Review Article

Python for Kids: Designing an Educational Literacy Application

[1,2,*]Tjahjo Adiprabowo, [1]Dani Ramdani, [2]Septian Maulana, [1]Sansan Nurdiansyah

[1] Department of Electrical Engineering, Langlangbuana University, Bandung, West Java, Indonesia
[2] Department of Communication Studies, Langlangbuana University, Bandung, Indonesia

* Corresponding Author: tjahjo.adiprabowo@unla.ac.id

Abstract: The ability to read and write (Literacy skill) is an essential thing that needs to be known early by children. Children can build applications such as simple games, desktop applications, and websites, several programming languages support the development of children's thinking skills such as C++ on Arduino boards, Raspberry Pi, and other Microcontrollers. Python and MicroPython are also easy to learn, as well as the development of Artificial Intelligence (AI) and its Platform which is so rich in libraries, applications, and other applications. In creating applications with certain extensions, compatibility, and also speed when applied to all PCs in general. This research discusses comprehensively how to build a simple Python-based application by converting voice and text, and this application can be applied to the general public, especially for Kids and also disabilities.

Keywords: Educational, literacy, Computer application, Python Programming, Kids, GUI

1. Introduction

Indonesia is a vast country, one of the things that until now in 2025 is education that is still uneven in areas such as eastern Indonesia, for example, Papua and East Nusa Tenggara, although there have been many government activities that try to improve the standard of living of the people there, especially education, but still need to be improved in terms of the quality of education. Educators also need to be certified to be able to teach according to the teaching criteria and curriculum. The main problem that still occurs and is experienced by some elementary schools in Indonesia is the delay in mastering language, reading, and writing. Several ways have been done to accelerate language and writing skills.

Moreover, in the current Prabowo-Gibran administration, many things are promoted, including free lunches and also curriculum reform for elementary schools, namely the implementation of programming languages or coding as a strong foundation for children to become computerized and capable programmers. This paper is one of the results of research where the program creation process is made using the Python programming language which is easy for children and can be applied and tested to help children learn to write, listen, and also interact, do trial and error on python programs and also learn debugging.

In this research, we will design a simple program that reads or converts text to voice and vice versa. The coding of this program will then be submitted to elementary schools where Langlangbuana University Bandung students conduct fieldwork practices or Real Work Lectures (KKN), as a form of application of a tool or system for the community as one of the student contributions to the community in real terms. One of the problems found by the Rector of Langlangbuana University when visiting KKN students in one of the elementary schools (SD) in Soreang, Bandung Regency, West Java is the lack of reading and writing skills. This problem is experienced by most students who are in class 3 of Karamat Mulya Elementary School, Soreang. This is a big homework for all KKN students from Langlangbuana University to unite in completing this small project, for the welfare of the community, especially elementary school children in helping to accelerate their reading and writing skills.

check for updates

Citation: Adiprabowo, T., Ramdani, D., Maulana, S., & Nurdiansyah, S. (2025). Python for kids: Designing an educational literacy application. Iota, 5(1). https://doi.org/10.31763/iota.v5i1.809

Academic Editor: Adi, P.D.P

Received: August 11, 2024

Accepted: November 21, 2024

Published: February 01, 2025

2. Literature Review

2.1 Python Programming Language

In its journey until now, Python continues to be one of the most popular open-source programming languages for programmers and many people who are beginning to learn this programming language. Previously there were C, Pascal, and other programming languages that were difficult to understand, but if mastery starts from these programming languages, it will accelerate the understanding of the Python programming language. Python is a high-level language that will make it easier for users to understand and also comes with the choice of whether Command Prompt or GUI like Visual Studio Code which can be installed for free.

Python programming language is simple compared to C++, Python and its derivative MicroPython can be applied to embedded systems to build Internet of Things (IoT) applications for example. Python is perfect for beginners and can also be developed to work on various projects that are very useful at this time, such as Artificial Intelligence (AI), automation, WEB development, and software development.

Python supports features such as data analysis, artificial intelligence, WEB development, automation, and software development. Python is equipped with code blocks that are easy to understand and also neat and Python supports object-oriented programming (OOP), functional programming, and imperative programming. Python is also supported by an extensive library to accelerate application development, Python is flexible and can be run on various platforms such as Windows, macOS, and Linux. Applications that support software development such as data science, machine learning, and Deep Learning [19].

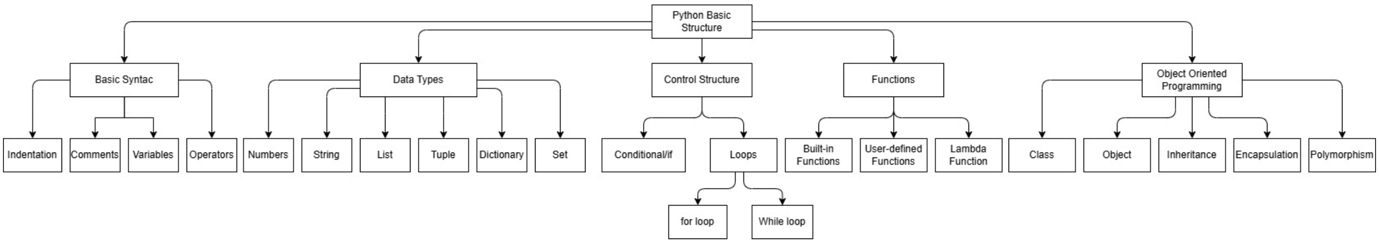


Figure 1. Python Basic Structure

Moreover, from Figure 1, we can see the structure of the Python programming language starting from Syntac, data types, structure control, functions, and OOP. With OOP Python can accomplish include Class, Object, Inheritance, Encapsulation, and Polymorphism. In detail, to practice Python Programming, it is necessary to continuously practice and also trial and error, if there is an error, you can investigate the location of the error in Syntac, or errors in identifying a program, program punctuation, and other errors. From this, students can also be trained to be creative in self-development through the Python programming language.

