P-ISSN : 2715-2448 | E-ISSN : 2715-7199
Vol.7 No.1 January 2026
Buana Information Technology and Computer Sciences (BIT and CS)

Implementation of the LSTM Model for Speech-to-Text Systems in the Recognition of the *Walikan* Language of Malang

Raynanda Dwi Pangestu[1]*, Aviv Yuniar Rahman[2], Istiadi[3]
[1,2,3]Widyagama University of Malang, Malang Jl. Taman Borobudur Indah No.35, Telp. 0341-492282,
Kec.Lowokwaru, Kota Malang, Jawa Timur 65128
E-mail: dwireynanda52@gmail.com[1]*, aviv@widyagama.ac.id[2], istiadi@widyagama.ac.id[3]

***Received:** 2025-08-12 | **Revised:** 2026-01-21 | **Accepted:** 2026-01-31*

Abstract

This study developed a Speech-to-Text (STT) system based on the Long Short-Term Memory (LSTM) model to recognize and convert speech in the Malang *Walikan* language into text. The Malang *Walikan* language has a unique linguistic structure in the form of word reversal, which poses a challenge in speech recognition. The data used consisted of 1,000 sentences collected from social media and direct recordings. The data was processed using Mel Frequency Cepstral Coefficients (MFCC) and then used to train the LSTM model. The system's performance was evaluated using the Word Error Rate (WER), Character Error Rate (CER), and Average Test Loss metrics. The best results obtained showed a WER value of 1.0 on a 699:300 data split, a CER of 0.78 on a 799:200 split, and an Average Test Loss of 11.0147 on a 299:700 split. The high Average Test Loss value indicates the model's difficulty in minimizing prediction errors, which may be caused by the model's mismatch with the data patterns or overfitting. To improve the model's performance, it is recommended to improve the quality of the training data, optimize the parameters, and apply regularization techniques.

Keywords: *Bahasa Walikan Malang*, Speech-to-Text, Long Short-Term Memory (LSTM), Mel Frequency Cepstral Coefficients (MFCC), Word Error Rate (WER).

I. Introduction

The rapid advancement of speech recognition technology, or Speech-to-Text (STT), has had a significant impact on various aspects of modern life, ranging from voice-based services to more efficient human–machine interactions. This technology enables the automatic conversion of spoken language into text, which can be utilized in applications such as virtual assistants, automated transcription, and voice-based communication [1]. Speech-to-Text allows computers to understand human language through voice commands.

Speech-to-Text (STT) is a technology capable of converting human speech into text. This system uses various speech signal processing methods, feature extraction, and machine learning algorithms to recognize and understand inputted speech. This technology has developed rapidly, with various applications such as virtual assistants, automatic transcription services, voice navigation systems, and other human-machine interactions. In computer science, there is a field called Speech to Text, and this theory will be useful in these situations [2].

One of the unique linguistic phenomena in Indonesia is the *Bahasa Walikan Malang*, a distinctive language variety used by the people of Malang. This language features an unusual structure in which words are reversed or altered, creating specific challenges for speech recognition systems. Considering the importance of preserving local culture and the lack of automated systems capable of recognizing *Bahasa Walikan Malang*, there is a need to develop a system that can convert speech in this language into text [3].

Long Short-Term Memory (LSTM), a variant of the Recurrent Neural Network (RNN), has shown promising results in handling long sequential data and retaining long-term dependencies, which are crucial in recognizing complex patterns in *Bahasa Walikan* Malang. Previous studies indicate that LSTM overcomes the vanishing/exploding gradient problem that affects traditional RNNs, enabling better performance in sequential prediction tasks. This makes LSTM an appropriate choice for developing a Speech-to-Text system tailored for *Bahasa Walikan* Malang.

While speech recognition technology has been widely applied to major languages such as Indonesian and English, its development for local languages remains limited. The phonetic complexity and unique linguistic structure of *Bahasa Walikan* require specialized approaches for accurate recognition. Therefore, the application of LSTM in recognizing reversed word patterns and contextual meaning in *Bahasa Walikan* offers potential advantages over traditional models, especially in cultural preservation and modern communication contexts.

The proposed system aims to provide practical benefits, such as assisting communities and cultural advocates in preserving the language, as well as supporting educational tools, cross-cultural communication, and the development of locally adapted voice assistants. However, challenges remain, including the absence of pre-existing datasets for *Bahasa Walikan* requiring manual data collection and the tendency of LSTM models to produce repetitive outputs when handling non-standard language variations. Moreover, little research has been conducted to compare LSTM with other architectures such as CNNs or Transformers for this specific use case, leaving a significant opportunity for further study.

This research focuses exclusively on recognizing speech in *Bahasa Walikan* Malang using the LSTM model, without comparisons to alternative architectures. The dataset is collected from predefined and limited sources, and evaluation metrics include Word Error Rate (WER) and Character Error Rate (CER). The ultimate goal is to create a robust, efficient, and culturally relevant STT system that can contribute to the preservation and revitalization of *Bahasa Walikan* Malang.

The LSTM model is designed with an input layer, a hidden layer, and an output layer. This model receives input in the form of voice features (MFCC) and processes it through the LSTM layers to produce output in the form of text predictions.

The results of this study are the accuracy values of *Walikan* language pronunciation classification. The testing process uses *Walikan* language pronunciation audio data for prediction data, producing prediction results consistent with their pronunciations [4].

II. Method

This study uses an experimental approach to develop a Long Short-Term Memory (LSTM)-based Speech-to-Text (STT) system for recognizing the *Walikan* Malang language. Recognition techniques enable speakers' voices to be used to verify their identities and control access to services such as voice calls, telephone banking, telephone shopping, database access services, information services, voice messages, security controls for confidential information areas, and remote access to computers. The acoustic parameters of voice signals used in recognition tasks have been widely studied and investigated, and can be categorized into two types of processing domains: the first group consists of spectral-based parameters, and the other is dynamic time series [5]. The Mel Frequency Cepstrum Coefficients method has several advantages, including being able to capture important information in sound signals, producing minimal data without eliminating existing information, and replicating the human hearing organ in perceiving sound signals [6]. The research design begins with the collection of voice data to be converted into text, followed by data preprocessing, LSTM model training, and model performance evaluation using specific metrics. The general stages of the research can be seen in the following figure.

