Machine Learning-Based Besemah Language Translator Model with Recurrent Neural Network (RNN) Model Algorithm

Muhamad Andika [1], Yesi Novaria Kunang [2], Ilman Zuhri Yadi [3], Susan Dian Purnamasari [4]
Information Systems Study Program, Faculty of Science Technology [1], [2], [3] [4]
Intelligent Systems Research Group
Bina Darma University
Palembang, Indonesia
191410165@student.binadarma.ac.id [1], yesinovariakunang@binadarma.ac.id [2], ilmanzuhriyadi@binadarma.ac.id [3], susandian@binadarma.ac.id [4]

***Abstract*—Indonesia consists of various tribes with their respective regional languages, one of which is the Besemah tribe in South Sumatra province with its language culture, namely Besemah Language. Until now Besemah language is still used by the Besemah tribe but over time the number of Besemah language speakers is decreasing not to mention most of the wider community do not know what Besemah language is. Machine translation is a tool that can switch one language to another. This research aims to collect datasets in the form of sentences and words from Besemah Language and then create a Besemah Language translation machine to Indonesian and vice versa. In this research, Neural Machine Translation (NMT) technology with Recurrent Neural Network (RNN) approach is applied. The results for val_accuracy besemah-indonesia is 0.8469 and for Indonesia-besemah get a val_accuracy value of 0.8492, in translation trials conducted using the RNN model, 100 epochs, 64 batch sizes and 0.2 validation split.**

Keywords—Neural Machine Translation, Besemah Language, Recurrent Neural Network.

I. INTRODUCTION

Machine Learning is a subset of artificial intelligence that is often used to solve various problems. Machine Learning involves the use of computers and mathematical algorithms that use data to make future predictions. The process involves training and testing stages, and continues to evolve through research, including in the field of language translation [1].

Machine translation is an active field of research, although specific research on translation from Indonesian to Besemah using artificial neural network-based methods is rare. [2]. Neural Machine Translation (NMT) is a term that refers to a translation method using artificial neural networks. Machine translation is responsible for automatically converting text from one language to another. Commonly used methods include RNN (Recurrent Neural Network), CNN (Convolutional Neural Network), and attention mechanism [3].

Artificial neural networks known as Recurrent Neural Network (RNN) are well suited for identifying patterns in sequentially arranged data, for example converting word order from Indonesian into Besemah This research aims to preserve the Besemah regional language so that it lives on from one generation to the next. The approach used in this research is to use Neural Machine Translation technology (NMT). This approach relies on Recurrent Neural Network (RNN) architecture in the translation process [4].

Besemah is the name of a tribe that has been in Indonesia for a very long time. The Besemah have their own local language, called Besemah, just as other tribes have their own languages. Until now, Besemah is still used by the people who speak the language as a means of communication and relationship between fellow community members [5].

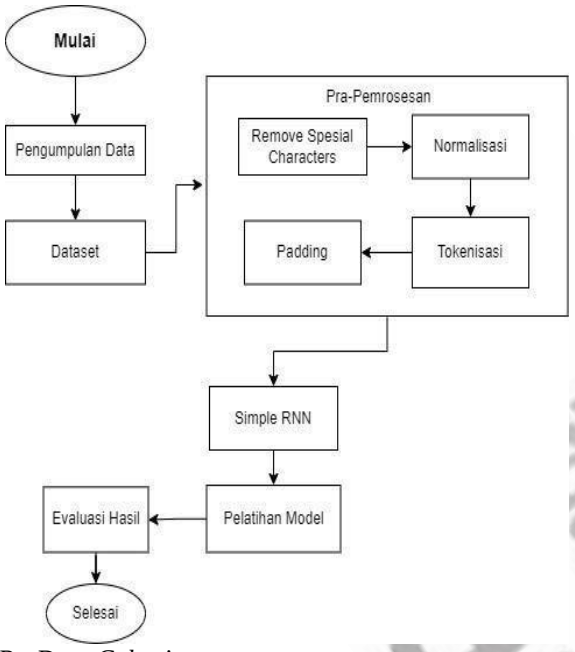
This research is intended to design or build a translation machine that can be a solution to increase knowledge, use and preservation of the Besemah regional language. Therefore, the author raises research with the title "machine learning-based besemah language translation machine with model algorithm Recurrent Neural Network (RNN)".

