ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika

ISSN(p): 2338-8323 | ISSN(e): 2459-9638 | Vol. 12 | No. 4 | Halaman 983 - 994
DOI : http://dx.doi.org/10.26760/elkomika.v12i4.983 Oktober 2024

Response Speed Analysis of Interactive Voicebot Receptionist

MUHAMMAD HAFIDZ UDZRI[1], NIZAR AINUL YAKIN[2] ,FIDA YUZIDA HASNAH[3], DENNY DARLIS[4], ERDIANSYAH REZAMELA[5], IFTITA FITRI[6], FARIS NUR FAUZI A[7], UNANG SUNARYA[8]

[1,2,3,4,8] School of Applied Sciences, Telkom University, Indonesia
[5,6,7] Research Institute for Tea and Cinchona, Gambung, Indonesia
Email: unangsunarya@telkomuniversity.ac.id

Received 1 September 2024 **|** *Revised* 30 September 2024 **|** *Accepted* 30 Oktober 2024

ABSTRAK

Pusat Penelitian Teh dan Kina (PPTK) Gambung menghadapi tantangan dalam memberikan informasi cepat kepada pengunjung. Untuk mengatasinya, dikembangkan voicebot interaktif berbasis teknologi Python Speech Recognition dan Pyttsx3, yang menggunakan metode *speech-to-text* dan *text-to-speech*. Pengujian dilakukan dengan variasi kondisi internet, intensitas kebisingan, perbedaan aksen, dan analisis keluaran suara. Hasil menunjukkan akurasi hingga 90% dengan rata-rata kecepatan tanggap 1,94 detik pada koneksi internet stabil dan suara yang jelas. Di lingkungan bising dengan kekuatan suara tinggi (105 dB), voicebot tetap mampu menanggapi. Voicebot ini juga menunjukkan akurasi yang sama (90%) untuk penutur asli dan non-asli bahasa Inggris. Solusi ini berpotensi meningkatkan aksesibilitas informasi di PPTK Gambung, meskipun kinerjanya dipengaruhi oleh kestabilan internet dan kondisi lingkungan.

Kata kunci*: voicebot, voice-to-text, text-to-speech, pyttsx3, speech recognition*

ABSTRACT

The Gambung Tea and Cinchona Research Center (PPTK) faces challenges in providing timely information to visitors. To address this, an interactive voicebot based on Python Speech Recognition technology and Pyttsx3 was developed, utilizing speech-to-text and text-to-speech methods. Tests were conducted with variations in internet conditions, noise intensity, accent differences, and voice output analysis. Results show accuracy of up to 90% with an average response speed of 1,94 seconds on a stable internet connection and clear voice. In noisy environments with high voice strength (105 dB), the voicebot was still able to respond. The voicebot also showed similar accuracy (90%) for native and non-native English speakers. This solution has the potential to improve information accessibility at PPTK Gambung, although its performance is affected by internet stability and environmental conditions.

Keywords*: voicebot, voice-to-text, text-to-speech, pyttsx3, speech recognition*

1. INTRODUCTION

Robots are machines programmed to perform specific functions, either autonomously or with human assistance. Today, robots are equipped with Natural Language Processing (NLP) and Artificial Intelligence (AI), enabling more sophisticated human-robot interactions, such as through voicebots **(Hameed, 2016)(Valerie et al., 2023)**. A voicebot is a program that performs simple automation tasks and can understand, interpret, and respond to a user's voice input **(Ionescu & Schlund, 2020)**.

Voicebots can be used in various fields, such as supporting children with attention deficit hyperactivity disorder (ADHD) in their daily activities, predicting behavior in e-commerce, and supporting automated counseling in healthcare **(Park et al., 2024)**. In fact, with recent advances, voicebots can receive input in the form of electrical signals from brain neurons through techniques such as pseudo-Wigner-Ville smooth distribution (SPWVD) and convolutional neural networks (CNN) **(Kamble et al., 2023)**. Voicebots continue to advance; however, they still encounter several limitations. This is mainly due to the speech recognition models used and the quality of their voice input.

The methodologies underlying speech-to-text and text-to-speech technologies play an important role in improving response accuracy. Moreover, these technologies are becoming key architectures in human-computer communication, especially in voicebot development **(Gonzales et al., 2024)**. With the support of these technologies, voicebots can understand the user's intent better. To further improve functionality and accuracy, effective domain-based QA prediction models, such as 1-layer Bi-GRU, can be used between speech-to-text and text-to-speech processes**(Swathi et al., 2024)**. However, these methods are often difficult to implement for users who have no experience in data science or machine learning, so they will find it difficult to create a reliable voicebot.