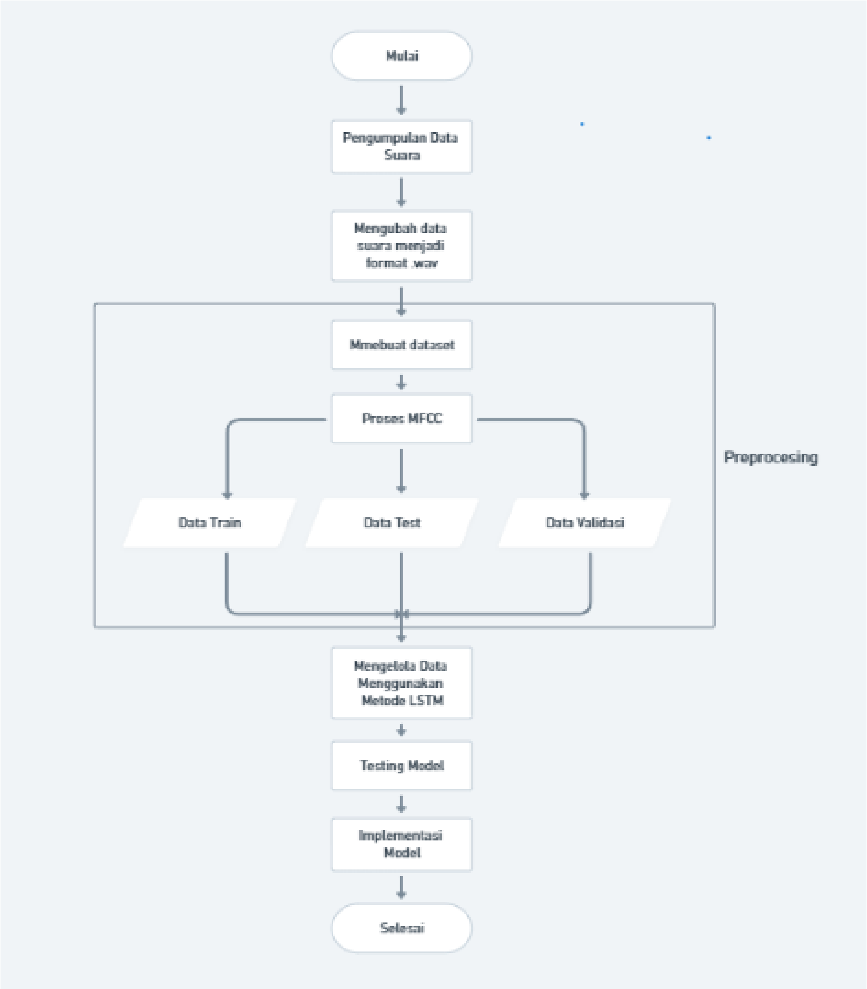


Figure 1. Research Design (Aini Lailla Asri et al., 2023)

The data used in this study consists of *Walikan Malang* language audio data obtained from original audio recordings. The data collected consists of sentences in the *Walikan Malang* language used by social media users. The filtering process was carried out using keywords specific to the *Walikan* language, such as “utapes” (shoes), “ilakes” (once), and other words commonly used in the language.

A total of 1,000 sentences in the *Walikan* language were collected. Each sentence was then translated into Indonesian to support the training and evaluation of the text-to-speech system. This translation aims to ensure that the meaning of each sentence in *Walikan* is clear and easy to understand. The collected data is stored in spreadsheet format (Excel) for easier management and processing. Before use, the data undergoes a pre-processing stage to remove irrelevant symbols or emojis and standardize the text. An example of the data used is shown in (Table 1).

Table 1. *Walikan* Language Datasets

Walikan Language	Indonesian Language
Kera Ngalam	*Arek Malang*
Ngalam	*Malang*
Kera	*Arek*
Asli Ngalam	*Asli Malang*

Umak	Kamu
Umak hebak sam	Kamu Semua Mas
Hebak	Semua

1. Literature Review

This section presents the theoretical foundation and relevant literature that support the development of a Speech-to-Text (STT) system based on Long Short-Term Memory (LSTM) for *Bahasa Walikan Malang*. The review begins with a general overview of speech recognition technology, the unique characteristics of *Bahasa Walikan Malang*, and the LSTM model, which is selected for its capability in handling sequential data and maintaining long-term contextual dependencies. In addition, relevant previous studies are examined to provide insights into the development trends and effectiveness of approaches used in speech recognition systems.

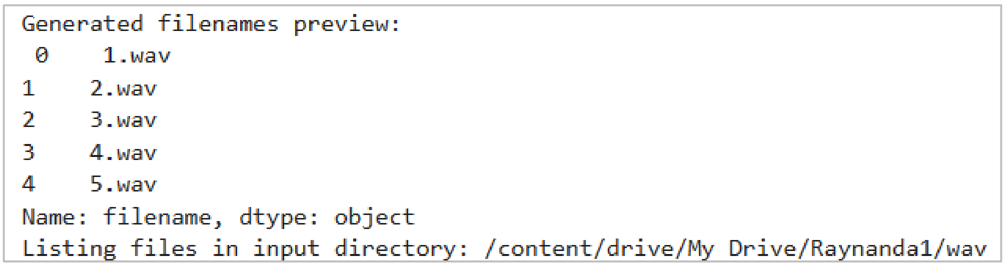
2. Data Collection

This study used 1,000 *Walikan Malang* language voice data obtained from recordings of native speakers and conversations on social media, particularly X (Twitter) and Facebook. The data was collected in the form of Boso *Kiwalan* sentences that are popular among the people of Greater Malang. All sentences were translated into Indonesian to ensure clarity of meaning and to support the system training and evaluation process.

Eighty percent of the data was used as training data to train the LSTM model, while the remaining 20% was used as test data to evaluate the model's performance. The data was stored in spreadsheet format (Excel) for easy management, then processed through a pre-processing stage, including the removal of irrelevant symbols or emojis and text standardization.

3. Data preprocessing

The process begins with a very important pre-processing stage for the data, namely the audio data and label data (text transcription). MFCC extraction aims to obtain parameter values. The class labels obtained from filename cleaning during dataset loading are converted from categorical to numeric. During the data separation process, the data is divided into training, validation, and testing data [7]. In the audio pre-processing stage, the original M4A format files are converted to WAV format using the ffmpeg tool. This conversion is essential to ensure the files are compatible and can be processed efficiently. Each converted audio file is then opened using the torch audio library. To maintain consistency, the sampling rate is standardized to 16000 Hz, and the audio is converted to a single channel (mono) if necessary. After that, the Mel-Frequency Cepstral Coefficients (MFCC) features are extracted from each audio file. The MFCC features, which have 40 coefficients per frame, are selected for their ability to effectively describe voice characteristics, making them suitable as input for the ASR model.



```
Generated filenames preview:
 0    1.wav
1    2.wav
2    3.wav
3    4.wav
4    5.wav
Name: filename, dtype: object
Listing files in input directory: /content/drive/My Drive/Raynanda1/wav
```

Figure 2. Preprocessing wav files

During label preprocessing, transcription data is extracted from an Excel file. Raw data is cleaned by removing unnecessary rows, leaving only the number and transcription columns. The number column is used to create corresponding WAV file names (e.g., 2.wav, 3.wav). ensuring accurate correspondence between audio files and text. Before training, each transcription text is converted into numerical form through label encoding. Since the duration of audio files varies, the length of audio feature data from each file is standardized using a padding method before being fed into the model.

4. Design System

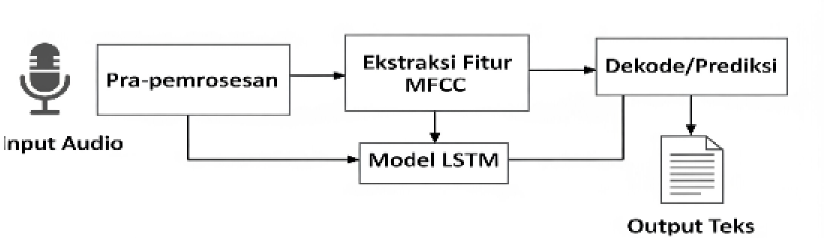


Figure 3. Design System

The system process begins with voice recording, which is then converted into audio signals. Next, preprocessing is performed to produce cleaner and more structured data. The processed audio data is then processed by the LSTM model, which is designed to handle the sequential nature of voice data. The output layer of the model will produce transcription text that can be used further.

The system requires information in the form of word patterns obtained through feature extraction. Feature extraction in speech recognition is the computational process of analyzing audio signals to generate feature vectors that represent the characteristics of those signals [8]. In its implementation, the system receives audio input, performs preprocessing to extract important features such as Mel Frequency Cepstral Coefficients (MFCC), then processes the data using the LSTM model to produce text output. This system is designed to accommodate the unique characteristics of the Malang *Walikan* language, including word reversal and distinctive phonetic variations.

5. Evaluation

The overall WER value reached 1.13, indicating a very high transcription error rate and low model accuracy. The box plot shows a wide distribution of WER with a number of outliers, while the histogram shows that the majority of data is at WER 1.0 with a sharp decline after 1.5. The WER distribution graph per sample index indicates large fluctuations and performance instability. Analysis of the relationship between predictions found two sentences with identical values, one of which had a uniform index distribution and varying WER, while the other was consistent at WER 1.0. This indicates differences in complexity between test sentences and the need for significant improvements to the model.

