



End-to-End Text-to-Speech for Minangkabau Pariaman Dialect Using Variational Autoencoder with Adversarial Learning (VITS)

Muhammad Dzaki Fakhrezi¹, Yusra², Muhammad Fikry³, Pizaini⁴, Suwanto Sanjaya⁵

¹⁻⁵Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia

Article Information

Article History:

Submitted: April 29, 2025

Revision: May 15, 2025

Accepted: June 11, 2025

Published: June 30, 2025

Keywords

Machine Learning

Natural Language Processing

Variational Inference with adversarial learning for end-to-end Text-to-Speech

Mean Opinion Score

Minangkabau

Pariaman

Correspondence

E-mail: yusra@uin-suska.ac.id

A B S T R A C T

Language serves as a medium of human communication to convey ideas, emotions, and information, both orally and in writing. Each language possesses vocabulary and grammar adapted to the local culture. One of the regional languages that enriches Indonesian as the national language is Minangkabau. This language has four main dialects, namely Tanah Datar, Lima Pulu Kota, Agam, and Pesisir. Within the Pesisir dialect, there are several variations, including the Padang Kota, Padang Luar Kota, Painan, Tapan, and Pariaman dialects. This study discusses the application of Text-to-Speech (TTS) technology to the Minangkabau language, specifically the Pariaman dialect, using the Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech (VITS) method. This dialect needs to be preserved to prevent extinction and supported through technological development that broadens its use. The VITS method was chosen because it is capable of producing natural and high-quality speech. The research stages include voice data collection and recording, VITS model training, and speech quality evaluation using the Mean Opinion Score (MOS). The final results show a score of 4.72 out of 5, indicating that the generated speech closely resembles the natural utterances of native speakers. This TTS technology is expected to support the preservation and development of the Minangkabau language in the Pariaman dialect, as well as enhance information accessibility for its speakers.

This is an open access article under the CC-BY-SA license



1. Introduction

Bahasa Language functions as a medium for humans to express ideas, emotions, and information, both in spoken and written forms. Each language has its own vocabulary and grammar adapted to the local culture. Indonesia has approximately 724 regional languages distributed across the country [1]. The Minangkabau

language is one of the regional languages that significantly contributes to the linguistic diversity of Indonesian and is among the languages with the largest number of speakers in the country [2]. The Minangkabau language has various linguistic variations distributed across different regions of West Sumatra. These variations are commonly referred to as dialects [3].

A dialect is a fundamental form of language variation that is often associated with particular individuals or groups within a society [4]. The Minangkabau language dialects are divided into four main groups, namely the Tanah Datar, Lima Puluh Kota, Agam, and Pesisir dialects. Specifically, within the Pesisir dialect, there are several sub-dialects, including Padang Kota, Padang Luar Kota, Painan, Tapan, and Pariaman [3]. Regional languages, such as the Minangkabau language in the Pariaman dialect, require development efforts to preserve their cultural heritage. However, these efforts face challenges, one of which is the tendency of parents to prioritize teaching the national language to their children, resulting in the gradual marginalization of regional languages [5].

This situation threatens the continuity of regional languages, including the Minangkabau language of the Pariaman dialect. Indonesia has as many as 742 regional languages spread from Sabang to Merauke. Of this number, 737 languages are still actively used by their speakers, while the remaining 5 are endangered due to a continuous decline in the number of speakers [6]. Therefore, efforts are needed to prevent the extinction of regional languages, including the Minangkabau language of the Pariaman dialect. With the advancement of technology and the internet, the Pariaman dialect of Minangkabau can be preserved through Text-to-Speech (TTS) technology, which converts text into speech [7]. TTS technology is a branch of knowledge within natural language processing. TTS is a system that converts text into speech, generated through cloud-based systems with application development using Amazon Polly. [8]. A significant development in TTS is marked by the emergence of Tacotron 2, which combines mel-spectrogram prediction with the WaveNet vocoder, enabling the generation of speech quality that closely resembles natural human utterances [9]. TTS can enhance the quality of speech synthesis, where the wav2vec2.0 model at the 9th layer demonstrates the best performance for both types of TTS, whether using structured reading or spontaneous speech. [10]. Speech recognition is a technology that enables devices to recognize and understand spoken words through the digitization of speech signals and pattern matching. At present, modern deep learning-based methods such as wav2vec 2.0 and Whisper are capable of improving accuracy through more efficient end-to-end approaches [11].

Non-autoregressive TTS models are capable of producing competitive speech quality, with synthesis speeds up to 46.7 times faster than autoregressive models such as Deep Voice 3 (DV3) [12]. TTS can improve reading skills by providing clear pronunciation, a wide variety of reading materials, and better accessibility for individuals who experience difficulties in reading. [13]. TTS also includes models such as FastSpeech 2, which is one of the non-autoregressive TTS models designed to improve speech quality. [14]. FastSpeech 2 is able to address the one-to-many mapping problem in TTS systems and produce better speech quality by incorporating more accurate variation information such as pitch, energy, and duration. [14]. In addition to FastSpeech, there is also Mellotron, which offers a different approach to speech synthesis. Mellotron is a multispeaker speech synthesis model capable of generating expressive and singing voices without requiring specific emotional or singing data, by explicitly utilizing rhythm and pitch contour information. [15]. In TTS technology, one of the methods for managing speech data is the Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech (VITS). This method employs a Variational Autoencoder (VAE) and adversarial training to improve waveform quality. [16].

