

PENERAPAN METODE NATURAL LANGUAGE PROCESSING (NLP) DALAM IMPLEMENTASI ASISTEN VIRTUAL CHATBOT DENGAN MEMANFAATKAN API CHATGPT DAN GRADIO APP

Andre Farhan Saputra¹, Kecitaan Harefa²

¹Teknik Informatika, Universitas Pamulang, Jl. Raya Puspitek, Buaran, Kec. Pamulang, Kota Tangerang Selatan, Indonesia, 15310
e-mail: ¹andrefarhansaputra@gmail.com

²Teknik Informatika, Universitas Pamulang, Jl. Raya Puspitek, Buaran, Kec. Pamulang, Kota Tangerang Selatan, Indonesia, 15310
e-mail: ²dosen00842@unpam.ac.id

Abstract

This research discusses the application of NLP (Natural Language Processing) methods in everyday language use in virtual assistant Chatbots. The issue at hand is the lack of interactivity when communicating with Chatbots, caused by the use of formal language without voice output and a complicated interface. This problem was identified from three respondents through a ChatGPT satisfaction questionnaire. The goal is to meet user needs by making the Chatbot more interactive by adding voice output and an easy-to-use interface. The development method used is the Waterfall Model. In this study, the author developed a Chatbot system using machine learning, specifically the Gradio App as the interface, Python as the programming language, Visual Studio Code as the text editor, MySQL as the database, and the ChatGPT API as the main engine. By applying rules to the Chatbot system, it produced everyday language output and voice output using the TTS (Text to Speech) feature from the ChatGPT API. The research results showed that from a questionnaire of 24 respondents, ChatGPT users who directly tested Chillbot provided positive feedback, with a user satisfaction rate of 87.17%. It is hoped that this system can help advance technology, assist many people in dealing with problems, and better understand user intentions in interactions with Chatbots.

Abstrak

Penelitian ini membahas penerapan metode NLP (Natural Language Processing) dalam penggunaan bahasa sehari-hari pada asisten virtual Chatbot. Permasalahan yang dihadapi adalah kurangnya interaktivitas saat berkomunikasi dengan Chatbot, yang disebabkan oleh penggunaan bahasa baku tanpa output suara dan antarmuka yang rumit. Permasalahan ini diambil dari 3 orang responden melalui kuesioner tingkat kepuasan ChatGPT. Tujuannya adalah untuk memenuhi kebutuhan pengguna yaitu membuat Chatbot menjadi lebih interaktif dengan menambahkan output suara serta antarmuka yang mudah digunakan oleh user. Metode pengembangan yang digunakan adalah Model waterfall. Dalam penelitian ini, penulis mengembangkan sistem Chatbot dengan menggunakan machine learning, yaitu Gradio App sebagai antarmuka, Python sebagai bahasa pemrograman, Visual Studio Code sebagai text editor, MySQL sebagai database, dan API ChatGPT sebagai mesin utama. Dengan menerapkan aturan-aturan pada sistem Chatbot, dihasilkan output bahasa sehari-hari, dan output suara menggunakan fitur TTS (Text to Speech) dari API ChatGPT. Hasil penelitian menunjukkan bahwa hasil kuesioner dari 24 responden, pengguna ChatGPT yang menguji langsung Chillbot memberikan hasil yang positif yaitu 87,17% kepuasan pengguna. Diharapkan sistem ini dapat membantu memajukan teknologi, membantu banyak orang menghadapi masalah, dan lebih memahami niat pengguna dalam interaksi dengan Chatbot.

Keywords: Chatbot; Natural Language Processing; Output Suara; Informatics Engineering;

1. PENDAHULUAN

Kemajuan teknologi kecerdasan buatan (Artificial Intelligence/AI) telah memungkinkan komputer untuk melakukan pengolahan berbasis AI. Salah satu teknologi AI yang sedang populer adalah ChatGPT, yang dikembangkan oleh OpenAI. ChatGPT menggunakan metode deep learning untuk memberikan output yang menyerupai respons manusia dalam memahami dan menanggapi bahasa alami. Teknologi ini mampu memberikan jawaban yang tersusun dengan baik, memiliki ketelitian antar kata, serta mampu mengingat percakapan sebelumnya. Bahkan, ChatGPT dapat menghasilkan jurnal atau artikel ilmiah dengan cepat [1].

Penggunaan Chatbot seringkali menghadapi kendala dalam hal interaktivitas karena penggunaan bahasa yang terlalu baku dan antarmuka yang rumit. Dengan menambahkan fitur output suara dan menyederhanakan antarmuka, diharapkan dapat meningkatkan interaktivitas dan kepuasan pengguna dalam berkomunikasi dengan Chatbot [2]. Penelitian sebelumnya menunjukkan bahwa ChatGPT memiliki kekurangan dalam menghasilkan output suara dan menggunakan bahasa yang terlalu baku. Berdasarkan kuesioner tingkat kepuasan ChatGPT dari 3 responden, kurangnya interaktivitas saat berkomunikasi dengan Chatbot menjadi masalah utama yang dihadapi.

Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan menerapkan metode Natural Language Processing (NLP) dalam mengembangkan Chatbot yang menggunakan API ChatGPT dan Gradio App. Fokus utama penelitian ini adalah memanfaatkan API ChatGPT untuk memahami niat pengguna serta menghasilkan tanggapan yang relevan dan kontekstual. Selain itu, Gradio App akan digunakan sebagai antarmuka pengguna untuk meningkatkan interaksi yang intuitif dan efektif antara pengguna dan Chatbot. Dengan demikian, penelitian ini diharapkan dapat meningkatkan pengalaman pengguna dan membuat Chatbot lebih interaktif dan mudah digunakan.

Peneliti menunjukkan pemahaman yang mendalam tentang teknologi ChatGPT dan NLP, serta bagaimana teknologi ini dapat diimplementasikan untuk meningkatkan interaksi

Chatbot. Penelitian ini melibatkan penggunaan metode waterfall dalam pengembangan sistem Chatbot, dengan Python sebagai bahasa pemrograman, Visual Studio Code sebagai text editor, dan MySQL sebagai database. Keunggulan dari aplikasi ini adalah membuat Chatbot menjadi lebih interaktif dengan menambahkan output suara serta antarmuka yang mudah digunakan. Diduga pengguna mengalami kesulitan dalam berinteraksi dengan Chatbot karena bahasa terlalu baku, tidak ada output suara, dan antarmuka yang rumit. Penulis berharap dapat memenuhi keinginan pengguna dan membantu pengguna dalam mengatasi masalah saat menggunakan Chatbot. Dengan latar belakang tersebut, penulis tertarik untuk melakukan penelitian dengan judul “Penerapan Metode Natural Language Processing (NLP) dalam Implementasi Asisten Virtual Chatbot dengan Memanfaatkan API ChatGPT dan Gradio App”.

2. PENELITIAN YANG 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 jurnal penelitian terdahulu yang berkaitan dengan skripsi, antara lain:

1. (Setiawan & Luthfiyani, 2023) yang berjudul “Penggunaan ChatGPT Untuk Pendidikan di Era Education 4.0: Usulan Inovasi Meningkatkan Keterampilan Menulis”. Permasalahan yang terjadi adalah Chatbot AI saat ini telah menjadi solusi yang populer dalam industri teknologi. Hal ini dikarenakan Chatbot AI mampu meningkatkan efisiensi dan produktivitas dalam berbagai bidang, seperti layanan pelanggan, pemasaran, dan bahkan kesehatan. Namun, masih banyak Chatbot AI yang mengalami kesulitan dalam menangani percakapan yang kompleks dan mengerti konteks [2].