6. Result Analysis

The quality and quantity of training data need to be improved by adding more *Walikan* Malang audio data with a variety of speakers, speaking speeds, and environmental conditions, accompanied by noise cleaning and volume normalization to make the data more consistent. The feature extraction process can be optimized by adjusting MFCC parameters such as the number of coefficients, frame length, and hop length to be more sensitive to *Walikan* phonetic features, as well as considering the addition of delta and delta-delta MFCC features to capture temporal dynamics.

The LSTM model architecture can also be strengthened by increasing the number of units and layers, as well as applying an attention mechanism to focus attention on relevant parts of the audio. Training strategies can be made more adaptive through the use of data augmentation techniques such as pitch shifting, time stretching, and the addition of synthetic noise, accompanied by learning rate scheduling and the application of early stopping to avoid overfitting. Additionally, given the uniqueness of the *Walikan* language, a specialized pre-processing module is required to detect and process word reversals, as well as training the model with normal *Walikan* word pairs to strengthen the distinctive phonetic mapping.

III. Results and Discussion

Based on the evaluation results described in the previous subsection, the Text-to-Speech system used in this study showed poor performance when applied to languages with limited resources such as *Walikan*, mainly because no fine-tuning was performed on specific data. This is reflected in the Word Error Rate (WER) and Character Error Rate (CER) values obtained from the audio data.

1. Word Split Data Testing

Training data is used to build and adjust model parameters, while testing data is used to evaluate how well the model performs on new data that it has never seen before, thereby objectively demonstrating the model's ability to generalize.

Table 2. Split data testing

Split Ratio	WER	CER	Loss
899:100	1.06	0.8	13.8313
799:200	1.06	0.78	14.2845
699:300	1.0	0.78	13.7803
599:400	1.19	0.93	13.6771
499:500	1.34	1.04	12.5776
399:600	1.05	0.77	12.7114
299:700	1.11	0.82	11.0147
199:800	1.13	0.85	12.2895
99:900	1.0	0.82	12.2217

(Table 2) Shows the results of evaluating the LSTM model for speech-to-text tasks in various training and testing data division scenarios. The first column shows the ratio of training data to testing data, such as 899:100, meaning that 899 data points were used for training and 100 for testing. The second column displays the Word Error Rate (WER), which measures the error rate at the word level. The smaller the WER value, the better the model's ability to correctly recognize words. The third column presents the Character Error Rate (CER), which measures the error rate at the character level, where a smaller value indicates better model performance.

The fourth column shows the model's accuracy on the test data in percentage form. From the results obtained, all scenarios show 0 percent accuracy, meaning the model's predictions do not fully align with the reference transcription at the word level. The last column displays the average loss on the test data, reflecting the average error of the model during the evaluation process. A smaller loss value indicates the model is more successful in reducing prediction errors.

Overall, although the WER and CER values differ in each scenario, the accuracy trend remaining at zero indicates that the model is not yet capable of producing transcriptions that are identical to the reference, even though in some cases the errors at the word and character level are relatively small.

2. Distribution Word Error Rate (WER)

Based WER is a metric used to measure the percentage of words incorrectly recognized by the system compared to the reference text. The lower the WER value, the better the system performance. WER measures errors at the word level.

The formula is:

$$WER = \frac{S+D+I}{N} \tag{1}$$

Explanation:

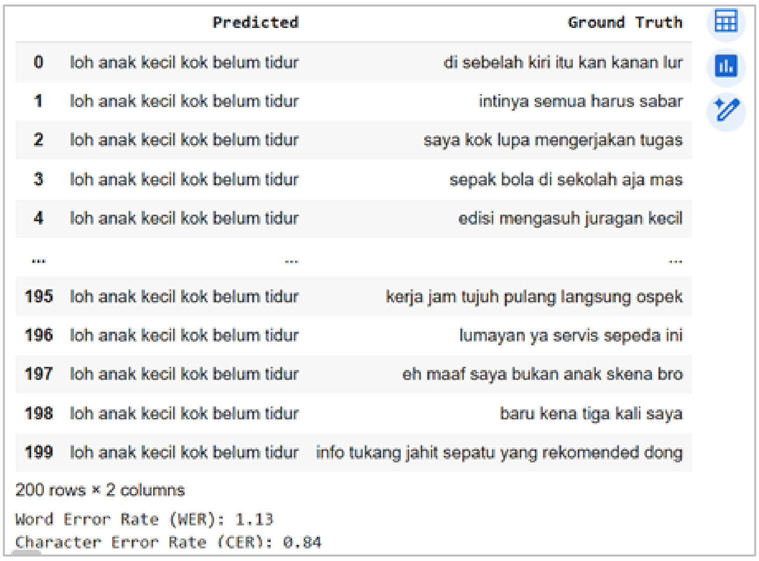
S : Number of words incorrectly substituted.

D : Number of words deleted from the reference text.

I : Number of words inserted into the predicted text.

N : Total number of words in the reference text (ground truth).

Manual WER calculation: WER calculation is performed based on the formula above. Overall WER calculation for the entire test data is performed automatically.



	Predicted	Ground Truth
0	loh anak kecil kok belum tidur	di sebelah kiri itu kan kanan lur
1	loh anak kecil kok belum tidur	intinya semua harus sabar
2	loh anak kecil kok belum tidur	saya kok lupa mengerjakan tugas
3	loh anak kecil kok belum tidur	sepak bola di sekolah aja mas
4	loh anak kecil kok belum tidur	edisi mengasuh juragan kecil
...	...	...
195	loh anak kecil kok belum tidur	kerja jam tujuh pulang langsung ospek
196	loh anak kecil kok belum tidur	lumayan ya servis sepeda ini
197	loh anak kecil kok belum tidur	eh maaf saya bukan anak skena bro
198	loh anak kecil kok belum tidur	baru kena tiga kali saya
199	loh anak kecil kok belum tidur	info tukang jahit sepatu yang rekomended dong

200 rows × 2 columns

```
Word Error Rate (WER): 1.13
Character Error Rate (CER): 0.84
```

Figure 4. Prediction Results and Basic Truths

To understand the calculation process, a manual evaluation was performed on one row of data as an example, namely row 0 with the following content:

a. Model Prediction: Why isn't the little kid asleep yet?
b. Ground Truth: That's on the right side, isn't it?

The first step is to separate each sentence into words:

a. Prediction: 6 words
b. Ground Truth: 7 words

Then, the positions of the words are matched one by one. From the comparison results, none of the words in the prediction match the ground truth, so:

a. Substitution (S): 6 words
b. Deletion (D): 1 word (the word “lur” was not predicted)
c. Insertion (I): 0

Total words in ground truth (N): 7

Therefore, the WER calculation for this row is:

$$\text{WER} = \frac{6 + 1 + 0}{7} = \frac{7}{7} = 1.0 \qquad (2)$$

From the manual calculation of one row of data, it is known that:

a. The model made 6 substitution errors and 1 word deletion out of a total of 7 words.

b. The WER value for that row is 1.0, indicating a total error at the word level.

c. This indicates that the model is not yet able to understand or generalize voice input effectively.

Analysis of 200 text entries revealed discrepancies between the predictions and ground truth, with matches only on lines 0 and 195. Predictions often deviated from the context, such as on lines 1, 2, 3, 4, 196, 197, 198, and 199. Evaluation shows a Word Error Rate (WER) of 1.23%, Character Error Rate (CER) of 0.98%, Test Accuracy of 0.0%, and Average Test Loss of 15.5946, indicating suboptimal model performance. It is recommended to enhance the training data and adjust the model to improve accuracy.