This model demonstrates better performance compared to baseline models in generating more natural multi-speaker emotional TTS. [17]. In speech synthesis experiments, the VITS model demonstrated superior performance compared to FastSpeech2+HiFi-GAN, particularly in terms of the naturalness of the generated speech. [18]. VITS was selected in this study due to its high synthesis speed and its ability to generate recording-level quality speech for both neutral and sad styles. VITS outperforms the best public TTS systems with a Mean Opinion Score (MOS) of 4.39, which is close to natural speech quality. [19]. In this study, the speech quality of the TTS system was evaluated using the Mean Opinion Score (MOS) method with the involvement of native speakers. [20]. The VITS method was implemented and evaluated using MOS with a 95% confidence interval on the LJ dataset, yielding a MOS score comparable to natural speech. [19]. The HiFi-GAN generator variants recorded MOS scores of 3.77, 3.69, and 3.61, outperforming AR- and flow-based models, and demonstrating strong generalization capability to unseen speakers. [21]. The high MOS score makes this method highly suitable for application in this study.

Based on various sources reviewed in the background, there is a need for a technology capable of producing speech with a high degree of naturalness. This is particularly important in the context of preserving

and developing local language technologies, especially within the field of natural language processing. Therefore, in this study, the Minangkabau language of the Pariaman dialect was selected as the primary object for TTS system development. This selection aims to build a single-speaker TTS model that can accurately represent the phonetic characteristics of the Minangkabau Pariaman dialect, while also supporting the broader advancement of regional language speech technologies.

2. Method

The research methodology covers the stages undertaken in conducting the TTS study for the Minangkabau language of the Pariaman dialect using the VITS method, as illustrated in Figure 1.

The research methodology includes the stages undertaken in conducting the TTS study for the Minangkabau language of the Pariaman dialect using the VITS (Variational Inference with Adversarial Learning for End-to-End Text-to-Speech) method developed by Kim et al. [19]. The research stages are presented in Figure 1.

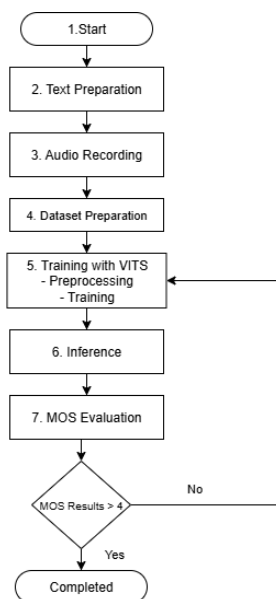


Figure 1. Research Stages

Figure 1 presents the complete flow of the research stages, covering the initial process of data collection through to the formulation of conclusions. A detailed description of each stage is provided in the following sections.

2.1. Text Preparation

The initial stage of this research involved data collection through the preparation of texts to be read by a speaker. The recorded speech from these readings was then used as the dataset for TTS model training. The language used in this study was Minangkabau with the Pariaman dialect. Speech data were obtained from an adult female native speaker of this dialect. The speaker was born, raised, and resided in Pariaman until the age of 21, thereby representing the accent and phonetic characteristics of the Pariaman dialect. In this process, the speaker read 500 sentences that had been prepared as a script. After the text preparation and research outputs were completed, a validation process was conducted by a Minangkabau language expert specializing in the Pariaman dialect, residing in the local area, to ensure the accuracy and appropriateness of the language used in this study.

2.2. Audio Recording

Subsequently, the audio recording process was carried out in WAV 16-bit mono PCM format, with each audio clip ranging from 1 to 10 seconds in duration. All recordings were maintained to be free from background noise and distortion, and to avoid long silent pauses at the beginning, middle, or end. After ensuring compliance with these technical requirements, the next stage was the voice recording process, which was conducted through the following steps:

- a. The first step was selecting a location away from noise sources to minimize excessive noise during voice recording.
- b. The recording process was then carried out using a script, with each sentence consisting of 4–5 words and ranging from 1 to 10 seconds in duration.
- c. After completing all the recordings, a cleaning process was performed using the Audacity web platform with an audio sample rate of 22,050 Hz, ensuring that the speech samples maintained good quality without disruptive background noise.
- d. Finally, after all recordings had been cleaned, the format was converted from MP3 to WAV and stored in a single folder.

2.3. Dataset Preparation

In this study, the dataset consisted of 500 recorded sentences in the Minangkabau language of the Pariaman dialect, divided into 450 training data and 50 testing data. All recordings were produced by a native speaker of the Pariaman dialect, an adult female aged 48, to ensure authenticity of accent and pronunciation. The audio files were recorded in WAV format to preserve sound quality. The dataset was organized into two main components: a Comma-Separated Values (CSV) file for the text and the audio dataset stored in a Google Drive directory. All audio recordings were placed in a folder named 'wav,' while the sentence texts were documented in a 'metadata.csv' file following the LJSPEECH format, which included columns for the file name, the text in Indonesian, and its translation in Pariaman. The rules for structuring the metadata.csv file are explained as follows:

- a. In the first column of metadata.csv, the file name is written without the prefix 'wavs/' and without a period ('.') at the end. For example, the file '1.wav' is simply written as '1'.
- b. The second column contains the original text exactly as written.
- c. The third column contains the pronunciation of the text in the second column. If there is no difference between the text and its pronunciation, simply copy the content of the second column into the third. For example, if the second column contains 'satu', then the third column is also written as 'satu'.
- d. Use the symbol | as the column separator instead of a semicolon (;). An example of a row format in metadata.csv is: '1 | Satu | Satu'.