2. (Rifaldi, 2023) yang berjudul “Analisis Sentimen Terhadap Aplikasi ChatGPT Pada Twitter Menggunakan Algoritma Naïve Bayes”. Permasalahan yang terjadi adalah berkembangnya teknologi informasi berbasis komputer yang pesat berdampak terhadap perubahan dalam berbagai macam pilar kehidupan manusia. Kecerdasan

buatan (artificial intelligence) atau yang lebih dikenal dengan sebutan AI merupakan teknologi terkini produk kemajuan teknologi yang pesat. AI membuat komputer dapat melakukan banyak pekerjaan seperti yang dilakukan oleh 10 manusia sehingga menjadi produk teknologi yang banyak dimanfaatkan dalam pembuatan aplikasi pada saat ini karena memudahkan manusia dalam memenuhi kebutuhannya[3].

3. (Prasetyo, 2021) yang berjudul “Implementasi Natural Language Processing Dalam Pembuatan Chatbot Pada Program Information Technology Universitas Surabaya”. Permasalahan yang terjadi adalah keterbatasan informasi pada halaman tersebut mengakibatkan kurangnya minat dari calon mahasiswa untuk mendaftar pada Program Information Technology, Universitas Surabaya. Keterbatasan informasi tersebut sebenarnya dapat diatasi dengan bertanya langsung ke Direktorat Marketing and Public Relations (MPR) yang dimiliki Universitas Surabaya melalui telepon, email, ataupun livechatting. Akan tetapi, keterbatasan staf yang ada di MPR menyebabkan masalah baru yaitu human delay, apabila banyak calon pendaftar ingin bertanya dalam waktu yang bersamaan. Selain itu, adanya office hour mengakibatkan calon pendaftar tidak memiliki keleluasaan waktu dalam bertanya. Oleh karena itu, Chatbot dipilih karena dapat menggantikan peran manusia untuk memberikan jawaban dengan cepat kepada pengguna tanpa harus membaca skimming atau scanning pada sumber data informasi. Pada penelitian ini, Chatbot dibangun dengan menggunakan pendekatan Natural Language Processing[4].

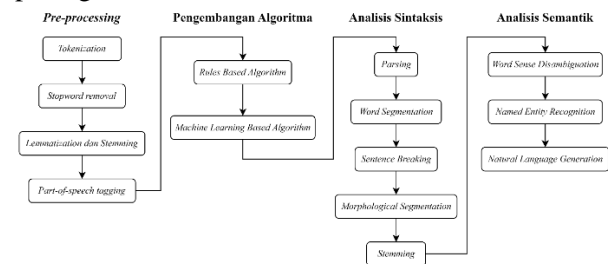
4. (Mashud, 2019) yang berjudul “Aplikasi Chatbot Berbasis Website Sebagai Virtual Personal Assistant Dalam Pemasaran Properti”. Permasalahan yang dihadapi adalah semakin hari jumlah pelanggan yang 11 bertanya semakin banyak dan membutuhkan respon yang cepat. Penelitian ini bertujuan merancang suatu aplikasi chatbot berbasis website yang dapat membantu menjawab pertanyaan pelanggan dengan cepat sehingga pelanggan mendapatkan informasi yang dibutuhkan tanpa membutuhkan waktu dan tenaga dari pegawai[5].

5. (Yunefri & Ersan, 2021) yang berjudul “Chatbot Pada Smart Cooperative Oriented Problem Menggunakan Natural Language Processing Dan Naive Bayes Classifier”. Kendala

yang dialami dosen dalam mengajar matakuliah Struktur Data adalah belum adanya model pembelajaran yang mendekatkan mahasiswa dengan teori abstrak yang sulit untuk dipahami mahasiswa, untuk mengatasi permasalahan tersebut maka perlu adanya model pembelajaran yang dapat meningkatkan kualitas belajar yang dimiliki mahasiswa dengan cara melakukan chatbot belajar[6].

3. METODE PENELITIAN

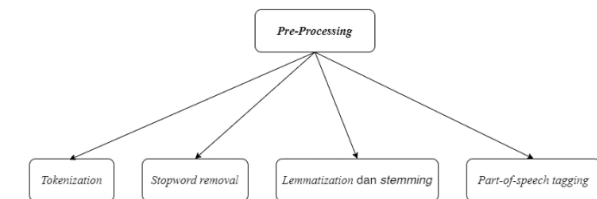
Pada penelitian ini, *Natural Language Processing (NLP)* diimplementasikan pada asisten virtual *Chatbot*[7]. Proses yang dilakukan terlihat pada gambar 5.1.



Gambar 1 Kerangka Kerja Metode NLP

3.1 Pre-Processing

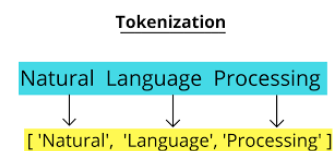
Pada tahap ini merupakan tahapan dimana teks diseragamkan bentuk dan formatnya, supaya dapat dipersiapkan menjadi data yang dapat diolah untuk hasil kalimat yang lebih alami.



Gambar 2 Tahapan Pre-Processing

3.1.1 Tokenization

Pada tahap ini, dilakukannya pembagian setiap kata menjadi lebih kecil. Tujuannya agar memudahkan kinerja sistem.



Gambar 3 Teknik Tokenization

3.1.2 Stopword Removal

Selanjutnya pada tahap ini, dilakukannya pemilihan kata yang dinilai menunjukkan data/informasi yang bisa diambil dari teks. Tujuannya untuk menghilangkan kata yang tidak memiliki arti.

Stop Words

["This", "is", "a", "test"]
 ✓ X X ✓

Gambar 4 Teknik Stopword Removal

3.1.3 Lemmatization dan Stemming

Berikutnya pada tahap ini, dilakukannya pengembalian kata ke bentuk kata dasar, dengan cara menghilangkan kata imbuhan. Tujuannya untuk mengenali kata berdasarkan kamus bahasa.

Word	Stemming	Lemmatization
information	inform	information
informative	inform	informative
computers	comput	computer
feet	feet	foot

Gambar 5.5 Teknik Lemmatization dan Stemming

3.1.4 Part-of-speech Tagging

Terakhir, pada tahap ini dilakukannya penandaan kata berdasarkan kelasnya. Tujuannya untuk mengelompokkan kata berdasarkan kelas kata seperti: kata benda, kata kerja, atau kata sambung.



