P-ISSN : 2338-2899
E-ISSN: 2807-1271

Implementasi Algoritma *Deep Artificial Neural Network* Menggunakan *Mel Frequency Cepstrum Coefficient* Untuk Klasifikasi Audio Emosi Manusia

Ajrana[1], Aynun Akbar[2], Armin Lawi[3]

[123]Program Studi Sistem Informasi, Departemen Matematika, Universitas Hasanuddin

Email : [1]ajrana18h@student.unhas.ac.id [2]akbara18h@student.unhas.ac.id [3]armin@unhas.ac.id

***Abstract* — Emotion is a state that is felt by each individual in a high intensity towards something. Emotions are difficult to understand and difficult to measure quantitatively. Emotions can be reflected in facial expressions and tone of voice. Sound contains physical properties that are unique to each speaker. Everyone has a different tone, tempo, and rhythm. Therefore, identification of human emotions is useful in the field of human and computer interaction. It helps develop software interfaces that can be implemented in community service centers, banks, education and more. In this study, a model based on deep artificial neural network (deep ann) is used in classifying sound emotions. The dataset used is the "toronto emotional speech set data" with 14 classes and 2,800 audio data. Deep ann is composed of 2 hidden layers with 100 and 7 neurons respectively using the rectified linear unit (relu) activation function. Feature extraction is applied to all audio files using the mel frequency cepstrum coefficient (mfcc) method. Based on the results obtained, this deep ann-based architecture with 100 epochs gets a very good level of accuracy with an accuracy value of 99.71%, an average precision of 99.97%, an average recall of 99.97%, and an average f1 score of 99.97%.**

***Keywords* — deep learning, classification, ANN, MFCC, Toronto Emotion Speech Set.**

Abstrak — Emosi merupakan keadaan yang dirasakan pada setiap individu dalam intensitas yang tinggi terhadap sesuatu hal. Emosi sulit dipahami dan sulit diukur secara kuantitatif. Emosi dapat tercermin dalam ekspresi wajah dan nada suara. Suara mengandung sifat fisik yang unik untuk setiap pembicara. Setiap orang memiliki warna nada, tempo, dan ritme yang berbeda. Oleh karena itu, Identifikasi emosi manusia berguna dalam bidang interaksi manusia dan komputer. Ini membantu mengembangkan antarmuka perangkat lunak yang dapat diterapkan di pusat layanan masyarakat, bank, pendidikan, dan lainnya. Pada penelitian ini, digunakan model berbasis Deep Artificial Neural Network (Deep ANN) dalam mengklasifikasikan emosi suara. Dataset yang digunakan ialah “Toronto Emotional Speech Set” dengan 14 class dan 2.800 data audio. Deep ANN tersusun dari 2 hidden layer dengan masing-masin 100 dan 7 neuron menggunakan fungsi aktivasi Rectified Linear Unit (ReLU). Ekstraksi fitur diberlakukan untuk semua file audio menggunakan metode Mel Frequency Cepstrum Coefficient (MFCC). Berdasarkan hasil yang diperoleh, arsitektur berbasis Deep ANN ini dengan 100 epoch mendapatkan tingkat akurasi yang sangat baik dengan nilai akurasi adalah 99.71%, presisi rata-rata 99.97%, recall rata-rata 99.97%, dan skor F1 rata-rata 99.97%.

Kata Kunci — Deep Learning, Classification, ANN, MFCC, Toronto Emotion Speech Set Data.

I. PENDAHULUAN

Emosi merupakan keadaan yang dirasakan pada setiap individu dalam intensitas yang tinggi terhadap sesuatu hal. Emosi juga bisa disebut sebagai reaksi akibat timbal balik atas tindakan seseorang ataupun kejadian yang dialami pemilik emosi. Seringkali emosi mengakibatkan perubahan perilaku yang berakibat terganggunya hubungan dengan lingkungan. Emosi dapat dikategorikan menjadi emosi positif dan negatif dalam jenisnya. Beberapa kategori emosi positif adalah senang, kepedulian, ketertarikan, antusias, kebosanan dan keingintahuan. Beberapa kategori emosi negatif adalah marah, sedih, takut, iri dan kebencian.