This research aims to develop an interactive voicebot using Python libraries for text-to-speech and speech-to-text technologies to enhance response accuracy. In the voice conversion process, this voicebot is built using speech-recognition and pyttsx3 libraries, MySQL as a database, and SQL queries for searching relevant data through "LIKE" and "SELECT" queries. The combination of these four systems could facilitate the development of interactive voicebots without requiring large computing power. To make voicebot development easier and more efficient, this research utilized existing libraries. This voicebot is expected to better meet the needs of users, as the system has been validated through user tests to ensure the quality of interaction and identify weaknesses that need to be improved.

2. MATERIAL AND METHODS

2.1 Quality Analysis Block Diagram of Voicebot Speed and accuracy.

Figure 2 shows the overall block diagram for voicebot speed and accuracy analysis with multiple tests. The system starts by entering data in the form of a voice captured by a microphone device then the voice is converted into text using the Speech-Recognition library. Next, the data is searched for answers that match the question using the SQL query "LIKE %text%" in the database. If no relevant match is found for the question in the database, the system redirects the query to Google for an answer. For comparison, a variety of internet conditions, noise intensity, accent differences, and voice output analysis were used to test the speed and accuracy of the voicebot.

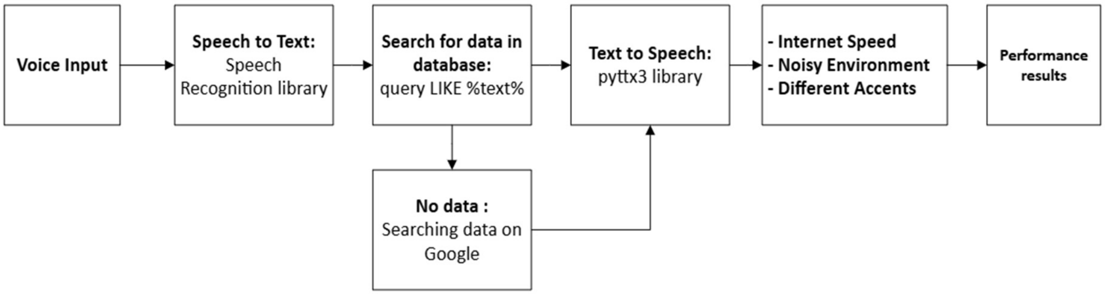
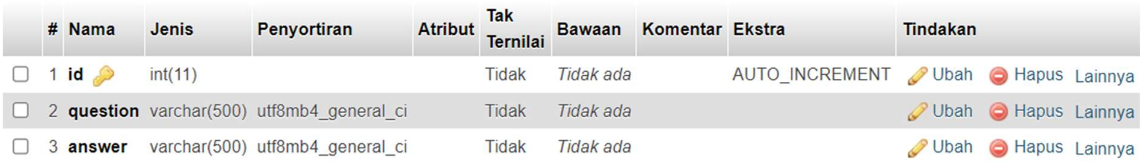

Voice Input
Speech to Text: Speech Recognition library
Search for data in database: query LIKE %text%
Text to Speech: pyttx3 library
- Internet Speed
- Noisy Environment
- Different Accents
Performance results
No data : Searching data on Google

Figure 1. Block Diagram of Voicebot Implementation

2.2 Datasheet

The data used for this research is data in the form of questions and answers that can be downloaded in SQL format at the address *https://drive.google.com/file/d/17U-GCeSOiV1EvEnAqbETW12qUZu4T4M/view?usp=sharing*. Data used for the sample there are 20 consisting of three columns, but the data used is only two columns, namely "question" and "answer". In this research, the data will be stored in a MySQL database, requiring the use of Structured Query Language (SQL) for data access. Figure 2 in the "question" column is used for comparison parameters between input questions in the form of voice, when comparing it turns out to have similarities, the data in the answer column is used as an answer that is issued in the form of voice.



	#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan		
☐	1	**id**	int(11)			Tidak	*Tidak ada*		AUTO_INCREMENT	Ubah	Hapus	Lainnya
☐	2	**question**	varchar(500)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah	Hapus	Lainnya
☐	3	**answer**	varchar(500)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah	Hapus	Lainnya

Figure 2. Tabel structure

2.3 Speech-Recognition Library (voice to text conversion)

The speech recognition library accepts voice input and converts it into text. This library works using an Application Programming Interface (API) system, so the computation is done on the API provider's server, not on the user's device. So, the user of this library needs to use the internet network. Converting voice into text has several stages so that the computer can understand what we say. At first, voice comes in the form of audio signals which are then converted into acoustic signals. Acoustic signals are very important in sound processing because the sound patterns become clearer and understandable to the machine **(Marinati et al., 2024)**. After that, the acoustic signal is mapped using an acoustic model to form phonemes (basic units of sound in language). Finally, the phonemes are processed by the language model to be mapped into words and sentences **(Alqadasi et al., 2024)**. The steps in implementing the speech recognition library include importing the library, initialization, and usage. Table 1 shows the usage of the speech recognition library and its explanation.