Part of Speech Tagging

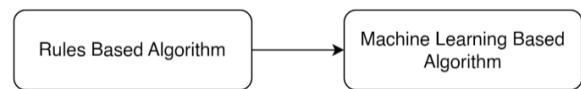
Gambar 6 Teknik Part of Speech Tagging

Tabel 5.1 Daftar Part of Speech Tags

Tag	Deskripsi	Artinya
PRP	Proper Noun	Kata benda yang tepat
VBZ	Verbs	Kata kerja
NNS	Nouns	Kata benda jamak
IN	Interjection	Kata seru
DT	Determiner	Kata sandang
NN	Noun	Kata benda

3.2 Pengembangan Algoritma

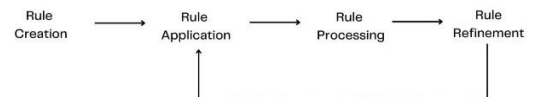
Sebelum melakukan analisis, perlu melakukan pembersihan data yaitu dengan pengembangan algoritma. Algoritma yang digunakan adalah:



Gambar 7 Tahapan Pengembangan Algoritma

3.2.1 Rule Based Algorithm

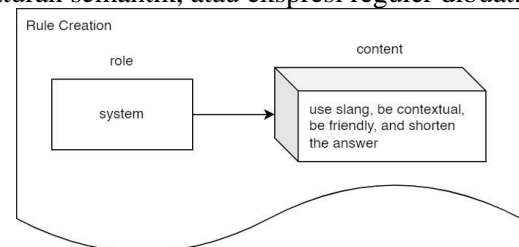
Pada tahap ini, melakukan analisis dan proses data teks dengan menerapkan aturan linguistik yang telah ditentukan sebelumnya. Fungsinya adalah untuk menangkap struktur khusus dalam teks, mengekstraksi informasi yang relevan, dan melakukan tugas-tugas seperti klasifikasi teks berdasarkan aturan atau pola tertentu.



Gambar 8 Rule Based Approach pada NLP

1. Rule Creation

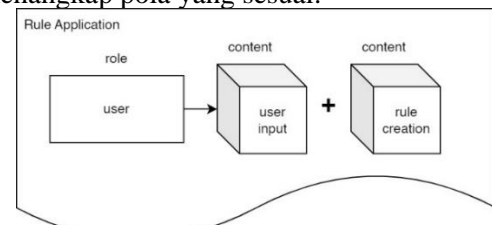
Berdasarkan tugas yang diinginkan, aturan-aturan linguistik spesifik domain seperti aturan tata bahasa, pola sintaksis, aturan semantik, atau ekspresi reguler dibuat.



Gambar 9 Rule Creation

2. Rule Application

Aturan-aturan yang telah ditetapkan diterapkan pada data yang dimasukkan untuk menangkap pola yang sesuai.



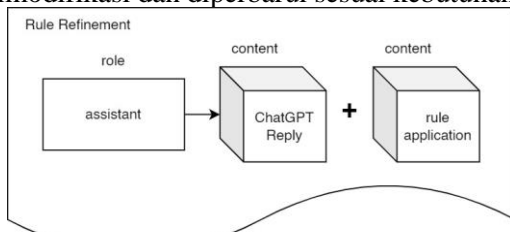
Gambar 10 Rule Application

3. Rule Processing

Data teks diproses sesuai dengan hasil dari aturan-aturan yang sesuai untuk mengekstraksi informasi, membuat keputusan, atau melaksanakan tugas lainnya.

4. Rule Refinement

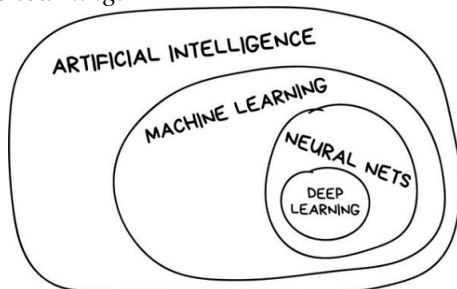
Aturan-aturan yang telah dibuat disempurnakan secara iterasi melalui pemrosesan berulang untuk meningkatkan akurasi dan kinerja. Berdasarkan umpan balik sebelumnya, aturan-aturan tersebut dimodifikasi dan diperbarui sesuai kebutuhan.



Gambar 11 Rule Refinement

3.2.2 Machine Learning Based Algorithm

Selanjutnya pada tahap ini, dilakukannya proses *training* terhadap data yang diberikan, dengan melakukan *training* secara berulang maka akan menghasilkan data yang lebih akurat. Menggunakan kombinasi dari *machine learning*, *Neural network* dan *deep learning*.

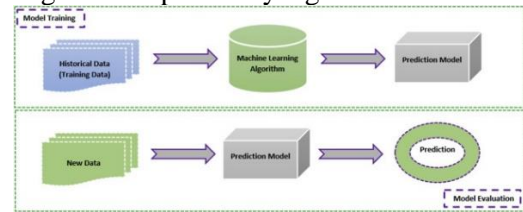


Gambar 12 Kombinasi dari Machine Learning, Neural Network dan Deep Learning

Hal yang dilakukan pada saat *training data* yaitu dengan melihat *history data (training data)* kemudian diproses melalui *machine learning* yang akan menghasilkan *prediction model*.

Selanjutnya melalui proses evaluasi data yaitu mengambil data baru, kemudian

disatukan dengan *prediction model*, yang akan menghasilkan prediksi yang lebih akurat.



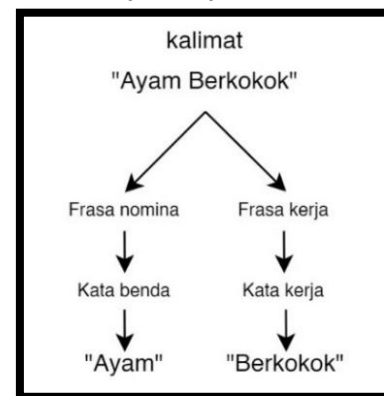
Gambar 13 Model Training dan Model Evaluation

3.3 Analisis Sintaks

Tahapan selanjutnya adalah analisis sintaksis yaitu teknik pengaturan pada suatu kalimat sehingga kalimat memiliki tata bahasa yang benar. Tujuannya untuk menilai arti dan makna dari sebuah kalimat.

3.3.1 Parsing

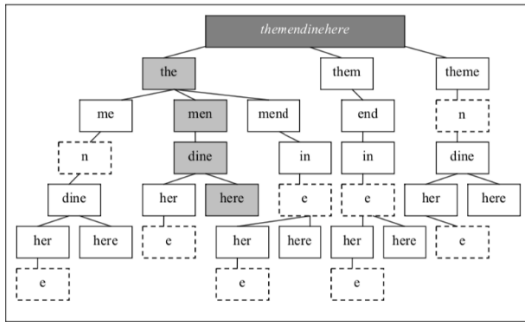
Pada tahap ini, melibatkan penentuan struktur kalimat, seperti subjek, predikat, objek, dan keterangan. Sebagai contoh “Ayam Berkokok”, saat di *parsing* akan menjadi 2 kata yaitu “Ayam” sebagai subjek dan “Berkokok” menjadi objek.



Gambar 14 Teknik Parsing

3.3.2 Word Segmentation

Tahapan selanjutnya yaitu dilakukannya proses penentuan batas kata dalam sebuah kalimat dengan algoritma komputer. Supaya penyebutan tiap kata menjadi tepat, sesuai dengan konteks dan aksennya.