II. RESEARCH METHODOLOGY

In the research conducted, the approach used is Experimental Research in Machine Learning. Experimental research in machine learning for language translation is a research approach that involves designing and conducting a series of experiments to test and validate the performance of a language translation model. In this context, experiments are conducted by adjusting various parameters and variables in the model, such as neural network architecture, number of layers, activation function, and so on. The flow of the research conducted can be seen through the following flowchart:

A. Flowchart

The research methodology for the development of the Besemah to Bahasa Indonesia machine translator is shown in Figure 1.



B. Data Colection

C. In this stage, the collection of Besemah language data to be translated is done by scanning the source data from the BESEMAH-INDONESIA-ENGLISH dictionary [6].

D. Dataset Creation

In this stage, a dataset is formed from the previously collected data. This dataset will then be divided into 2 parts, namely the Indonesian language dataset and the Basemah language dataset. Both datasets will be created in the form of files with txt format.

E. Pra Processed

- Special Characters Removal: A step to remove or replace unwanted characters in a string. Special characters are characters that are different from letters, numbers, or spaces, such as punctuation marks, symbols, or non-ASCII characters.
- Normalization: This process converts the text to a standardized format by converting the text to lowercase for easier processing. The goal is to make varied texts uniform, thus making further processing easier and improving data consistency.
- Tokeniation: This step decomposes the text into separate words. This separation is done by taking into account the spaces between words or by applying certain rules, such as separation based on punctuation or special characters.
- Padding: In this stage, additional elements are added to the input data to ensure that the data size is uniform or meets certain requirements. This is done with the aim that all samples in the dataset have the same dimensions, thus facilitating efficient batch processing.

F. Rnn Model

At this stage, modifications are made to the RNN model structure with sequentially connected Recurrent Neural Network layers. This is a step to prepare the data that will be used in the RNN model training process).

The formulas of the RNN algorithm include the following:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) dan\ y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

Wax, Waa, Wya, ba, and by are factors adjusted over time, and g1 and g2 are activation functions. In training a simple artificial neural network (RNN), the backpropagation through time (BPTT) method is used. BPTT calculates how the gradient of the loss function L relates to the model parameters, using a chain rule. This gradient is then utilized to update the model parameters using optimization algorithms such as stochastic gradient descent (SGD).

G. Model Training

After obtaining the dataset and the model is formed, the next step is model training. The model that has previously been formed will be given a dataset that has been prepared to train the model. The goal is for the model to understand the relationship between sentences in Besemah and Indonesian.

H. Model Evaluation

In this evaluation stage, it involves testing the model and making improvements to achieve the best results. At this stage, the model is evaluated using a test dataset to measure the performance and quality of the predictions.

III. RESULTS AND DISCUSSION

A. Data Collection

The data collected comes from scans of the besemah-indonesian-English dictionary by dr. sutyono mahdi D.rs., M.Hum. The scanned data is saved into an excel file for later processing. In the excel file, there are 5104 data that will be used later. Of the 5104 data, there are 3375 data in the form of words and 1729 data in the form of sentences.

5083	Ndaq ngape die	Mau apa dia
5084	Diq keruan li ku	Tidak tahu saya
5085	ısan dimane nining	Dari mana nenek
5086	Ndaq tuape die	Ingin apa dia
5087	Nganuka sape die	Mengganggu siapa dia
5088	Njemugh tuape kabah	Apa yang engkau jemur
5089	Aku dang kurang sihat	Saya sekarang kurang sehat
5090	Kalu aku lah ghadu	Bila saya sudah sembuh
5091	kabah ka kukanceghi	engkau akan saya temani
5092	Die nde pacaq gale nga jeme di situ	Dia yang kenal semua dengan orang di sana
5093	Mertuaku akan hajatan besok malam	Mertuaku akan hajatan besok malam
5094	Die ka bekiaji	Dia akan naik haji
5095	Mamaq mbuat ghumah di dusun kamu	Paman membuat rumah di desamu
5096	Ghumah itu kandiq anaqe nde empai tunaq	Rumah itu untuk anaknya yang baru menikah
5097	Banyaq pisang di bawah ghumah	Di bawa rumah banyak pisan
5098	Pisang itu empai nebang petang tadi	Pisang itu baru diambil sore tadi
5099	Jeme ghumah itu lah agung sandi baghi	Orang itu sudah kaya sejak lama
5100	Anye cakagh duit tu masih katah nemane	Tetapi mencari uang masih sangat giat
5101	Masuqka gale ikan ni ke kulkas	Masukan semua ikan ini ke dalam kulkas
5102	Kalu dide pagi busuq gale	Kalau tidak besok busuk semuanya
5103	Die tu lah tue samegi nga aku	Dia itu sudah tua sama dengan saya
5104	Anye kinaqane masih muda	Tetapi kelihatannya masih muda

Fig. 2. The scanned dictionary data is stored in an Excel file.

