

# KLASIFIKASI AUDIO UCAPAN EMOSIONAL MENGUNAKAN MODEL LSTM

Raynaldy Arief<sup>1</sup>, Nur Aviva Iriawan<sup>2</sup>, dan Armin Lawi<sup>3</sup>

<sup>123</sup>Program Studi Sistem Informasi, Departemen Matematika, Fakultas MIPA, Universitas Hasanuddin, Makassar

Email Copresponder Author : <sup>1</sup>[ariefr18h@student.unhas.ac.id](mailto:ariefr18h@student.unhas.ac.id), <sup>2</sup>[iriawanna18h@student.unhas.ac.id](mailto:iriawanna18h@student.unhas.ac.id),  
<sup>3</sup>[armin@unhas.ac.id](mailto:armin@unhas.ac.id)

**Abstract** — Emotion is an important element in the synthesis of human expressive speech. There are various kinds of emotions, such as sad, angry, happy, and other forms of emotion. The purpose of this study is to assist psychologists by providing automatic classification recommendations for a person's emotional speech based on sound. Therefore, in this study used the dataset *Toronto Emotional Speech (TESS)* with a total of 2,800 data and a total of 7 emotional classes with each class having a lot of data as much as 400, the classification of a person's emotional speech will be carried out. This paper used the *Long Short-Term Memory (LSTM)* method with distribution 70% training data and 30% test data from the total amount of data. With 100 epochs and 64 batch size producing a train accuracy rate of 98,47% and test accuracy of 97,02% (*best-fitting*) and 99,9% ROC.

**Keyword** — Classification, Deep Learning, Emotional Speech, Long Short-Term Memory (LSTM), Toronto Emotional Speech Set (TESS).

**Abstrak** — Emosi adalah elemen penting dalam sintesis ucapan ekspresif manusia. Emosi ada bermacam macam, seperti sedih, marah, bahagia, dan bentuk emosi lainnya. Tujuan dari penelitian ini adalah membantu tenaga psikologi dengan menyajikan rekomendasi klasifikasi otomatis untuk ucapan emosional seseorang berdasarkan suara. Oleh karena itu, pada penelitian kali ini menggunakan dataset *Toronto Emotional Speech (TESS)* dengan jumlah data 2.800 serta jumlah kelas emosional sebanyak 7 dengan masing masing kelas memiliki banyak data sebanyak 400, akan dilakukan klasifikasi ucapan emosional seseorang. Makalah ini menggunakan metode *Long Short-Term Memory (LSTM)* dengan pembagian data latihan 70% dan data pengujian 30% dari total jumlah data. Dengan jumlah epoch 100 dan batch size 64 menghasilkan tingkat akurasi train 98,47% dan akurasi test 97,02% (*best-fitting*) serta ROC 99,9%.

**Kata kunci** — Deep Learning, Klasifikasi, Long Short-Term Memory (LSTM), Toronto Emotional Speech Set (TESS), Ucapan Emosional.

## I. PENDAHULUAN

Ucapan adalah metode komunikasi yang paling alami antara manusia, bahkan antara manusia dan mesin. Namun, kita masih tidak dapat mencapai interaksi alami antara manusia dan mesin, karena mesin saat ini tidak dapat cukup memahami status emosional manusia. Pengenalan ucapan emosi otomatis telah menjadi hotspot penelitian di bidang interaksi manusia-komputer selama dekade terakhir. Namun, karena kurangnya penelitian tentang hubungan temporal yang melekat dari bentuk gelombang ucapan, akurasi pengenalan saat ini perlu ditingkatkan. Selama berbicara, diketahui bahwa orang mungkin terpengaruh oleh kondisi fisik mereka di lingkungan mereka sendiri, dan kondisi dunia luar di

sekitar mereka, emosi secara inheren dapat menunjukkan keragaman dan variasi. Pengenalan ucapan emosi merupakan tantangan utama yang menarik bagi para peneliti. Sejumlah besar upaya telah dilakukan untuk mengklasifikasikan keadaan emosional yang akurat, namun sampai saat ini masih belum diketahui, metode apa yang paling tepat untuk digunakan.