Table 1. Pseudocode for the usage of Speech Recognition Library

Pseudocode	Description
FUNCTION takeCommand(): INITIALIZE recognizer as sr.Recognizer() OPEN microphone as source PRINT "Listening..." SET pause_threshold to 1 LISTEN to audio from source TRY: PRINT "Recognizing..." SET query to result of recognize_google(audio, language='en-in') PRINT "User said: " + query EXCEPT Exception as e: PRINT "Say that again please..." WHILE true: CALL takeCommand()	○ takeCommand() : This function handles voice recognition. ○ initialize the voice recognition object (Recognizer). ○ use the microphone to listen to voice input. ○ set the pause time limit before speech recognition starts. ○ If an error occurs (for example, not recognizing the voice), a message will be displayed to prompt the user to repeat. ○ Looping :The program will keep running the takeCommand() function forever, so it is always ready to listen for new commands from the user.

2.4 SQL Query

To determine the answer contained in the "answer" column in the database table, it is necessary to match the question received from the voice input, with the question in the MySQL database in the "question" column. This use of MySQL requires SQL to access the data in the database table. When compared to databases that use NoSQL structures such as MongoDB, MySQL has a high speed in performing "SELECT" and "UPDATE" queries **(Naufal et al., 2022)**. Meanwhile, the use of databases with NoSQL structures is more suitable for storing real-time data, such as Internet of Things (IoT) data that has fast data input **(Kenitar et al., 2023)**.

To determine matches, SQL queries "LIKE" and "SELECT" are required. This query will search for data in the "question" column that matches the question input, if a similarity is found, the data will be retrieved and proceed to the text to voice conversion process. By using this query, it is expected to simplify the implementation of a simple voicebot. The following is an example of the SQL query used:

query = SELECT * FROM answer WHERE question LIKE '%*teks*%';

2.5 Pyttsx3 library (text to voice conversion)

pyttsx3 is the second library used in this research for text to speech conversion. Unlike alternative libraries, it works offline and is compatible with Python 2.0 and 3.0 **(Bhat, 2024)**. The text to be converted to voice is taken from the "answer" column in the database table. There are several important components in text to voice conversion, namely speech synthesis technology, text processing, and voice processing. Speech synthesis is used to produce sounds that apply human vocals, one method is Deep Learning-Based Synthesis **(Zhang et al., 2019)**, but it should be noted that sometimes speech synthesis is very influential in the text-to-voice conversion process because sometimes there is latency and poor quality of synthesized sound **(Saeki et al., 2021)**.

Table 2. Pseudocode for using the pyttsx3 library

Pseudocode	Description
INITIALIZE engine using pyttsx3 with 'sapi5' GET voices from engine GET current speech rate from engine SET voice to voices[1].id SET speech rate to 125 CALL engine.say("Hello world") CALL engine.runAndWait()	○ Engine Initialization: initialize the engine object using pyttsx3 with sapi5 settings. ○ Retrieving Voices: Get the list of available voices via engine.getProperty('voices'). ○ Retrieve Rate: Retrieve the current speech rate via engine.getProperty('rate'). ○ Set voice: set the voice to be used by selecting the second voice from the list (voices[1].id). ○ Adjust the speed of speech: adjust the speed of speech to 125. ○ Speak a sentence: call engine.say() to say "Hello World". ○ Running the engine: call engine.runAndWait() to run the voice command and wait for it to finish.

In text processing part, the text is converted into small units such as words or phonemes, which is the reverse of the speech-to-text process **(Chen et al., 2016)**. Finally, speech processing is the conversion of phonemes into sounds that are combined with speech synthesis components to produce sounds that are synchronized with the text. The implementation of the pyttsx3 library can be seen in the pseudocode in Table 2.

3. RESULTS AND DISCUSSION

3.1 Voice Output Quality Results

To determine the quality of the voice produced by the voicebot, proper analysis is required. One way to facilitate the analysis of voice output is to convert the text to be converted into voice, into a WAV file. The WAV file is then analyzed using two methods, namely by looking at the waveform and spectrogram.