2.4. Training with VITS

At the training stage, the collected data were uploaded to Google Drive to facilitate access through Google Colab. The training process employed the Python programming language to process the previously divided dataset, aiming for the model to deliver the best performance among all candidates in terms of speech naturalness, with scores comparable to the reference voice, thereby demonstrating synthesis quality close to that of native speakers [13]. This resulted in a model capable of producing clear and natural speech in accordance with the research requirements. The workflow stages for training the VITS model are as follows:

a. *Text Encoder*

This stage begins by converting the text into phoneme representations, after which the Text Encoder transforms them into latent representations in the form of numerical data containing essential phoneme information. The phonemes, as the smallest units of sound, are then processed in the subsequent stages.

b. *Projection*

The output of the Text Encoder is then projected into latent distribution parameters during the Projection stage, enabling the model to statistically link text representations with audio.

c. *Monotonic Alignment Search*

Next, this stage seeks alignment between the text representations from the encoder and the audio by mapping each phoneme to its precise duration and position within the audio sequence.

d. *Flow*

After alignment, the flow stage transforms the latent distribution into a distribution that matches the audio characteristics, allowing random inputs to be converted into more realistic audio outputs.

e. *Stochastic Duration Predictor*

In this stage, the model stochastically predicts the duration of each phoneme by adding noise, resulting in audio signal lengths that are more natural and less rigid.

f. *Posterior Encoder*

The Posterior Encoder processes the audio's linear spectrogram to generate latent representations (z), which are then sliced to create smaller, more manageable signals for encoding, ultimately producing speech corresponding to the input text.

g. *Decoder*

In the final process, the decoder reconstructs the audio signal from the encoding to produce raw speech waves that correspond to the input text and sound natural.

h. *Raw Waveform*

The raw waveform is the final output of the Text-to-Speech process, representing the speech signal in its original form.

2.5. Evaluation Using the Mean Opinion Score (MOS)

The evaluation stage is the final part of this research process, conducted after the successful implementation of the system. In this study, the evaluation was performed using the Mean Opinion Score (MOS) method to assess the quality of the generated speech. MOS is a subjective evaluation method that involves human judgment to measure speech quality based on the average scores provided by listeners. This method is commonly used in TTS systems to evaluate the accuracy and naturalness of speech. In this study, the researchers applied a MOS rating scale from 1 to 5 to assess speech quality and the correspondence of the dataset to the original voice, while also determining the final score of the system-generated speech [22]. However, although MOS is effective, this method has limitations due to its subjective nature and the substantial resources it requires, making it necessary to complement it with objective evaluation metrics based on neural networks [23]. The MOS method is effectively used in evaluating TTS systems, as demonstrated in studies showing that the TTS model achieved the highest score of 4.09, followed by Transformer-L with a score of 4.06, and P-FastSpeech-L with a score of 3.78 [24].

MOS evaluation was conducted on 50 test samples converted into audio using the trained model to assess speech quality and accuracy. Five listeners were involved as evaluators, following common practice in TTS research, where 5–10 listeners are considered sufficient to obtain representative subjective assessments, provided the evaluation is consistent and controlled. If the obtained scores fell below 4, the model training was repeated until scores exceeded 4, indicating that the speech was natural and met the expected quality. Furthermore, the practice of using MOS has also been applied in recent research on speech quality assessment by Coldenhoff et al. (2024) [25]. As shown in Table 1, the MOS rating scheme refers to the standard 1–5 MOS scale, as described by Jiang et al. (2020) [26].

Table 1. Score MOS

MOS	Quality Rating	Score Degradation
5	Excellent	No noticeable errors
4	Good	Errors present but not disruptive
3	Fair	Errors present and slightly disruptive
2	Poor	Errors present and disruptive
1	Bad	Errors present and highly disruptive

3. Results and Discussion

3.1. Data Collection

Dataset collection is a crucial initial stage in the implementation of this research. This step was undertaken to ensure that the data used were not only relevant but also of sufficient quality to optimally support the VITS model training process. Accuracy, completeness, and alignment of the data with the characteristics of the Minangkabau Pariaman dialect were prioritized in this process, as non-representative data could significantly affect the model's outcomes and performance. Additionally, a common challenge in TTS systems is interference from multi-speaker environments, which can reduce speech recognition accuracy and synthesis clarity. The VoiceFilter study proposed a speaker-conditioned speech separation approach to isolate the target voice from multi-speaker mixtures, thereby maintaining the quality of the input [27]. Therefore, the dataset collection process was conducted systematically and through several structured stages to ensure the validity and integrity of the acquired data. The limited amount of data represents a major challenge in developing TTS systems for regional languages, which are considered low-resource. However, recent studies have shown that weakly labeled data can still be effectively utilized to build competitive speech translation models for low-resource languages [28]. The stages carried out in this process are described as follows.

3.1.1. Text

The Minangkabau Pariaman dialect data used in this study were obtained from two main sources. The first source came from interviews with the Head of the Nagari Consultative Body (Bamus Nagari) of Koto Dalam, Mr. Burhanudin, resulting in approximately 500 sentences in the Pariaman dialect along with their Indonesian equivalents. The second source consisted of written texts, namely the folk tale books *Kaba Siti Baheram* and *Kaba Sabai Nan Aluih*. The sentences from these books were reviewed together with the speaker to ensure their consistency with the Pariaman dialect. The combination of these two sources aimed to enrich the linguistic variation and contextual diversity in the dataset.

Table 2. Minangkabau Pariaman Dialect Texts

No.	Minangkabau Pariaman Dialect	Indonesian Translation
1.	Jan Makan dibiliak Makan tu dilua	Jangan makan dikamar makan tu diluar
2.	Mukasui awak datang kamari alah taniak dari dulu	Maksud saya datang kesini sudah terniat dari dulu
3.	Kasiko lah duduak, duduak baselo dimuko mamak, danga an bana	Kesinilah duduk, duduk bersila didepan paman, dengarkan benar
4.	Rundiangan elok-elok sabalun habih kato urang	Rundingkan baik-baik sebelum selesai kata orang
5.	Pai lah ang mandi lu, lah ba baun badan ang mah	Pergi lah kamu mandi dulu, sudah bau badan kamu ini
6.	Kok baranak dirumah urang, anak diasuah ka nan elok	Kalau beranak dirumah orang, anak diasuh ke arah yang baik
7.	Anak di pangku kamanakan dibimbiang	Anak dipangku keponakan dibimbing
8.	Nan pandai awak surang, kok nan buruak parangai urang	Yang pandai kita sendiri, kalau yang buru parangai orang
9.	Jan mandi hujan beko ang damam	Jangan mandi hujan nanti kamu demam
10.	Jan mamauang beko hilang pangana ang	Jangan bermenung, nanti hilang pikiran kamu

Table 2 presents a small portion of the complete dataset of 500 sentences, encompassing various types of everyday expressions such as greetings, statements, questions, proverbs, and narratives related to local cultural contexts. These sentences were carefully selected to represent the structural and lexical variations characteristic of the Minangkabau Pariaman dialect, aiming to enhance both the representational coverage and the accuracy of the developed TTS model.