Ada dua representasi emosi yang banyak digunakan dalam beberapa penelitian, yaitu kontinu dan diskrit. Dalam representasi kontinu, emosi suatu ucapan dapat diekspresikan sebagai nilai-nilai berkelanjutan sepanjang berbagai dimensi psikologis. Menurut Ayadi, Kamel, & Karray (2011) , "emosi dapat dicirikan dalam dua dimensi: aktivasi dan valensi." Aktivasi adalah "jumlah energi yang dibutuhkan untuk mengekspresikan emosi tertentu" dan penelitian telah menunjukkan bahwa kegembiraan, kemarahan, dan ketakutan dapat dikaitkan dengan energi dan nada tinggi dalam ucapan, sedangkan kesedihan dapat dikaitkan dengan energi rendah dan bicara lambat. Valensi memberi lebih banyak nuansa dan membantu membedakan antara emosi seperti marah dan bahagia karena peningkatan aktivasi dapat menunjukkan keduanya. Dalam representasi diskrit, emosi dapat diekspresikan secara diskrit sebagai kategori tertentu, seperti marah, sedih, bahagia, dan lain-lain.

Kinerja sistem pengenalan emosi murni bergantung pada fitur / representasi yang diekstraksi dari audio. Mereka secara luas diklasifikasikan ke dalam fitur berbasis waktu dan frekuensi. Penelitian ekstensif telah dilakukan untuk menimbang pro dan kontra dari feature. Tidak ada fitur suara tertentu yang dapat berkinerja baik di semua tugas pemrosesan sinyal suara. Selain itu, fitur

dibuat dengan tangan agar sesuai dengan persyaratan masalah yang ada.

Berbagai eksperimen/percobaan yang telah dilakukan peneliti sebelumnya untuk mengidentifikasi emosi dari ucapan untuk berbagai bahasa dan aksen. Chenchah dan Lachiri [3] mempelajari kinerja Mel-Frequency Cepstral Coefficients dan Linear Frequency Cepstral Coefficient (LPCC) dalam mengidentifikasi emosi menggunakan Hidden Markov Model (HMM) dan Support Vector Machines (SVM). Model yang dikembangkan menghasilkan akurasi 61% pada Database Surrey Audio-Visual Expressed Emotion (SAVEE). Parthasarathy dan Tashev [5] telah membandingkan model DNN, RNN, dan 1D-CNN pada fitur MFCC dari kumpulan data bahasa Cina. Mereka telah mencapai akurasi 56% dengan model CNN 1D.

Emosi sulit dipahami dan sulit diukur secara kuantitatif. Emosi dapat tercermin dalam ekspresi wajah dan nada suara. Identifikasi emosi manusia berguna dalam bidang interaksi manusia dan komputer. Ini membantu mengembangkan antarmuka perangkat lunak yang dapat diterapkan di pusat layanan masyarakat, bank, pendidikan, dan lainnya. Dengan demikian, digunakan Metode Deep Learning yaitu Metode Artificial Neural Network (ANN) untuk mengklasifikasikan data audio emosi manusia.

Metode Deep Learning yang saat ini memiliki hasil paling signifikan dalam pengenalan suara yaitu Artificial Neural Network (ANN). Oleh sebab itu fokus dari penelitian ini adalah implementasi algoritma deep artificial neural network menggunakan mel frequency cepstrum coefficient untuk klasifikasi data audio emosi manusia pada dataset Toronto Emotional Speech Set dengan jumlah 2.800 data audio dengan hasil akurasi yang sangat bagus.

II. Penelitian Terkait

(T.M.Rajisha, A.P. Sunija, K.S. Riyas) melakukan pengenalan otomatis emosi dari ucapan oleh mesin telah menjadi salah satu bidang penelitian yang paling menantang di bidang interaksi mesin manusia. Sistem pengenalan emosi otomatis dengan ucapan untuk memantau dan mengidentifikasi keadaan emosi atau fisiologis seseorang dari ucapannya. Database emosional ucapan untuk bahasa Malayalam (salah satu bahasa India Selatan) dan sistem untuk mengenali emosi. Sistem yang digunakan adalah Mel Frequency Cepstral Coefficients (MFCC), Short Time Energy (STE) dan Pitch sebagai teknik ekstraksi ciri. Dua pengklasifikasi yaitu ANN dan SVM digunakan untuk klasifikasi pola. Percobaan menunjukkan bahwa metode ini memberikan akurasi yang tinggi sebesar 88.4% untuk JST dan 78,2% untuk SVM.