The waveform analysis focuses on two parameters: amplitude (indicating intensity) and duration of pronunciation. Meanwhile, in the spectrogram, the two parameters analyzed are frequency and pronunciation time. For instance, to evaluate the voicebot's sound quality, text from the 'answer' column in the database ('This laboratory is known as the Greentech Laboratory, Faculty of Applied Sciences') is used.

Figure 3 shows the shape of the sound wave signal generated by the voicebot. Waveform analysis is conducted by examining the Y-axis, which indicates sound intensity at any given moment, the amplitude approaches a value of 1.00, indicating that higher amplitudes correspond to louder sound at specific points in time. The X-axis shows time (in seconds), the longer the sound wave range, the longer the audio duration.

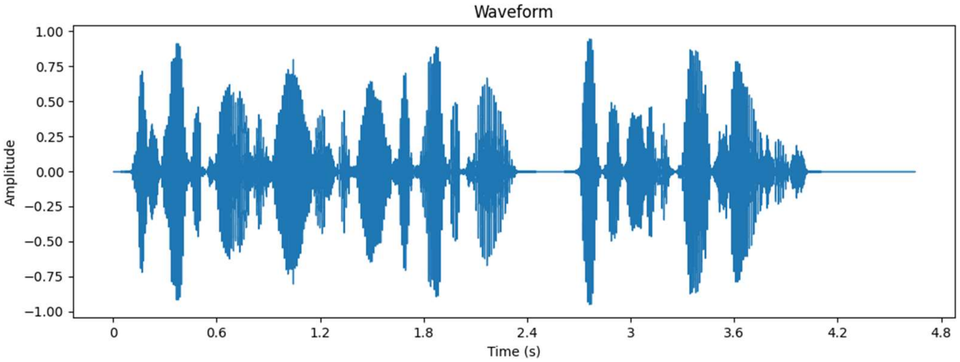


Figure 3. Waveform Result

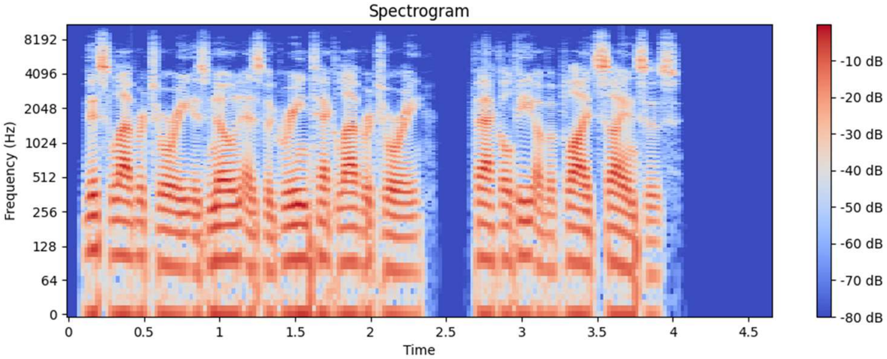
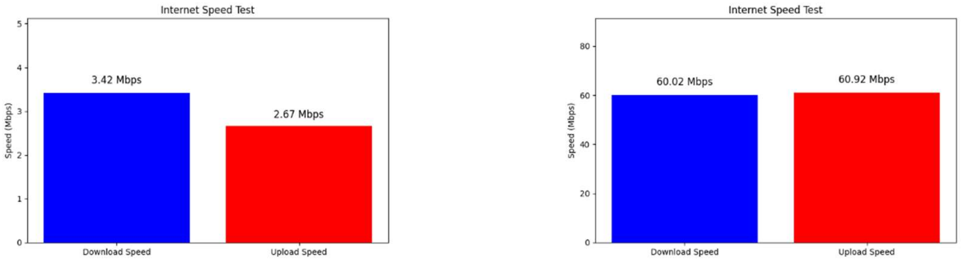


Figure 4. Spectogram Analysis Results

Figure 4 shows the shape of the spectrogram generated by the voicebot. In the spectrogram graph, there are two main parameters, namely frequency and time. The Y-axis shows the measurement based on frequency, the higher the frequency, the higher the tone produced. On the graph, there is a moment where the sound reaches the maximum limit of 8192 Hz, but the decibel of the sound is not high, which means the sound is not too loud.

3.2 Voicebot Accuracy Testing Based on Internet Speed and Stability

This test assesses the voicebot's accuracy by examining variations in internet speed and stability. Low and high internet speed conditions. This test was conducted because the speech recognition library requires internet access to connect to the application's speech recognition API. Figure 5(a) shows the first internet speed, with a download speed of 3.42 Mbps and an upload speed of 2.67 Mbps; Figure 5(b) shows the second internet speed, with a download speed of 60.02 Mbps and an upload speed of 60.92 Mbps. During the test, the voicebot device was connected to these two types of internet speeds consecutively. The results show that there are variations in accuracy and response speed.