Gambar 15 Teknik Word Segmentation

5.3.3 Sentence Breaking

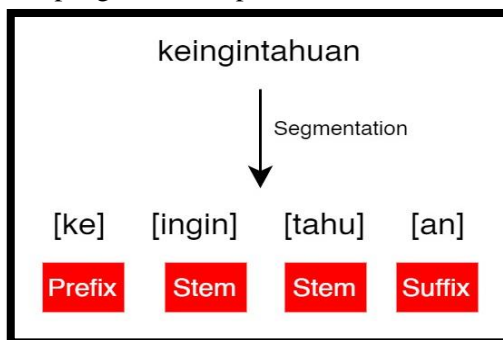
Selanjutnya pada tahap ini, dilakukannya teknik pemecahan sebuah kalimat menjadi beberapa kalimat. Yaitu dengan mengenali tanda koma sebagai pemisahannya.



Gambar 16 Teknik Sentence Breaking

3.3.4 Morphological Segmentation

Pada tahap ini, dilakukannya pembagian kata menjadi bagian yang lebih kecil atau morfem. Tujuannya agar komputer dapat mengenali kata imbuhan, selain itu juga untuk pengenalan ucapan.

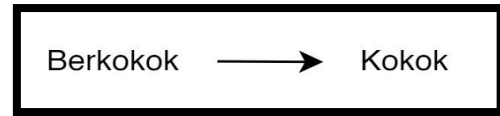


Gambar 17 Teknik Morphological Segmentation

3.3.5 Stemming

Tahapan terakhir pada analisis sintaksis, pada tahap ini yaitu melakukan pembagian kata yang mempunyai imbuhan

akan diubah menjadi kata dasar. Tujuannya agar komputer dapat mengenali kata dasarnya.



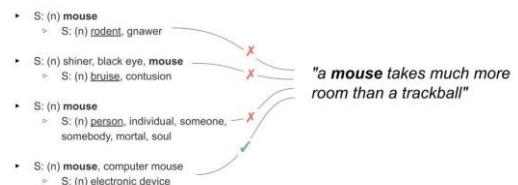
Gambar 18 Teknik Stemming

3.4 Analisis Semantik

Terakhir, pada tahap ini berfokus pada makna dan fungsi dari sebuah kata dalam kalimat. Tujuannya agar kata yang dihasilkan memiliki informasi, relevan, dan kontekstual.

3.4.1 Word Sense Disambiguation

Pada tahap ini, melakukan penalaran dari sebuah kata berdasarkan konteksnya. Tujuannya agar komputer memahami arti kata yang dimaksud sesuai dengan konteksnya.

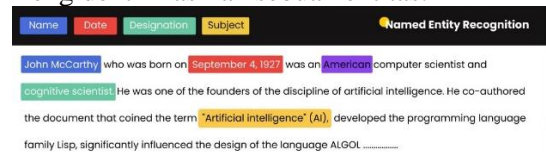


Gambar 19 Teknik Word Sense Disambiguation

Dapat dilihat pada gambar 5.19 bahwa kata *mouse* merujuk pada *mouse* komputer, bukan *mouse* hewan pengerat, orang, ataupun memar.

3.4.2 Named Entity Recognition

Selanjutnya pada tahap ini, melakukan pengkategorian pada kata-kata ke dalam beberapa kelompok kata. Tujuannya untuk mengidentifikasi sebuah entitas.

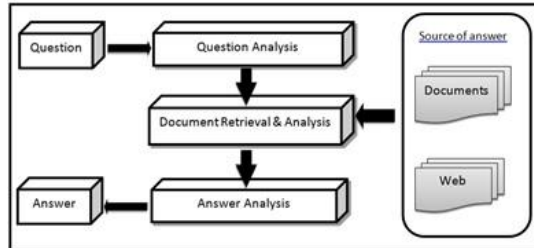


Gambar 20 Teknik Named Entity Recognition

Pada gambar 5.20 dapat dilihat bahwa kata yang disorot dengan warna memiliki identitas. Seperti "September 4, 1927" yang disorot warna merah adalah sebuah tanggal.

3.4.3 Natural Language Generation

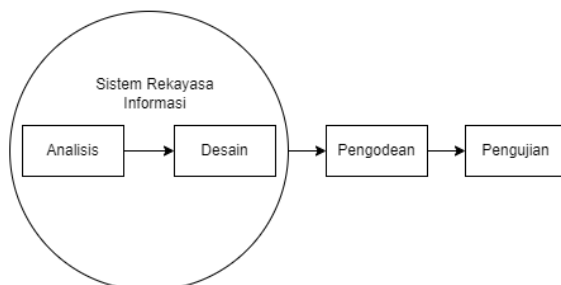
Terakhir, pada tahap ini dilakukannya penggalian makna dari sebuah teks, dengan melakukan pengenalan hubungan antara kata-kata tertentu yang memungkinkan komputer untuk memahami dan menafsirkan frasa, paragraf, atau bahkan seluruh manuskrip. Untuk menghasilkan ringkasan tekstual, data yang diambil berdasarkan *database* dan kumpulan data yang telah dikumpulkan sebelumnya.



Gambar 21 Teknik *Natural Language Generation*

Model Waterfall

Menurut (Handrianto & Sanjaya, 2020) dalam Sukanto & Shalahuddin, 2018. *Model Waterfall* adalah “model menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau teratur dimulai dari analisis, desain, pengodean, dan pengujian”[9]. Berikut gambar dari *Model waterfall* dalam SDLC:



Gambar 22 Ilustrasi *Model Waterfall*

Berikut ini adalah tahapan dari *Model waterfall* yaitu:

1. Analisis

Melakukan analisis kebutuhan perangkat lunak, fungsi dan proses dari web yang dibuat, pengidentifikasian kendala dalam pembuatan web, menganalisis keandalan, kelemahan, dan teknologi yang dipakai.

2. Desain

Desain perangkat lunak adalah proses beberapa tahapan langkah pada rancangan pembuatan program perangkat lunak meliputi struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean. Tahap ini melakukan translasi kebutuhan perangkat lunak dari tahapan analisis kebutuhan ke representasi rancangan agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Pada tahap ini, hasil dari desain perangkat lunak yang telah ada didokumentasikan.

3. Pengodean

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai desain yang telah dibuat pada tahap desain. Atau tahapan penulis membuat program dengan bahasa program seperti *Python*, *PHP* dan lain-lain.

4. Pengujian

Tahapan ini penulis melakukan pengujian terhadap program yang telah dibuat untuk mengetahui kekurangan dari program tersebut.

4. HASIL DAN PEMBAHASAN

Implementasi merupakan proses penerapan ide, konsep, kebijakan, dalam bentuk tindakan praktis, sehingga memberikan dampak, baik berupa perubahan pengetahuan, keterampilan, maupun nilai dan sikap[10].

Dalam mengimplementasikan program *Chatbot* (Chillbot), diperlukan lingkungan pengembangan dengan menginstall *Python* serta library nya seperti *Gradio*, *OpenAI*, *Pygame*, dan *MySQL*. Dengan *Gradio* sebagai tampilan antarmuka, *OpenAI* sebagai mesin pemroses bahasa alami yang akan menghasilkan respon pesan, *Pygame* sebagai auto-play audio, dan *MySQL* sebagai database nya.

Kemudian, menginstall *XAMPP* sebagai local database, setelah itu membuat database dengan nama *db_chillbot* pada *PHPMyAdmin*. Database ini akan digunakan sebagai penyimpanan data riwayat percakapan.

Selanjutnya, menginstall *Visual Studio Code* sebagai code editor untuk merancang kode dari *Chatbot* (Chillbot). *Visual Studio Code* berfungsi sebagai alat untuk menulis, mengedit, dan menjalankan program dengan mudah dan efisien.