3.1.2. Audio Recording

The voice recordings were conducted by a native speaker of the Minangkabau Pariaman dialect, Yetri Martini, a 47-year-old adult female who works as a homemaker. She was selected as the speaker due to her deep understanding of and active use of the Minangkabau Pariaman dialect in daily life. All collected texts were read aloud and recorded by the speaker using a smartphone device.

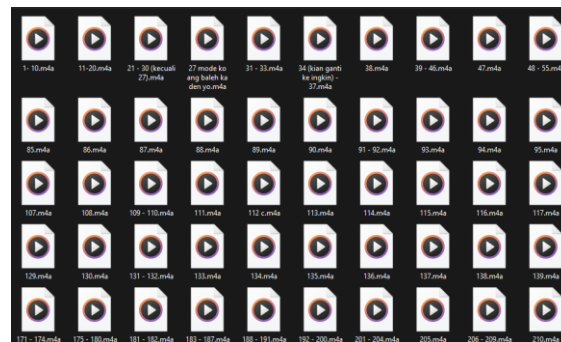


Figure 2. Voice Recording

Figure 2 illustrates the results of the voice recordings, which were saved in the .m4a audio format, corresponding to the default format of the device used.

3.1.3. Cleaning Data Audio

The audio cleaning process was performed manually using Audacity software. The cleaning stages included trimming sections of the recordings containing long pauses at the beginning, middle, or end of the audio files. Additionally, adjustments were made to technical parameters such as sample rate, encoding, number of channels, and audio format. The resulting audio recordings were saved in .wav format with 16-bit resolution, using a single mono audio channel and Pulse-Code Modulation (PCM) encoding, commonly employed in digital audio signal processing. The audio files were set to a sample rate of 22,050 Hz, which is technically considered sufficient to capture human speech quality without generating excessively large file sizes. This configuration was deemed suitable for training a machine learning-based TTS system, as it balances computational efficiency with the clarity of audio data necessary for building accurate and natural-sounding speech models.

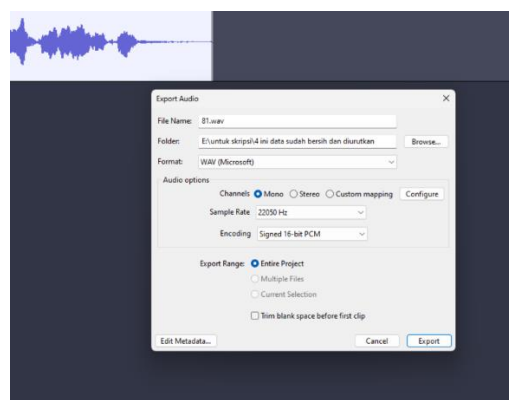


Figure 3. Cleaning Data

Figure 3 illustrates the audio cleaning process, in which long pauses at the beginning, middle, or end of sentences were manually removed using the cut-to-cut method. Once the cleaning process was completed and the results were considered optimal, the researchers adjusted the configuration to export the cleaned .wav files. This configuration included using 16-bit PCM audio format and a sample rate of 22,050 Hz, indicating the number of audio samples taken per second.

3.2. Dataset Preparation

The dataset, consisting of cleaned and format-adjusted audio files, was divided into 450 training samples and 50 testing samples for the model evaluation stage. A 90% training and 10% testing split was chosen following common practices in machine learning, where the majority of data is allocated for training to allow the model to learn optimally, while a smaller portion is used to test the model's generalization. Meanwhile, the text dataset obtained from the transcripts of the recordings in the previous Excel file was converted into Comma Separated Values (CSV) format and saved as metadata.csv.

3.2.1. Metadata Creation

Table 3. Training Data Metadata

1	Jan Makan dibiliak Makan tu dilua	jan makan dibiliak makan tu dilua
2	Mukasuik awak datang kamari alah taniek dari dulu	mukasuik awak datang kamari alah taniek dari dulu
3	Kasiko lah duduak, duduak baselo dimuko mamak, danga an bana	kasiko lah duduak, duduak baselo dimuko mamak, danga an bana
4	Rundiangan elok-elok sabalun habih kato urang	rundiangan elokelok sabalun habih kato urang
5	Pai lah ang mandi lu, lah ba baun badan ang mah	pai lah ang mandi lu, lah ba baun badan ang mah
6	Kok baranak dirumah urang, anak diasuah ka nan elok	kok baranak dirumah urang, anak diasuah ka nan elok
7	Anak di pangku kamanakan dibimbiang	anak di pangku kamanakan dibimbiang
8	Nan pandai awak surang, kok nan buruak parangai urang	nan pandai awak surang, kok nan buruak parangai urang
9	Jan mandi hujan beko ang damam	jan mandi hujan beko ang damam
10	Jan mamauang beko hilang pangana ang	jan mamauang beko hilang pangana ang

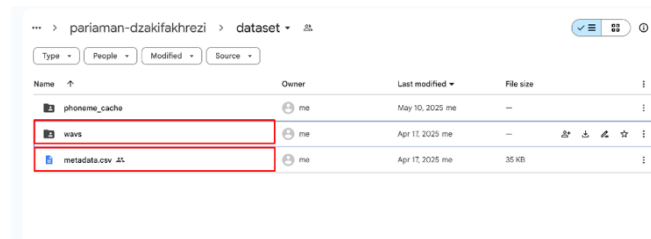
Table 3 presents a small portion of the complete training dataset, which consists of 450 samples in total. These data were extracted from the metadata.csv file, organized according to the LJSpeech standard format, a common format used in TTS system development. In its original form, the metadata.csv file is not a table but a text file where each line contains three main components separated by the | character: (1) the audio file name, (2) the original or prepared transcript, and (3) the cleaned transcript. The use of the | separator allows the TTS system to recognize the relationships between components in a structured manner, thereby supporting the efficiency of the model training and inference processes.