(Mituk Kumar Ahirwal & Mangesh Ramaji Kose, 2019) Pada Penelitian ini melakukan klasifikasi emosi dengan sinyal elektroensefalografi (EEG). Stimulasi audio-visual digunakan untuk membangkitkan emosi pada saat percobaan. Setelah merekam sinyal EEG, ekstraksi fitur dan klasifikasi diterapkan untuk mengklasifikasikan emosi (senang, marah, sedih dan santai). sorotan utama dari studi ini yaitu, identifikasi/karakterisasi rangsangan audio-visual yang menghasilkan emosi berbahaya dan pendekatan yang diusulkan untuk mengurangi jumlah saluran EEG untuk klasifikasi emosi. Tujuan dari identifikasi tersebut yang bertanggung jawab atas emosi berbahaya seperti sedih dan marah untuk mengontrol akses mereka atas media sosial dan flatfrom publik lainnya. Saluran EEG dipilih berdasarkan kemungkinan aktivasinya, dihitung dari matriks korelasi saluran EEG. Tiga jenis fitur diekstraksi dari sinyal EEG, domain waktu, domain frekuensi, dan berbabasis entropi. Setelah ekstraksi fitur, tiga algoritma berbeda, support vector machine (SVM), artificial neural network (ANN) dan naive bayes (NB) digunakan untuk mengklasifikasikan emosi. Studi ini dilakukan melalui database DEAP (Database untuk analisisi emosi menggunakan sinyal fisiologis) dari sinyal EEG yang direkam pada keadaan emosi yang berbeda subjek. Setelah dilakukan analisis, JST ditemukan sebagai pengklasifikasi terbaik dengan rata-rata akurasi 97.74%. Di antara fitur yang terdaftar, fitur berbasis entropi ditemukan sebagai fitur terbaik dengan akurasi rata-rata 90.53%.

(Xianxin Ke, Yjiao Zhu, Lei Wen, and Wenzhen Zhang, 2018) Pengenalan emosi ucapan terutama mencakup ekstraksi fitur emosi, pengukuran fitur dan ucapan model pengenalan emosi. pada penelitian ini memilih emosi yang valid fitur dan mengekstrak nilai statistik emosional fitur. Model pengenalan emosi ucapan dibangun berdasarkan SVM dan ANN dan efek pengenalan reduksi fitur masing-masing pada dua jenis model tersebut dibandingkan. Hasil percobaan menunjukkan bahwa, berdasarkan fitur emosi yang dieksekusi oleh korpus emosi CASIA, pengukuran fitur dapat meningkatkan akurasi pengenalan dan efek pengenalan suara model SVM lebih baik dari model ANN, dengan akurasi SVM 85.83% dan akurasi ANN sebesar 75%.

III. Bahan dan Metode

A. Dataset

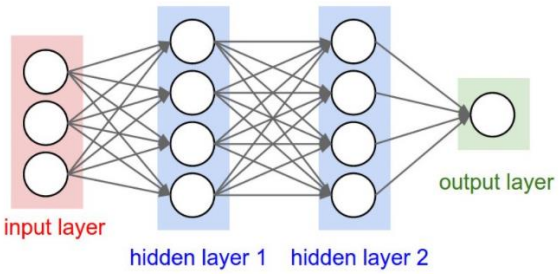
Pada penelitian ini, kami mengambil dataset dari repository kaggle. Datasetnya ialah “TESS Toronto Emotional Speech Set Data”. Link website dataset yang kami gunakan: https://www.kaggle.com/ejlok1/toronto-emotional-speech-set-tess. Pada dataset ini terdiri dari 14 class dengan jumlah file data audio sebanyak 2.800 dengan rincian class OAF_angry, OAF_disgust, OAF_Fear, OAF_happy, OAF_neutral, OAF_Pleasant_surprise, OAF_sad, YAF_angry, YAF_disgust, YAF_fear, YAF_happy, YAF_neutral, YAF_pleasant-surprised, YAF_sad.

No.	Class	Data Audio
1.	OAF_angry	200
2.	OAF_disgust	200
3.	OAF_Fear	200
4.	OAF_happy	200
5.	OAF_neutral	200
6.	OAF_Pleasant_surprise	200
7.	OAF_Sad	200
8.	YAF_angry	200
9.	YAF_disgust	200
10.	YAF_fear	200
11.	YAF_happy	200
12.	YAF_neutral	200
13.	YAF_pleasant_surprised	200
14.	YAF_sad	200
	Total	2.800

Tabel 1. Dataset Toronto emotional speech set (TESS)

B. Artificial Neural Network (ANN) Architecture

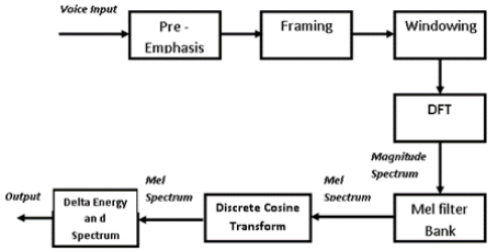
Artificial Neural Network (ANN) atau jaringan syaraf tiruan merupakan sebuah teknik atau pendekatan pengolahan informasi yang terinspirasi oleh cara kerja sistem saraf biologis, khususnya pada sel otak manusia dalam memproses informasi. Elemen kunci dari teknik ini adalah struktur sistem pengolahan informasi yang bersifat unik dan beragam untuk tiap aplikasi. Neural Network terdiri dari sejumlah besar elemen pemrosesan informasi (neuron) yang saling terhubung dan bekerja bersama-sama untuk menyelesaikan sebuah masalah tertentu, yang pada umumnya adalah masalah klasifikasi ataupun prediksi.