3. Results Character Error Rate (CER)

CER measures the percentage of errors at the character level and is often used to evaluate speech recognition performance with higher accuracy. CER measures errors at the character level. The formula is:

$$\text{CER} = \frac{\text{Sc} + D\text{c} + I\text{c}}{N\text{c}} \qquad (3)$$

Explanation:

Sc : Number of incorrectly substituted characters.

Dc : Number of characters deleted from the reference text.

Ic : Number of characters inserted into the predicted text.

Nc : Total number of characters in the reference text (ground truth).

This evaluation is performed by comparing the transcribed text with the spoken reference text. Character Error Rate (CER) is a metric used to evaluate the accuracy of sequence-to-sequence models, particularly in tasks such as speech recognition or machine translation. CER is calculated by comparing the number of incorrect characters (including substitutions, deletions, and additions) to the total number of characters in the reference text, then converting it to a percentage.

CER was recorded at 0.98%, indicating that only 0.98% of the total characters in the reference text had errors. This value indicates that the model has a very good level of accuracy at the character level, with relatively minimal errors. However, despite the low CER, this does not fully reflect contextual or semantic accuracy, as this metric focuses solely on character matching and does not consider word order or overall understanding. Therefore, even though the CER is only 0.98%, the contextual mismatches observed in the predictions suggest that errors are more prevalent at the level of language comprehension or sentence structure.

4. Average Test Loss

The Average Test Loss was recorded at 15.5946. This value is quite high, indicating that the model has significant difficulty in minimizing prediction errors in the test data. This large number indicates that there is a big difference between the prediction results and the ground truth, which could be caused by a lack of model fit with the data patterns, overfitting, or suboptimal training data quality. Therefore, the high Average Test Loss value (15.5946) indicates the need for model adjustments, such as increasing the training data, optimizing parameters, or using regularization techniques to improve accuracy and reduce errors.

5. System Evaluation Methods

System performance evaluation was conducted using standard metrics in speech recognition, namely transcription comparison, overall Word Error Rate (WER) and Character Error Rate (CER), WER per sentence, and visualization of WER distribution. Transcription comparison is performed by creating a DataFrame named comparison_df containing two main columns: Predicted, which holds the model's predicted transcription results, and Ground Truth, which contains the actual transcription from the test data. This Data Frame is displayed to facilitate direct comparison between the predicted results and the reference transcription, allowing the model's accuracy to be observed visually.

The overall WER and CER calculations are performed using the jiwer library. WER is calculated by summing the number of substitutions, insertions, and deletions required to convert the predicted transcription into the ground truth transcription, then dividing it by the number of words in the ground truth transcription. CER is calculated using a similar method but at the character level. The lower the values of these two metrics (closer to 0), the better the model's performance. These overall WER and CER values are then printed as an overview of the model's performance across the entire test data.

WER per sentence is calculated for further analysis, with the results stored in the Data Frame *wer_per_sentence_df*, which contains the columns Ground Truth, Predicted, and WER. This Data Frame is displayed to identify sentences that are difficult to transcribe (have high WER) and those that are transcribed well (low WER or 0). The WER distribution is visualized using two types of plots. The WER histogram is used to see the frequency of occurrence of specific WER values in the test data, while the WER box plot displays a summary of the distribution, including the median, quartiles, and outliers. This visualization provides additional insight into the distribution of the model's performance on each sentence.

Overall, this evaluation includes direct comparison between predictions and references, calculation of WER and CER both aggregated and per sentence, as well as visual analysis of WER distribution to understand patterns and variations in system performance.

6. Analysis and Word Error Rate (WER) Variation

Evaluation of the Speech to Text (STT) model transcription results on 200 test data showed a Word Error Rate (WER) of 1.13 and a Character Error Rate (CER) of 0.84, indicating very low text recognition performance. Errors are dominated by mass substitutions, where the model consistently replaces entire sentences with phrases, disregarding context or original meaning. No significant patterns were found in word deletions or insertions, so the primary errors focus on comprehensive changes. The high CER indicates a high probability of errors in nearly every character, consistent with the detected substitution pattern. The likely causes stem from limitations in training data variability or model bias, so improvements are recommended through dataset expansion and algorithm adjustments to enhance accuracy and adaptability to diverse contexts.

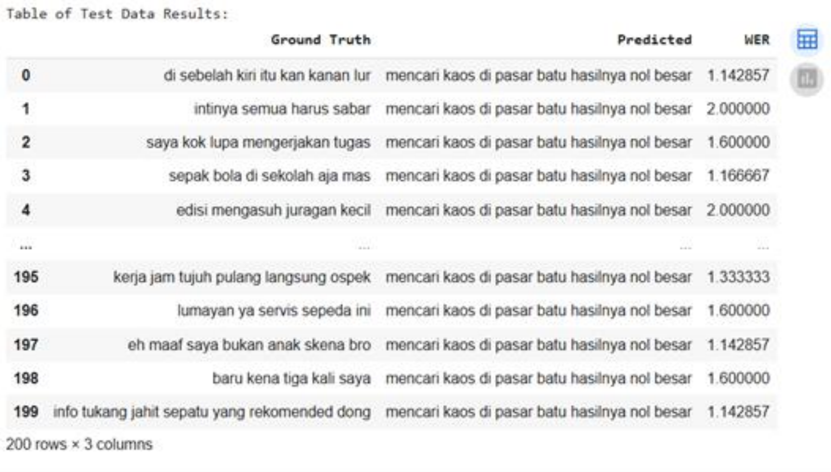


Table of Test Data Results:

	Ground Truth	Predicted	WER
0	di sebelah kiri itu kan kanan lur	mencari kaos di pasar batu hasilnya nol besar	1.142857
1	intinya semua harus sabar	mencari kaos di pasar batu hasilnya nol besar	2.000000
2	saya kok lupa mengerjakan tugas	mencari kaos di pasar batu hasilnya nol besar	1.600000
3	sepak bola di sekolah aja mas	mencari kaos di pasar batu hasilnya nol besar	1.166667
4	edisi mengasuh juragan kecil	mencari kaos di pasar batu hasilnya nol besar	2.000000
...	...	...	...
195	kerja jam tujuh pulang langsung ospek	mencari kaos di pasar batu hasilnya nol besar	1.333333
196	lumayan ya servis sepeda ini	mencari kaos di pasar batu hasilnya nol besar	1.600000
197	eh maaf saya bukan anak skena bro	mencari kaos di pasar batu hasilnya nol besar	1.142857
198	baru kena tiga kali saya	mencari kaos di pasar batu hasilnya nol besar	1.600000
199	info tukang jahit sepatu yang rekomended dong	mencari kaos di pasar batu hasilnya nol besar	1.142857

200 rows × 3 columns

Figure 5. WER test sentence

The examples were randomly selected to represent the variation in Word Error Rate (WER) values from moderate to high.

Row Number 0: Original Truth: *di sebelah kiri itu kan kanan lur* (on the left is on the right),

Predicted*: mencari kaos di pasar batu hasilnya nol besar* (searching for T-shirts at the stone market results in a big zero), WER: 1.142857

In the first row, the original sentence (True) is "*di sebelah kiri itu kan kanan lur*." However, the model produces a prediction that is very different, namely "*mencari kaos di pasar batu hasilnya nol besar*." Due to this significant difference, the Word Error Rate (WER) reaches approximately 1.14, indicating that the number of word errors exceeds the total number of words in the original sentence.

Line Number 1: Original Truth: the point is, everyone must be patient, predicted: searching for a shirt at the stone market yields zero results, WER: 2.000000

In the second line, the original sentence (Original Truth) is "*intinya semua harus sabar*." Similar to the previous line, the model again predicts "*mencari kaos di pasar batu hasilnya nol besar*." Due to the significant deviation from the original sentence, the WER increases to 2.00, meaning the number of word errors is twice the number of words in the original sentence.