B. Dataset

In this stage, the dataset is split from the previous excel file. This dataset split is broken down into 2, namely the Indonesian language dataset and the Besemah language dataset. Each of these datasets will be saved back into a txt file and will be used in data processing in machine learning later.

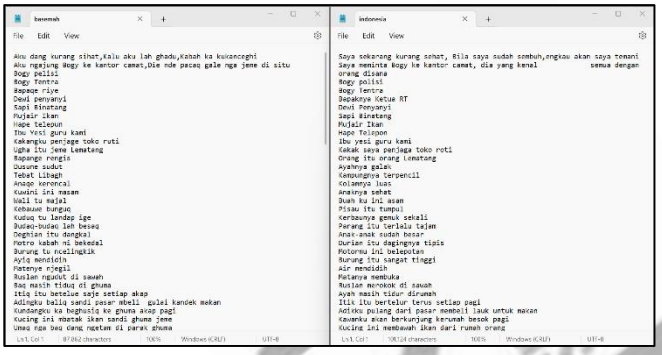


Fig. 3. Besemah language dataset and Indonesian language dataset

C. Pre Processed

In this step, the dataset that has been created will undergo pre-processing first. The pre-processing stage includes four steps, namely removing special characters, normalization, tokenization, and padding.

- Remove Special Characters

  In this stage, the text will be cleaned from any special characters. These special characters are punctuation marks, symbols or special characters that are not very important and are not needed in data processing later.

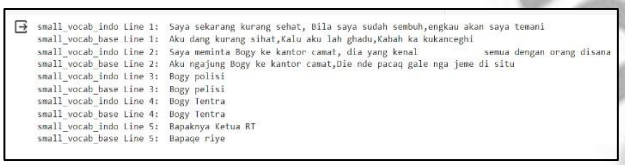


Fig. 4. Data before removing special characters

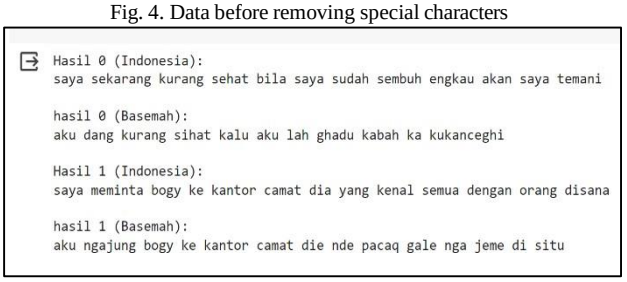


Fig. 5. Data after the process of removing special characters

Based on Figure 4 and Figure 5, it can be seen that there are changes in the text after going through the remove special characters process. In the initial text, there are still punctuation marks such as "," located after certain words. After undergoing this process, the "," sign is lost and deleted.

- Normalisasi

  This stage is in the form of simplifying the text by converting all letters into lowercase letters. This form of normalization is done so that the letters in the text become the same and simpler so that it can facilitate further processing.

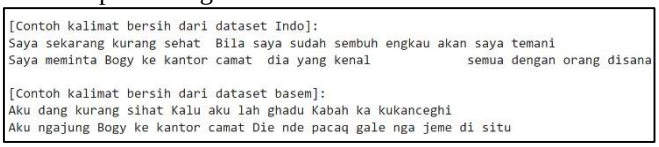


Fig. 6. Data before normalization process

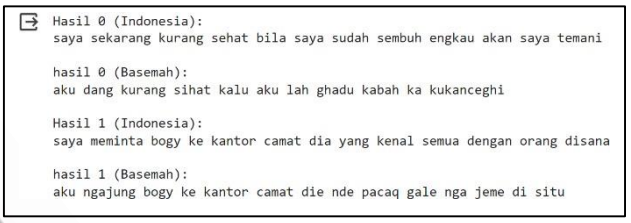


Fig. 7. Data after normalization process

From Figure 6 and Figure 7, there are changes that occur in Indonesian and Besemah texts after going through the normalization process, namely changing all letters to lowercase letters and removing excessive spaces.

- Tokenization

This process is done by converting data from text into numeric by assigning token values to each word in the sentence. This token value is given to break the text into several small units so that data processing will be easier to do later.