Untuk memanfaatkan sepenuhnya perbedaan saturasi emosional antara kerangka waktu, metode baru diusulkan untuk pengenalan ucapan menggunakan fitur ucapan yang dikombinasikan dengan jaringan saraf berulang berbasis Long Short-Term Memory (LSTM) [1]. Pengenalan ucapan emosional bertujuan untuk secara otomatis mengklasifikasikan unit ucapan (mis., Ucapan) ke dalam keadaan emosional, seperti marah, bahagia, netral, sedih, muak, dan terkejut.

## II. PENELITIAN TERKAIT

(Ververidis, Dimitrios dan Constantine Kotropoulos, 2015), pada penelitian ini berfokus pada klasifikasi ucapan emosional berdasarkan informasi gender menggunakan metode Sequential Forward Selection. Ketika pengklasifikasi Bayes dengan PDF Gaussian digunakan, tingkat klasifikasi yang benar sebesar 61,1% diperoleh untuk subjek laki-laki dan tingkat yang sesuai sebesar 57,1% untuk subjek perempuan. Dalam eksperimen yang sama, pengenalan ucapan Emosional acak bertujuan untuk secara otomatis mengklasifikasikan unit ucapan (misalnya ucapan) ke dalam keadaan emosional, seperti kemarahan, kebahagiaan, netral, kesedihan, dan kejutan [2].

(Tamulevičius G., Karbauskaitė R. and Dzemyda G. (2019), melakukan penelitian klasifikasi ucapan emosional menggunakan fitur berbasis dimensi fraktal. Hasil penelitian menunjukkan akurasi rata-ratanya adalah 96,5 % [3].

(Yi-Lim Lin, Gang Wei, 2005), melakukan penelitian pengenalan ucapan emosional menggunakan Hidden Markov Model (HMM) dan Support Vektor Machine (SVM). Dengan cara mengklasifikasikan keadaan emosi seseorang berdasarkan gender. Dari hasil penelitian tersebut, tingkat pengenalan oleh pengklasifikasi HMM adalah 98,9% untuk subjek perempuan, 100% untuk subjek laki-laki, dan 99,5% untuk kasus independen gender. Ketika klasifikasi SVM dan vektor fitur yang diusulkan digunakan, tingkat klasifikasi yang benar dari 89,4%, 93,6% dan 88,9% diperoleh masing-

masing untuk kasus independen gender laki-laki dan perempuan [4].

### III. BAHAN DAN METODE

#### A. Dataset

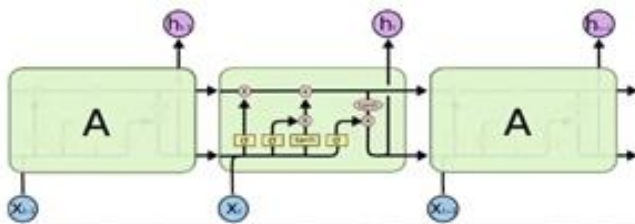
Data yang digunakan pada penelitian ini merupakan dataset tentang ucapan emosional seseorang, yang diambil dari website Kaggle [5]. Dataset *Toronto Emotional Speech Set (TESS)* ini terdiri dari 2.800 file audio, dengan 14 kelas yang masing-masing kelas berisi 200 file audio.

Dataset	<i>Toronto Emotional Speech Set (TESS)</i>	
Type	Wav	
Jumlah File	2.800	
Kelas	1. OAF Fear	(200)
	2. OAF Pleasant Surprise	(200)
	3. OAF Sad	(200)
	4. OAF Angry	(200)
	5. OAF Disgust	(200)
	6. OAF Happy	(200)
	7. OAF Neutral	(200)
	8. YAF Angry	(200)
	9. YAF Disgust	(200)
	10. YAF Fear	(200)
	11. YAF Happy	(200)
	12. YAF Neutral	(200)
	13. YAF Pleasant Surprise	(200)
	14. YAF Sad	(200)

Tabel 1. Dataset Toronto Emotional Speech Set (TESS)

