

**LAPORAN FINAL PROJECT**  
**PENGANTAR PEMROSESAN DATA MULTIMEDIA**



**Disusun Oleh:**  
**Kelompok 2**

- |                                       |                  |
|---------------------------------------|------------------|
| 1. Made Putra Teguh Pramana           | (2108561078 / C) |
| 2. Luh Arimas Pertiwi                 | (2018561088 / C) |
| 3. Ni Luh Putu Ayu Siwastuti Cayadewi | (2108561103 / C) |

**Pengampu Mata Kuliah:**  
Dr. AAIN Eka Karyawati, S.Si., M.Eng.

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS UDAYANA**  
**2023**

## DAFTAR ISI

DAFTAR ISI .....	i
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Problem Komputasi .....	2
1.3 Tujuan .....	2
1.4 Manfaat .....	2
BAB II .....	3
PEMBAHASAN .....	3
2.1 Manual Aplikasi .....	3
2.2.1 Fitur Sistem dan Antarmuka sistem .....	3
2.2 Source code .....	6
BAB III.....	9
PENUTUP .....	9
3.1 Kesimpulan .....	9
DAFTAR PUSTAKA.....	10

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Audio preprocessing adalah serangkaian teknik dan langkah langkah yang digunakan untuk mempersiapkan data audio sebelum diolah atau dianalisis lebih lanjut. Proses preprocessing ini bertujuan untuk meningkatkan kualitas data audio, mengurangi kebisingan atau gangguan, dan mengekstrak fitur fitur penting dari sinyal audio. preprocessing memegang peran penting dalam pengolahan sinyal Audio audio dengan tujuan mempersiapkan data audio sebelum menjalankannya melalui algoritma pemrosesan lebih lanjut. Keberhasilan aplikasi seperti sistem pengenalan suara, pemrosesan bahasa alami, pemrosesan audio, dan komunikasi suara sangat tergantung pada kualitas audio yang baik dan dapat diandalkan.

Dalam situasi dunia nyata, sinyal audio sering kali terpengaruh oleh beragam gangguan atau noise. Sumber gangguan ini bisa berasal dari lingkungan sekitar, peralatan perekam yang tidak sempurna, atau metode perekaman yang kurang ideal. Kehadiran noise tersebut dapat menyebabkan kabur pada informasi penting dalam sinyal audio, mempengaruhi kemampuan sistem untuk mengenali suara, atau merusak kualitas audio secara keseluruhan. Selain itu, karakteristik audio seperti frekuensi, amplitudo, dan durasi juga bisa berbeda secara signifikan antara sinyal audio yang berbeda. Perbedaan ini dapat menghambat analisis dan pemrosesan data audio. Sebagai contoh, ketika suara yang direkam memiliki tingkat amplitudo yang tidak konsisten, hal ini dapat menyulitkan identifikasi pola atau fitur audio yang penting.

Dengan melakukan audio preprocessing secara efektif, data audio dapat dibersihkan dari gangguan yang tidak diinginkan, tingkat amplitudo dapat disesuaikan agar seragam, dan karakteristik audio lainnya dapat dioptimalkan. Hal ini membantu meningkatkan kualitas audio, mengurangi kesalahan dalam analisis atau pemrosesan data audio, serta menghasilkan hasil yang lebih akurat dan dapat diandalkan. Dalam perkembangan teknologi terkini, audio preprocessing juga menjadi fokus utama dalam penelitian. audio preprocessing selesai, data audio siap untuk diolah lebih lanjut, seperti pengenalan suara, pemrosesan bahasa alami, pengenalan ucapan, pemrosesan audio, atau aplikasi lainnya yang berkaitan dengan analisis audio.

## **1.2 Problem Komputasi**

Computing Problem yang akan dibahas pada laporan ini adalah membuat program untuk mengidentifikasi emosi atau sentimen dari beberapa suara, yaitu :

- a. Menggunakan bahasa pemrograman Python.
- b. Dataset yang digunakan merupakan data audio dari suara orang.
- c. terdapat dua sentimen yang digunakan yaitu positif (happy) dan sentimen negatif (sad).
- d. Menyertakan dokumentasi (penjelasan) program.