Gambar 1. Skema arsitektur Artificial Neural Network

C. Mel-Frecuency Cepstral Coefficients (MFCC)

MFCC adalah ekstraksi ciri yang sering digunakan pada pemrosesan suara, karena dapat merepresentasikan sinyal dengan baik. MFCC memiliki cara kerja yang didasarkan pada perbedaan frekuensi yang sesuai dengan pendengaran manusia sehingga dapat merepresentasikan sinyal suara seperti manusia mempresentasikannya.



Gambar 2. Ekstraksi Fitur MFCC

D. Confusion Matrix

Confusion matrix adalah suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi pada konsep data mining atau Sistem Pendukung Keputusan. Pada pengukuran kinerja menggunakan confusion matrix, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN).

- True Negative (TN) merupakan jumlah data negatif yang terdeteksi dengan benar.
- False Positive (FP) merupakan data negatif namun terdeteksi sebagai data positif.
- True Positive (TP) merupakan data positif yang terdeteksi benar.
- False Negative (FN) merupakan kebalikan dari True Positive, sehingga data positif, namun terdeteksi sebagai data negatif.

		True Values	
		True	False
Prediction	True	TP Correct result	FP Unexpected result
	False	FN Missing result	TN Correct absence of result

Tabel 2. Confussion Matrix

i. Presisi

Presisi adalah data yang diambil berdasarkan informasi yang kurang. Dalam klasifikasi biner, presisi dapat dibuat sama dengan nilai prediksi positif.

$$\text{Precision} = (TP / (TP + FP)) * 100\%$$

Persamaan 1. Rumus mencari Presisi

ii. Recall

Recall adalah data penghapusan yang berhasil diambil dari data yang relevan dengan kueri. Dalam klasifikasi biner, recall dikenal sebagai sensitivitas. Munculnya data relevan yang diambil adalah menyetujui dengan query dapat dilihat dengan recall. Berikut ini adalah peran recall.

$$\text{Recall} = (TP / (TP + FN)) * 100\%$$

Persamaan 2. Rumus mencari Recall

iii. Akurasi dan Validasi Akurasi

Akurasi digunakan untuk mengukur kinerja algoritma dengan cara yang dapat ditafsirkan. Akurasi suatu model biasanya ditentukan setelah parameter model dan dihitung dalam bentuk persentase. Ini adalah ukuran seberapa akurat prediksi model dibandingkan dengan data sebenarnya dan Akurasi (acc) berada pada train. Berikut ini adalah aturan akurasi.

Akurasi = (TP + TN) / (TP + TN + FP + FN)) *100%

Persamaan 3. Rumus mencari Akurasi

Sedangkan validasi acc ada di data validasi. Yang terbaik adalah mengandalkan val_acc untuk representasi yang adil dari kinerja model karena neural network yang baik pada akhirnya akan menyesuaikan data train pada 100%, tetapi hal ini akan dapat berkinerja buruk pada data yang tidak terlihat.

iv. Loss dan Validasi Loss

Loss function digunakan untuk mengoptimalkan algoritma Machine Learning. Loss function dihitung berdasarkan training data dan validasi dataserta 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 xi dan membandingkan output jaringan yi dengan nilai-nilai kebenaran dasar yi menggunakan loss function.

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

Persamaan 4. Fungsi kerugian individu

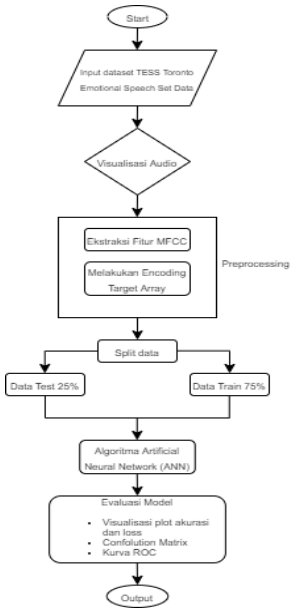
IV. IMPLEMENTASI

A. Lingkungan Komputasi Eksperimen

Jenis perangkat yang digunakan untuk melakukan implementasi algoritma ANN menggunakan MFCC (Mel-Frecuency Cepstral Coefficients) untuk klasifikasi Emotion adalah laptop Acer E5-476G-599H dan Lenovo ideapad 330S dengan spesifikasi Processor Intel® Core(TM) i5-8250U CPU @1.60GHz (8 CPUs), ~1.8GHz dan AMD A4-5000 APU with Radeon(TM) HD Graphics, RAM : 8 GB,Operating System Windows 10. Untuk membangun dan menguji model kami menggunakan Google Colab akselerator hardware GPU dan library yang di gunakan yaitu keras dengan basis tensorflow-GPU. Dengan penggunaan *google colab* ini di harapkan waktu komputasi akan lebih efisien di karenakan *google colab* yang memberikan fasilitas RAM, memori penyimpanan, dan GPU yang disediakan oleh *google.*

B. Alur Kerja Implementasi

Berikut merupakan flowchart atau alur kerja dari implementasi.