Based on these two lines, it is clear that the model tends to generate the phrase "*mencari kaos di pasar batu hasilnya nol besar*" even though the original sentence is different.

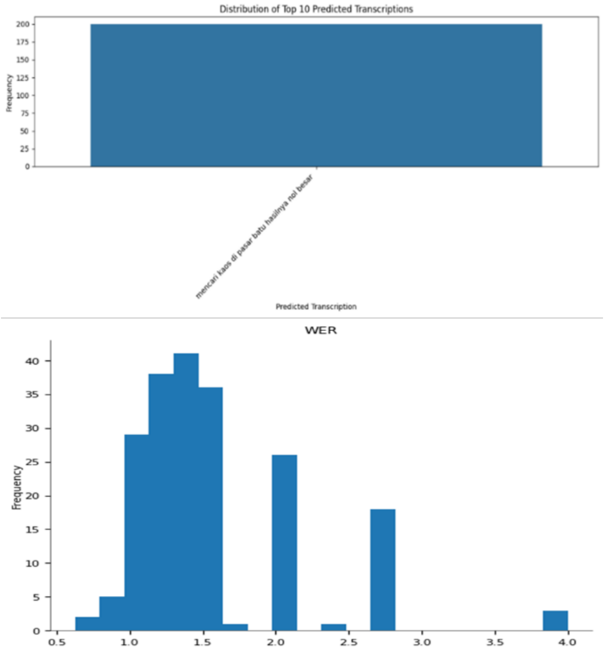


Figure 6. Predicted transcription graph and WER graph of test data results

This is consistent with previous observations in the distribution graph, which shows the dominance of these predictions. The high WER values in both rows indicate that the model is not yet capable of accurately transcribing these sentences.

Predicted Transcription Distribution Graph (Bar Chart): This graph displays a list of unique transcriptions most frequently predicted by your model on the test data (on the horizontal axis). The height of each bar (on the vertical axis) indicates the number (frequency) of how often the model predicts that transcription. The contents of this graph show which phrases are most frequently output by your model as transcription results. If there is one or more bars that are significantly higher than the others, as we saw earlier, it means the model tends to predict that phrase repeatedly, even for audio that should have a different transcription. This indicates bias in the model's output.

Word Error Rate (WER) Distribution Graph (Histogram): This graph displays the distribution of WER values for each test sentence. The horizontal axis is divided into several intervals of WER values. The height of each bar (on the vertical axis) indicates the number of sentences with WER values within that range. The content of this graph indicates how frequently your model makes errors at various accuracy levels. If many tall bars are near the left axis (low WER values, close to 0), this means many sentences are transcribed with high accuracy. If tall bars are in the middle or right (high WER values), this means many sentences have significant error rates. This graph illustrates the consistency of the model's performance.

a. Testing with a split ratio of 0.1

Testing with a ratio of 0.1 means that the dataset is divided into 90% for training and 10% for testing. In this context, out of a total of 999 data points, 899 are used to train the model, while 100 data points are used as a test set to evaluate its performance. The goal is for the model to have more data to learn from so it can better capture patterns, with sufficient test data to measure its ability to handle new data.

Table 3. Test table ratio 0.1

Aspect	Description
Model Structure	LSTM with 40 feature inputs, 2 hidden layers each with 128 units, followed by a linear output layer with 999 classes.
Data Type	Indonesian speech audio dataset that has been processed into numerical features (feature extraction).
Training Results	Initial loss of 7.1392 decreased to 6.2182 by epoch 10.
Evaluation Results	WER: 1.06, CER: 0.80, Average Test Loss: 13.8313.
Brief Analysis	The model was able to reduce loss during training, but accuracy remained at zero, indicating that predictions did not align with ground truth despite moderate word and character error rates.

(Table 4.3) LSTM model testing with training and testing data divided in a ratio of 0.1. In the model structure section, it is explained that the architecture used is LSTM with 40 input features, two hidden layers each consisting of 128 units, and ending with a linear layer that produces 999 classes of output. The data used is a collection of Indonesian speech audio that has undergone feature extraction to be processed by the model.

The training results show that the initial loss value was 7.1392, then decreased to 6.2182 at epoch 10, indicating a learning process during training. During the evaluation phase, the model produced a Word Error Rate of 1.06, a Character Error Rate of 0.80, and an average test loss of 13.8313. Although the loss value decreased during training, the accuracy value remained at zero, meaning the model's predictions did not align with the reference transcription, despite the word and character error rates being at a moderate level. A brief analysis concludes that the model is capable of learning from the training data but fails to generalize well to the test data.

Therefore, some suggestions for improvement include increasing the amount and variety of data, applying regularization or dropout techniques to reduce overfitting, experimenting with learning rate values and optimizer types, and considering the use of fine-tuning or adding attention mechanisms to improve speech recognition performance

b. Randomized Search Ratio 60:40

The 0.2 ratio test means that the dataset is divided into 80% for training and 20% for testing. In this context, out of a total of 999 data points, 799 are used to train the model, while 200 data points are used as a test set to evaluate performance. With a larger proportion of test data compared to a ratio of 0.1, the evaluation can provide a broader picture of the model's generalization capabilities. However, the consequence is that the amount of training data becomes smaller, which may affect the model's ability to learn patterns optimally.

Nevertheless, the final loss value is still quite high, indicating that the model's performance is not yet optimal. This may be due to the large number of classes, noise in the data, or the need to adjust training parameters. Therefore, it is recommended to continue training with more epochs, check and adjust parameters, and consider using regularization techniques and more effective label representation strategies to enable the model to produce more accurate transcriptions.

Table 4. Test table ratio 0.2

Aspect	Description
Model Structure	LSTM with 40 feature inputs, 2 hidden layers each with 128 units, followed by a linear output layer with 999 classes.
Data Type	Indonesian speech audio dataset that has been processed into numerical features (feature extraction).
Training Results	Initial loss of 7.2276 decreased to 6.3589 by epoch 10.
Evaluation Results	WER: 1.06, CER: 0.78, Average Test Loss: 14.2845.
Brief Analysis	The model experienced a decrease in loss during training, but accuracy remained at zero. The word error rate decreased slightly compared to CER, but the test loss was higher than the 0.1 ratio, indicating that the model was less optimal when faced with larger test data.

(Table 4.4) Explains various important things such as model structure, types of data used, training results, evaluation results, and suggestions for improvement. At the beginning, it is mentioned that this model is a combination of LSTM and linear layers, which are commonly used for tasks involving data sequences. The model accepts 40 features as input, which is common in speech signal processing. There are two LSTM layers, each containing 128 units, indicating that the model has a fairly deep structure. At the end, the model has 999 classes or tokens, meaning the model is designed to choose from a wide range of possible outcomes. In terms of data, the total amount of data used is 999, which is relatively small for training a deep learning model. The data is divided into 80% for training and 20% for testing, which is a standard division in machine learning experiments.

During the training process, the initial loss value was quite high in the first epoch, at around 7.7908. After 10 epochs, the loss value dropped to 6.2652. This decrease indicates a learning process, although the decrease is not very significant, which may indicate that the model is not learning optimally. The evaluation results on the test data show that the model provides repetitive and unvaried prediction results. For example, the same phrase often appears as a prediction for different inputs, such as "*utamakan cari pemain kecil buat masa depan*" and *"saya yang di borgol melanggar hukum mas"* This pattern indicates that the model does not understand or generalize the input effectively. As a result, the model's performance is considered a complete failure, and metrics such as the Word Error Rate (WER) are estimated to be very high.

c. Testing with a split ratio of 0.3

Testing with a ratio of 0.3 means that the dataset is divided into 70% for training and 30% for testing. In this case, out of a total of 999 data points, 699 are used to train the model, while 300 are used to evaluate its performance. This ratio provides a sufficiently large amount of test data, allowing the evaluation to better reflect the model's performance in real-world scenarios. However, since the amount of training data is reduced, the model may not receive enough variety of information to create an accurate representation, which could lead to decreased generalization quality if the architecture and parameters are not properly tuned.