2.2 Graphical User Interface

Furthermore, if computers in the past such as MS.DOS used a black layer, even used in some machines withdrawing money at ATMs, this is called a Non-Graphical User Interface (Non-GUI) because it only runs on the Command Prompt platform or black layer, although there are several commands, all based on text, we input a text or syntax program, then we will switch automatically. But this is time-consuming, less dynamic, less interesting, and less flexible, so GUI-based emerged.

GUI (Graphical User Interface) is an interface that is a graphical element such as icons, buttons, menus, and even some images that attract the attention of users so that users can interact in the system properly. The opposite of GUI is Non-GUI or CLI (Command Line Interface), Where each command must be typed first before moving to another facility or menu. Operating systems that have used the GUI are Windows, macOS, and Android, as well as Linux not all use the CLI, some of the initial Linux Ubuntu displays, for example, use the GUI to enter certain menus.

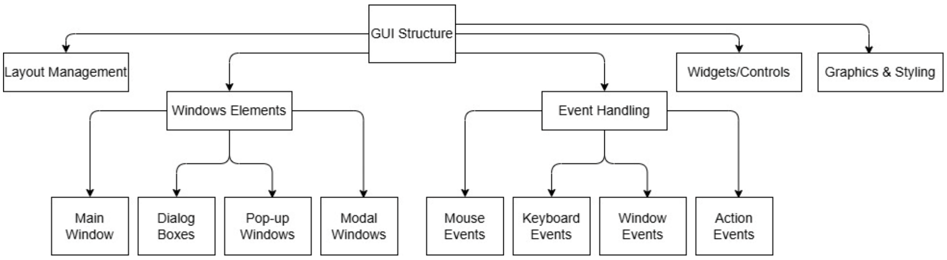


Figure 2. GUI Structure 1

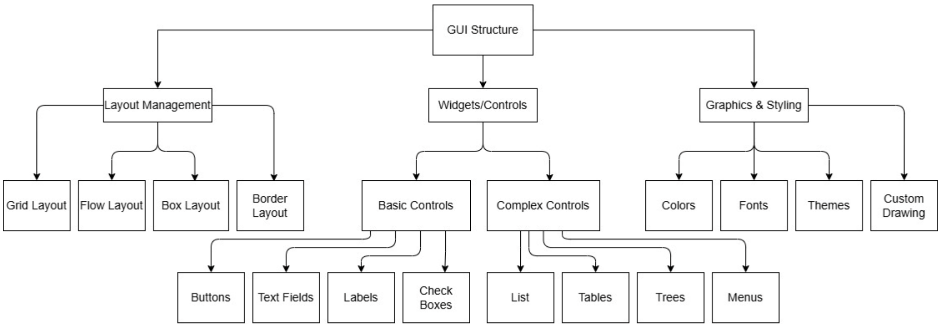


Figure 3. GUI Structure 2

Moreover, from Figures 2 and 3 it can be seen that the GUI structure is so complex, that each of these elements needs to be achieved so that the GUI can be built properly, starting from Windows Elements, Layout Elements, Event Handling, Widgets/Controls, and Graphics & Styling, and each of these elements has components that are related and complement each other into a unified system that builds the GUI to the maximum.

3. Method

3.1 *Conversion Technique*

In this paper we will discuss the method of converting Text to Speech System [1,2,3] and Speech to Text System using Python Programming, Flowchart System in detail can be seen in Figure 4 and Figure 5. The process of converting text to audio through several stages, namely text processing which consists of normalization and tokenization, then proceed with text analysis which consists of part of speech and syllable structure, then proceed with phonetic analysis which consists of phoneme sequence and pronunciation [16,17,18,20], then proceed with prosody generation which consists of pitch, duration, and intensity, then proceed with waveform synthesis which consists of concatenative and neural TTS, then from these results audio output is produced. Python programming language with several libraries can produce audio output from text input.

Moreover, the conversion of speech to to-text system starts from the input audio which then goes through the audio preprocessing process which consists of Noise reduction and signal enhancement [4,5,6], then continues the feature extraction process which consists of MFCC and Spectogram, then continues with Acoustic Model Processing which consists of Neural Network and Hidden Markov Model, then continued with Language Model Processing which consists of Grammar rules and Word sequence, then the Decode & Recognition process, bypassing these stages the output text can be created.