## **1.3 Tujuan**

Adapun tujuan – tujuannya yaitu:

- a. Memenuhi tugas dari mata kuliah Pengantar Pemrosesan Data Multimedia.
- b. Membuat program untuk mengidentifikasikan sentimen positif (happy) atau negatif (sad) dari suatu audio.

## **1.4 Manfaat**

Manfaat dari hasil yang kami buat yaitu:

- a. Mengetahui cara mengimplementasikan teknik-teknik audio processing pada Bahasa python.
- b. Menambah pengetahuan dan wawasan tentang cara membuat program untuk mengidentifikasikan sentiment positif (happy) atau negatif (sad) dari suatu audio.

## BAB II

### PEMBAHASAN

#### 2.1 Manual Aplikasi

##### 2.2.1 Fitur Sistem dan Antarmuka sistem

###### a. Tampilan awal

### Audio Analysis App

Upload an audio file



Drag and drop files here

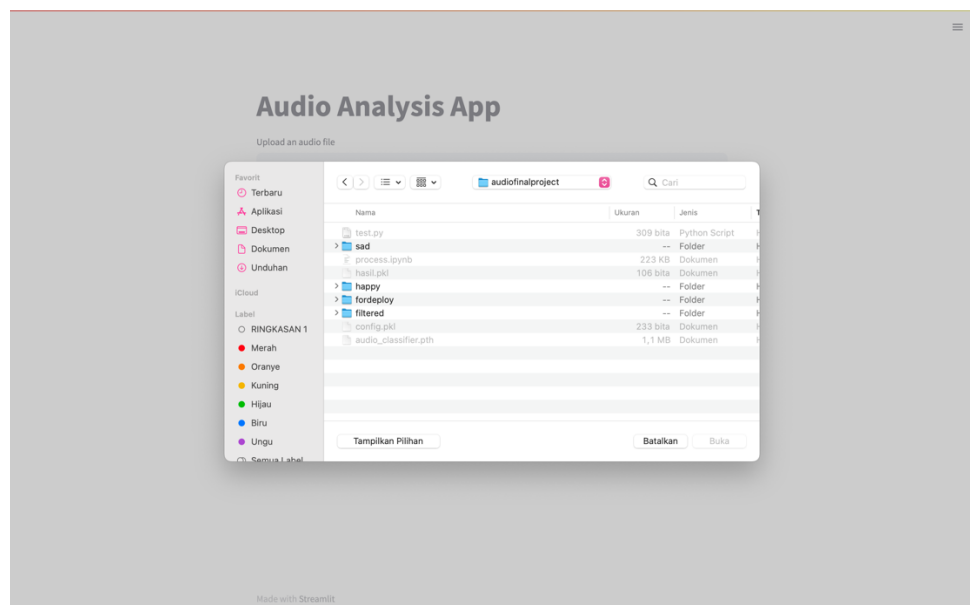
Limit 200MB per file • MP3, WAV

Browse files

Tampilan ini merupakan tampilan awal saat web dibuka. Fitur yang ada dalam aplikasi ini ada fitur menginput data. Data yang dapat diinput dalam sistem adalah data dengan format MP3, WAV

Silahkan pilih file sentimen yang ingin dianalisis dengan menekan tombol browse file lalu pilih file dengan format MP3, WAV

###### b. Tampilan memilih dataset

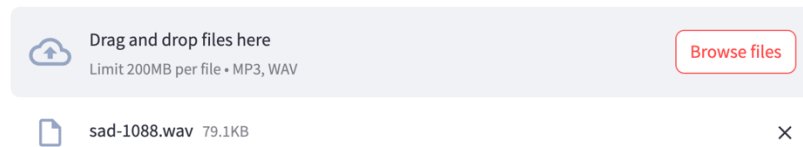


Saat user menekan tombol browse file, maka pilih dataset sad/happy yang anda miliki.