Table 5. Test table ratio 0.3

Aspect	Details
Data Type	Indonesian audio, transcription of everyday conversation sentences.
Data Quantity	999 audio files
Data Distribution	Train: 699 (70%), Test: 300 (30%)
Model Architecture	LSTM (input=40, hidden=128, 2 layer) + FC (128 → 999)
Training Epochs	10
Loss Function	Cross Entropy Loss
Optimizer	Adam
Feature Extraction	MFCC (40 coefficients)
Training Results (Loss)	Epoch 1: 7.1956 → Epoch 10: 5.5395
Word Error Rate (WER)	1.00
Character Error Rate (CER)	0.78
Average Test Loss	13.7803
Error Pattern	The model output frequently repeats the same phrase across all test data.
Conclusion	The model has not achieved generalization, likely due to overfitting and insufficient data variation

Testing with a ratio of 0.3 using an audio dataset. There are a total of 999 audio files, divided into 699 data for training and 300 data for testing. The model used has an LSTM structure with an input size of 40, a hidden size of 128, two layers, and a fully connected layer sized 128 to 999 outputs. Training was conducted for 10 epochs using the Cross Entropy Loss loss function and the Adam optimizer. The feature used to process the input was MFCC with 40 coefficients. The training results showed a decrease in the loss value from 7.1956 in the first epoch to 5.5395 in the tenth epoch.

Evaluation on the test data resulted in a Word Error Rate of 1.00 and a Character Error Rate of 0.78, with an average loss during testing of 13.7803. During testing, an error pattern was found where the model often repeated the same phrase across all test data. This indicates that the model is not yet able to generalize well, possibly due to overfitting and a lack of variation in the training data.

7. Overall Evaluation Based on WER

The Overall Word Error Rate (WER) of your model is approximately 1.13. Evaluation: An Overall WER value of 1.13 is considered high. As a benchmark, speech-to-text models that perform well on general tasks typically have a WER below 10-15%, and can even be less than 5% on clean data and specific tasks. A WER of 1.13 indicates that, on average, the number of word errors (replacements, additions, or omissions) exceeds the number of words in the original sentence. This suggests that the model is not yet sufficiently accurate in transcribing audio on the test data.

Key Points from WER-Based Evaluation:

a. Low Accuracy: A high WER value directly reflects the frequency with which your model makes mistakes in recognizing and generating words from audio.

b. Need for Improvement: These results clearly indicate the need for significant improvements to the model, data, or training process in order to achieve better accuracy.

c. Further Analysis Required: To understand the causes of the high WER, it is important to review the WER table per sentence and the WER distribution graph.

This step will help identify the most dominant types of errors and determine whether the difficulties occur in all sentences or only in certain parts. Overall, the evaluation based on WER shows suboptimal performance on this test data, so additional efforts are needed to improve its accuracy.

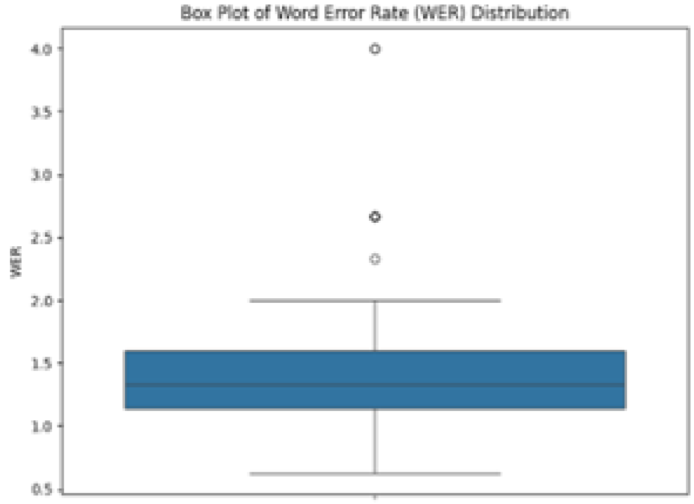


Figure 7. Boxplot WER

Meanwhile, (Figure 6) shows a visual representation of the distribution of Word Error Rate (WER) values for each sentence in your test data. Let's take a look at its parts. The center line inside the box shows the median of the WER values. This is the middle value of all test sentence WERs. About half of the sentences have a WER below this value, and half above it. Blue Box This part of the box represents the middle 50% of WER data. The lower boundary of the box is the First Quartile (Q1). 25% of sentences have a WER below this value. The upper boundary of the box is the Third Quartile (Q3). 75% of sentences have a WER below this value. The height of the box indicates the Interquartile Range

(IQR), which is the spread of WER for 50% of the data in the middle. Vertical Lines (Whiskers) The lines extending above and below the box indicate the range of WER values within the normal limits (typically 1.5 times the IQR from the edges of the box).

This shows the spread of data outside the middle 50%, excluding outliers. Points Outside the Lines (Outliers) Individual points above or below the whiskers are outliers. These are sentences with WER values that are significantly higher or lower than most other sentences. In the box plot shown, these points indicate sentences that are very difficult for the model to transcribe.

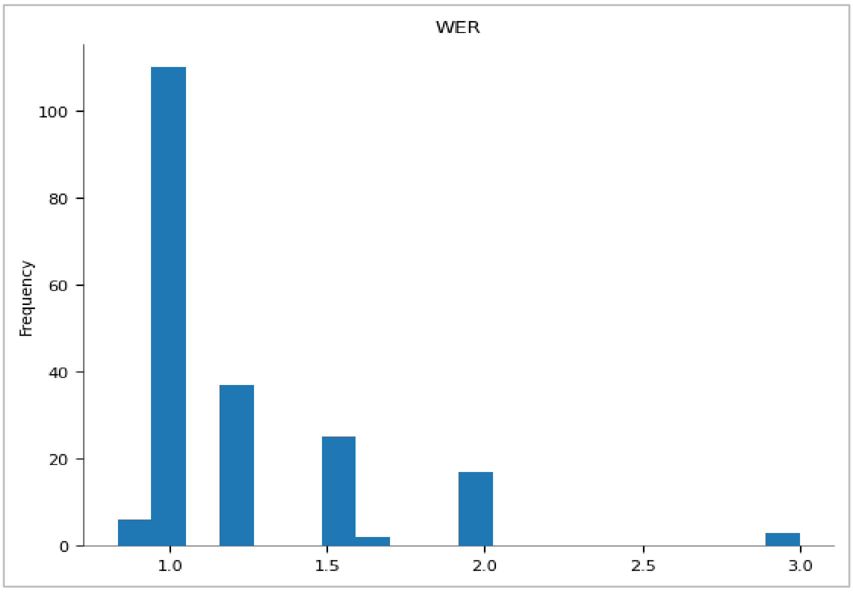


Figure 8. Histogram WER

1. Frequency Distribution:
   a. The histogram records a dominant peak at WER 1.0 with a frequency close to 100, far exceeding the previous histogram which only reached a peak of around 40 in the 1.0-1.5 range.
   b. At WER 1.5, the histogram shows a frequency of around 40, while the previous histogram also had a significant peak in this range, though lower (around 35).
   c. The frequency in the histogram drops sharply starting from WER 2.0 (around 20 and continues to decrease), similar to the decreasing pattern in the previous histogram, but with a smaller frequency (around 25 at 2.0).
2. Key Differences:
   a. The latest histogram shows a much greater concentration of data at WER 1.0, indicating that most model predictions are now consistent at an error rate of around 1.0, in contrast to the more scattered distribution in the previous histogram.
   b. The WER range above 2.0 shows a very low frequency in both histograms, but the new histogram experiences a steeper decline after WER 1.5, indicating fewer incidents of high error.