Gambar 3. Flowchart Pengujian Algoritma ANN

C. Implementasi pada Google Colab

i. Import Dataset

```
from google.colab import drive
drive.mount('/content/drive')
```

```
# Changint the directory to where the zip file of the data is stored
%cd /content/drive/MyDrive/Deep/TESS Toronto emotional speech set data
```

```
/content/drive/MyDrive/Deep/TESS Toronto emotional speech set data
```

Kode di atas meng-upload Dataset ke Google Drive lalu memuat dataset ke mesin virtual Google Collab.Adapun dataset yang kami gunakan adalah dataset *Toronto Emotional Speech Set Data.*

ii. Import Libraries

Berikut merupakan beberapa libraries yang kami gunakan

```
import librosa
import librosa.display
import matplotlib.pyplot as plt
import os
import pandas as pd
import glob
import re
import numpy as np
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
import tensorflow as tf
from keras.models import Model, Sequential
from keras.layers import Flatten, Dense, Dropout, Activation
```

iii. Pembuatan Dataframe

Membuat kerangka data dari file dalam dataset

```
df = pd.DataFrame(columns=["Path","Age","Emotion","Class"])
for file in glob.glob("*.wav"):
  age = re.findall("YAF|OAF",file)
  emotion = re.findall("[_][a-z]*",file)[1][1:]
  if age == "OAF":
    age = "Old"
    category = "Old and "+emotion
  else:
    age = "Young"
    category = "Young and "+emotion
  df = df.append({"Path":file,"Age":age,"Emotion":emotion,"Class":category},ignore_index=True)
```

iv. Penampilkan Dataframe

Dalam dataset ini menampilkan 4 kolom diantaranya Path yang merupakan kolom file audio dalam bentuk file .wav, Age merupakan kolom yang manapilkan umur dari orang yang berbicara, Emotion adalah kolom yang menampilkan jenis emosi pada suara dan Class merupakan kolom yang menampilkan suara orang tua dan suara anak muda.

```
df
```

	Path	Age	Emotion	Class
0	YAF_pass_fear.wav	Young	fear	Young and fear
1	YAF_loaf_fear.wav	Young	fear	Young and fear
2	YAF_gun_fear.wav	Young	fear	Young and fear
3	YAF_sure_fear.wav	Young	fear	Young and fear
4	YAF_chalk_fear.wav	Young	fear	Young and fear
...	...	...	...	...
2795	OAF_pick_angry.wav	Young	angry	Young and angry
2796	OAF_ditch_disgust.wav	Young	disgust	Young and disgust
2797	OAF_hole_disgust.wav	Young	disgust	Young and disgust
2798	OAF_mode_angry.wav	Young	angry	Young and angry
2799	OAF_join_disgust.wav	Young	disgust	Young and disgust

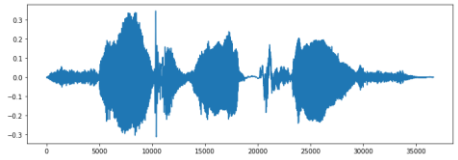
2800 rows × 4 columns

v. Visualisasi Audio

```
audio_file_path=df.iloc[0]["Path"]
librosa_audio_data,librosa_sample_rate=librosa.load(audio_file_path)

plt.figure(figsize=(12, 4))
plt.plot(librosa_audio_data)
```

```
[<matplotlib.lines.Line2D at 0x7f8e73cb8190>]
```



vi. Ekstraksi fitur Menggunakan MFCC

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

```
def features_extractor(file):
    audio, sample_rate = librosa.load(file, res_type='kaiser_fast')
    mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
    mfccs_scaled_features = np.mean(mfccs_features.T,axis=0)
    return mfccs_scaled_features
```

vii. Menampilkan Hasil Ekstraksi Fitur

Setelah mengekstraksi feature Terdapat tambahan kolom feature pada Dataframe. Kolom tersebut menampilkan parameter-parameter audio.

```
df["Features"] = df["Path"].apply(lambda x: features_extractor(x))
df
```