**c. Tampilan dataset yang dipilih**

## Audio Analysis App

Upload an audio file

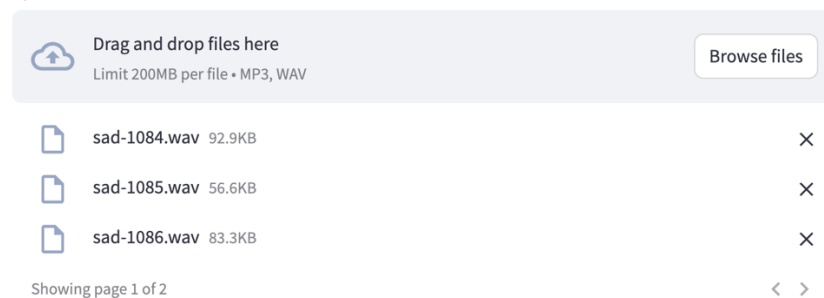


Ini merupakan tampilan dataset yang dipilih. Jika ingin mengganti file yang akan diolah maka silahkan tekan [x] pada sebelah kanan input file.

**d. Tampilan dataset saat di proses/running**

## Audio Analysis App

Upload an audio file



Dataset yang dipilih bisa langsung memilih lebih dari 1 file juga, seperti tampilan diatas. File akan otomatis diproses saat dimasukkan.

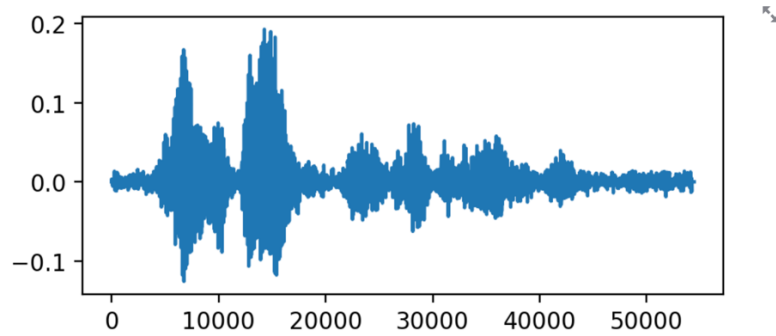
**e. Tampilan suara dataset**

### sad-1088.wav



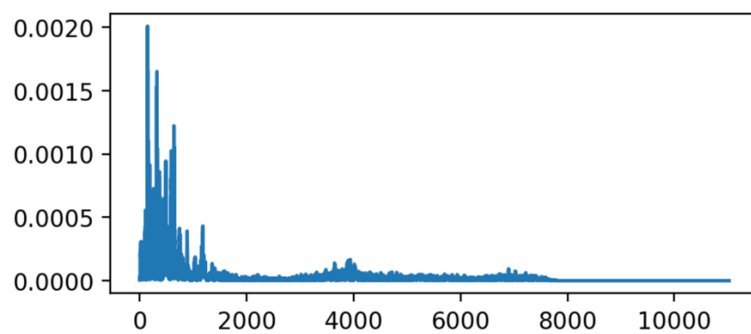
Output pertama yang dihasilkan seperti pada gambar diatas. Dimana user bisa mendengar audio yang diproses tadi.

**f. Time Series**



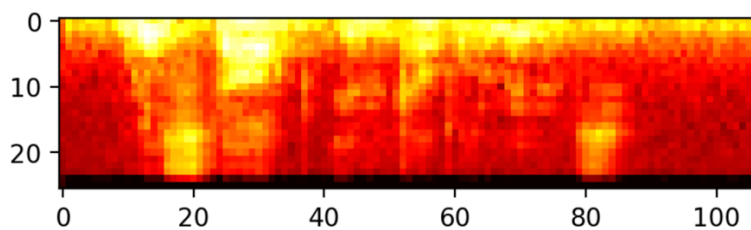
Output kedua yang dihasilkan adalah time series. Dimana x sebagai waktu dan y sebagai amplitudo

#### g. Fourier Transforms



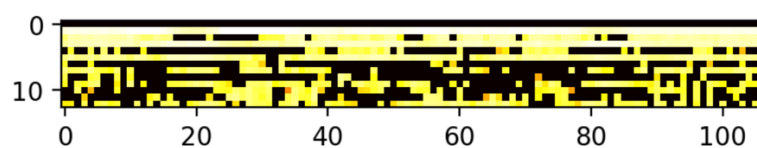
Output ketiga yaitu fourier transforms. Dimana x sebagai waktu, sedangkan y juga sebagai amplitudo

#### h. Filter Bank Coefficient



Output keempat yaitu bank coefficient. Dimana x sebagai waktu, dan y sebagai Amplitudo.