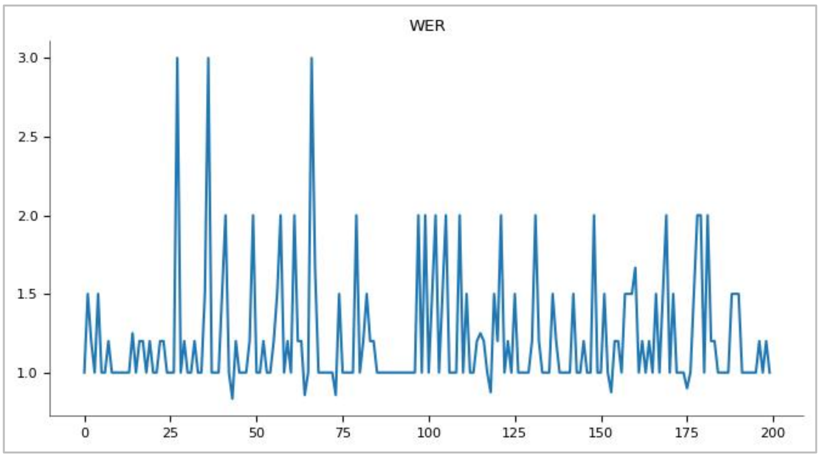


Figure 9. WER data value graph

The graph shows WER (Word Error Rate) data plotted against time or number of samples, ranging from 0 to 200. The x-axis (horizontal) represents time or number of samples, while the y-axis (vertical) shows WER values between 0 and 3.0. The data exhibits significant fluctuations, with sharp spikes approaching 3.0 at several points, followed by declines and minor variations, indicating inconsistent system performance, possibly due to noise, sound variations, or complex input data. The decline following the spikes may indicate system adaptation or improved input quality, though the ongoing fluctuations suggest instability.

This graph is relevant for analyzing the performance of automatic speech or text recognition systems, discussing factors such as audio quality, algorithm models, or data processing, and proposing improvements based on visible patterns.

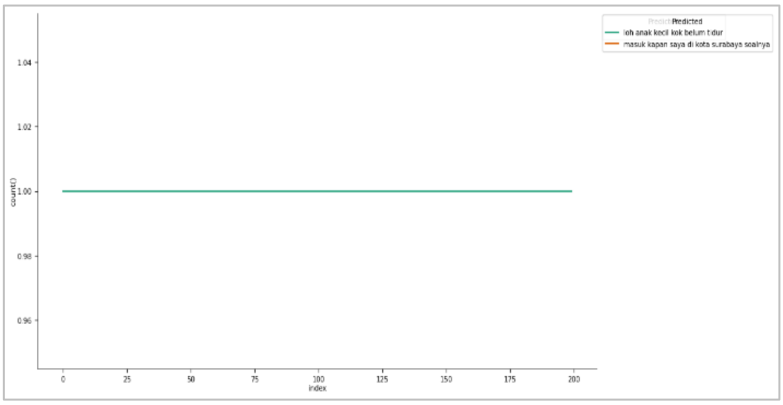


Figure 10. Time series data graph

(Figure 9) shows a linear relationship between two variables. The only straight green line visible in this graph illustrates that the count () value for the two text predictions*, “loh anak kecil kok belum tidur”* and *“masuk kapan saya di kota surabaya soalnya,”* is exactly the same, namely 1.00. This value remains unchanged throughout the entire index range (0-200). The fact that only one line appears indicates that both predictions consistently produce identical outputs.

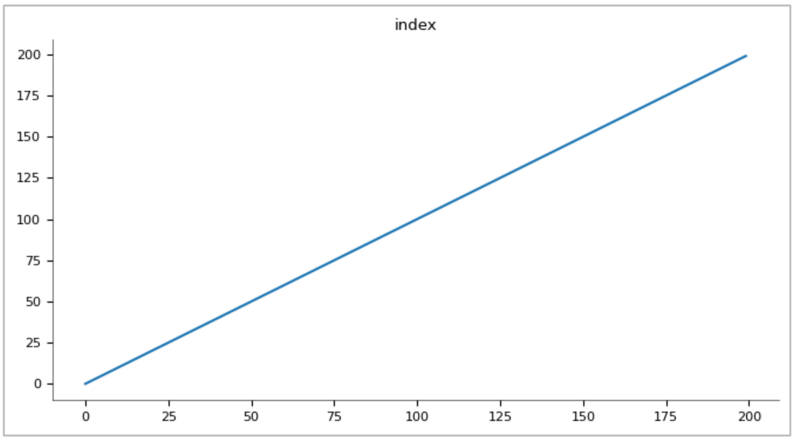


Figure 11. Linear Graph

(Figure 10) This graph shows a positive linear relationship between the x-axis and the y-axis. As the value on the x-axis increases, the value on the y-axis also increases at a constant rate. The line starts at the point (0, 0) and ends around the point (200, 200). This shows a proportional relationship, where the value on the y-axis is equal to the value on the x-axis.

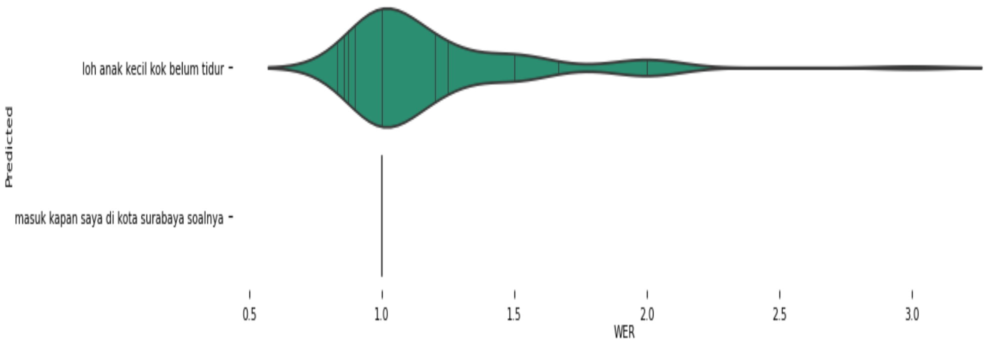


Figure 12. Violin plot graph 1

(Figure 11) A data visualization graph showing the distribution of data (probability density) for each category represented by the Y-axis (Predicted). The width of the "violin" shape indicates the density of data at a specific value on the X-axis (index). The wider the violin, the more data is concentrated at that point. In this case, the graph shows that the prediction *"loh anak kecil kok belum tidur?"* has a wide and even distribution of data across the entire index range from approximately 0 to 200. Conversely, the prediction *"masuk kapan saya di kota surabaya soalnya?"* only has data at a single index point, represented by a thin vertical line.

