

Working With Audio

i Introduction

In this hands-on, we will learn how to work with audio using Python. Make sure the active environment in this jupyter notebook is appropriate.

□ Learn Code

IF 3024

□ Learn Date

Wednesday, 11 September 2024

□ Library Tool Command

```
!pip install matplotlib numpy wave
```

Buatlah sebuah library yang dibutuhkan terlebih dahulu, contohnya matplotlib.pyplot, numpy, os, wave

```
import matplotlib.pyplot as plt
import numpy as np
import os
import wave
```

Memuat File Audio

File audio dapat didownload melalui Youtube yang diconvert dari mp4 ke mp3 melalui extension. playlist file audio Youtube dapat diakses di https://www.youtube.com/playlist?list=PLTlghO_nzap2Jty3v5V6jFdQU_RA8wb5O

```
file_path = os.path.join(os.getcwd(), 'data', 'audio1.wav') #filePath
if not os.path.exists(file_path): #tandaTidakDitemukanFile
    print('File tidak ditemukan')
    exit()
```

Menggunakan `wave` untuk membuka file audio yang di download.

Apa Itu Waveform?

Waveform adalah representasi visual dari gelombang suara atau sinyal audio dalam domain waktu. Dalam grafik waveform, sumbu horizontal (x) biasanya mewakili waktu, sedangkan sumbu vertikal (y) mewakili amplitudo sinyal audio pada waktu tertentu. Dengan kata lain, waveform menunjukkan bagaimana amplitudo suara berubah seiring berjalannya waktu.

Waveform memberikan informasi penting tentang karakteristik audio, seperti:

- **Amplitudo:** Ketinggian atau intensitas gelombang suara. Amplitudo tinggi biasanya terkait dengan suara yang lebih keras.
- **Frekuensi:** Jumlah siklus gelombang per detik. Meskipun waveform sendiri tidak langsung menunjukkan frekuensi, pola-pola dalam waveform bisa memberikan indikasi tentang frekuensi sinyal.
- **Durasi:** Panjang waktu sinyal audio.

Tentang Package `wave`

`import wave` adalah modul dalam Python yang digunakan untuk membaca dan menulis file audio dalam format WAV. Format WAV adalah format audio yang tidak terkompresi, yang berarti data audio disimpan dalam bentuk aslinya tanpa pengurangan kualitas.

```
with wave.open(file_path, 'r') as wav_file:                                # membuka
    file wav
    # membaca semua frame
    frames = wav_file.readframes(-1)
    # mengubah frame ke array numpy
    sound_data = np.frombuffer(frames, dtype=np.int16)
    frame_rate = wav_file.getframerate()                                    #
mendapatkan sample rate

print(f"Isi dari sound_data: {sound_data}")
print(f"Panjang dari sound_data: {len(sound_data)}")
print(f"Isi data pada elemen ke 500000 - 500020:
{sound_data[500000:500020]}")
print(f"Sample rate: {frame_rate}")

Isi dari sound_data: [  0  0  0 ... 1806 1883 1518]
Panjang dari sound_data: 1924842
Isi data pada elemen ke 500000 - 500020: [4082 2179 4290 3743 3451
2573 4427 2378 4995 3717 3710 3866 4481 3951
6125 4215 3898 2548 973 633]
Sample rate: 48000

time_axis = np.linspace(0, len(sound_data) / frame_rate,
num=len(sound_data))

# menampilkan plot dengan matplotlib
plt.figure(figsize=(18, 4))

# Buat judul "Visualisasi Audio"
plt.title('Visualisasi Audio', fontsize=16)
plt.plot(time_axis, sound_data)
plt.xlabel('Waktu (detik)')
plt.ylabel('Amplitudo')
# Menampilkan grid yang halus
plt.grid(linestyle='-', linewidth=0.5)
plt.show()
```