	Path	Age	Emotion	Class	Features
0	YAF_pass_fear.wav	Young	fear	Young and fear	[-305.61902, 39.874496, -16.49049, 18.730463, ...
1	YAF_loaf_fear.wav	Young	fear	Young and fear	[-317.14862, 35.641407, -25.952332, 17.346666,...
2	YAF_gun_fear.wav	Young	fear	Young and fear	[-288.7624, 41.99878, -26.804163, 20.576271, -...
3	YAF_sure_fear.wav	Young	fear	Young and fear	[-275.55334, 23.893112, -24.31273, 13.755532, ...
4	YAF_chalk_fear.wav	Young	fear	Young and fear	[-239.25475, 46.68057, -22.907618, 22.335852, ...
...	...	...	...	...	...
2795	OAF_pick_angry.wav	Young	angry	Young and angry	[-449.99377, 49.416267, -11.046038, 1.6740906,...
2796	OAF_ditch_disgust.wav	Young	disgust	Young and disgust	[-450.3471, 65.96312, 12.325116, 18.956816, 10...
2797	OAF_hole_disgust.wav	Young	disgust	Young and disgust	[-459.00064, 97.41574, 25.254267, 9.648966, -1...
2798	OAF_mode_angry.wav	Young	angry	Young and angry	[-384.0347, 83.048805, -6.405364, -4.797825, -...
2799	OAF_join_disgust.wav	Young	disgust	Young and disgust	[-439.95352, 92.3705, 19.522247, 8.601505, -1....

2800 rows × 5 columns

viii. Label Encoding

Label encoding digunakan untuk mengubah label yang berbentuk kategorikal menjadi numerik. Dimana angka 1 untuk yang berumur muda dan angka 0 untuk yang berumur tua.

```
X = np.array(df["Features"].to_list())
y = df["Class"].values
encoder = LabelEncoder()
y = to_categorical(encoder.fit_transform(y))
y
```

```
array([[0., 0., 1., ..., 0., 0., 0.],
       [0., 0., 1., ..., 0., 0., 0.],
       [0., 0., 1., ..., 0., 0., 0.],
       ...,
       [0., 1., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       [0., 1., 0., ..., 0., 0., 0.]], dtype=float32)
```

ix. Normalisasi Array

Membagi Dataset menjadi Data Testing dan Training

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
```

x. Membuat Model dan summary Model

```
[ ]  # Creating a sequential model
     model = Sequential()
     # Adding a dense layer of 100 neurons
     model.add(Dense(100))
     # Applying 'relu' activation
     model.add(Activation('relu'))
     # Applying dropout with probability of 0.2
     model.add(Dropout(0.2))
     model.add(Flatten())
     # Adding a dense layer of 7 neurons
     model.add(Dense(7))
     # Applying 'softmax' activation
     model.add(Activation('softmax'))
```

xi. Compile Model

Kompilasi model dengan 'categorical_crossentropy' sebagai loss dan 'rmsprop' sebagai pengoptimal.

```
[ ] model.compile(loss='categorical_crossentropy',metrics=['accuracy'],optimizer='rmsprop')
```

xii. Train Model

Dibawah ini digunakan Epoch 100 dalam melatih model.

```
[ ] num_epochs = 100
    num_batch_size = 64
    # Fitting the model
    history = model.fit(X_train, y_train, batch_size=num_batch_size,
                        epochs=num_epochs, validation_data=(X_test, y_test))
```

Berikut 10 Epoch terakhir dari hasil Train model dengan epoch = 100.

```
Epoch 91/100
33/33 [==============================] - 0s 3ms/step - loss: 0.0071 - accuracy: 0.9943 - val_loss: 0.0083 - val_accuracy: 0.9957
Epoch 92/100
33/33 [==============================] - 0s 2ms/step - loss: 0.0063 - accuracy: 0.9978 - val_loss: 0.0070 - val_accuracy: 0.9971
Epoch 93/100
33/33 [==============================] - 0s 3ms/step - loss: 0.0064 - accuracy: 0.9980 - val_loss: 0.0069 - val_accuracy: 0.9971
Epoch 94/100
33/33 [==============================] - 0s 3ms/step - loss: 0.0041 - accuracy: 0.9983 - val_loss: 0.0105 - val_accuracy: 0.9971
Epoch 95/100
33/33 [==============================] - 0s 2ms/step - loss: 0.0027 - accuracy: 0.9990 - val_loss: 0.0100 - val_accuracy: 0.9971
Epoch 96/100
33/33 [==============================] - 0s 3ms/step - loss: 0.0091 - accuracy: 0.9977 - val_loss: 0.0078 - val_accuracy: 0.9971
Epoch 97/100
33/33 [==============================] - 0s 3ms/step - loss: 0.0046 - accuracy: 0.9988 - val_loss: 0.0086 - val_accuracy: 0.9957
Epoch 98/100
33/33 [==============================] - 0s 2ms/step - loss: 0.0026 - accuracy: 0.9992 - val_loss: 0.0112 - val_accuracy: 0.9971
Epoch 99/100
33/33 [==============================] - 0s 3ms/step - loss: 0.0070 - accuracy: 0.9981 - val_loss: 0.0071 - val_accuracy: 0.9971
Epoch 100/100
33/33 [==============================] - 0s 2ms/step - loss: 0.0033 - accuracy: 0.9994 - val_loss: 0.0134 - val_accuracy: 0.9971
```

Dari hasil tersebut diketahui CPU times dari training dengan waktu rata-rata setiap epoch adalah 11 detik.

xiii. Menghitung Akurasi Model