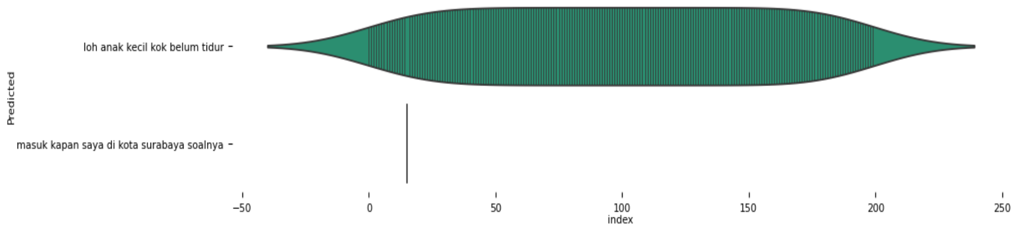


Figure 13. Violin plot graph 2

(Figure 12) Similar to the previous graph, this is the second Violin Plot, but with a different X-axis, namely WER (Word Error Rate). This graph visualizes the distribution of WER values for each text prediction. For the prediction *"loh anak kecil kok belum tidur"* (Why isn't the little kid asleep yet?), the graph shows a varied distribution with most of the data concentrated around a WER value of 1.0, but with a “tail” extending to around 3.0, indicating that there are some cases with higher WER values. For the prediction *“masuk kapan saya di kota surabaya soalnya”,* the graph only shows a single vertical line at a WER value of 1.0. This indicates that the data for this prediction is highly consistent and always has the same WER value, which is 1.0.

Table 6. Comparison of Regional Language TTS Studies

No.	Researcher	Language	Method	Accusation / WER	Dataset
1.	[1]	*Indonesia & Jawa*	Deep Learning (MLP + CTC)	*Indonesia: 65%* *Jawa: 57%*	50 words, single-word recognition
2.	Our Porposed	*Walikan Malang*	LSTM	WER Average: 0.69 (69%)	1000 *Walikan* language sentence audio samples

Two studies discussed local language speech recognition in Indonesia, namely Malang *Walikan* and Indonesian/Javanese. The first study used LSTM with 1,000 *Walikan* audio sentences, resulting in a WER of 69%, which indicates low accuracy due to data limitations and language complexity. The second study by Teguh used an MLP+CTC model with a dataset of 50 single words, achieving an accuracy of 65% for Indonesian and 57% for Javanese, influenced by dialectal and phonetic variations.

Both studies faced similar challenges in terms of dataset size and quality limitations. Improvements can be made by expanding data collection through community involvement, using synthetic data, or integrating cross-language data. From a model perspective, selecting appropriate architectures such as Transformer or Conformer, along with adding phonetic or dialect features, can enhance the performance of speech recognition systems for local languages.

IV. Conclusion

This study aims to develop a Long Short-Term Memory (LSTM)-based Speech-to-Text (STT) system for recognizing the *Walikan* Malang language. Based on literature review, methodological planning, and expected results, it can be concluded that the use of the LSTM model has great potential in recognizing complex patterns of the *Walikan* language. This model offers the ability to process sequential data and remember context, which is highly suitable for handling the phonetic variations and unique structure of the *Walikan* Malang language.

The use of authentic voice data as training and testing data is expected to improve the system's accuracy in recognizing and converting *Walikan* speech into text. With effective model training, this system is expected to reduce errors in speech recognition, which can be measured through metrics such as Word Error Rate (WER) and Character Error Rate (CER). This research develops a Long Short-Term Memory (LSTM) model-based Speech-to-Text (STT) system to recognize and convert *Walikan* Malang speech into text. The Malang *Walikan* language has a unique linguistic structure in the form of word reversal, which poses a challenge in speech recognition. The data used consists of 1,000 sentences collected from social media and direct recordings. The data is processed using Mel Frequency Cepstral Coefficients (MFCC) and then used to train the LSTM model.

System performance was evaluated using Word Error Rate (WER), Character Error Rate (CER), and Average Test Loss metrics. The best results obtained showed a WER value of 1.0 on a data split of 699:300, a CER of 0.78 on a split of 799:200, and an Average Test Loss of 11.0147 on a split of 299:700.

The high Average Test Loss value indicates the model's difficulty in minimizing prediction errors, which is likely due to the model's mismatch with the data patterns or overfitting. To improve the model's performance, it is recommended to improve the quality of the training data, optimize the parameters, and apply regularization techniques. However, this research still faces challenges in terms of data availability and the diversity of speakers used in the training process. Therefore, further development is needed to ensure the system can operate optimally under various conditions and environments.

V. Acknowledgment

I would like to express my sincere and heartfelt gratitude to Widyagama University Malang for the opportunity, facilities, and supportive academic environment provided to me to pursue my education and successfully complete this research.

References

[1] T. P. Laksono, "Speech To Text Untuk Bahasa Indonesia," *Skripsi*, 2018, [Online]. Available: https://dspace.uii.ac.id/handle/123456789/10756

[2] I Komang Setia Buana, "Implementasi Aplikasi Speech to Text untuk Memudahkan Wartawan Mencatat Wawancara dengan Python," *J. Sist. dan Inform.*, vol. 14, no. 2, pp. 135–142, 2020, doi: 10.30864/jsi.v14i2.293.

[3] L. Vinanda and M. J. Lelono, *Aku, kami, dan mereka : mensyukuri perbedaan*. 2013.

[4] N. Aini Lailla Asri, R. Ibnu Adam, and B. Arif Dermawan, "Speech Recognition Untuk Klasifikasi Pengucapan Nama Hewan Dalam Bahasa Sunda Menggunakan Metode Long-Short Term Memory," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 2, pp. 1242–1247, 2023, doi: 10.36040/jati.v7i2.6744.

[5] C. Ittichaichareon, S. Suksri, and T. Yingthawornsuk, "Pengenalan Ucapan menggunakan MFCC," pp. 28–29, 2012.

[6] Riakesdas, "Bab 1 pendahuluan," *Pelayanan Kesehat.*, no. 2018, pp. 3–13, 2018, [Online]. Available: http://repository.usu.ac.id/bitstream/123456789/23790/4/Chapter I.pdf

[7] W. Gunawan, H. Sujaini, and T. Tursina, "Analisis Perbandingan Nilai Akurasi Mekanisme Attention Bahdanau dan Luong pada Neural Machine Translation Bahasa Indonesia ke Bahasa Melayu Ketapang dengan Arsitektur Recurrent Neural Network," *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 3, p. 488, 2021, doi: 10.26418/jp.v7i3.50287.

[8] J. Oruh, S. Viriri, A. Senior, and D. A. N. Lainnya, "Memori Jangka Panjang Jangka Pendek Saraf Berulang Jaringan untuk Pengenalan Ucapan Otomatis," vol. 10, pp. 30069–30079, 2022.

[9] R. G. Gunawan, Erik Suanda Handika, and Edi Ismanto, "Pendekatan Machine Learning Dengan Menggunakan Algoritma Xgboost (Extreme Gradient Boosting) Untuk Peningkatan Kinerja Klasifikasi Serangan Syn," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 3, no. 3, pp. 453–463, Dec. 2022, doi: 10.37859/coscitech.v3i3.4356.

[10] M. Zlobin and V. Bazylevych, "Bayesian Optimization For Tuning Hyperparametrs Of Machine Learning Models: A Performance Analysis In Xgboost," *Computer systems and information technologies*, no. 1, pp. 141–146, Mar. 2025, doi: 10.31891/csit-2025-1-16.

[11] P. Fajar and Y. I. Aviani, "Hubungan Self-Efficacy dengan Penyesuaian Diri: Sebuah Studi Literatur," vol. 6, no. 1, pp. 2186–2194, 2022.

[12] A. Akbar *et al.*, “Pelatihan Dan Pengembangan Sdm Dalam Perspektif Ilmu Manajemen: Sebuah Studi Literatur,” 2023.

[13] U. Mufidah and M. Siahaan, “Perancangan Aplikasi Perbanndingan Harga Produk (Historical Data) Menggunakan Teknik Scraping Web,” 2021.

[14] M. Rizqi, A. Rustiawan, and P. T. Prasetyaningrum, “Analisis Sentimen Terhadap Klinik Natasha Skincare di Yogyakarta Dengan Metode Google Review,” *Journal of Information Technology Ampera*, vol. 5, no. 1, pp. 2774–2121, 2024, doi: 10.51519/journalita.v5i1.556.