#### B. Arsitektur LSTM

*Long Short-Term Memory (LSTM)* merupakan salah satu jenis dari *Recurrent Neural Network (RNN)* dimana dilakukan modifikasi pada *RNN* dengan menambahkan *memory cell* yang dapat menyimpan informasi untuk jangka waktu yang lama. *LSTM* diusulkan sebagai solusi untuk mengatasi terjadinya *vanishing gradient* pada *RNN* saat

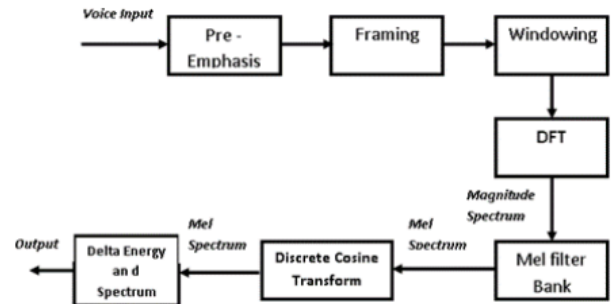


memproses data sequential yang panjang.

Gambar 1. Struktur LSTM

#### C. Mel-Frequency Cepstral Coefficients (MFCC)

*Mel Frequency Cepstral Coefficients (MFCC)* merupakan salah satu metode yang banyak digunakan dalam bidang speech technology, baik speaker recognition maupun speech recognition. Metode ini digunakan untuk melakukan feature extraction, sebuah proses yang mengkonversikan sinyal suara menjadi beberapa parameter.



Gambar 2. Ekstraksi Fitur MFCC

#### D. Confusion Matrix

Confusion matrix adalah salah satu tools analitik prediktif yang menampilkan dan membandingkan nilai aktual atau nilai sebenarnya dengan nilai hasil prediksi model yang dapat digunakan untuk menghasilkan metrik evaluasi seperti Accuracy (akurasi), Precision, Recall dan F1-Score atau F-Measure [7].

Ada empat nilai yang dihasilkan di dalam tabel confusion matrix, di antaranya True Positive (TP), False Positive (FP), False Negative (FN) dan True Negative (TN). Ilustrasi tabel confusion matrix dapat dilihat pada gambar berikut.

Nilai Prediksi	Positive	Negative
	Positive	Negative
	TP	FP
	FN	TN

Tabel 2. Nilai Confusion Matrix

##### i. Precision

Precision adalah peluang kasus yang diprediksi positif yang pada kenyataannya termasuk kasus kategori positif.

$$Presisi = \frac{TP}{TP+FP} \quad (1)$$

##### ii. Recall

Recall adalah peluang kasus dengan kategori positif yang dengan tepat diprediksi positif.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

##### iii. F1-Score

Nilai F1-Score atau dikenal juga dengan nama F-Measure didapatkan dari hasil presisi dan recal antar kategori hasil prediksi dengan kategori sebenarnya.

$$F1 = \frac{2 \times Recall \times Presisi}{Recall+Presisi} \quad (3)$$

##### iv. Accuracy

Nilai akurasi didapatkan dari jumlah data bernilai positif yang diprediksi positif dan data bernilai negatif yang diprediksi negatif dibagi dengan jumlah seluruh data di dalam dataset.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (4)$$

v. Loss dan Validasi Loss

Loss function digunakan untuk mengoptimalkan algoritma Machine Learning. Loss function dihitung berdasarkan training data dan validasi data serta interpretasinya didasarkan pada seberapa baik kinerja model dalam dua set ini. Ini adalah jumlah kesalahan yang dibuat untuk setiap contoh dalam set training atau validasi. Nilai loss menyiratkan seberapa buruk atau baiknya suatu model berperilaku setelah setiap iterasi optimasi.