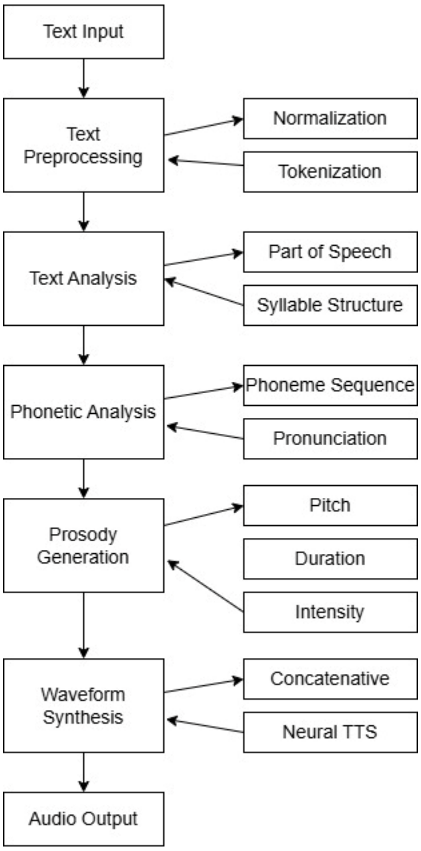


Figure 4. Text to Speech System

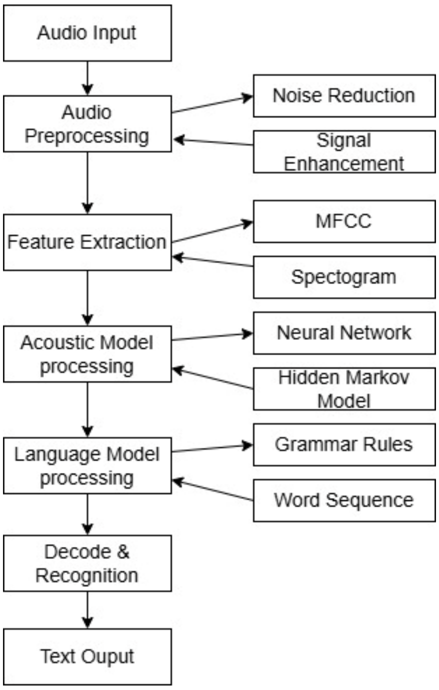


Figure 5. Speech to Text System

3.2 *Hidden Markov Model*

3.2.1 *Speech-to-Text (Speech Recognition)*

This Hidden Markov Model (HMM) works by converting starting Speech to text recognition by doing the Pro-processing step which is converting the raw sound signal with the noise cleaning and normalization process. Next is Feature extraction which is converting the sound signal into MFCC (Mel-Frequency Cepstral Coefficients), this feature represents the spectral characteristics of sound in the frequency domain. The next process is HMM Training which is the process of training data consisting of audio text and representing it to sound units usually phonemes. The model learns the probability of transition between states and the probability of feature emission. Moreover, the next process is Decoding which uses Viterbi Algorithm to find the most likely sequence state and combine it with the language model to produce a plausible text.

3.2.2 *Text-to-Speech (Speech Synthesis)*

The next process is Text-to-speech [7,8,9], starting with the recognition of the text, which converts the text to the predicted sequence of phonemes and prosody (intonation, stress, duration). The next step is HMM Synthesis, which is the process of converting phonemes to a sequence of HMM states and then changing the acoustic parameters (pitch, spectrum) generated from the model. The next process is Waveform generation, which is the process of changing acoustic parameters converted to sound signals [10,11,12], and then a vocoder is used to produce natural sounds [13,14,15]. The details of the Hidden Markov Model are shown in Figure 6.

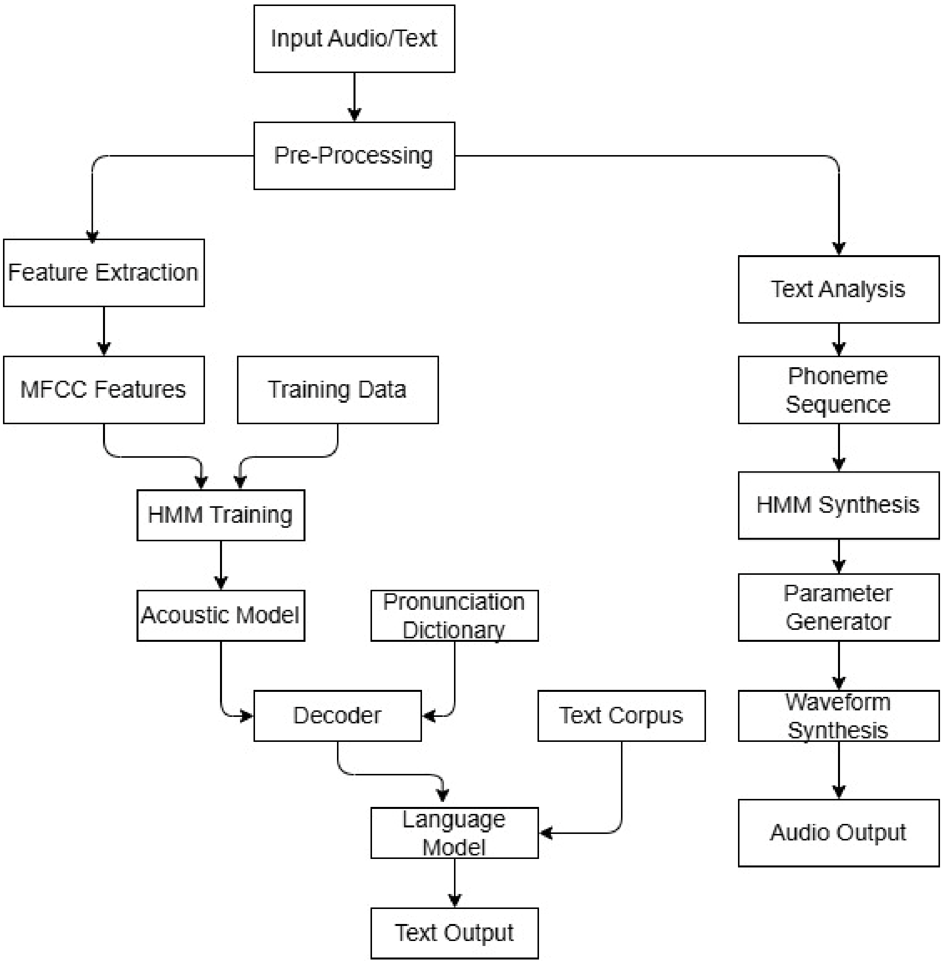


Figure 6. Hidden Markov Model

4. Result and Analyzes

4.1 Direct approach and data collection

The next step is to collect data directly, namely by conducting a direct survey at Karamat Mulya Elementary School as shown in Figure 7, and entering the classroom. In this case, the Practical Students directly asked a question that had to do with reading and writing. From the survey results and data collection, the value is obtained as shown in Figure 8.

Moreover, from the data collection in Figure 8, it is quite clear that there are children who cannot read and write as shown in class 3 of Karamat Mulya Primary School. From the data, 10 children still get less than equal to 6 for their score, while 11 children can read and write. The highest score is 9, obtained by 2 students, a score of 8 is obtained by 3 students, and a score of 7 is the most by 6 children. In the next step, the students coded a program from Python and then gave it to the school, namely 1 computer with Speech recognition coding added with a Graphical User Interface (GUI) so that elementary students would find it easier to do the conversion process.