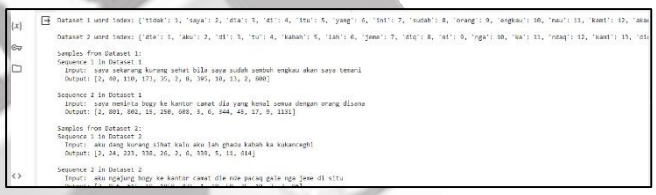


Fig. 8. Tokenization Result

- Pading

In this stage, the token values that have been determined in the previous tokenization process will be equalized through this padding process. The purpose of the padding technique is to adjust the length of the data so that it can be incorporated into an algorithm or model with a fixed input size. Padding is usually applied to data that has varying lengths so that each example or sample has a uniform size.

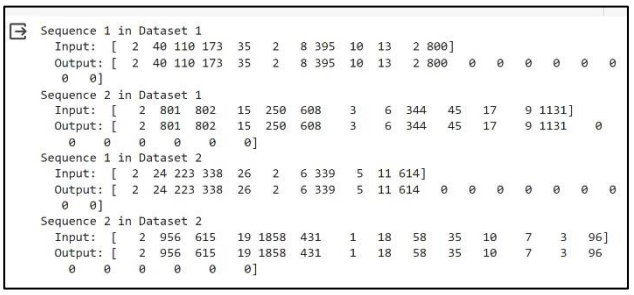


Fig. 9. Padding Result

The purpose of this process is to prepare the data in a suitable format to be fed into the machine learning model. By equalizing the length of each sentence, the model can process the data more efficiently and uniformly. This padding process is necessary because machine learning models require inputs with consistent lengths to be able to perform the learning process properly.

D. Rnn Model

This stage is in the form of building an RNN model for the Besemah language translation trial. A Recurrent Neural Network (RNN) is a type of neural network architecture specifically designed to handle sequential data, such as text or time series. In the context of language translation engines, the RNN model is used to understand and translate sequences of words or phrases from one language to another. The following is the code used to create the RNN model:

```
def simple_rnn_model1(input_shape, output_sequence_length, indonesia_vocab_size, basemah_vocab_size):
    learning_rate = 1e-3
    input_seq = Input(input_shape[1:])
    rnn = GRU(64, return_sequences = True)(input_seq)
    logits = TimeDistributed(Dense(basemah_vocab_size))(rnn)
    model = Model(input_seq, Activation('softmax')(logits))
    model.compile(loss = sparse_categorical_crossentropy,
                  optimizer = Adam(learning_rate),
                  metrics = ['accuracy'])

    return model
    tests.test_simple_model(simple_rnn_model)
    print(tmp_x.shape)

tmp_x = pad(preproc_indonesia_sentences, max_basemah_sequence_length)
tmp_x = tmp_x.reshape((-1, preproc_basemah_sentences.shape[-2], 1))

# Train the neural network
simple_rnn_model1 = simple_rnn_model1(
    tmp_x.shape,
    max_basemah_sequence_length,
    indonesia_vocab_size,
    basemah_vocab_size
    )
print(simple_rnn_model1.summary())
```

Fig. 10. Model making code rnn Indonesia- Besemah

```
def simple_rnn_model1b(input_shape, output_sequence_length, basemah_vocab_size, indonesia_vocab_size):
    learning_rate = 1e-3
    input_seq = Input(input_shape[1:])
    rnn = GRU(64, return_sequences = True)(input_seq)
    logits = TimeDistributed(Dense(indonesia_vocab_size))(rnn)
    model = Model(input_seq, Activation('softmax')(logits))
    model.compile(loss = sparse_categorical_crossentropy,
                  optimizer = Adam(learning_rate),
                  metrics = ['accuracy'])
    #print(model.summary())
    return model
    print(tmp_x2.shape)
   # tests.test_simple_model(simple_rnn_model)
tmp_x2 = pad(preproc_basemah_sentences, max_indonesia_sequence_length)
tmp_x2 = tmp_x2.reshape((preproc_indonesia_sentences.shape[-2],-1, 1))


# Train the neural network
simple_rnn_model1b = simple_rnn_model1b(
    tmp_x2.shape,
    max_indonesia_sequence_length,
    basemah_vocab_size,
    indonesia_vocab_size
    )
print(simple_rnn_model1b.summary())
```

Fig. 11. Model making code rnn besemah-indonesia

The code implements a Recurrent Neural Network (RNN) model using a Gated Recurrent Unit (GRU) layer to perform language translation. The model accepts a set of sentences in Indonesian as input and aims to produce a translation in Besemah. The model consists of a GRU layer with 64 units, followed by TimeDistributed and Dense layers that are in charge of generating outputs for each time step. A softmax activation function was applied to convert the output values into probability distributions. The model was trained using the sparse categorical crossentropy loss function and optimized with Adam's method on the training dataset. During training, the model was run for 100 epochs with a batch size of 64, and validation was performed on 20% of the data. After training, the model was tested by predicting the output for one input instance and the results were converted to text using the logits_to_text function.