Validation loss adalah metrik yang sama dengan Training Loss, tetapi tidak digunakan untuk memperbarui bobot. Ini dihitung dengan cara yang sama dengan menjalankan jaringan maju melalui input  $x_i$  dan membandingkan output jaringan  $y_i$  dengan nilai-nilai kebenaran dasar  $y_i$  menggunakan loss function.

$$J = \frac{1}{N} \sum_{i=1}^N l(\hat{y}_i, y_i) \tag{5}$$

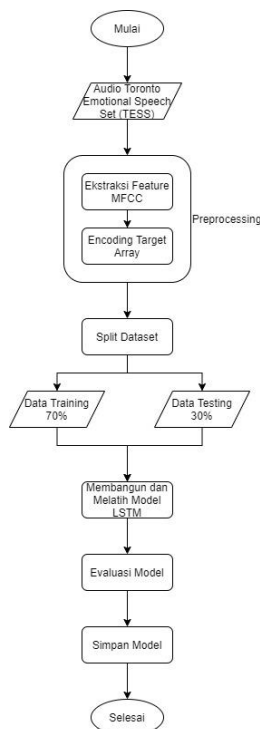
IV. IMPLEMENTASI

A. Instrumen Penelitian

Perangkat keras yang digunakan dalam penelitian ini adalah laptop tipe Acer Nitro 5 dengan CPU intel core i5 8300H, GPU Nvidia GTX 1050 4 GB, RAM 16 GB DDR4 dan sistem operasi Windows 10 Home 64-Bit. Sedangkan perangkat lunak yang digunakan adalah *Google Colab* untuk membangun dan menguji model pada penelitian ini.

B. Alur Kerja Implementasi

Berikut adalah flowchart alur kerja dari implementasi penelitian ini.



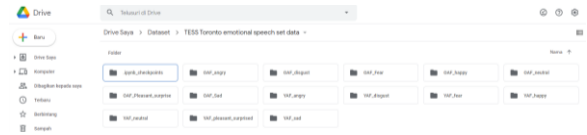
Gambar 3. Alur Kerja Pengujian LSTM

C. Implementasi

Untuk implementasinya menggunakan *Google Colab* dengan bahasa pemrograman Python pada browser *Google Chrome*.

i. Import Dataset

Hal pertama yang dilakukan adalah meng-upload Dataset *TESS* ke *Google Drive*, yang nantinya akan digunakan pada *Google Colab*.



Selanjutnya dataset *TESS* yang sudah berada di *Google Drive* dipanggil ke *Google Colab* dengan memasukkan alamat dari Dataset.

```
directory_root = "/content/drive/MyDrive/Dataset/TESS Toronto emotional speech set data/"
```

ii. Import Libraries

Berikut merupakan librari yang digunakan untuk melakukan pemodelan dan pengujian pada penelitian ini:

```
1 import numpy as np
2 import pickle
3 import cv2
4 import random as rd
5 import tensorflow as tf
6 import os, random, time
7 import matplotlib.pyplot as plt
8 import matplotlib.image as mpimg
9 import seaborn as sns
10
11 from os import listdir
12 from tensorflow import keras
13 from tensorflow.keras import layers
14 from tensorflow.keras import backend as K
15 from tensorflow.keras import regularizers, activations
16 from tensorflow.keras.optimizers import Adam
17 from tensorflow.keras.preprocessing import image
18 from tensorflow.keras.preprocessing.image import ImageDataGenerator
19 from tensorflow.keras.preprocessing.image import img_to_array
20 from tensorflow.keras.layers import Input, Convolution2D, MaxPooling2D
21 from tensorflow.keras.layers import Dense, Dropout, Flatten, Activation, LSTM
22 from tensorflow.keras.models import Model, Sequential
23 from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, ModelCheckpoint, TensorBoard, ProgbarLogger
24 from tensorflow.keras.utils import to_categorical
25
26 from sklearn.utils import shuffle
27 from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc
28 from sklearn.model_selection import train_test_split
29 from sklearn.preprocessing import MultiLabelBinarizer
30 from sklearn.preprocessing import LabelBinarizer
31 from sklearn import metrics
32 from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
33 from sklearn.preprocessing import LabelEncoder
34 import itertools
35 import librosa
36 import librosa.display
```

iii. Pembuatan Dataframe

Setelah dataset sudah terhubung dengan *Google Colab*, dataset audio *TESS* tersebut dibaca kemudian dibuatkan *Dataframe*.