(a) **(b)**

Figure 5. Internet Speed for Testing: (A) Low speed and (B) High Speed

During the first test, the device was connected to a smartphone hotspot network with download and upload speeds of 3.42 Mbps and 2.67 Mbps, respectively. Figure 6 shows the results of 10 trials of running the voicebot, where there were three failures to respond. Thus, the accuracy of the first test was 70%, and the average response time was 3.6 seconds. This is due to the unstable and low-speed internet network. The accuracy may continue to decrease if the internet stability and speed remain low.

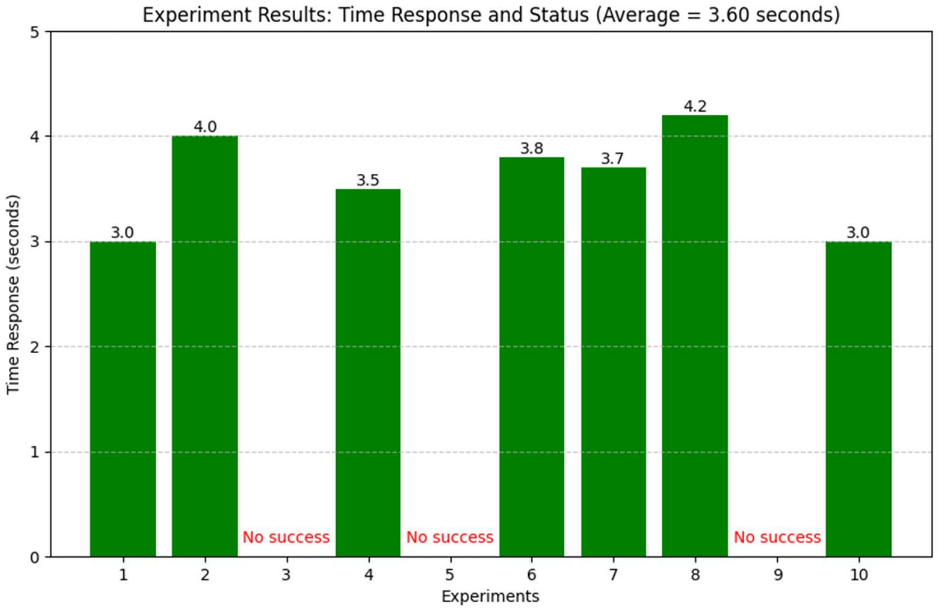


Figure 6. Voicebot Testing Results with Low Internet Speeds

In the second test, the device used a Wi-Fi router network with internet speeds of 60.02 Mbps for download and 60.92 Mbps for upload. Figure 7 shows the results of another 10 trials with much higher speeds, where only one failure occurred. With these results, the accuracy of the second test reached 90%, and the average response time was 1.94 seconds. This accuracy can be improved with better internet stability and speed.