API ChatGPT adalah layanan *API* yang disediakan oleh *OpenAI* untuk mengintegrasikan model secara efisien. *API ChatGPT*

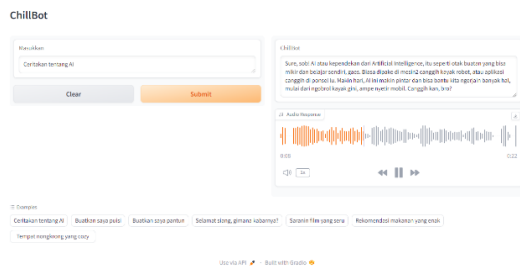
memungkinkan untuk mengakses fitur dan fungsionalitas *ChatGPT* secara langsung dari aplikasi, yang bisa meningkatkan kinerja dan mengurangi waktu tunggu[11].

Terakhir, membuat API pada website OpenAI yang akan digunakan untuk mengakses layanan pemrosesan bahasa alami. Dengan semua komponen ini terinstal dan dikonfigurasi, pengembangan dan integrasi Chatbot (Chillbot) dapat dilakukan, memastikan bahwa setiap bagian berfungsi dengan baik dan saling terhubung.

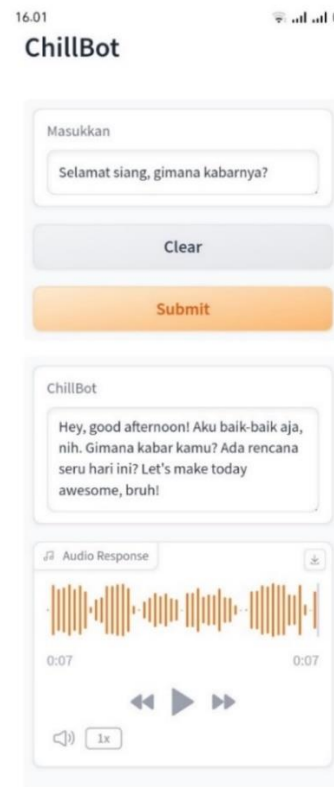
Tampilan Antarmuka

Gradio App adalah pustaka *Python* sumber terbuka untuk menciptakan antarmuka pengguna yang memungkinkan demonstrasi model pembelajaran mesin, alur kerja ilmu data, serta interaksi dengan pengguna melalui *browser*[12].

Berikut merupakan tampilan dari antarmuka *Chillbot*, terdiri dari *form input*, *form output text*, *audio player* untuk *output* suara, dan beberapa teks contoh *input* yang bisa digunakan.



Gambar 23 Tampilan Antarmuka Pada Desktop



Gambar 24 Tampilan Antarmuka Pada Mobile

Kode Program

Python adalah bahasa pemrograman interpretatif yang dianggap mudah dipelajari serta berfokus pada keterbacaan kode. *Python* secara umum berbentuk pemrograman berorientasi objek, pemrograman imperatif dan pemrograman fungsional[13].

Visual Studio Code adalah code editor sumber yang dikembangkan oleh Microsoft untuk Windows, Linux dan macOS. Ini termasuk dukungan untuk debugging, kontrol git yang tertanam dan GitHub, penyorotan sintaksis, penyelesaian kode cerdas, snippet, dan refactoring kode. Ini sangat dapat disesuaikan, memungkinkan pengguna untuk mengubah tema, pintasan keyboard, preferensi, dan menginstal ekstensi yang menambah fungsionalitas tambahan[14].

MySQL merupakan *database* yang cukup terkenal karena hampir sebagian besar aplikasi berbasis website seperti *WordPress* dilengkapi dengan *MySQL*. *SQL (Structured Query Language)*

adalah bahasa yang digunakan untuk mengelola data pada RDBMS (Relational Database Management System)[15].

Kode program yang digunakan pada aplikasi Chatbot (Chillbot) yaitu menggunakan bahasa pemrograman Python. Berikut adalah kode programnya:

1. Library Python, berdasarkan gambar 3.3 diperlihatkan beberapa library yang digunakan seperti, Gradio sebagai antarmuka, Pygame sebagai pemutar audio otomatis, MySQL.connector sebagai database, dan OpenAI sebagai API.

```
# Import libraries
import gradio as gr
import pygame
import mysql.connector # type: ignore
from openai import OpenAI
```

Gambar 25 Library Python

2. Database MySQL, berikut adalah database yang digunakan pada Chatbot. Jika tabel tidak ada maka akan dibuat tabel baru dengan nama chat_history pada database db_chillbot.

```
# Inisialisasi ke database mysql
conn = mysql.connector.connect(host="localhost", user="root", password="", database="db_chillbot")
cursor = conn.cursor()

cursor.execute("""
CREATE TABLE IF NOT EXISTS chat_history (
id INT AUTO_INCREMENT PRIMARY KEY,
role VARCHAR(255),
content TEXT
)
""")
conn.commit()
```

Gambar 26 Database MySQL

3. Fungsi Respon Pada Chatbot
a. Mendefinisikan fungsi Respon pada Chatbot.

```
# Mendefinisikan Fungsi Respon pada Chatbot
def generate_writing_prompt(Masukkan):
```

Gambar 27 Mendefinisikan Fungsi Respon pada Chatbot

- b. Menyimpan pesan user ke dalam tabel, dan merespon pesan berdasarkan pesan sebelumnya.

```
# Menjalankan pesan user ke tabel
cursor.execute("INSERT INTO chat_history (role, content) VALUES (%s, %s)", ("user", Masukkan))
conn.commit()
messages.append({"role": "user", "content": Masukkan})
response = client.chat.completions.create(
model="gpt-3.5-turbo",
messages=messages
)
```

Gambar 28 Menyimpan Pesan User

- c. Memproses respon dari Chatbot dan menghasilkan suara dari respon tersebut.

```
# Memproses Respon dari Chatbot
chatgpt_reply = response.choices[0].message.content
messages.append({"role": "assistant", "content": chatgpt_reply})

# Menghasilkan Audio dari Respon Chatbot
response2 = client.audio.speech.create(
model="tts-1",
voice="alloy",
input=chatgpt_reply
)
```

Gambar 29 Respon Pesan teks dan suara dari Chatbot

```
# Mengembalikan Hasil Pada Antarmuka
return chatgpt_reply, output_filename
```

Gambar 30 Mengembalikan Hasil Pada Antarmuka

4. Fungsi Auto-play Audio, membuat fungsi Auto-play audio menggunakan library Pygame.

```
# Fungsi untuk memainkan audio menggunakan pygame
def play_audio(audio_file):
pygame.mixer.music.load(audio_file)
pygame.mixer.music.play()
```

Gambar 31 Fungsi Auto-play Audio

5. Antarmuka Menggunakan Gradio App, menampilkan antarmuka menggunakan Gradio App, dan beberapa contoh teks input bila ingin digunakan.