Additionally, the LJSpeech format implements additional rules such as writing numbers in word form (e.g., the number 35 is written as 'thirty-five') and prohibiting abbreviations (e.g., 'dgn' must be written as 'dengan', and 'sbg' as 'sebagai'). These rules aim to maintain consistency and cleanliness of the text data. The tabular presentation in this article is intended to visualize the data structure more clearly, facilitating readability and manual verification, even though the original format is not in table form.

3.2.2. Dataset Folder Structure

After the audio data were divided into 450 samples for use as training data in the model training process, the next step was to upload the data to Google Drive as part of the training preparation. The data

were stored in a folder named 'dataset.' Organizing the data in a structured and orderly manner aims to minimize errors and ensure ease of access during the model training process.



Name	Owner	Last modified	File size
phoneme_cache	me	May 10, 2025	—
wavs	me	Apr 17, 2025	—
metadata.csv	me	Apr 17, 2025	35 KB

Figure 4. Dataset Folder Structure

Figure 4 shows the dataset directory structure, consisting of the 'wavs' folder and the metadata.csv file, stored on Google Drive and used during the model training stage in Google Colab. The 'wavs' folder contains 450 .wav audio files in the Minangkabau Pariaman dialect, all of which have undergone the cleaning process. Organizing the data in this structure is designed to facilitate integration into the training environment and ensure compatibility with the model architecture used.

3.3. Training with VITS

The training process is a crucial stage in model development, as without this step, the model cannot be built or used to generate text-to-speech output. In the initial training stage, the model was trained using an initial batch consisting of 32 samples. The batch size selection follows the windowed generator training approach, in which short segments of the latent representations are randomly extracted with a window size of 32 to optimize memory efficiency and training time [15]. The training used inputs in the form of phoneme sequences as token IDs and spectral representations as spectrograms. The training log recorded initial information as Epoch 0/20000, Step 0/14, and Global Step 0/280000, indicating that the process started from the initial index and was designed to continue until Epoch 19999, Step 13, and Global Step 279999. The training process was conducted gradually from May 1 to May 5, 2025, resulting in the production of the trained model.

```

--> EPOCH: 13/20000
--> /content/drive/MyDrive/tts-minangkabau-pariaman/vits_training-May-25-2025_07:34AM-0000000

> TRAINING (2025-05-25 07:43:40)

> EVALUATION

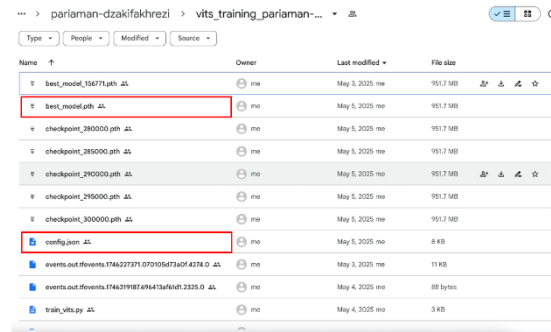
--> STEP: 0
| > loss_disc: 2.2762746810913086 (2.2762746810913086)
| > loss_disc_real_0: 0.16543975472450256 (0.16543975472450256)
| > loss_disc_real_1: 0.279990553855896 (0.279990553855896)
| > loss_disc_real_2: 0.20075233280658722 (0.20075233280658722)
| > loss_disc_real_3: 0.15350839495658875 (0.15350839495658875)
| > loss_disc_real_4: 0.12991014122962952 (0.12991014122962952)
| > loss_disc_real_5: 0.19593439996242523 (0.19593439996242523)
| > loss_0: 2.2762746810913086 (2.2762746810913086)
| > loss_gen: 2.06430983543396 (2.06430983543396)
| > loss_kl: 1.35691499710083 (1.35691499710083)
| > loss_feat: 3.939444065093994 (3.939444065093994)
| > loss_mel: 34.4723777709961 (34.4723777709961)
| > loss_duration: 1.9258242710113525 (1.9258242710113525)
| > loss_l: 43.75807189941406 (43.75807189941406)

--> EVAL PERFORMANCE
| > avg_loader_time: 0.20776033401489258 (-0.1877801418304434)
| > avg_loss_disc: 2.2762746810913086 (+0.14697051048278899)
| > avg_loss_disc_real_0: 0.16543975472450256 (-0.003767266869544983)
| > avg_loss_disc_real_1: 0.279990553855896 (+0.16635464876890182)
| > avg_loss_disc_real_2: 0.20075233280658722 (-0.006591424345970154)
| > avg_loss_disc_real_3: 0.15350839495658875 (-0.0019411742687225342)
| > avg_loss_disc_real_4: 0.12991014122962952 (-0.00012458860874176025)
| > avg_loss_disc_real_5: 0.19593439996242523 (+0.026315972208976746)
| > avg_loss_0: 2.2762746810913086 (+0.14697051048278899)
| > avg_loss_gen: 2.06430983543396 (+0.005469799041748047)
| > avg_loss_kl: 1.35691499710083 (+0.4615433488045825)
| > avg_loss_feat: 3.939444065093994 (-0.43309085380249023)
| > avg_loss_mel: 34.4723777709961 (-2.6284523010253906)
| > avg_loss_duration: 1.9258242710113525 (+0.03506253791809082)
| > avg_loss_l: 43.75807189941406 (-2.565296173095703)

```

Figure 5. Model Training Process

Figure 5 shows the training process, displaying the recorded Epoch, Step, and loss values. Upon completion of the model training, two files were obtained: best_model.pth and config.json.