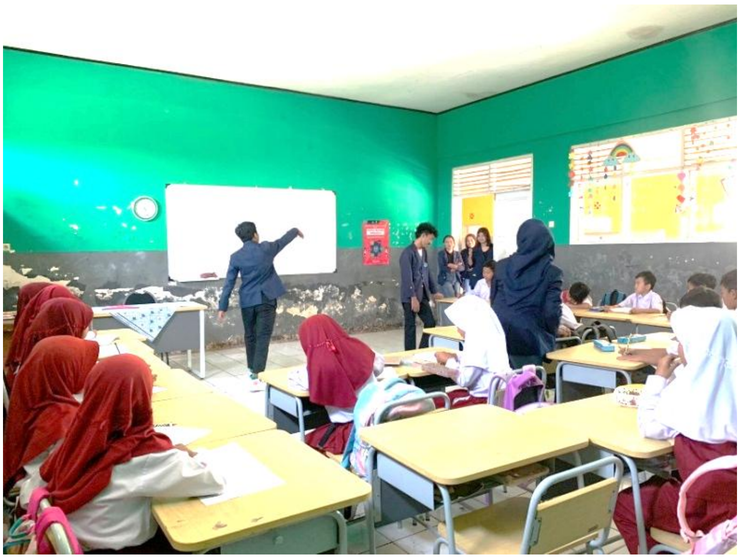


Figure 7. Literacy testing in Karamat Mulya Elementary School, Soreang, West Java

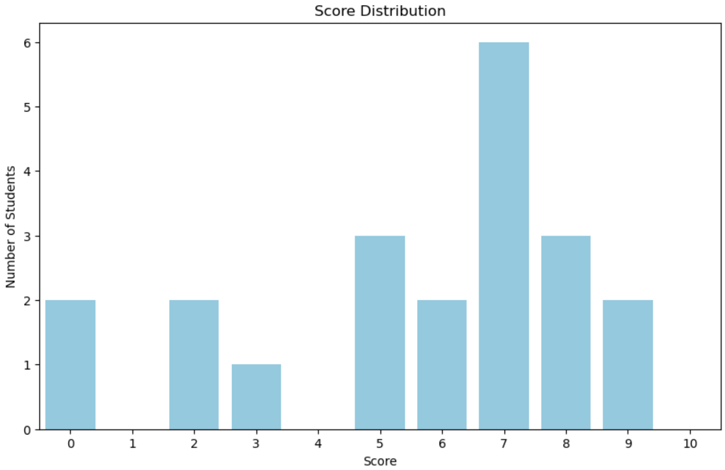
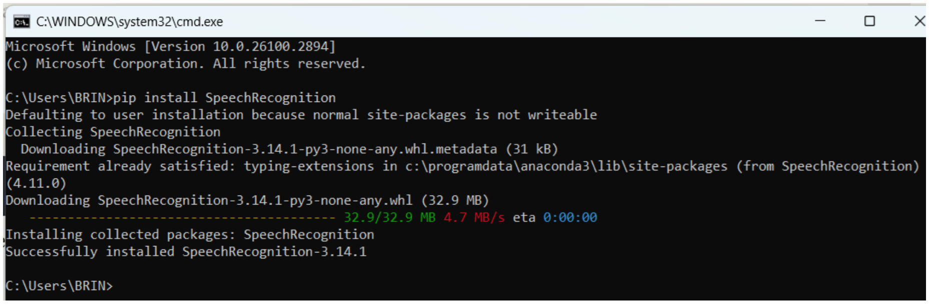


Figure 8. The resulting value of Literacy testing

4.2 Data Conversion Processing using Python

Furthermore, the main step in running the SpeechRecognition program is the installation of the Library. As shown in Figure 9.



C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.26100.2894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\BRIN>pip install SpeechRecognition
Defaulting to user installation because normal site-packages is not writeable
Collecting SpeechRecognition
  Downloading SpeechRecognition-3.14.1-py3-none-any.whl.metadata (31 kB)
Requirement already satisfied: typing-extensions in c:\programdata\anaconda3\lib\site-packages (from SpeechRecognition)
(4.11.0)
Downloading SpeechRecognition-3.14.1-py3-none-any.whl (32.9 MB)
   ---------------------------------------- 32.9/32.9 MB 4.7 MB/s eta 0:00:00
Installing collected packages: SpeechRecognition
Successfully installed SpeechRecognition-3.14.1

C:\Users\BRIN>
```

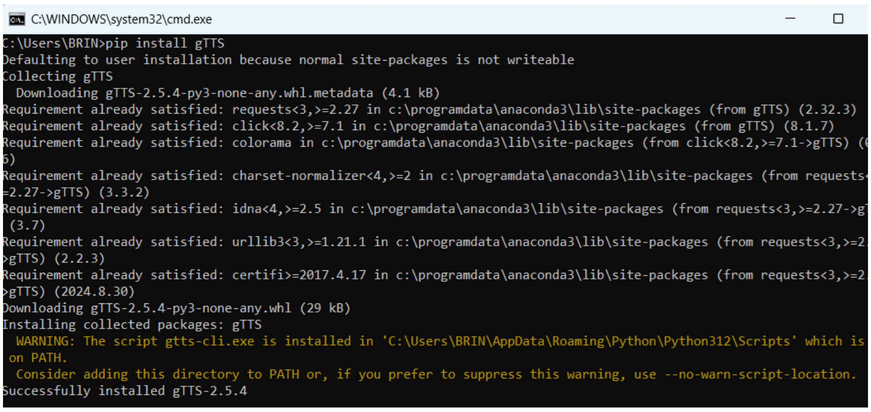
Figure 9. SpeechRecognition Library in Python

Moreover, after the SpeechRecognition installation process is successful, the next step is to run the SpeechRecognition and TextRecognition programs that have been built using the Python Language. For SpeechRecognition coding and vice versa can be seen in Pseudocode 1 and 2.