E. Model Training

After the RNN model is created, in this process the model training will be carried out so that the previously created model can translate Besemah language. This model training is done with 100 epochs and using 64 batch sizes.

```
Epoch 91/100
64/64 [==============================] - 1s 12ms/step - loss: 0.9389 - accuracy: 0.8472 - val_loss: nan - val_accuracy: 0.9463
Epoch 92/100
64/64 [==============================] - 1s 12ms/step - loss: 0.9369 - accuracy: 0.8472 - val_loss: nan - val_accuracy: 0.9463
Epoch 93/100
64/64 [==============================] - 1s 12ms/step - loss: 0.9345 - accuracy: 0.8477 - val_loss: nan - val_accuracy: 0.9464
Epoch 94/100
64/64 [==============================] - 1s 12ms/step - loss: 0.9323 - accuracy: 0.8475 - val_loss: nan - val_accuracy: 0.9465
Epoch 95/100
64/64 [==============================] - 1s 13ms/step - loss: 0.9306 - accuracy: 0.8477 - val_loss: nan - val_accuracy: 0.9462
Epoch 96/100
64/64 [==============================] - 1s 12ms/step - loss: 0.9284 - accuracy: 0.8478 - val_loss: nan - val_accuracy: 0.9465
Epoch 97/100
64/64 [==============================] - 1s 12ms/step - loss: 0.9260 - accuracy: 0.8479 - val_loss: nan - val_accuracy: 0.9464
Epoch 98/100
64/64 [==============================] - 1s 12ms/step - loss: 0.9244 - accuracy: 0.8478 - val_loss: nan - val_accuracy: 0.9463
Epoch 99/100
64/64 [==============================] - 1s 12ms/step - loss: 0.9221 - accuracy: 0.8480 - val_loss: nan - val_accuracy: 0.9463
Epoch 100/100
64/64 [==============================] - 1s 14ms/step - loss: 0.9199 - accuracy: 0.8479 - val_loss: nan - val_accuracy: 0.9463
1/1 [==============================] - 0s 486ms/step
aku dang sihat sihat dang aku lah <PAD> jeme ke aku <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>

[17] print(logits_to_text(simple_rnn_model1.predict(tmp_x[:1])[0],basemah_tokenizer))

1/1 [==============================] - 0s 28ms/step
aku dang sihat sihat dang aku lah <PAD> jeme ke aku <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
```

Fig. 12. Results of rnn model training indonesia - besemah

```
64/64 [==============================] - 1s 12ms/step - loss: 0.9962 - accuracy: 0.8415 - val_loss: nan - val_accuracy: 0.9281
Epoch 92/100
64/64 [==============================] - 1s 12ms/step - loss: 0.9934 - accuracy: 0.8415 - val_loss: nan - val_accuracy: 0.9281
Epoch 93/100
64/64 [==============================] - 1s 12ms/step - loss: 0.9912 - accuracy: 0.8416 - val_loss: nan - val_accuracy: 0.9282
Epoch 94/100
64/64 [==============================] - 1s 12ms/step - loss: 0.9897 - accuracy: 0.8414 - val_loss: nan - val_accuracy: 0.9282
Epoch 95/100
64/64 [==============================] - 1s 11ms/step - loss: 0.9870 - accuracy: 0.8418 - val_loss: nan - val_accuracy: 0.9282
Epoch 96/100
64/64 [==============================] - 1s 13ms/step - loss: 0.9856 - accuracy: 0.8417 - val_loss: nan - val_accuracy: 0.9282
Epoch 97/100
64/64 [==============================] - 1s 13ms/step - loss: 0.9826 - accuracy: 0.8420 - val_loss: nan - val_accuracy: 0.9282
Epoch 98/100
64/64 [==============================] - 1s 13ms/step - loss: 0.9809 - accuracy: 0.8418 - val_loss: nan - val_accuracy: 0.9282
Epoch 99/100
64/64 [==============================] - 1s 14ms/step - loss: 0.9784 - accuracy: 0.8421 - val_loss: nan - val_accuracy: 0.9282
Epoch 100/100
64/64 [==============================] - 1s 11ms/step - loss: 0.9772 - accuracy: 0.8419 - val_loss: nan - val_accuracy: 0.9282
1/1 [==============================] - 0s 295ms/step
saya tidak saya saya makan saya sudah <PAD> itu tidak <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>

[21] print(logits_to_text(simple_rnn_model1b.predict(tmp_x2[:1])[0],indonesia_tokenizer))
    #print(tmp_x2[:1])

1/1 [==============================] - 0s 17ms/step
saya tidak saya saya makan saya sudah <PAD> itu tidak <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
```