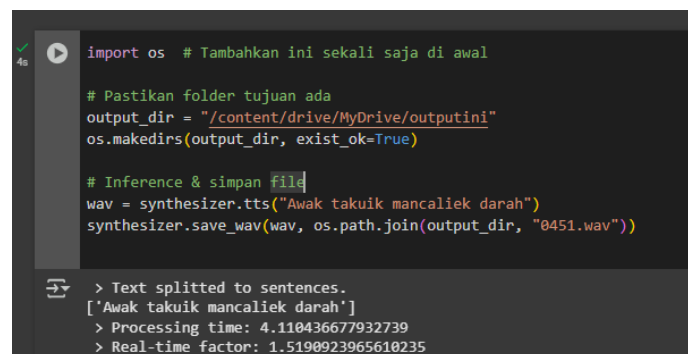
Name	Owner	Last modified	File size
best_model_156778.pth	me	May 3, 2025	951.7 MB
best_model.pth	me	May 5, 2025	951.7 MB
checkpoint_280000.pth	me	May 5, 2025	951.7 MB
checkpoint_285000.pth	me	May 5, 2025	951.7 MB
checkpoint_290000.pth	me	May 5, 2025	951.7 MB
checkpoint_295000.pth	me	May 5, 2025	951.7 MB
checkpoint_300000.pth	me	May 5, 2025	951.7 MB
config.json	me	May 5, 2025	8 KB
events.out.fevents.1746227271.070105673d0f.4274.D	me	May 3, 2025	11 KB
events.out.fevents.1746227271.070105673d0f.4274.D	me	May 4, 2025	88 bytes
train_vits.py	me	May 4, 2025	3 KB

Figure 6. Model Training Results

Figure 6 shows the results of the successfully trained model, yielding two files that will later be used in the inference stage.

3.4. Audio Inference with VITS

For inference, two files obtained from the previous training, namely best_model.pth and config.json, were used. These files were uploaded to Google Drive as the data source for generating audio outputs, allowing the inference process to be executed directly in Google Colab, with the resulting audio automatically saved to Google Drive. As the final stage of the process, the audio data were stored in .wav file format, based on the test data previously organized in Excel, as evidenced by the successfully saved audio files for 50 consecutive test samples, namely data numbered 451 to 500.



```
import os # Tambahkan ini sekali saja di awal

# Pastikan folder tujuan ada
output_dir = "/content/drive/MyDrive/outputini"
os.makedirs(output_dir, exist_ok=True)

# Inference & simpan file
wav = synthesizer.tts("Awak takuik mancaliek darah")
synthesizer.save_wav(wav, os.path.join(output_dir, "0451.wav"))
```

> Text splitted to sentences.
['Awak takuik mancaliek darah']
> Processing time: 4.110436677932739
> Real-time factor: 1.5190923965610235

Figure 7. Inference Results

Figure 7 shows the results of the text-to-audio inference process using the synthesizer.tts() function, as well as the saving of the resulting audio into .wav files using synthesizer.save_wav().

3.5. Evaluation

At this stage, the Minangkabau Pariaman dialect TTS system based on VITS was evaluated to ensure the performance and quality of the audio output. The evaluation employed the subjective MOS method with five native speaker respondents. Prior to testing, the data were validated, and each respondent listened to 50 audio clips. MOS is a subjective evaluation method used to assess audio quality by asking respondents to score the audio they hear, typically on a scale from 1 to 5, where the average score reflects the comfort or naturalness of the speech generated by the system. MOS scores are considered effective for studying the prediction of room

acoustic parameters and speech quality metrics, providing a comprehensive overview of user satisfaction with telecommunication services [25]. The results of the MOS evaluation are presented in Table 4.

Table 4. MOS Evaluation

No.	Sentence	Respondent	Respondent	Respondent	Respondent	Respondent	Total	Average
		1	2	3	4	5		
1.	Awak takuik mancalie darah	5	5	5	5	5	25	5
2.	Sajak ketek awak alah pai marantau mah	5	5	5	5	5	25	5
3.	Pai lah ang dari siko	5	5	5	5	5	25	5
4.	Sabananyo awak anak inyo mah	5	5	5	5	5	25	5
:	:	:	:	:	:	:	:	:
7.	Indak buliah mangecek kuek – kuek ka urang tuo doh	4	4	5	4	4	21	4,2
8.	Marasai awak dek anak kini ko	5	5	4	4	5	23	4,6
9.	Disabuik bana apo gunonyo	4	5	4	5	5	23	4,6
:	:	:	:	:	:	:	:	:
23.	Sagalo nan alah tajadi dinampak an dek tuhan disinan	5	5	5	5	5	25	5
24.	Kok buruak bana kulikaik awak, lun buruak bana do lai	4	4	5	4	4	21	4,2
25.	Awak maraso basuo di badan diri wak wakatu tu	5	5	5	5	5	25	5
:	:	:	:	:	:	:	:	:

34.	Jan cangok bana makan tu	5	5	5	5	5	25	5
35.	Kok bakawan tu dicaliek caliek	5	5	5	5	5	25	5
38.	Ba a sampik bana tampek ko?	5	5	5	5	5	25	5
39.	Dima awak kini ko?	5	5	5	5	5	25	5
40.	Dima inyo tingga kini?	5	5	5	4	5	24	4,8
48.	Latiah bana badan awak raso e	5	5	5	5	5	25	5
49.	Sia nan tibo ka umah patang tu?	5	5	5	5	5	25	5
50.	Lamak banak samba nan ang buck ko	5	5	5	5	5	25	5
							MOS	4,72

The results summarized in Table 4 show an average MOS score of 4.72. This value indicates that the synthesized speech quality is considered very good to nearly natural (Excellent) by the majority of respondents, suggesting that the developed TTS system is capable of producing natural and pleasant-sounding speech. For respondent selection, five individuals aged between 27 and 43 years were chosen, all of whom are native speakers of the Minangkabau Pariaman dialect. Based on the respondents' assessments, there was variation in the scores for the generated speech quality, which were then calculated using the MOS method to obtain the average perception of speech quality.