```
ALGORITHM SpeechToText – [ Pseudocode 1 ]

1.  Global Variable
DECLARE recognizer
DECLARE root_window
DECLARE status_label
DECLARE hasil_text
DECLARE hasil_label

2.  The main function of converting voice to text
      PROCEDURE suara_ke_teks()
    BEGIN
        CREATE new recognizer object
        TRY
            OPEN microphone as audio_source
                SET status_label text = "Mendengarkan..."
                UPDATE root_window display

          ADJUST recognizer for ambient noise using audio_source
                RECORD audio from audio_source

          CONVERT audio to text using Google Speech Recognition
(Indonesian)
           SET hasil_text = "Anda mengatakan: " + converted_text
           SET status_label text = "Selesai"

        CATCH UnknownValueError
           SET hasil_text = "Google Speech Recognition tidak dapat
memahami audio."
           SET status_label text = "Kesalahan"

        CATCH RequestError as error
           SET hasil_text = "Permintaan gagal: " + error_message
           SET status_label text = "Kesalahan"
    END
```

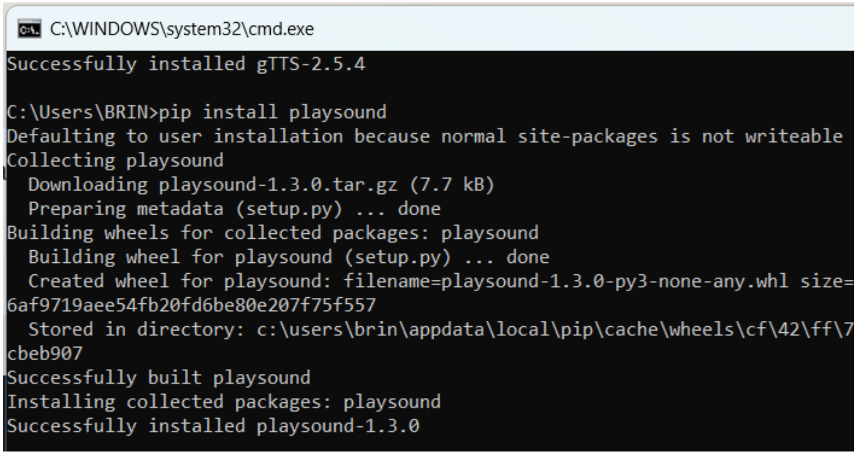
```
3.  Handler function for a start button
    PROCEDURE on_start_button_click()
    BEGIN
        CALL suara_ke_teks()
    END

4.  Main Program
    PROCEDURE Main()
    BEGIN
        4.1 Main window initialization
        CREATE main window as root_window
        SET window title = "Suara ke Teks"

        4.2 Creating a frame
        CREATE frame in root_window
        SET frame padding = 40 pixels

        4.3 Creating GUI Components
        CREATE label as status_label
        SET status_label text = "Klik tombol untuk mulai merekam"
        SET status_label vertical padding = 40 pixels

        CREATE button as start_button
        SET button text = "Mulai Rekam"
        SET button command = on_start_button_click
        SET button vertical padding = 40 pixels

        CREATE variable hasil_text
        CREATE label bound to hasil_text
        SET label word wrap = 1000 pixels
        SET label vertical padding = 40 pixels

        4.4 Running the app
        START main event loop
    END

5.  Starting the program
    CALL Main()
```

Furthermore, Pseudocode 2 shows the conversion process from text to voice. It is necessary to install gTTS beforehand.



```
C:\Users\BRIN>pip install gTTS
Defaulting to user installation because normal site-packages is not writeable
Collecting gTTS
  Downloading gTTS-2.5.4-py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: requests<3,>=2.27 in c:\programdata\anaconda3\lib\site-packages (from gTTS) (2.32.3)
Requirement already satisfied: click<8.2,>=7.1 in c:\programdata\anaconda3\lib\site-packages (from gTTS) (8.1.7)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-packages (from click<8.2,>=7.1->gTTS) (
6)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests
=2.27->gTTS) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.27->g
 (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2
>gTTS) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2
>gTTS) (2024.8.30)
Downloading gTTS-2.5.4-py3-none-any.whl (29 kB)
Installing collected packages: gTTS
  WARNING: The script gtts-cli.exe is installed in 'C:\Users\BRIN\AppData\Roaming\Python\Python312\Scripts' which is
 on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed gTTS-2.5.4
```

Figure 10. Text to Speech Library in Python, i.e. gTTS



C:\WINDOWS\system32\cmd.exe

```
Successfully installed gTTS-2.5.4

C:\Users\BRIN>pip install playsound
Defaulting to user installation because normal site-packages is not writeable
Collecting playsound
  Downloading playsound-1.3.0.tar.gz (7.7 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: playsound
  Building wheel for playsound (setup.py) ... done
  Created wheel for playsound: filename=playsound-1.3.0-py3-none-any.whl size=
6af9719aee54fb20fd6be80e207f75f557
  Stored in directory: c:\users\brin\appdata\local\pip\cache\wheels\cf\42\ff\7
cbeb907
Successfully built playsound
Installing collected packages: playsound
Successfully installed playsound-1.3.0
```

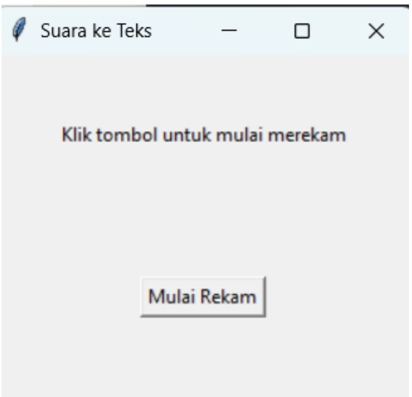
Figure 11. Text to Speech Library in Python, i.e. play sound

```
ALGORITHM TextToSpeech

1.  Global Variable
DECLARE root_window
DECLARE entry_teks
DECLARE entry_direktori
DECLARE entry_nama_file

2.  Function to convert text to voice
PROCEDURE teks_ke_suara(teks, direktori, nama_file)
   BEGIN
       CREATE path object from direktori

       IF the directory doesn't exist THEN
           CREATE directory with parents
       END IF

       SET full_path = combine directory path and filename

3.  Text-to-audio conversion
       CREATE gTTS object with:
           text = teks
           language = "Indonesian"

       SAVE audio to full_path
       PRINT "File audio telah disimpan sebagai " + full_path

       PLAY audio file from full_path
   END

4.  Function to select a directory
PROCEDURE pilih_direktori()
   BEGIN
       OPEN directory selection dialog
       IF directory selected THEN
           CLEAR entry_direktori
           INSERT selected directory to entry_direktori
       END IF
   END
```

```
5.  Functions to handle conversion
PROCEDURE konversi_teks()
   BEGIN
       GET text from entry_teks
       GET directory from entry_direktori
       GET filename from entry_nama_file