```
[ ] scores = model.evaluate(X_test, y_test)
    print(f"Test Accuracy: {scores[1]*100}")
```

xiv. Menampilkan Plot

```
[ ]  # Visualizing the loss and accuracy
     plt.figure(figsize=(12, 6), dpi=80)
     plt.subplot(1, 2, 1)
     plt.plot(history.history['accuracy'])
     plt.plot(history.history['val_accuracy'])
     plt.title('Model accuracy')
     plt.ylabel('Accuracy')
     plt.xlabel('Epoch')
     plt.legend(['Train', 'Test'], loc='upper left')
     plt.plot()

     plt.subplot(1, 2, 2)
     plt.plot(history.history['loss'])
     plt.plot(history.history['val_loss'])
     plt.title('Model loss')
     plt.ylabel('Loss')
     plt.xlabel('Epoch')
     plt.legend(['Train', 'Test'], loc='upper left')
     plt.plot()
     plt.show()
```

xv. Confusion Matriks

```
[ ] import seaborn as sns
    from sklearn.preprocessing import MultiLabelBinarizer
    from sklearn.preprocessing import LabelBinarizer

    model_ANN = confusion_matrix(np.argmax(y_test, axis=1), model_predicted)
    np.set_printoptions(precision=2)
    print(model_ANN)
    plt.figure()
    ax = plt.subplot()
    sns.heatmap(model_ANN, annot= True, ax = ax)

    ax.set_xlabel('Predict labels')
    ax.set_title('Confusion Matrix')
```

xvi. ROC

```
[ ] fpr, tpr, thresholds = roc_curve(y_test.ravel(), model_pred.ravel())
    auc_ = auc(fpr, tpr)