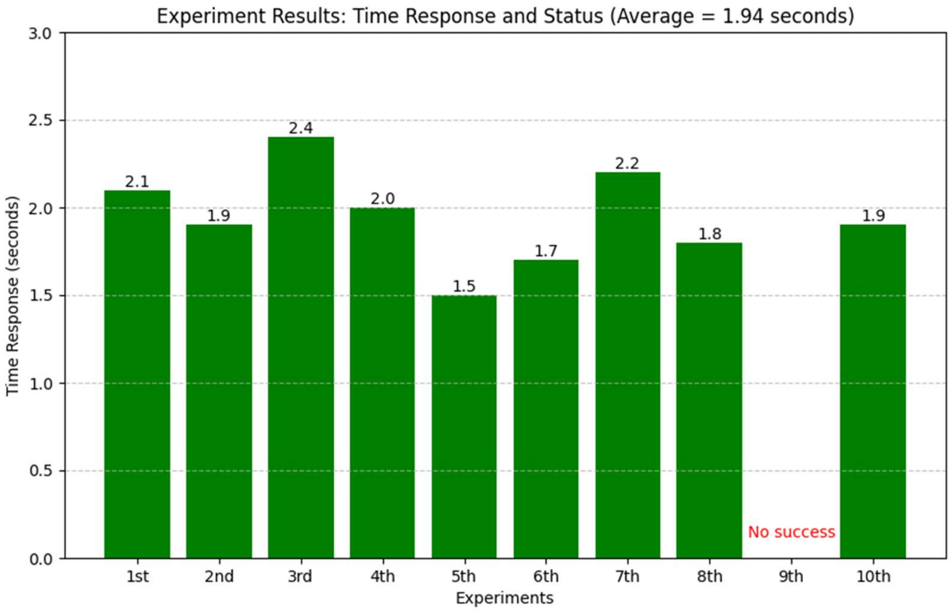


Figure 7. Voicebot Testing Results with High Internet Speeds

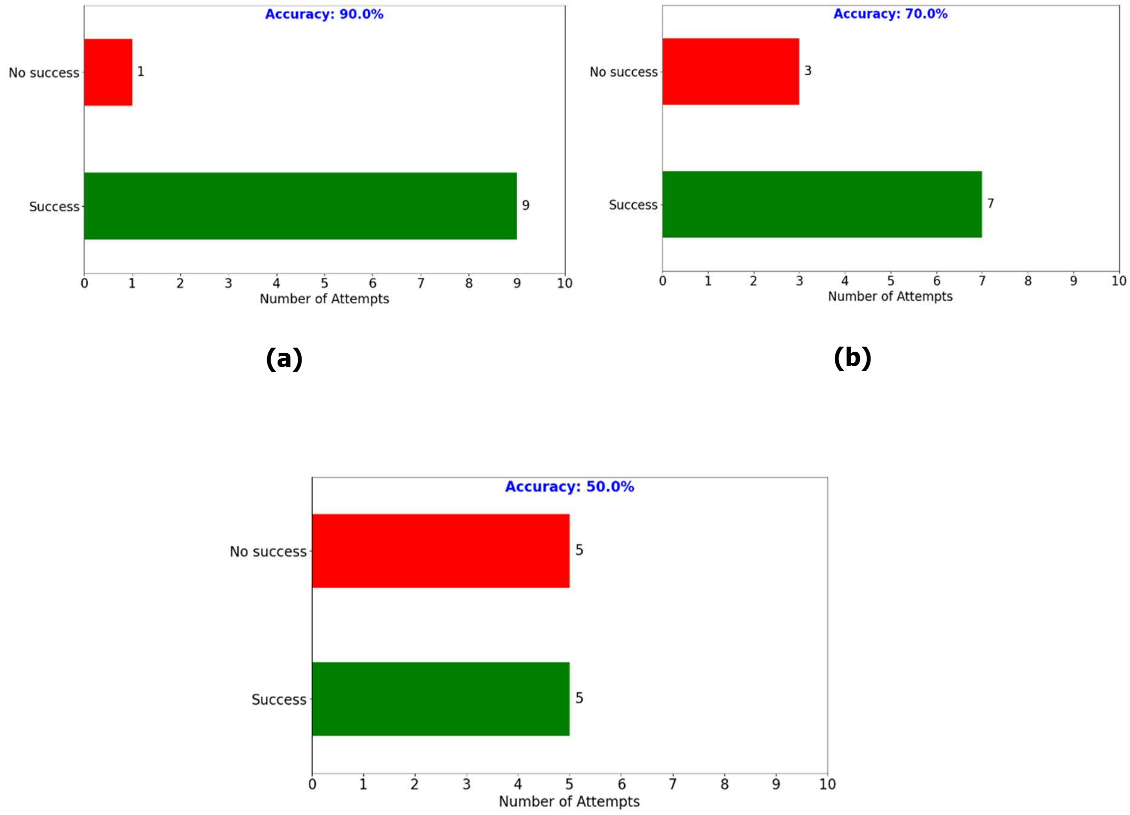
3.3 Comparative Testing of Responses in a Noisy Environment

In Table 3 of this voicebot testing phase, three types of noise interference with different decibel levels were prepared, namely casual conversation noise with a sound intensity of 80 dB, crowd noise with an intensity of 90 dB, and rock concert noise with an intensity of 105 dB. The noise disturbance testing was not conducted in real conditions; in this study, simulations were used through videos that have been measured according to their noise levels in decibel (dB) units. Each noise interference is tested in 10 trials to measure the accuracy and maximum limit of the voicebot in receiving voice input in noise conditions. To support voicebot testing, a stable and high-speed internet connection is used.

Table 3. Types of Noise for the Testing

No.	Noise Type	Sound Intensity (dB)
1.	Casual Conversation	80 dB
2.	Crowd Noise	90 dB
3.	Rock Concert	105 dB

Figure 8(a) shows the results of the first test of the voicebot under conditions of casual conversation noise with an intensity of 80 dB, where out of 10 trials there was only one failure, resulting in an accuracy of 90% which is equivalent to testing on a stable internet connection and shows that with 80 dB noise interference, the voicebot is still able to respond to answers well. The second test result, shown in Figure 1(b), was conducted with 90 dB intensity crowd noise; the results show that out of 10 trials there were three failures with 70% accuracy, indicating that when the voicebot receives 90 dB noise interference, its accuracy starts to degrade. In the third test, the voicebot was tested with 105 dB rock concert noise, as shown in Figure 1(c); although the voicebot was still able to respond, out of 10 trials only five were successful with 50% accuracy, indicating a drastic decrease in accuracy as 105 dB noise significantly affects the performance of the voicebot.