       IF text AND directory AND filename are not empty THEN
           CALL teks_ke_suara(text, directory, filename)
           SHOW info message "Konversi selesai dan audio telah diputar!"
       ELSE
           SHOW warning message "Semua field harus diisi!"
       END IF
   END

6.  Main Program
PROCEDURE Main()
   BEGIN
       6.1 Main window initialization
       CREATE main window as root_window
       SET window title = "Teks ke Suara"

       6.2 Create GUI components
       CREATE label "Masukkan teks:"
       CREATE entry field as entry_teks with width 50

       CREATE label "Pilih direktori untuk menyimpan:"
       CREATE entry field as entry_direktori with width 50
       CREATE button "Browse" with command pilih_direktori

       CREATE label "Nama file: tambahkan *.(mp3)"
       CREATE entry field as entry_nama_file with width 50

       CREATE button "Konversi ke Suara" with command konversi_teks

       6.3 Set the padding for all components
       SET padding x=10, y=5 for all components

       6.4 Running the app
       START main event loop
   END

7.   Starting the program
CALL Main()
```

4.3 *Graphical User Interface of the Conversion System*

Furthermore, here is the result of running the program, the Speech to Text GUI, to start you can click the Start Record button. Furthermore, Figure 12, will continue with the display in Figure 13, namely Listening to sound. The results can be seen in Figure 14 where the results said are at the bottom, namely, You Say: "testing", for example.


Suara ke Teks

Klik tombol untuk mulai merekam

Figure 12. GUI Speech to Text start

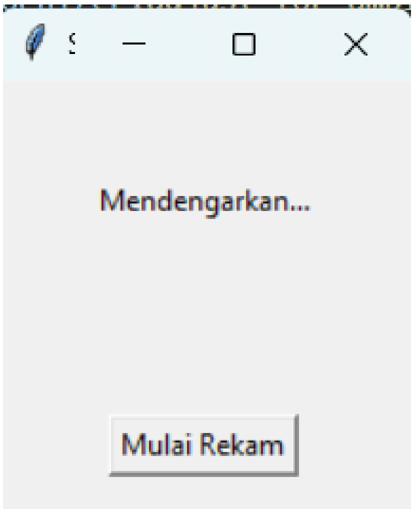
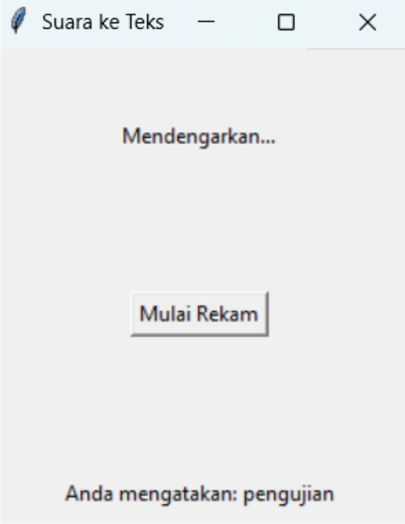

Mendengarkan...

Figure 13. GUI Speech to Text, Listening to sound


Suara ke Teks

Mendengarkan...

Anda mengatakan: pengujian

Figure 14. GUI Speech to Text, provide voice-listened result text

Furthermore, Figure 15 is the Text to Voice display, which is to enter the text that will be made into a voice, then specify the storage directory and also add a voice extension directory such as *.mp3. as shown in Figure 16. After the conversion process has been carried out, a notification will appear as in Figure 17. You can see and listen to the converting sound in Figure 18. Usually, it is stored according to the desired extension *.mp3.

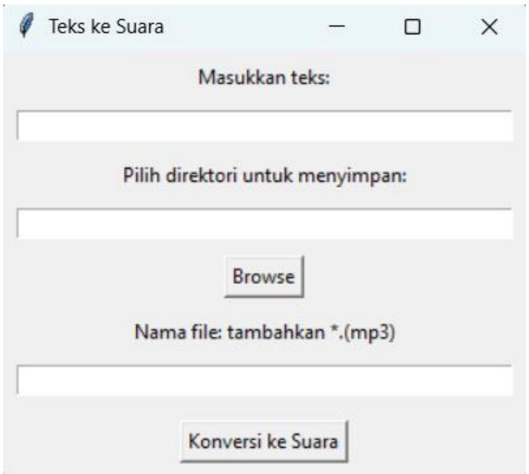
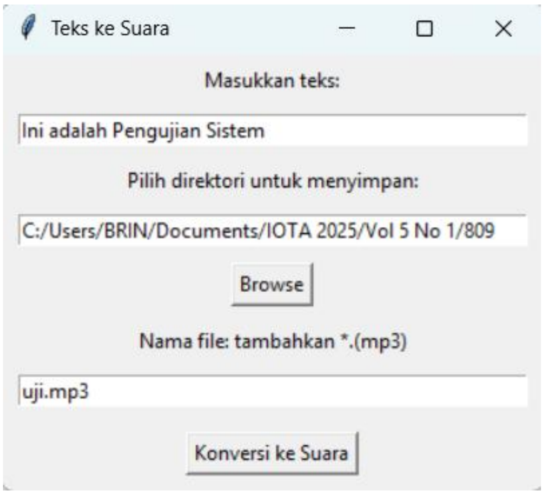

Teks ke Suara
Masukkan teks:
Pilih direktori untuk menyimpan:
Browse
Nama file: tambahkan *.(mp3)
Konversi ke Suara

Figure 15. GUI text-to-speech



Masukkan teks:

Ini adalah Pengujian Sistem

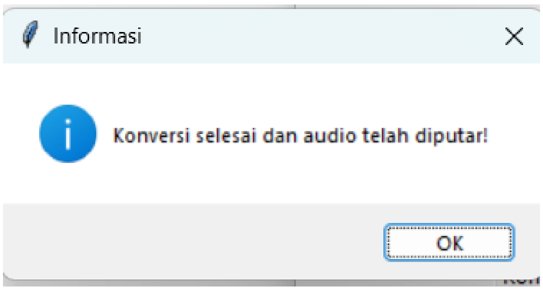
Pilih direktori untuk menyimpan:

C:/Users/BRIN/Documents/IOTA 2025/Vol 5 No 1/809

Nama file: tambahkan *.(mp3)

uji.mp3

Figure 16. File names and directories in GUI Text to Speech



Konversi selesai dan audio telah diputar!

Figure 17. Notification that the text-to-audio conversion has been played and also saved

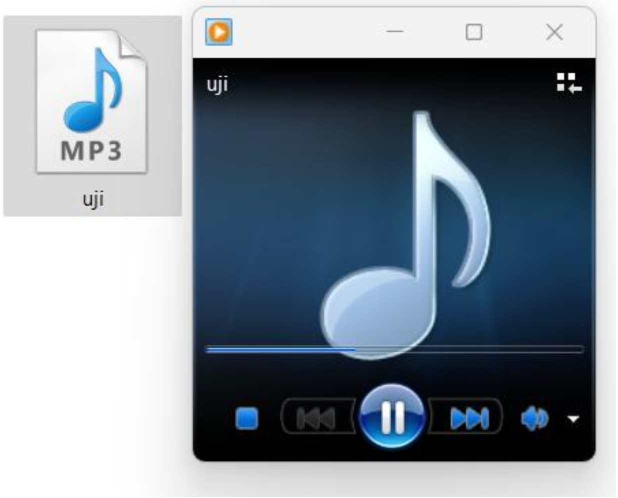


Figure 18. Audio files that have been saved can be re-run.

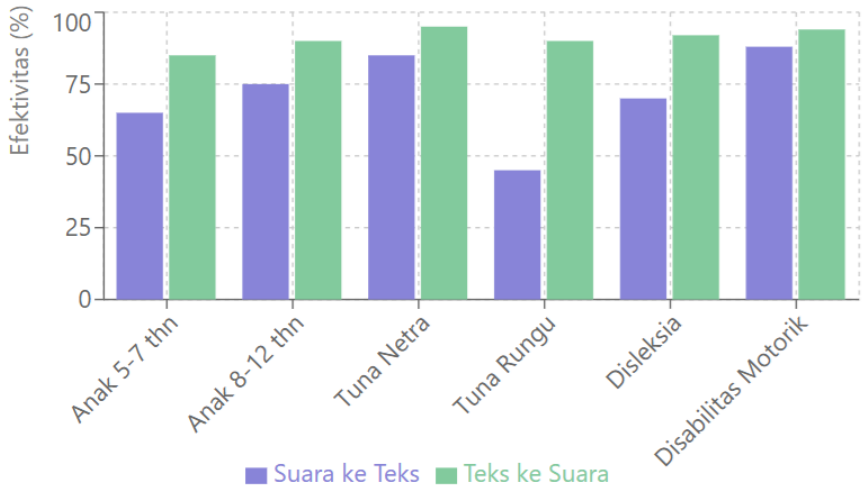


Figure 19. Technology Effectiveness by Usage Group

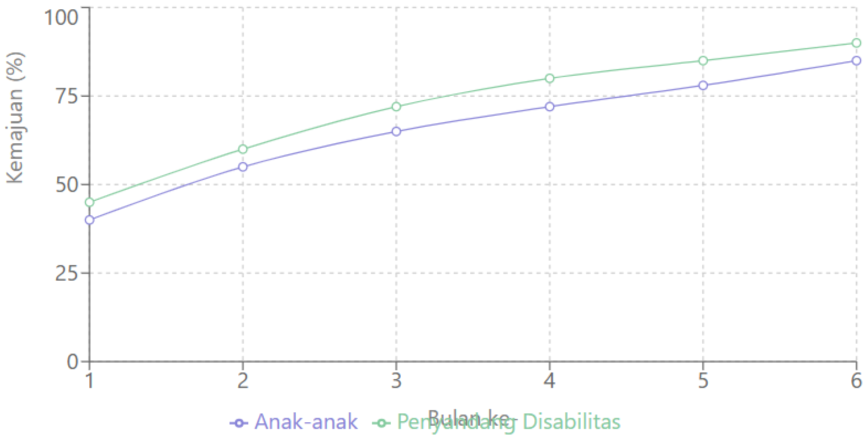


Figure 20. Learning Progress for 6 months of use

Figure 19 is a prediction of the use of this program if applied to various age conditions and conditions of a person's disability. This figure is not true, because the condition of each person with a disability and children who continue to improve their learning patterns and understanding will make this graph floating or still changeable. While Graph Figure 20 shows the increase in the learning process from day to day in the x data above in months, it has increased even though it is not significant, between children and disabilities are almost the same, but disabilities are still superior because children are still experiencing growth in understanding day by day, while disabilities already understand the learning process but only because of physical deficiency problems such as blindness for example.