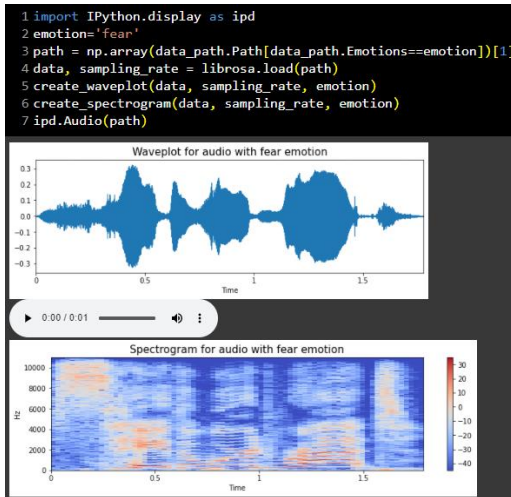
```
1 tess_directory_list = os.listdir(directory_root)
2
3 file_emotion = []
4 file_path = []
5
6 for dir in tess_directory_list:
7     directories = os.listdir(directory_root + dir)
8     for file in directories:
9         part = file.split('.')[0]
10        part = part.split('_')[2]
11        if part=='ps':
12            file_emotion.append('surprise')
13        else:
14            file_emotion.append(part)
15        file_path.append(directory_root + dir + '/' + file)
16
17 # dataframe for emotion of files
18 emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])
19
20 # dataframe for path of files.
21 path_df = pd.DataFrame(file_path, columns=['Path'])
22 Tess_df = pd.concat([emotion_df, path_df], axis=1)
23 Tess_df.head()
```

Hasil dari *Dataframe* tersebut kemudian disimpan dalam format csv dengan nama `data_path`.

```
1 data_path = pd.concat([Tess_df], axis = 0)
2 data_path.to_csv("data_path.csv", index=False)
3 data_path.head()
```

#### iv. Visualisasi Data Audio

Dari *Dataframe* yang telah dibuat, dapat ditampilkan visualisasi *waveplot* dan *spectrogram*nya. Berikut adalah contoh visualisasi salah satu kelas audio.



#### v. Ekstraksi Fitur MFCC

Ekstraksi ciri *MFCC* mempunyai tujuan untuk mendapatkan fitur berupa parameter-parameter. *MFCC* mempunyai tujuh tahap. Tahap pertama adalah pre-emphasis, kedua frame blocking, ketiga windowing, keempat *Fast Fourier Transform (FFT)*, kelima *Mel Frequency Wrapping (MFW)*, keenam *Discrete Cosine Transform (DCT)* dan ketujuh *Cepstral Liftering*.

```
1 #mengextrak koefisien Mel-frequency cepstral
2 def features_extractor(file):
3     audio, sample_rate = librosa.load(file, res_type='kaiser_fast')
4     mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
5     mfccs_scaled_features = np.mean(mfccs_features.T, axis=0)
6     return mfccs_scaled_features
```

#### vi. Label Encoding

Label encoding bertujuan untuk mengubah label kelas yang berbentuk kategorial menjadi bentuk numerik, jadi yang kelas awalnya adalah *angry* berubah menjadi 0.

```
1 # Encoding the categorical target array
2 encoder = LabelEncoder()
3 y = to_categorical(encoder.fit_transform(y))
4 y
```

#### vii. Split Data

Pembagian data menjadi data training dan data testing dengan rasio pembagian 70:30, dimana 70% untuk data training dan 30% untuk data testing.

```
1 # Splitting the data into training and testing
2 X_train, X_test, y_train, y_test = train_test_split(X, y,
3                                                    test_size=0.3,
4                                                    random_state = 42,
5                                                    stratify=y)
6 print(len(X_test), len(X_train))
```