#### i. Mfcc



Output kelima yaitu berupa hasil mfcc. Dimana x sebagai waktu, sedangkan y sebagai decibel.

#### j. Hasil akhir

##### Audio Analysis

the label is : sad

Hasil diakhir akan diberikan output berupa analisis dataset apakah dia benar sad atau sebaliknya (happy).

## 2.2 Source code

Berikut merupakan penjelasan dari source code program :

1. Ini digunakan untuk konfigurasi database dan konektivitas antar file atau folder lainnya.

```
12
13 class Config:
14     def __init__(self, mode='conv', nfilt=26, nfeat=13, nfft=512, rate=16000):
15         self.mode=mode
16         self.nfilt=nfilt
17         self.nfeat=nfeat
18         self.rate=rate
19         self.nfft=nfft
20         self.step=int(rate/10)
21         self.min=float('-inf')
22         self.max=-float('inf')
```

#### 2. Model Audio

- model = AudioClassifiers() : untuk membuat kelas model untuk pengklasifikasian audio.
- model.load\_state\_dict(torch.load("audio\_classifier.pth")) : memuat parameter (state dictionary) yang telah disimpan sebelumnya dari file "audio\_classifier.pth" ke dalam model audio classifier. State dictionary berisi informasi tentang parameter-parameter model seperti bobot dan bias yang telah ditentukan selama proses pelatihan model.
- model.eval() : dalam konteks PyTorch, fungsi eval() digunakan untuk mengubah mode model menjadi mode evaluasi. Saat model berada dalam mode evaluasi, beberapa modul atau lapisan seperti dropout atau batch normalization dapat berperilaku berbeda dibandingkan saat mode pelatihan (training). Dalam kasus ini, fungsi eval() digunakan untuk memastikan bahwa model beroperasi dalam mode evaluasi sebelum melakukan prediksi atau inferensi pada data audio.



```

94
95     model = AudioClassifiers()
96     model.load_state_dict(torch.load("audio_classifier.pth"))
97     model.eval()
98
99     with open("config.pkl", 'rb') as file:

```

### 3. Konfigurasi dari file

- open() untuk membuka file "config.pkl" dalam mode pembacaan biner. 'as' digunakan untuk memberikan alias atau nama variabel "file" pada objek file yang dibuka.
- config = pickle.load(file) : fungsi ini membaca data objek dari file yang dibuka sebelumnya dan mengembalikan objek Python yang sesuai. Objek ini kemudian ditugaskan ke variabel config.

```

98
99     with open("config.pkl", 'rb') as file:
100         config = pickle.load(file)

```

### 4. Fungsi build\_rand\_feat(rate, wav) yang diberikan digunakan untuk membangun fitur acak (random features) dari data audio.

```

101
102     def build_rand_feat(rate, wav):
103         _min=config.min
104         _max=config.max
105         X=[]
106         length=(wav.shape[0]-config.step)//32
107         rand_index=np.random.randint(0, wav.shape[0]-config.step)
108         sample=wav[rand_index:rand_index+config.step]
109         X_sample=librosa.feature.mfcc(y=sample, sr=rate, n_mels=config.nfilt, n_fft=config.nfft, n_mfcc=config.nfeat)
110         X_sample=librosa.power_to_db(X_sample)
111         X.append(X_sample)
112         X=np.array(X)
113         X=(X-_min)/(_max-_min)
114         if config.mode == 'conv':
115             print(X.shape)
116             X=X.reshape(X.shape[0], X.shape[1], X.shape[2], 1)
117             print(X.shape)
118         elif config.mode == 'time':
119             X=X.reshape(X.shape[0], X.shape[1], X.shape[2])
120         return X
121

```

### 5. st.title("Audio Analysis App") : mengindikasikan bahwa kita menggunakan library Streamlit (dikenal sebagai st) untuk membangun aplikasi analisis audio. dapat membuat judul yang akan ditampilkan di aplikasi kita. Judul ini akan muncul sebagai elemen teks dengan format yang lebih besar atau menonjol untuk menarik perhatian pengguna.

```

121
122     # Set Streamlit app title
123     st.title("Audio Analysis App")
124