From these calculations, the total score amounted to 236.2, which was divided by the 50 test sentences, resulting in a final MOS value of 4.72, indicating that the model's quality is very good. This result is consistent with ESPnet-TTS studies, which also reported high MOS scores on the LJSpeech dataset [29]. However, recent studies emphasize that although MOS is effective and widely used for evaluating TTS system quality, this method has limitations due to its subjective nature and the required time and cost. Consequently, more research has focused on developing objective neural network-based evaluation metrics to complement subjective assessment [30].

The evaluation results show that most respondents gave high scores, indicating that the TTS speech sounds natural and is easily understood. However, some words such as 'doh', 'lapau', 'jauah', and 'onggok' were pronounced less clearly or inaccurately, affecting the naturalness and meaning of the sentences. Certain intonations also did not fully match the authentic Minangkabau Pariaman dialect. This indicates that the model still requires refinement, particularly in intonation variation. Nevertheless, the TTS system is on the right track and only requires minor adjustments to improve the results. As a next step, the MOS-based evaluation used can be complemented with statistical analysis or objective testing in future studies to

strengthen the validity of the findings. This aligns with recent studies stating that although MOS is effective for evaluating speech quality, the method is subjective and requires considerable time and cost. Therefore, it is recommended to be combined with objective neural network-based evaluation metrics [31].

Furthermore, this study has strong relevance to the preservation of regional languages, particularly the Minangkabau Pariaman dialect. By developing a TTS system, the regional language is not only documented but also brought to life in spoken form, thereby strengthening cultural identity in the digital era. Moreover, this technology has potential applications in various real-world domains, such as education, language training, and local voice assistants, which can help the community recognize, use, and preserve the Minangkabau language in daily life. Thus, the contribution of this study is not only technical but also supports digital language preservation efforts to ensure that regional languages remain sustainable and relevant amidst technological advancements.

4. Conclusion

Based on the results of the study on the Minangkabau Pariaman dialect TTS system using the VITS method, this research successfully developed a TTS system for the Minangkabau Pariaman dialect employing VITS. The system utilized a dataset consisting of 450 training sentences and 50 test sentences from a native adult female speaker. The resulting model was capable of generating synthetic speech closely resembling the original speaker. Evaluation with five native speaker respondents using the MOS scale yielded an average score of 4.72 out of 5, indicating high-quality and natural-sounding speech.

The limitations of this study include the use of a dataset sourced from a single speaker, specifically an adult female native of the Minangkabau Pariaman dialect, which restricts the speech characteristics to one speaker only. For future research, it is recommended to use a multi-speaker dataset with variations in gender and age range to enhance the model's ability to handle variations in intonation, expression, and speech clarity from different types of speakers.

References

- [1] F. Lafamane, "Fenomena Penggunaan Bahasa Daerah di Kalangan Remaja," *OSF*, Jul. 2020, doi: 10.31219/osf.io/jubxp.
- [2] R. S. Velini and M. Suryadi, "Usaha Pemertahanan Bahasa Minangkabau melalui Permainan dan Tradisi Budaya Lokal di Kota Padang, Sumatera Barat," *Jurnal Sastra Indonesia*, vol. 12, no. 1, pp. 71–80, Apr. 2023, doi: 10.15294/jsi.v12i1.59370.
- [3] S. BAHRI, "Eufemisme Bahasa Minangkabau Dialek Pariaman," *Jurnal Bahasa Unimed*, no. 84, 2012, doi: <https://doi.org/10.24114/bhs.v0i84%20TH%2038.2328>.
- [4] A. Sutra, Z. Rida Rahayu, M. Putri, and H. Fikri, "Variasi Dialek Bahasa Minang (Pasisia) Siswa Kelas X SMA NEGERI 3 Painan," *Jurnal Edukasi dan Literasi Bahasa*, vol. 4, no. 1, Apr. 2023, doi: <https://doi.org/10.36665/jelisa.v4i1.711>.
- [5] H. Hernawati, "Penggunaan Bahasa Ibu Sebagai Pengantar Dalam Pembelajaran Bahasa," *Jurnal Ilmiah Program Studi Pendidikan Bahasa dan Sastra Indonesia*, vol. 4, Sep. 2015, doi: <https://doi.org/10.22460/semantik.v4i2.p83%20-%2091>.
- [6] A. Triandana, Y. Mestika Putra, S. Fitriah, and A. Kartika Putri, "Strategi Pemertahanan Bahasa Daerah Sebagai Bentuk Pelestarian Bahasa Pada Generasi Muda Di Kalangan Mahasiswa Sastra Indonesia Universitas Jambi," *Jurnal Pengabdian Masyarakat*, vol. 2, no. 1, Apr. 2023, doi: 10.22437/est.v2i1.24576.
- [7] N. Arrizqi, I. Santoso, D. Yosua, and A. A. Soetrisno, "Implementasi Google Text To Speech Pada Aplikasi Pendeteksi Uang Berbasis Android," *Transient*, vol. 10, no. 3, pp. 2685–0206, Sep. 2021, doi: <https://doi.org/10.14710/transient.v10i3.510-516>.
- [8] K. D. Fatmawati and I. Tahyudin, "Teknologi Text To Speech Menggunakan Amazon Polly Untuk Meningkatkan Kemampuan Membaca Pada Anak Dengan Gejala Disleksia," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 6, pp. 1351–1360, Dec. 2024, doi: 10.25126/jtiik.2024117426.
- [9] J. Shen *et al.*, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," *ArXiv*, Feb. 2018, doi: <https://doi.org/10.48550/arXiv.1712.05884>.
- [10] S. Wang, G. E. Henter, J. Gustafson, and É. Székely, "A Comparative Study of Self-Supervised Speech Representations in Read and Spontaneous TTS," *ArXiv*, Jul. 2023, doi: <https://doi.org/10.48550/arXiv.2303.02719>.