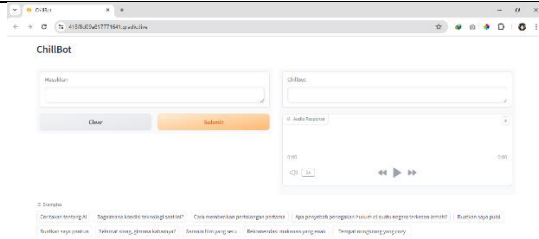
```
# Gradio user interface
interface = gr.Interface(
fn=generate_writing_prompt,
inputs="textbox",
outputs=[
gr.Textbox(label="ChillBot"),
gr.Audio("output.mp3", type="filepath", label="Audio Response", autoplay=True)
],
title="ChillBot",
allow_flagging="auto",
)

examples=[
["Ceritakan tentang AI"],
["Bagaimana kondisi teknologi saat ini?"],
["Cara memberikan pertolongan pertama"],
["Apa penyebab pengekangan hukum di suatu negara terkesan lemah?"],
["Buatkan saya puisi"],
["Buatkan saya pantun"],
["Selamat siang, gimana kabarnya?"],
["Saranin film yang seru"],
["Rekomendasi makanan yang enak"],
["Tempat nongkrong yang cozy"],
]
```

Gambar 32 Antarmuka Menggunakan Gradio App

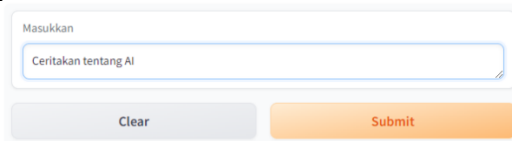
Demo Aplikasi

Demo aplikasi ini bertujuan untuk menunjukkan cara kerja dan fitur-fitur utama dari aplikasi Chatbot yang telah dikembangkan. Demonstrasi ini akan memberikan gambaran umum mengenai fungsionalitas dan antarmuka pengguna dari aplikasi Chillbot. Setelah mengakses link akan tampil halaman utama Chillbot.



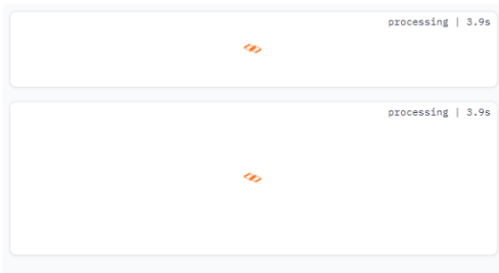
Gambar 33 Halaman Utama Demo Aplikasi Chillbot

Selanjutnya masuk ke bagian *form input* yang akan digunakan oleh *user*. Setelah memasukkan teksnya kemudian klik tombol submit.



Gambar 34 Form Input User Chillbot

Kemudian *Chillbot* akan memproses hasil yang dimasukkan oleh *user*. Proses hanya berjalan beberapa detik saja.

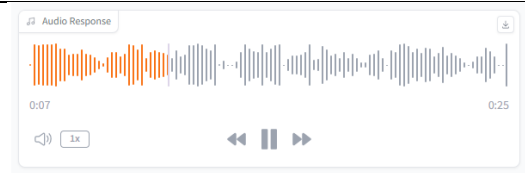


Gambar 35 Aplikasi Chillbot Saat Memproses Teks

Setelah proses selesai *Chillbot* akan menampilkan *output* / hasil berupa teks dan suara dari teks yang dimasukkan oleh *user*.

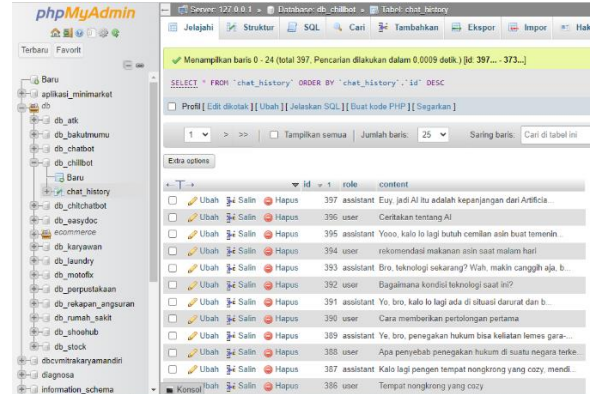


Gambar 36 Output Teks dari Chillbot



Gambar 37 Output Suara dari Chillbot

Terakhir memastikan *input* dari *user* dan *output* dari *Chillbot* tersimpan dalam *database db_chillbot*. Berikut tampilan *database* dari *db_chillbot*.



Gambar 38 Tampilan Database dari db_chillbot

Pengujian Program

Black Box Testing

Pengujian *Black Box* dilakukan untuk memastikan bahwa aplikasi bekerja sesuai dengan harapan tanpa melihat kode sumber[16]. Berikut adalah detail dari pengujian *Black Box*:

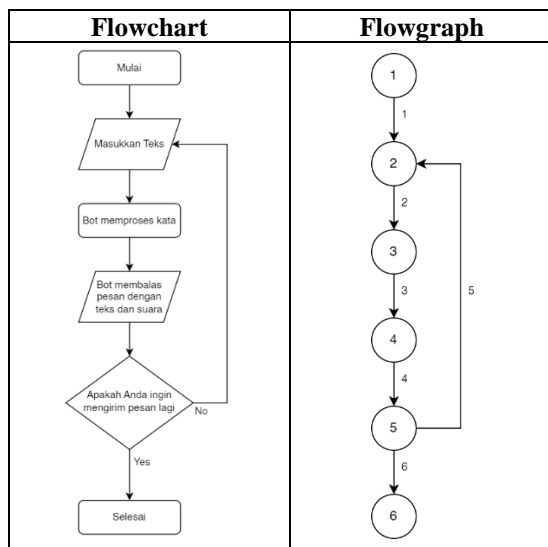
Tabel 6.1 Pengujian *Black Box* Pada *Chillbot*

No	Detail Pengujian	Output yang Diharapkan	Output yang Diharaskan	Kesimpulan
1	Pengguna memasukkan teks pertanyaan di form "Masukkan", kemudian	Chillbot merespons dengan jawaban yang relevan	Chillbot merespons dengan jawaban yang relevan	Berhasil

	an menekan tombol "Submit" pada <i>Gradio App</i>						kembali riwayat percakapan sebelumnya melalui <i>Gradio App</i>	es kembali riwayat percakapan sebelumnya	mengakses kembali riwayat percakapan	
2	Pengujian respon waktu saat mengirimkan input pengguna ke <i>API ChatGPT</i> dan menerima respon	Respon waktu yang cepat	Respon waktu kurang dari 5 detik (bergantung pada koneksi internet)	Berhasil		6	Memilih teks dari <i>dataset</i> yang tersedia pada "example"	Teks yang dipilih masuk ke dalam <i>form</i> "Masukkan"	Teks yang dipilih masuk ke dalam <i>form</i> "Masukkan"	Berhasil
3	Memeriksa pemutaran audio otomatis menggunakan <i>Pygame</i> setelah <i>Chillbot</i> memberikan respon	Audio terkait diputar secara otomatis	Audio diputar secara otomatis	Berhasil		7	Menguji ketahanan sistem dengan memberikan input secara bersamaan dengan <i>device</i> berbeda	Sistem tetap stabil dan memberikan jawaban yang tepat tanpa <i>error</i>	Sistem stabil dan memberikan jawaban yang tepat tanpa <i>error</i>	Berhasil
4	Memeriksa penyimpanan riwayat percakapan di <i>database MySQL</i>	Riwayat percakapan tersimpan dengan benar di <i>database</i>	Riwayat percakapan tersimpan dengan benar di <i>database</i>	Berhasil		8	Fitur <i>download</i> audio	Audio dapat di <i>download</i>	Audio dapat di <i>download</i>	Berhasil
5	Pengguna mengakses	Pengguna tidak dapat mengakses	Pengguna tidak dapat	Berhasil		9	Pengguna menekan tombol <i>clear</i> untuk menghapus riwayat percakapan	Riwayat percakapan di antarmuka terhapus	Riwayat percakapan di antarmuka terhapus	Berhasil

White Box Testing

Pengujian *white box testing* yang dilakukan adalah menggunakan teknik *Basic Path*. Pengujian dilakukan dengan *test case* yang didapatkan melalui jalur independen [16]. Dengan cara menghitung nilai *cyclomatic complexity* dari *flowgraph*. *Flowgraph* dibuat berdasarkan *flowchart* analisa sistem usulan.