Fig.13. Results of rnn model training besemah-indomesia

From the given output results, it appears that the val_accuracy value recorded during the training of the besemah - indonesia model is about 0.8469 in the training set. model indonesia-besemah accuracy 0.8492.

F. Model Evalution

This stage is a final evaluation of the translation trials conducted using the pre-trained model. In this stage, a translation test will be conducted using the pre-trained RNN model and an evaluation of the translation results produced.

```
[790, 155]
20
bogy polisi
1/1 [==============================] - 0s 34ms/step
<PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
```

Fig. 14. Translation trial results indonesia-besemah

```
[614, 120]
[[614 120   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0]]
5104
bogy pelisi
1/1 [==============================] - 0s 22ms/step
tidak kelapa <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
```

Fig. 15. Translation trial results besemah-indonesia

From the output, it can be seen that the model has gone through training with the loss and accuracy values reported for

1 epoch. Although the accuracy in the validation set (val_accuracy) seems quite stable, there are some "<PAD>" tokens in the prediction results, which indicates that the model has not been able to fully generate the appropriate words in translation. In sample 1, the prediction results only produced "<PAD>" tokens, while in sample 2, the model produced some appropriate tokens but not perfectly. Therefore, the model may need to be adjusted or improved through parameter tuning or other approaches to improve the translation quality.

IV. CONCLUSION

Translation trials between Besemah and Indonesian using the developed Recurrent Neural Network (RNN) model resulted in significant achievements. In a series of experiments with 100 epochs, batch size 64, and validation split 0.2, the model managed to achieve a val_accuracy rate of 0.8469 for besemah - indonesia and for Indonesia-besemah it got a val_accuracy of 0.8492. Nonetheless, the prediction results show some limitations, especially in replicating the corresponding reference sentences. The overuse of the token "<PAD>" indicates the model's difficulty in understanding the Besemah language structure. Therefore, it is necessary to conduct an in-depth review of the model configuration, dataset size, and language structure complexity to improve the performance and accuracy of the developed translator model.

REFERENCES

[1] A. Roihan, P. A. Sunarya, and A. S. Rafika, "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," *IJCIT*, vol. 5, no. 1, May 2020, doi: 10.31294/ijcit.v5i1.7951.

[2] A. Setiawan and H. Sujaini, "Implementasi Optical Character Recognition (OCR) pada Mesin Penerjemah Bahasa Indonesia ke Bahasa Inggris," vol. 5, no. 2, 2017.

[3] P. A. Wismoyo and R. Kusumaningrum, "MESIN PENERJEMAH BAHASA INGGRIS–INDONESIA BERBASIS JARINGAN SARAF TIRUAN DENGAN MEKANISME ATTENTION MENGGUNAKAN ARSITEKTUR TRANSFORMER," PhD Thesis, Universitas Diponegoro, 2018.

[4] M. Y. Aristyanto and R. Kurniawan, "Pengembangan Metode Neural Machine Translation Berdasarkan Hyperparameter Neural Network," *semnasoffstat*, vol. 2021, no. 1, pp. 935–946, Nov. 2021, doi: 10.34123/semnasoffstat.v2021i1.789.

[5] H. Saputra, "UPAYA PEMERTAHANAN BAHASA DAERAH BESEMAH SEBAGAI BAGIAN PELESTARIAN KEARIFAN LOKAL," *MEDAN MAKNA*, vol. 16, no. 1, p. 88, Jun. 2018, doi: 10.26499/mm.v16i1.2275.

[6] S. Mahdi, *KAMUS BAHASA BESEMAH-INDONESIA-INGGRIS FULL_unlocked*. Unpad Press, 2014.