#### viii. Membuat Model dan Melakukan Compile LSTM

```
1 print('Build LSTM RNN model ...')
2 model = Sequential()
3 model.add(LSTM(units=64, dropout=0.2, return_sequences=True, input_shape=(X_train.shape[1],)))
4 model.add(LSTM(units=32, dropout=0.2, return_sequences=True))
5 model.add(LSTM(units=16))
6 model.add(Dense(units=64, activation='relu'))
7 model.add(Dense(len(emotion), activation='softmax'))
8 model.compile(loss='categorical_crossentropy', optimizer='Adam', metrics=['accuracy'])
9 model.summary()
```

#### ix. Melatih Model

Setelah model *LSTM* selesai di compile, langkah selanjutnya adalah melatih model *LSTM* dengan menggunakan data training dan data testing, serta jumlah epoch sebanyak 100 dan batch size sebesar 64.

```
1 num_epochs = 100
2 num_batch_size = 64
3 # Fitting the model
4 history = model.fit(
5     X_train, y_train,
6     batch_size=num_batch_size,
7     epochs=num_epochs,
8     validation_data=(X_test, y_test),
9     callbacks=[callback],
10    verbose=1)
```

#### x. Menghitung Akurasi

```
1 # Keras reported accuracy:
2 score = model.evaluate(X_train, y_train, verbose=0)
3 print ("model train data score: ", round(score[1]*100, 2), "%")
4
5 score = model.evaluate(X_test, y_test, verbose=0)
6 print ("model test data score: ", round(score[1]*100, 2), "%")
```

#### xi. Menampilkan Plot Akurasi dan Loss

```
1 acc = history.history['accuracy']
2 val_acc = history.history['val_accuracy']
3 loss = history.history['loss']
4 val_loss = history.history['val_loss']
5 epochs = range(1, len(acc) + 1)
6
7 #Train dan validation accuracy
8 plt.plot(epochs, acc, 'b', label='Training accuracy')
9 plt.plot(epochs, val_acc, 'r', label='Validation accuracy')
10 plt.title('Training and Validation accuracy LSTM')
11 plt.legend()
12
13 plt.figure()
14 #Train and validation loss
15 plt.plot(epochs, loss, 'b', label='Training loss')
16 plt.plot(epochs, val_loss, 'r', label='Validation loss')
17 plt.title('Training and Validation loss LSTM')
18 plt.legend()
19 plt.show()
```

#### xii. Menampilkan Confusion Matrix

```
1 pred = model.predict(X_test)
2 pred = np.argmax(pred, axis=1)
3 y_test2 = np.argmax(y_test, axis=1)
4 cm = confusion_matrix(y_test2, pred)
5 np.set_printoptions(precision=2)
6 ax = plt.subplot()
7 sns.heatmap(cm, annot=True, ax = ax, fmt=""); #annot=True to annotate cells
8
9 # labels, title and ticks
10 ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
11 ax.set_title('Confusion Matrix');
12 ax.xaxis.set_ticklabels(label_binarizer.classes_, rotation=90);
13 ax.yaxis.set_ticklabels(label_binarizer.classes_, rotation=0);
```

#### xiii. Menampilkan Plot ROC AUC

```
1 fpr, tpr, thresholds = roc_curve(y_test.ravel(), model_pred.ravel())
2 auc_ = auc(fpr, tpr)
3
4 plt.figure(1)
5 plt.plot([0, 1], [0, 1], 'k--')
6 plt.plot(fpr, tpr, label='ROC (area = {:.3f})'.format(auc_))
7 plt.xlabel('False positive rate')
8 plt.ylabel('True positive rate')
9 plt.title('ROC curve')
10 plt.legend(loc='best')
11 plt.show()
```

xiv. Menampilkan Hasil Classification Report

```
1 from sklearn.metrics import classification_report, plot_roc_curve
2 print(classification_report(y_test2, pred))
```

V. HASIL DAN PEMBAHASAN

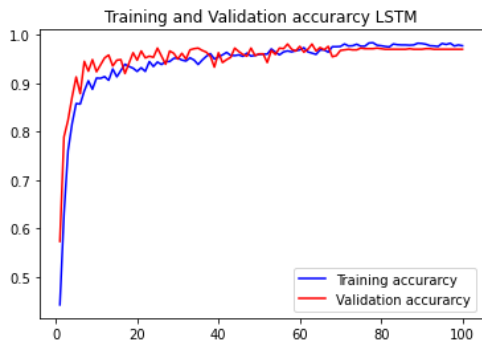
Dari hasil training dan testing pada model LSTM, didapatkan akurasi model data train 98,47% dan akurasi model data test 97,02%.

```
model train data score: 98.47 %
model test data score: 97.02 %
```