(a) **(b)**

(c)

Figure 8. Voicebot Accuracy Results Against the Effect of Noise: (a) casual conversation noise, (b) crowd noise, (c) rock concert noise

3.4 Testing with Different Accents and Intonations

In this research, the voicebot uses English to communicate. To support the implementation of voicebot in PPTK Gambung, which is usually visited by many people and has different accents in English pronunciation, this test will involve two types of speakers: native English speakers and non-native speakers who are Indonesian. The native speakers in this study will use Google Translate simulation to pronounce the tested phrases. Each speaker will be tested 10 times with two different intonation speeds, namely high and low. With this test, it is expected to know whether the response to speaker differences will affect the accuracy of the voicebot.

Figure 9 shows the accuracy results obtained in the first experiment using native English speakers. Out of 10 trials, there were only two failures with fast intonation. Meanwhile, for the results with low intonation, there was only one failure in the experiment. Thus, the accuracy obtained for fast intonation is 80%, while for low intonation it reaches 90%.

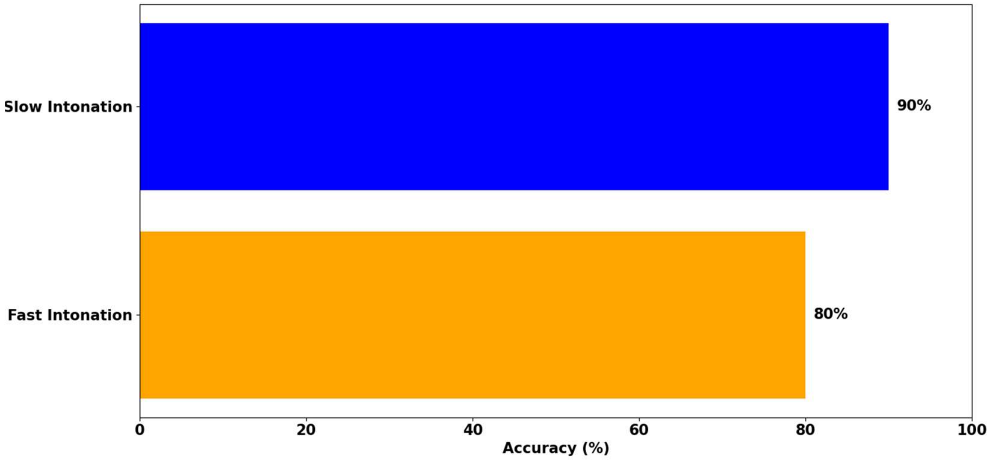


Figure 9. Comparison of Fast and Slow Intonation Accuracy in English Native Speakers

For the second experiment, which can be seen in Figure 10, a non-native English speaker who is Indonesian was used. The results show that in 10 trials, there were only three failures with fast intonation. For the results with low intonation, there was only one failure in the experiment. Thus, the accuracy obtained for fast intonation was 70%, and 90% for low intonation. This accuracy shows similarity in low intonation, so there is no problem if the speaker is not a native speaker.

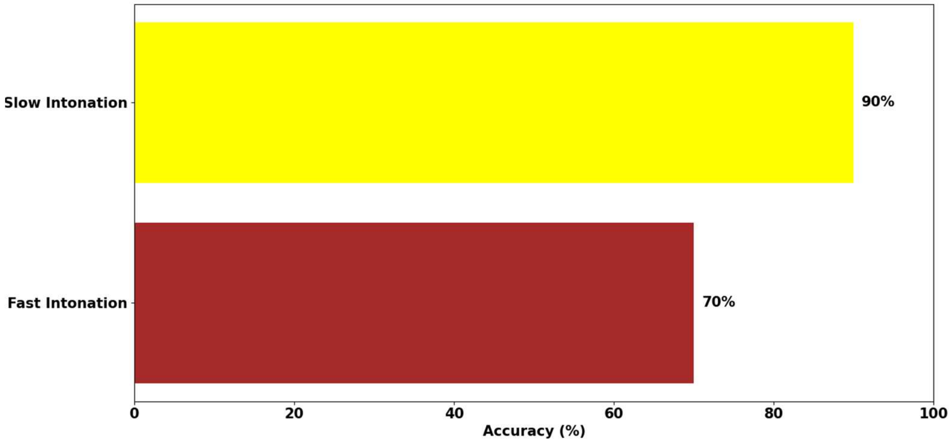


Figure 10. Comparison of Fast and Slow Intonation Accuracy in Non-Native English Speakers – Indonesian