5. Conclusions

The Speech-to-text and text-to-speech conversion program given to the children of Karamat Mulya Elementary School is one example of a simple product that will train children in writing and also deliver the results automatically. This will train children's independence in learning and also the process of automatic proofreading. This system will help in improving discipline and regular practice in writing and also listening to the results of their writing whether understood by the computer or not. This system will continue to be developed to make children able to read and write easily, with a system that can guide children in writing, recognize the type of keyboard on the computer, pour the results of handwriting into a regular book, and then copy it on a computer such as Ms.Office Word or Notepad.

6. Future Research

This system still uses a super simple Graphical User Interface (GUI), in the future, a more attractive GUI will be made, and an analysis system will also be added in the form of a percentage of errors (%) writing, and also a percentage of word understanding (%), when children read and write, these percentages will appear which are converted into numbers. This number ranges from 0-10, which shows the value of the student's reading and writing results.

Acknowledgments: Thank you to all academic staff, students, and lecturers at the Department of Electrical Engineering, Langlangbuana University, Bandung, hopefully, this research can continue to be developed and can be cited to add and further foster science, especially in the field of Artificial Intelligence this is c applicable tools such as speech-to-text conversion and vice versa, to be implemented in the community.

Author contributions: The authors were responsible for building Conceptualization, Methodology, analysis, investigation, data curation, writing—original draft preparation, writing—review and editing, visualization, supervision of project administration, funding acquisition, and have read and agreed to the published version of the manuscript.

Funding: The study was conducted without any financial support from external sources.

Availability of data and Materials: All data are available from the authors.

Conflicts of Interest: The authors declare no conflict of interest.

Additional Information: No Additional Information from the authors.

References

1. A. Gabryś et al., "Voice Filter: Few-Shot Text-to-Speech Speaker Adaptation Using Voice Conversion as a Post-Processing Module," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, 2022, pp. 7902-7906, doi: 10.1109/ICASSP43922.2022.9747239.

2. A. R. Singh, D. Bhardwaj, M. Dixit and L. Kumar, "An Integrated Model for Text to Text, Image to Text and Audio to Text Linguistic Conversion using Machine Learning Approach," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-7, doi: 10.1109/ISCON57294.2023.10112123.
3. M. Moeini, R. Ahmadian and M. Ghatee, "Calibrated SVM for Probabilistic Classification of In-Vehicle Voices into Vehicle Commands via Voice-to-Text LLM Transformation," 2024 8th International Conference on Smart Cities, Internet of Things and Applications (SCIoT), Mashhad, Iran, Islamic Republic of, 2024, pp. 180-188, doi: 10.1109/SCIoT62588.2024.10570106
4. X. YU and B. Mak, "Non-Parallel Many-To-Many Voice Conversion by Knowledge Transfer from a Text-To-Speech Model," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 2021, pp. 5924-5928, doi: 10.1109/ICASSP39728.2021.9414757.
5. T. N. Charanya and T. C. Sankar, "Voice Assisted Text Summarizer Using NLP," 2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI), Chennai, India, 2023, pp. 1-5, doi: 10.1109/ICDSAAI59313.2023.10452662.