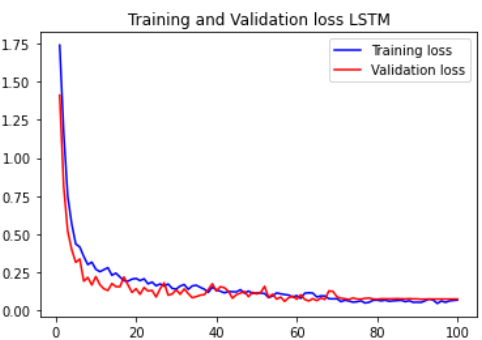
Untuk hasil dari *training* model bisa dilihat pada gambar berikut dengan 10 Epoch pertama.

```
Epoch 1/100 ..... - 11s 41ms/step - loss: 1.7485 - accuracy: 0.4429 - val_loss: 1.4128 - val_accuracy: 0.5738
Epoch 2/100 ..... - 0s 11ms/step - loss: 1.1636 - accuracy: 0.6296 - val_loss: 0.8116 - val_accuracy: 0.7893
Epoch 3/100 ..... - 0s 11ms/step - loss: 0.7485 - accuracy: 0.7607 - val_loss: 0.5202 - val_accuracy: 0.8238
Epoch 4/100 ..... - 0s 12ms/step - loss: 0.5665 - accuracy: 0.8168 - val_loss: 0.3983 - val_accuracy: 0.8714
Epoch 5/100 ..... - 0s 12ms/step - loss: 0.4362 - accuracy: 0.8582 - val_loss: 0.3157 - val_accuracy: 0.9131
Epoch 6/100 ..... - 0s 12ms/step - loss: 0.4366 - accuracy: 0.8571 - val_loss: 0.3379 - val_accuracy: 0.8786
Epoch 7/100 ..... - 0s 12ms/step - loss: 0.3548 - accuracy: 0.8842 - val_loss: 0.1918 - val_accuracy: 0.9452
Epoch 8/100 ..... - 0s 12ms/step - loss: 0.35397 - accuracy: 0.9051 - val_loss: 0.2172 - val_accuracy: 0.9258
Epoch 9/100 ..... - 0s 12ms/step - loss: 0.3178 - accuracy: 0.8878 - val_loss: 0.1673 - val_accuracy: 0.9488
Epoch 10/100 ..... - 0s 12ms/step - loss: 0.2781 - accuracy: 0.9187 - val_loss: 0.2225 - val_accuracy: 0.9238
```

Berdasarkan hasil *training* sebanyak 100 Epoch model LSTM tadi, bisa dibuatkan plot akurasi dan lossnya dengan hasil sebagai berikut.