Gambar 39 Flowgraph Analisa Sistem Susulan

Dari gambar 6.17 menunjukkan bahwa *flowchart* dan *flowgraph* memiliki *cyclomatic complexity* dari modul integrasi *API ChatGPT*, berikut adalah rumusnya:

$$V(G) = \text{Jumlah Predicate Node} + 1$$

$$= 0 + 1$$

$$= 1$$

$$V(G) = E - N + 2$$

$$= 6 - 6 + 2$$

$$= 2$$

$$V(G) = \text{Jumlah Region}$$

$$= 1$$

Keterangan:

E = Edge (Panah)

N = Node (Simpul)

Jadi jalurnya seperti ini:

Path I = 1 – 2 – 3 – 4 – 5 – 6

Path II = 1 – 2 – 3 – 4 – 5 – 2 – 3 – 4 – 5 – 6

User Acceptance Testing

User Acceptance Testing dilakukan oleh pengguna langsung dengan tujuan untuk mengevaluasi sejauh mana aplikasi *Chillbot* ini memenuhi kebutuhan dan harapan mereka. Metode pengujian ini melibatkan *user* dalam situasi nyata untuk berinteraksi dengan *Chillbot* dan memberikan *feedback* tentang pengalaman mereka. Pengujian dilakukan dengan menggunakan kuesioner yang diisi oleh *user* dengan bobot penilaian sebagai berikut:

Tabel 6.2 Bobot Penilaian *User*

Bobot	Keterangan
5	Sangat Setuju
4	Setuju
3	Cukup
2	Tidak Setuju
1	Sangat Tidak Setuju

Kuesioner ini berisi 10 pertanyaan yang kemudian diisi oleh *user* sebagai respondennya. Sebanyak 24 *user* yang menjadi responden, responden tersebut dipilih berdasarkan kriteria berikut:

1. *User* merupakan pengguna aplikasi *Chatbot* dari *ChatGPT*
2. *User* memiliki akses ke internet
3. *User* memahami dasar-dasar *AI (Artificial Intelligent)*

Berikut merupakan 10 pertanyaan *User Acceptance testing*:

Tabel 6.3 Pertanyaan dari Kuesioner UAT

No.	Pertanyaan
1	Seberapa mudah Anda menemukan dan mengakses aplikasi web Chillbot?
2	Bagaimana pengalaman Anda dengan antarmuka pengguna (UI) aplikasi web Chillbot?
3	Seberapa intuitif Anda menemukan navigasi dan fitur-fitur yang ada di aplikasi web Chillbot?
4	Bagaimana kualitas suara yang dihasilkan oleh Chillbot pada aplikasi web ini?
5	Apakah Chillbot memberikan jawaban yang relevan dan sesuai dengan pertanyaan yang Anda ajukan?

6	Bagaimana kecepatan respon dari Chillbot setelah Anda mengajukan pertanyaan?
7	Seberapa puas Anda dengan keseluruhan pengalaman menggunakan aplikasi web Chillbot?
8	Apakah Anda merasa Chillbot berguna dalam membantu kebutuhan Anda?
9	Seberapa besar kemungkinan Anda merekomendasikan Chillbot kepada teman atau keluarga Anda?
10	Secara keseluruhan, bagaimana penilaian Anda terhadap aplikasi Chillbot ini?

P3	9 x 5 = 45	12 x 4 = 48	3 x 3 = 9	0 x 2 = 0	0 x 1 = 0	102
P4	14 x 5 = 70	8 x 4 = 32	2 x 3 = 6	0 x 2 = 0	0 x 1 = 0	108
P5	10 x 5 = 50	10 x 4 = 40	4 x 3 = 12	0 x 2 = 0	0 x 1 = 0	102
P6	12 x 5 = 60	9 x 4 = 36	2 x 3 = 6	1 x 2 = 2	0 x 1 = 0	104
P7	14 x 5 = 70	8 x 4 = 32	2 x 3 = 6	0 x 2 = 0	0 x 1 = 0	108
P8	13 x 5 = 65	8 x 4 = 32	3 x 3 = 9	0 x 2 = 0	0 x 1 = 0	106
P9	8 x 5 = 40	14 x 4 = 56	2 x 3 = 6	0 x 2 = 0	0 x 1 = 0	102
P10	15 x 5 = 75	8 x 4 = 32	0 x 3 = 0	1 x 2 = 2	0 x 1 = 0	109

Data dari tabel diatas kemudian digunakan untuk mencari nilai rata-rata dan persentase kesetujuan *user*. Berikut adalah rumusnya:

$$\text{Nilai rata - rata} = \frac{\text{Jumlah Bobot Nilai Responden}}{\text{Total Responden}}$$

$$\text{Persentase} = \frac{\text{Nilai Rata - rata}}{\text{Bobot Maximum}} \times 100\%$$

Berikut perhitungan nilai rata-rata dan persentase di dalam tabel:

Tabel 6.6 Perhitungan Nilai Persentase

Per tanyaan	Nilai Rata-rata	Persentase
P1	103 / 24 = 4.2916	4.2916 / 5 * 100% = 85.83%
P2	102 / 24 = 4.25	4.25 / 5 * 100% = 85.00%
P3	102 / 24 = 4.25	4.25 / 5 * 100% = 85.00%
P4	108 / 24 = 4.5	4.5 / 5 * 100% = 90.00%
P5	102 / 24 = 4.25	4.25 / 5 * 100% = 85.00%
P6	104 / 24 = 4.333	4.333 / 5 * 100% = 86.67%
P7	108 / 24 = 4.5	4.5 / 5 * 100% = 90.00%
P8	106 / 24 = 4.416	4.416 / 5 * 100% = 88.33%
P9	102 / 24 = 4.25	4.25 / 5 * 100% = 85.00%
P10	109 / 24 = 4.5416	4.5416 / 5 * 100% = 90.83%

Berikut merupakan tabel hasil dari jawaban responden:

Tabel 6.4 Hasil Jawaban dari Responden

Jawaban Responden					
Pertanyaan	Sangat Setuju (SS)	Setuju (S)	Cukup (C)	Tidak Setuju (TS)	Sangat Tidak Setuju (STS)
P1	13	6	4	1	0
P2	9	12	3	0	0
P3	9	12	3	0	0
P4	14	8	2	0	0
P5	10	10	4	0	0
P6	12	9	2	1	0
P7	14	8	2	0	0
P8	13	8	3	0	0
P9	8	14	2	0	0
P10	15	8	0	1	0