- [11] S. R. U. A. S. S. P. Muhamad M.I. Putra, "Implementasi Speech Recognition pada Aplikasi Pembelajaran Bahasa Inggris untuk Anak," *Jurnal Teknik Informatika*, vol. 15, no. 4, Oct. 2020, doi: <https://doi.org/10.35793/jti.v15i4.30426>.
- [12] K. Peng, W. Ping, Z. Song, and K. Zhao, "Non-Autoregressive Neural Text-to-Speech," *ArXiv*, Jun. 2019, doi: <https://doi.org/10.48550/arXiv.1905.08459>.
- [13] S. Sarif and A. AR, "Efektivitas Artificial Intelligence Text to Speech dalam Meningkatkan Keterampilan Membaca," *Jurnal Naskhi Jurnal Kajian Pendidikan dan Bahasa Arab*, vol. 6, no. 1, pp. 1–8, Apr. 2024, doi: 10.47435/naskhi.v6i1.2697.
- [14] Y. Ren *et al.*, "FastSpeech 2: Fast and High-Quality End-to-End Text to Speech," *ArXiv*, Aug. 2022, doi: <https://doi.org/10.48550/arXiv.2006.04558>.
- [15] R. Valle, J. Li, R. Prenger, and B. Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," *ArXiv*, Oct. 2019, doi: <https://doi.org/10.48550/arXiv.1910.11997>.
- [16] W. Zhao and Z. Yang, "An Emotion Speech Synthesis Method Based on VITS," *Applied Sciences (Switzerland)*, vol. 13, no. 4, Feb. 2023, doi: 10.3390/app13042225.
- [17] Y. Shirahata, R. Yamamoto, E. Song, R. Terashima, J.-M. Kim, and K. Tachibana, "Period VITS: Variational Inference with Explicit Pitch Modeling for End-to-end Emotional Speech Synthesis," *ArXiv*, Feb. 2022, doi: <https://doi.org/10.48550/arXiv.2210.15964>.
- [18] K. Liang, B. Liu, Y. Hu, R. Liu, F. Bao, and G. Gao, "Comparative Study for Multi-Speaker Mongolian TTS with a New Corpus," *Applied Sciences (Switzerland)*, vol. 13, no. 7, Apr. 2023, doi: 10.3390/app13074237.
- [19] J. Kim, J. Kong, and J. Son, "Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech," *ArXiv*, Jun. 2021, doi: <https://doi.org/10.48550/arXiv.2106.06103>.
- [20] H. Guo *et al.*, "QS-TTS: Towards Semi-Supervised Text-to-Speech Synthesis via Vector-Quantized Self-Supervised Speech Representation Learning," *ArXiv*, Aug. 2023, doi: <https://doi.org/10.48550/arXiv.2309.00126v1>.
- [21] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis," *ArXiv*, Oct. 2020, doi: <https://doi.org/10.48550/arXiv.2010.05646>.
- [22] J. Frnda, J. Nedoma, R. Martinek, and M. Fridrich, "Predicting Perceptual Quality in Internet Television Based on Unsupervised Learning," *Symmetry (Basel)*, vol. 12, no. 9, p. 1535, Sep. 2020, doi: <https://doi.org/10.3390/sym12091535>.
- [23] S. Gururani, K. Gupta, D. Shah, Z. Shakeri, and J. Pinto, "Prosody Transfer in Neural Text to Speech Using Global Pitch and Loudness Features," *ArXiv*, May 2020, doi: <https://doi.org/10.48550/arXiv.1911.09645>.
- [24] Y. Choi, Y. Jung, Y. Suh, and H. Kim, "Learning to Maximize Speech Quality Directly Using MOS Prediction for Neural Text-to-Speech," *ArXiv*, May 2022, doi: <https://doi.org/10.48550/arXiv.2011.01174>.
- [25] J. Coldenhoff, A. Harper, P. Kendrick, T. Stojkovic, and M. Cernak, "Multi-Channel MOSRA : Mean Opinion Score And Room Acoustics Estimation Using Simulated Data And Teacher Model," *ArXiv*, Mar. 2024, doi: <https://doi.org/10.48550/arXiv.2309.11976>.
- [26] J. Frnda, J. Nedoma, R. Martinek, and M. Fridrich, "Predicting perceptual quality in internet television based on unsupervised learning," *Symmetry (Basel)*, vol. 12, no. 9, Sep. 2020, doi: 10.3390/SYM12091535.
- [27] Q. Wang *et al.*, "VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking," *ArXiv*, Jun. 2019, doi: <https://doi.org/10.48550/arXiv.1810.04826>.
- [28] A. Pothula, B. Akkiraju, S. Bandurupalli, C. D. S. Kesiraju, and A. K. Vuppala, "End-to-End Speech Translation for Low-Resource Languages Using Weakly Labeled Data," *ArXiv*, Jun. 2025, doi: <http://arxiv.org/abs/2506.16251>.
- [29] T. Hayashi *et al.*, "ESPnet-TTS: Unified, Reproducible, and Integratable Open Source End-to-End Text-to-Speech Toolkit," *ArXiv*, Feb. 2020, doi: <https://doi.org/10.48550/arXiv.1910.10909>.
- [30] G. Zhu, Y. Wen, and Z. Duan, "A Review on Score-based Generative Models for Audio Applications," *ArXiv*, Jun. 2025, doi: <http://arxiv.org/abs/2506.08457>.
- [31] S.-F. Huang, C.-J. Lin, D.-R. Liu, Y.-C. Chen, and H. Lee, "Meta-TTS: Meta-Learning for Few-Shot Speaker Adaptive Text-to-Speech," *ArXiv*, Jul. 2022, doi: <https://doi.org/10.1109/TASLP.2022.3167258>.