4. CONCLUSIONS

Based on the test results of this study, the voicebot showed good performance in providing information to visitors, with accuracy reaching 90% on a fast and stable internet connection, while accuracy decreased to 70% on a slow connection. Voice quality analysis demonstrated clear output, and testing in noisy environments achieved 90% accuracy at 80 dB noise levels. At higher noise levels (105 dB), the voicebot could still respond, although accuracy decreased. In addition, the voicebot functioned well despite accent differences, with native English speakers and non-native speakers showing similar accuracy. These results suggest that this voicebot has the potential to improve information accessibility at PPTK Gambung, although its performance remains affected by environmental conditions and internet signal quality.

ACKNOWLEDGMENTS

The research with the theme Comparative Analysis of Interactive Voiceobot Response Speed at PPTK Gambung is supported by the Directorate of Research and Community Service (PPM) No. KWR4.109/LIT03/PPM-LIT/2023, Telkom University Bandung, Indonesia.

REFERENCES

Alqadasi, A. M. A., Zeki, A. M., Sunar, M. S., Salam, M. S. B. H., Abdulghafor, R., & Khaled, N. A. (2024). Improving Automatic Forced Alignment for Phoneme Segmentation in Quranic Recitation. *IEEE Access*, *12*, 229–244.

Bhat, N. M. (2024, September 27). *pyttsx3 2.98*. Retrieved from https://pypi.org/project/pyttsx3/

Chen, L., Mao, X., & Yan, H. (2016). Text-Independent Phoneme Segmentation Combining EGG and Speech Data. *IEEE/ACM Transactions on Audio Speech and Language Processing*, *24*(6), 1029–1037.

Gonzales, M. G., Corcoran, P., Harte, N., & Schukat, M. (2024). Joint Speech-Text Embeddings for Multitask Speech Processing. *IEEE Access*, *12*, 145955–145967.

Hameed, I. A. (2016). Using natural language processing (NLP) for designing socially intelligent robots. *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, (pp. 268–269).

Ionescu, T. B., & Schlund, S. (2020). Programming cobots by voice: A human-centered, web-based approach. *Procedia CIRP*, *97*, 123–129.

Kamble, A., Ghare, P. H., & Kumar, V. (2023). Deep-Learning-Based BCI for Automatic Imagined Speech Recognition Using SPWVD. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–10

Kenitar, S. B., Arioua, M., & Yahyaoui, M. (2023). A Novel Approach of Latency and Energy Efficiency Analysis of IIoT With SQL and NoSQL Databases Communication. *IEEE Access*, *11*, 129247–129257.

Marinati, R., Coelho, R., & Zao, L. (2024). FRS: Adaptive Score for Improving Acoustic Source Classification from Noisy Signals. *IEEE Signal Processing Letters*, *31*, 671–675.

Naufal, N., Nurkhodijah, S., Anugrah, G. B., Pratama, A., Rabbani, M. I., Dilla, F. A., Anggraeni, T. N., & Firmansyah, R. (2022). Analisa Perbandingan Kinerja Response Time Query Mysql Dan Mongodb. Jurnal Informatika Dan Teknologi Komputer, *2*(2), 158–166.

Park, D. E., Lee, J., Han, J., Kim, J., & Shin, Y. J. (2024). A Preliminary Study of Voicebot to Assist ADHD Children in Performing Daily Tasks. *International Journal of Human–Computer Interaction*, *40*(10), 2711–2724.

Saeki, T., Takamichi, S., & Saruwatari, H. (2021). Incremental Text-to-Speech Synthesis Using Pseudo Lookahead with Large Pretrained Language Model. *IEEE Signal Processing Letters*, *28*, 857–861.

Swathi, B. P., Geetha, M., Attigeri, G., Suhas, M. V., & Halaharvi, S. (2024). Optimizing Question Answering Systems in Education: Addressing Domain-Specific Challenges. *IEEE Access*, *12*, 156572–156587.

Valerie, M., Salamah, I., & Lindawati. (2023). Innovative Personal Assistance: Speech Recognition and NLP-Driven Robot Prototype. *Jurnal Nasional Teknik Elektro*, *12*(2), 181–187.

Zhang, W., Yang, H., Bu, X., & Wang, L. (2019). Deep Learning for Mandarin-Tibetan Cross-Lingual Speech Synthesis. *IEEE Access*, *7*, 167884–167894.