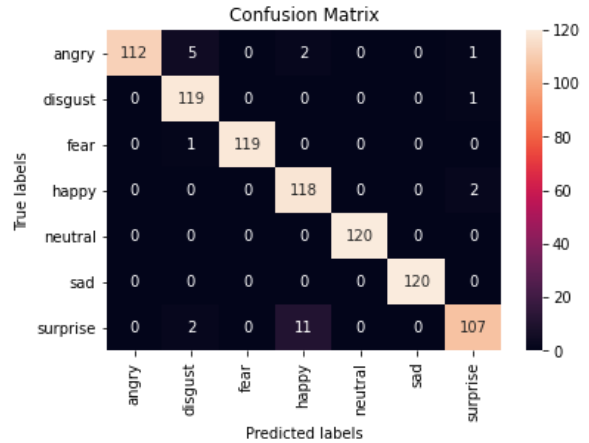


Gambar 4. Plot Akurasi Model LSTM



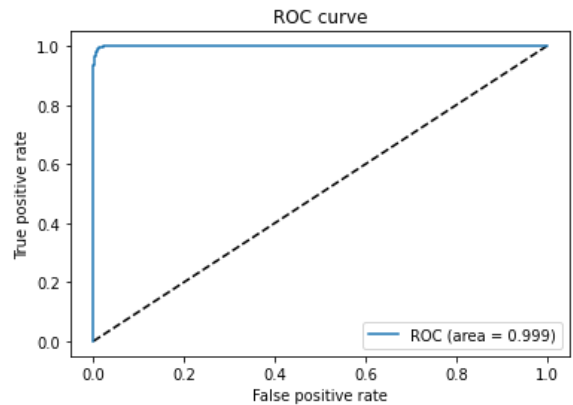
Gambar 5. Plot Loss Model LSTM

Untuk hasil klasifikasi dapat dilihat pada Confusion Matrix berikut.



Gambar 6. Confusion Matrix

Hasil plot kurva ROCnya sebagai berikut.



Gambar 7. Kurva ROC

Untuk hasil classification report dari Confusion Matrix tersebut bisa dilihat pada hasil dibawah.

	precision	recall	f1-score	support
0	1.00	0.93	0.97	120
1	0.94	0.99	0.96	120
2	1.00	0.99	1.00	120
3	0.90	0.98	0.94	120
4	1.00	1.00	1.00	120
5	1.00	1.00	1.00	120
6	0.96	0.89	0.93	120
accuracy			0.97	840
macro avg	0.97	0.97	0.97	840
weighted avg	0.97	0.97	0.97	840

Gambar 8. Hasil Classification Report Dari Confusion Matrix

VI. KESIMPULAN

Penelitian dengan melakukan klasifikasi ucapan emosional menggunakan metode *Long Short-Term Memory (LSTM)* pada dataset audio *Toronto Emotional Speech Set (TESS)* yang terdiri dari 7 kelas audio, yaitu Angry, Disgust, Fear, Happy, Neutral, Pleasant Surprise (disingkat jadi Surprise) dan Sad dengan jumlah data audio yang digunakan adalah 2.800 data audio yang dibagi menjadi data *training* dan data *testing* dengan perbandingan 70:30. Jumlah data *training*

sebanyak 1960 dan data *testing* sebanyak 840 dengan masing-masing setiap kelas audio terbagi sebanyak 120. Untuk hasil akurasi yang didapatkan dengan jumlah Epoch 100 dan batch size 64 adalah 98,47% untuk model data train dan 97,02% untuk model data test serta nilai ROC Curvenya 99,9%. Hasil prediksi yang dari total 840 data audio, hanya 25 data audio saja yang terjadi salah prediksi dengan kelas audio Surprise mengalami paling banyak salah prediksi, yaitu 13 data (recall 89%).

#### DAFTAR ACUAN

- [1] Yue Xie; Ruiyu Liang; Zhuang Liang; Chengwei Huang; Cairong Zou; Bjorn Schuller. “*Speech Emotion Classification Using Attention-Based LSTM*” Volume: 27, Issue: 11,1675 – 1685.2019.
- [2] Ververidis, Dimitrios dan Constantine Kotropoulos, “*Automatic speech classification to five emotional states based on gender information*”, 2015.
- [3] Tamulevičius G., Karbauskaitė R. and Dzemyda G. “*Speech emotion classification using fractal dimension-based features*”, “*Nonlinear Analysis: Modelling and Control*” 24(5), pp. 679–695, 2019.
- [4] Yi-Lim Lin, Gang Wei, *Speech emotion recognition based on HMM and SVM*, 2005.
- [5] Eu Jin Lok, *Toronto Emotional Speech Set (TESS)*, Available : <https://www.kaggle.com/ejlok1/toronto-emotional-speech-set-tess>
- [6] Devi Hardianti, *Implementasi Long Short-Term Memory (LSTM) Pada Pengenalan Tulisan Tangan*, 2020.
- [7] Lutfia Afifah, Apa Itu Confusion Matrix di Machine Learning?, Available : <https://ilmudatapy.com/apa-itu-confusion-matrix/>