Selanjutnya menghitung jumlah bobot dari setiap jawaban responden:

Tabel 6.5 Perhitungan Jumlah Bobot

Jawaban Responden						
Pertanyaan	Sangat Setuju (SS) x5	Setuju (S) x4	Cukup (C) x3	Tidak Setuju (TS) x2	Sangat Tidak Setuju (STS) x1	Jumlah Bobot
P1	13 x 5 = 65	6 x 4 = 24	4 x 3 = 12	1 x 2 = 2	0 x 1 = 0	103
P2	9 x 5 = 45	12 x 4 = 48	3 x 3 = 9	0 x 2 = 0	0 x 1 = 0	102

Hasil persentase keseluruhan menunjukkan hasil dari *User Acceptance Testing*, yang dapat dilihat dari kriteria interpretasi skor menggunakan Skala *Likert* sebagai berikut:

Tabel 6.7 Skala *Likert*

Persentase	Keterangan
0% - 20%	Sangat Tidak Puas
21% - 40%	Tidak Puas
41% - 60%	Cukup Puas
61% - 80%	Puas
81% - 100%	Sangat Puas

Persentase keseluruhan menunjukkan hasil dari pengujian tingkat kepuasan *user* secara menyeluruh dengan menghitung total persentase yang dibagi dengan jumlah pertanyaan, yaitu sebagai berikut:

$$\text{Persentase Keseluruhan} = \frac{\text{Total Persentase}}{\text{Jumlah Pertanyaan}}$$

$$\frac{(85.83\% + 85\% + 85\% + 90\% + 85\% + 86.67\% + 90\% + 88.33\% + 85\% + 90.83\%)}{10} = 87.17\%$$

Dari perhitungan tersebut, didapatkan hasil 87.17% yang berarti *user* sangat puas dan sistem *Chillbot* memenuhi kebutuhan *user*.

5. KESIMPULAN

Berikut merupakan kesimpulan yang didapat berdasarkan rumusan masalah dalam penelitian ini:

- a. Permasalahan terkait rancangan Chatbot yang kurang interaktif, dengan membuat Chatbot menggunakan bahasa sehari-hari agar mampu memberikan respons yang lebih relevan dan kontekstual, sehingga komunikasi menjadi lebih interaktif.
- b. Teknik yang digunakan untuk menghasilkan output suara pada *Chillbot*, dengan menggunakan teknik Text to Speech (TTS) yaitu, mengubah hasil chat menjadi output suara.
- c. Tampilan antarmuka yang lebih interaktif, dengan menggunakan *Gradio App* sebagai antarmuka akan menghasilkan tampilan yang responsif dan mudah digunakan oleh user. Penelitian ini berhasil meningkatkan kemampuan Chatbot dalam mengenali niat pengguna dengan metode Natural Language Processing (NLP) serta memanfaatkan API ChatGPT dan *Gradio App*. Dibuktikan dengan

hasil dari kuesioner yang diuji langsung oleh 24 responden dan menghasilkan 87.17% kepuasan pengguna.

DAFTAR PUSTAKA

- [1] F. Kurnia Ramadhan, M. Irfan Faris, I. Wahyudi, and M. Kamayani Sulaeman, "PEMANFAATAN CHATGPT DALAM DUNIA PENDIDIKAN".
- [2] 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>.
- [3] M. Ilmar Rifaldi, Y. Raymond Ramadhan, and I. Jaelani, "Analisis Sentimen Terhadap Aplikasi Chatgpt Pada Twitter Menggunakan Algoritma Naïve Bayes," 2023.
- [4] V. R. Prasetyo, N. Benarkah, and V. J. Chrisintha, "Implementasi Natural Language Processing Dalam Pembuatan Chatbot Pada Program Information Technology Universitas Surabaya," *Teknika*, vol. 10, no. 2, pp. 114–121, Jul. 2021, doi: 10.34148/teknika.v10i2.370.
- [5] K. Akuntansi STMIK AKBA, "Mashud, Aplikasi Chatbot Berbasis Website Sebagai Virtual Personal Assistant Dalam Pemasaran Properti APLIKASI Chatbot Berbasis Website Sebagai Virtual Personal Assistant Dalam Pemasaran Properti." Y. Yunefri And Y. Ersan Fadrial, "Chatbot Pada Smart Cooperative Oriented Problem Menggunakan Natural Language Processing Dan Naive Bayes Classifier Chatbot On Smart Cooperative Oriented Problems Using Natural Language Processing And Naive Bayes Classifier," *Journal Of Information Technology and Computer Science (INTECOMS)*, vol. 4, no. 2, p. 2021.
- [7] D. Sulisty, F. Ahda, and V. A. Fitria, "Epistemologi dalam Natural Language Processing," *Jurnal Inovasi Teknologi dan Edukasi Teknik*, vol. 1, no. 9, pp. 652–664, Sep. 2021, doi: 10.17977/um068v1i92021p652-664.
- [8] Y. Handrianto and B. Sanjaya, "Jatiwaringin Raya No. 18 Jakarta Timur."
- [9] I. Choldun and R. Rahmadewi, "Penerapan Metode Waterfall Pada Aplikasi Pembelajaran Seni Budaya Berbasis Website Menggunakan Framework Reactjs," *Jurnal Ilmiah Wahana Pendidikan, Juli*, vol. 2023, no. 13, pp. 335–348, doi: 10.5281/zenodo.8151254.
- [10] M. Rouf Akhmad Said Sekolah Tinggi Agama Islam Ma And Had H. Aly Al-Hikam Malang

-
- [11] Dedi Eko Riyadi, "Pengembangan Kurikulum Sekolah: Konsep, Model Dan Implementasi."
- [12] "Pengenalan Chatgpt Tips Dan Trik Bagi Pemula."
- [13] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, And J. Zou, "Gradio: Hassle-Free Sharing And Testing Of ML Models In The Wild," Jun. 2019, [Online]. Available: [Http://Arxiv.Org/Abs/1906.02569](http://Arxiv.Org/Abs/1906.02569)
- [14] Regina Lo *Et Al.*, "Penggunaan Bahasa Pemrograman Python Dalam Menganalisis Hubungan Kualitas Kopi Dengan Lokasi Pertanian Kopi," *Jurnal Publikasi Teknik Informatika*, Vol. 2, No. 2, Pp. 100–109, May 2023, Doi: 10.55606/Jupti.V2i2.1752.
- [15] W. Joni Kurniawan, "Sistem E-Learning Do'a Dan Iqro' Dalam Peningkatan Proses Pembelajaran Pada TK Amal Ikhlas," *Jurnal Mahasiswa Aplikasi Teknologi Komputer Dan Informasi*, Vol. 1, No. 3, Pp. 154–159, 2019.
- [16] S. Tita Faulina, M. Asia, J. A. Yani No, And S. Selatan, "Perancangan Sistem Informasi Kepegawaian Pondok Pesantren Darul Muttaqin Menggunakan Embarcadero XE2 Berbasis Client Server," 2021.
- [17] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, Q. A. Giansyah, And M. L. Hamzah, "Pengujian Black Box Dan White Box Sistem Informasi Parkir Berbasis Web Black Box And White Box Testing Of Web-Based Parking Information System."