```

6. Fungsi `st.file_uploader` untuk membuat tombol unggah file audio. Ketika tombol ini diklik, pengguna dapat memilih file audio dalam format "mp3" atau "wav" untuk diunggah.

```
124
125 # Upload audio file
126 audio_file = st.file_uploader("Upload an audio file", type=["mp3", "wav"])
127 model=AudioClassifiers()
128 if audio_file:
129     # Load audio data using Librosa
130     signal, rate = librosa.load(audio_file)
131     fig, ax = plt.subplots(nrows=4, ncols=1, figsize=(5, 10))
132     mask = envelope(signal,rate,0.0005)
133     signal=signal[mask]
134     fft=calc_fft(signal,rate)
135     logfbank = librosa.feature.melspectrogram(y=signal,sr=rate,n_mels=26,n_fft=512)
136     fbank=librosa.power_to_db(logfbank)
137     temp=librosa.feature.mfcc(y=signal,sr=rate,n_mels=128,n_fft=512,n_mfcc=13)
138     mfcc=librosa.power_to_db(temp)
139     ax[0].plot(signal)
140     Y,freq=fft
141     ax[1].plot(freq,Y)
142     ax[2].imshow(fbank,cmap='hot',interpolation='nearest')
143     ax[3].imshow(mfcc,cmap='hot',interpolation='nearest')
144     # Display audio waveform plot
```

7. NumPy adalah library Python yang digunakan untuk melakukan komputasi numerik yang efisien, terutama dalam konteks array multidimensi. NumPy digunakan untuk melakukan operasi matematika seperti penjumlahan atau perkalian, melakukan indeksing dan pemotongan pada array, serta menggunakan fungsi statistik untuk menganalisis data numerik.

```
1 import numpy as np
2
3 # Define your target labels
4 labels = np.array([1, 2, 0, 2]) # Example labels
5
6 # Determine the number of classes in your dataset
7 num_classes = np.max(labels) + 1
8
9 # Perform one-hot encoding
10 one_hot = np.eye(num_classes)[labels]
11
12 # Print the one-hot encoded array
13 print(one_hot)
```

## BAB III

### PENUTUP

#### 3.1 Kesimpulan

Program website ini dibangun menggunakan algoritma jaringan saraf tiruan *Convolutional Neural Network* (CNN), *Mel-Frequency Cepstral Coefficient* (MFCC) untuk ekstraksi fitur. Aplikasi ini menyediakan fitur untuk mengetahui sentimen atau emosi dari positif (atau emosi *happy*) dan sentimen negatif (atau emosi *sad*) file audio dengan berbasis web. Pada aplikasi ini terdapat :

1. Fitur *Preprocessing* menganalisis sentimen dengan memilih secara random file lalu melabeli dengan sentimen *happy* atau *sad*, menampilkan *time series*, *fourier transforms*, *filter bank coefficient*, dan *hasil mfcc*.

Dengan melakukan audio preprocessing secara efektif, data audio dapat dihindarkan dari gangguan yang tidak diinginkan, tingkat amplitudo dapat disesuaikan agar seragam, dan karakteristik audio lainnya dapat dioptimalkan. Hal ini membantu meningkatkan kualitas audio, mengurangi kesalahan dalam analisis atau pemrosesan data audio, serta menghasilkan hasil yang lebih akurat dan dapat diandalkan. Dalam perkembangan teknologi terkini, audio preprocessing juga menjadi fokus utama dalam penelitian.

## DAFTAR PUSTAKA

- [1] Suartika E. P, I Wayan., Yudhi Wijaya, Arya., Soelaiman, Rully. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101. Jurnal Teknik ITS, 5(1), 65 - 69.
- [2] K. Fukushima, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," Biological Cybernetics, 1980.
- [3] Y. LeCun, "Handwritten Digit Recognition with a Back Propagation Network," 1990.
- [4] Susanti, Meri., Susilo, Boko., Andreswari, Desi. (2018). Aplikasi Speech-To-Text Dengan Metode Mel Frequency Cepstral Coefficient (MFCC) dan Hidden Markov Model (HMM) dalam Pencarian Kode ICD-10. Jurnal Rekursif, 6(1), 48 - 58.
- [5] Manunggal, HS. 2005. Perancangan dan Pembuatan Perangkat Lunak Pengenalan Suara Pembicara dengan Menggunakan Analisa MFCC Feature Extraction. Surabaya : Universitas Kristen Petra