## LAPORAN TUGAS PENGOLAHAN SINYAL DIGITAL “FILTER DAN ANALISA SUARA KUCING DAN ANJING”

Sebuah gambar berisi papan klip, Grafis, simbol, logo

Deskripsi dibuat secara otomatis

**Oleh:**

**Muhammad Fathi Farhat**

**5024211065**

**GitHub:**

[**https://github.com/Blindheroes/FILTER\_DAN\_ANALISA\_SUARA\_KUCING\_DAN\_ANJING.git**](https://github.com/Blindheroes/FILTER_DAN_ANALISA_SUARA_KUCING_DAN_ANJING.git)

**Sumber dataset:**

[**https://www.kaggle.com/datasets/mmoreaux/audio-cats-and-dogs**](https://www.kaggle.com/datasets/mmoreaux/audio-cats-and-dogs)

## Langkah 1: Prepare Library dan File Audio

*import* numpy *as* np

*import* matplotlib.pyplot *as* plt

*from* scipy.io *import* wavfile

*from* scipy.fft *import* fft, ifft

*from* IPython.display *import* Audio

sample\_rate, data = wavfile.read("cat\_2.wav")

data = data.astype(float)

## Langkah 2: Menghitung Transformasi Fourier Cepat (FFT)

signal\_length = len(data)

fft\_data = fft(data)

## Langkah 3: Filtering

frequencies = np.linspace(0.0, sample\_rate, signal\_length)

dog\_frequency = 500

cat\_frequency = 2000

dog\_index = np.abs(frequencies - dog\_frequency).argmin()

cat\_index = np.abs(frequencies - cat\_frequency).argmin()

filtered\_data = np.copy(fft\_data)

filtered\_data[:dog\_index] = 0

filtered\_data[cat\_index:] = 0

## Langkah 4: Transformasi Fourier Cepat Balik (Inverse FFT)

filtered\_signal = ifft(filtered\_data)

filtered\_signal = filtered\_signal.astype(int)

## Langkah 5: Visualisasi

plt.figure(figsize=(10, 4))

plt.title("Sinyal Sebelum Filter")

plt.xlabel("Waktu (detik)")

plt.ylabel("Amplitudo")

plt.plot(np.arange(signal\_length) / sample\_rate, data)

plt.show()

Sebuah gambar berisi cuplikan layar, diagram, Plot, garis

Deskripsi dibuat secara otomatis

plt.figure(figsize=(10, 4))

plt.title("Sinyal Setelah Filter")

plt.xlabel("Waktu (detik)")

plt.ylabel("Amplitudo")

plt.plot(np.arange(signal\_length) / sample\_rate, filtered\_signal)

plt.show()

Sebuah gambar berisi cuplikan layar, Plot, garis, diagram

Deskripsi dibuat secara otomatis

## Langkah 6: Output Pemutar Audio

*# Memainkan file audio asli*

wavfile.write("filtered\_output.wav", sample\_rate, filtered\_signal)

Audio(data, rate=sample\_rate)

*# Memainkan file audio yang telah difilter*

filtered\_audio = wavfile.read("filtered\_output.wav")

Audio(filtered\_audio[1], rate=sample\_rate)

## Langkah 7: Analisis suara (anjing atau kucing)

dog\_amplitude = np.max(np.abs(filtered\_signal[:dog\_index]))

cat\_amplitude = np.max(np.abs(filtered\_signal[cat\_index:]))

*if* dog\_amplitude > cat\_amplitude:

    print("Suara hasil adalah suara anjing.")

*else*:

    print("Suara hasil adalah suara kucing.")