    plt.figure(1)
    plt.plot([0, 1], [0, 1], 'k--')
    plt.plot(fpr, tpr, label='ROC (area = {:.3f})'.format(auc_))
    plt.xlabel('False positive rate')
    plt.ylabel('True positive rate')
    plt.title('ROC curve')
    plt.legend(loc='best')
    plt.show()
```

xvii. Classification report

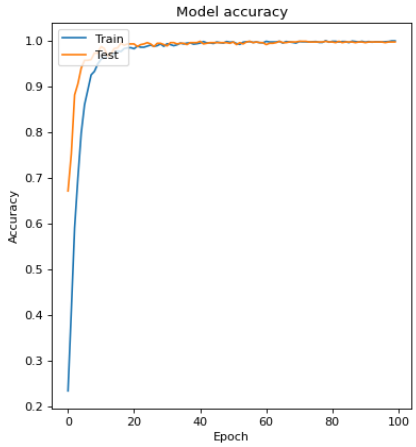
```
[ ] from sklearn.metrics import classification_report
    print (classification_report(y_test,y_pred))
```

V. HASIL DAN PEMBAHASAN

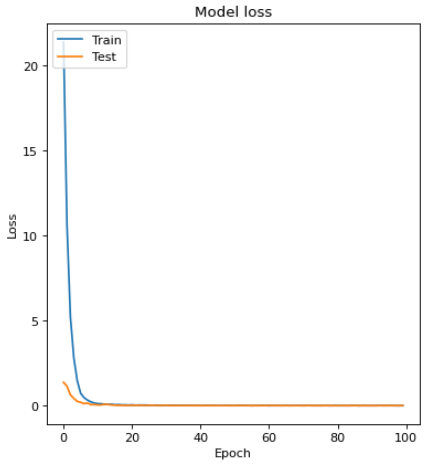
A. Training Dataset dengan Epoch = 100

Setelah dilakukan *training* data, didapatkan akurasi dan plot seperti pada gambar di bawah.

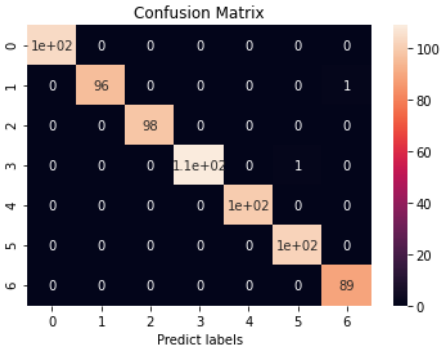
```
22/22 [==============================] - 0s 1ms/step - loss: 0.0134 - accuracy: 0.9971
Test Accuracy: 99.71428513526917
```



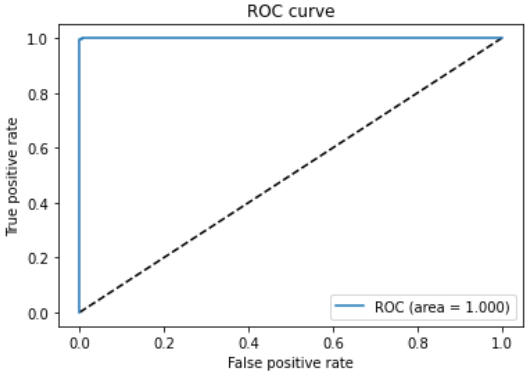
Gambar 4. Plot Akurasi Model untuk Epoch = 100



Gambar 5. Plot Loss Model untuk Epoch = 100



Gambar 6. Confusion Matrix untuk Epoch = 100



Gambar 7. Kurva ROC untuk Epoch = 100

Pada hasil yang di dapatkan, pada epoch ke-83 akurasi validasi sempat mencapai angka 99,86%, akan tetapi akurasinya menurun pada epoch ke-100 yang mana didapatkan hasil sebesar 99.71% pada akurasi validasi. Untuk kurva ROC (*Receiver Operating Characteristics)* dapat dilihat sangat baik (bernilai 100).

	precision	recall	f1-score	support
0	1.00	1.00	1.00	104
1	1.00	0.99	0.99	97
2	1.00	1.00	1.00	98
3	1.00	0.99	1.00	110
4	1.00	1.00	1.00	100
5	0.99	1.00	1.00	102
6	0.99	1.00	0.99	89
accuracy			1.00	700
macro avg	1.00	1.00	1.00	700
weighted avg	1.00	1.00	1.00	700

Gambar 8. Classification Report

VI. KESIMPULAN

Pada penelitian ini menggunakan dataset Toronto Emotion Speech Set (TESS) yang berjumlah 2.800 data audio dengan empat belas kelas, yaitu kelas OAF_angry, OAF_disgust, OAF_Fear, OAF_happy, OAF_neutral, OAF_Pleasant_surprise, OAF_sad, YAF_angry, YAF_disgust, YAF_fear, YAF_happy, YAF_neutral, YAF_pleasant-surprised, YAF_sad. Dalam mengklasifikasikan audio digunakan metode deep Artificial Neural Network. Pada tahapannya digunakan teknik pengolahan data seperti ektraksi fitur menggunakan MFCC dan melakukan label encoding.

Dari hasil penelitian yang dilakukan pada dataset ini, dengan proses training dan testing 75:25, didapatkan evaluasi model dengan akurasi yang diperoleh menggunakan 100 epoch adalah sebesar 99.71%. Waktu komputasi yang dibutuhkan untuk menjalankan 100 epoch adalah 2 detik. Untuk kurva ROC (Receiver Operating Characteristics) dapat dilihat bahwa hasilnya sangat baik dimana bernilai 100.

DAFTAR ACUAN

[1] Ahirwal Mitul Kumar & Mangesh Ramjii Kose. ”Audio-visual stimulation based emotion classification by correlated EEG Channels”. Available: https://link.springer.com/article/10.1007/s12553-019-00394-5.

[2] Ayadi, M. E, Karnel, M. S., & Karray, F. (1011). Survey on speech emotion recognition: Features, classification schemes, and database. Pattern Recognition, 44(3):572-587. DOI: 10.1016/j.patcog.

[3] Chenchah, Farah, and Zied Lachiri. "Acoustic emotion recognition using linear and nonlinear cepstral coefficients." International Journal of Advanced Computer Science and Applications 6.11 (2015): 135-138.

[4] Ke Xianxin, Yujiao Zhu, Lei Wen, and Wenzhen Zhang. “Speech Emotional Recognition Based on SVM and ANN”. International Journal of Machine Learning and Computing. 2018.

[5] Srinivas Parthasarathy and Ivan Tashev, “Convolutional Neural Network Techniques For Speech Emotion Recognition”, Microsoft Research, 2018.

[6] T.M. Rajisha, A.P. Sunija, K.A. Riyas. Performance Analysis of Ma,ayalam Languange Speech Emotion Recognition System Using ANN/SVM. Available: https://www.sciencedirect.com/science/article/pii/S2212017316303334.