6. K. Thopate, K. Amrutkar, T. Kasliwal, S. Karvir and O. Kumbhar, "Vision Voice: A Raspberry Pi-Based Text-to-Audio Converter for the Visually Impaired," 2024 International Conference on Emerging Innovations and Advanced Computing (INNOCOMP), Sonipat, India, 2024, pp. 231-234, doi: 10.1109/INNOCOMP63224.2024.00045.
7. J. Seong, W. Lee and S. Lee, "Multilingual Speech Synthesis for Voice Cloning," 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju Island, Korea (South), 2021, pp. 313-316, doi: 10.1109/BigComp51126.2021.00067.
8. T. -H. Kim, S. Cho, S. Choi, S. Park and S. -Y. Lee, "Emotional Voice Conversion Using Multitask Learning with Text-To-Speech," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 7774-7778, doi: 10.1109/ICASSP40776.2020.9053255.
9. M. A. Shaik, A. Kethireddy, S. Nerella, S. Pinninti, V. Kathare and P. Pitta, "Sound Wave Scribe: Bridging Spoken Language and Written Text," 2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT), Delhi, India, 2024, pp. 413-417, doi: 10.1109/IC2SDT62152.2024.10696694.
10. M. G. Gonzales, P. Corcoran, N. Harte and M. Schukat, "Joint Speech-Text Embeddings for Multitask Speech Processing," in IEEE Access, vol. 12, pp. 145955-145967, 2024, doi: 10.1109/ACCESS.2024.3473743.
11. A. J. James, N. D. Vangapalli, J. Siripurapu and Y. R. Chinnamallu, "Integration of Voice Assistant with ChatGPT and DALL-E," 2024 International Conference on Emerging Techniques in Computational Intelligence (ICETCI), Hyderabad, India, 2024, pp. 95-101, doi: 10.1109/ICETCI62771.2024.10704161.
12. M. Babiański, K. Pokora, R. Shah, R. Sienkiewicz, D. Korzekwa and V. Klimkov, "On Granularity of Prosodic Representations in Expressive Text-to-Speech," 2022 IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar, 2023, pp. 892-899, doi: 10.1109/SLT54892.2023.10022793.
13. J. Wu, A. Polyak, Y. Taigman, J. Fong, P. Agrawal and Q. He, "Multilingual Text-To-Speech Training Using Cross Language Voice Conversion And Self-Supervised Learning Of Speech Representations," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, 2022, pp. 8017-8021, doi: 10.1109/ICASSP43922.2022.9746282.
14. Y. -Y. Chen, Y. -W. Bai, C. -Y. Tsai, J. -F. Wang and B. -W. Chen, "Voice-customizable text-to-speech for intelligent home-care system," 2013 1st International Conference on Orange Technologies (ICOT), Tainan, Taiwan, 2013, pp. 239-242, doi: 10.1109/ICOT.2013.6521201.
15. R. Gupta, P. Kumar, P. K. Swain, D. Kumar and N. Garg, "Neural Voice Replication: Multispeaker Text-to-Speech Synthesizer," 2024 International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications (ICETCS), Bengaluru, India, 2024, pp. 1-6, doi: 10.1109/ICETCS61022.2024.10543403.
16. S. Kadam, A. Jikamade, P. Mattoo and V. Hole, "ReVoice: A Neural Network based Voice Cloning System," 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, 2024, pp. 1-6, doi: 10.1109/I2CT61223.2024.10543448.

17. A. L. Sinha, H. Muley, J. Ghosh and P. Sarode, "AI based Desktop Voice Assistant for Visually Impared Persons," 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2023, pp. 882-886, doi: 10.1109/ICCES57224.2023.10192894.
18. W. Weber, "Text Visualization - What Colors Tell About a Text," 2007 11th International Conference Information Visualization (IV '07), Zurich, Switzerland, 2007, pp. 354-362, doi: 10.1109/IV.2007.108.
19. V. M. Reddy, T. Vaishnavi and K. P. Kumar, "Speech-to-Text and Text-to-Speech Recognition Using Deep Learning," 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 2023, pp. 657-666, doi: 10.1109/ICECAA58104.2023.10212222.
20. K. Azizah, "Zero-Shot Voice Cloning Text-to-Speech for Dysphonia Disorder Speakers," in IEEE Access, vol. 12, pp. 63528-63547, 2024, doi: 10.1109/ACCESS.2024.3396377.