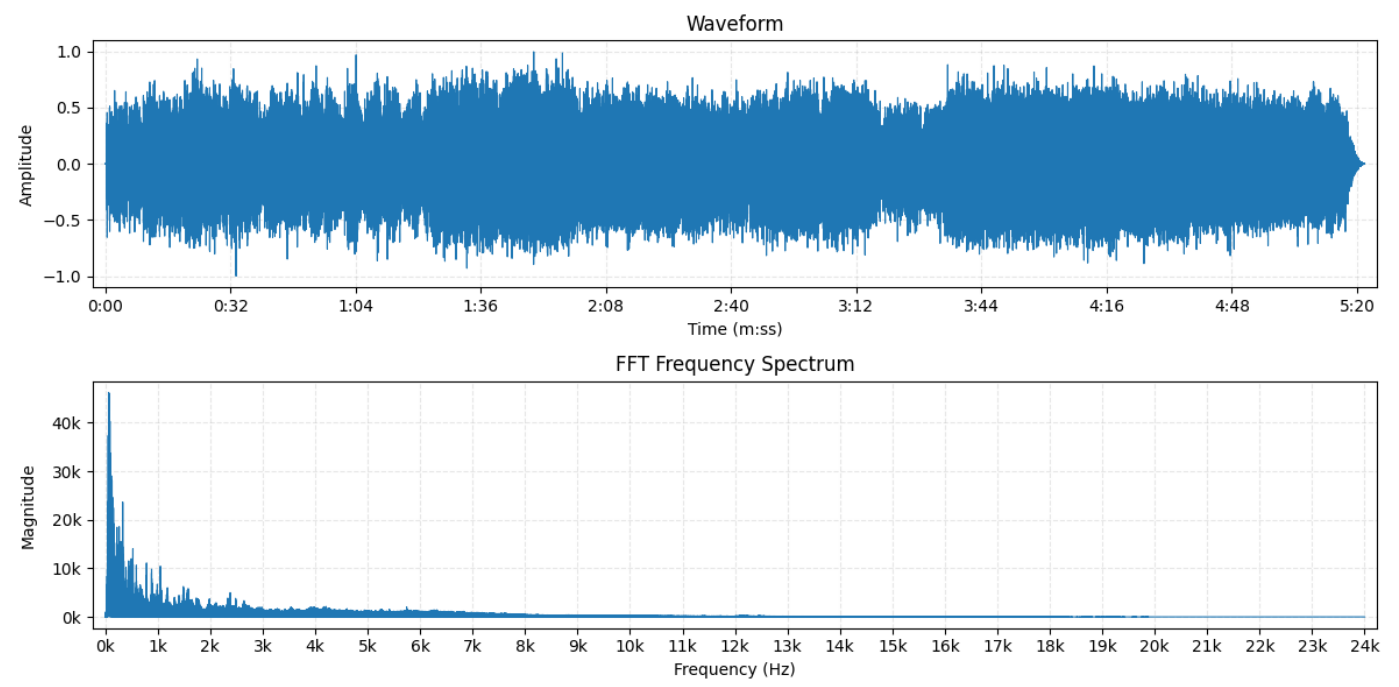


Grafik Fast Fourier Transform

+ Waveform



Diambil dari lagu “Dewa - Dua Sedjoli” di YouTube  
[https://youtu.be/U8girfB5Syc?si=zolT\\_zL8nXNqT1GF](https://youtu.be/U8girfB5Syc?si=zolT_zL8nXNqT1GF)

Source code bisa dilihat dan dicek di GitHub  
<https://github.com/nurfaiz-nuh/Kuliah-Isyarat>



Source Code 1<sup>st</sup> VER:

```
import numpy as np
import matplotlib.pyplot as plt
import librosa

# 1. Load any audio file (mp3, wav, flac, etc.)
filename = "Dewa - Dua Sedjoli.mp3" # <<<< input file here
data, fs = librosa.load(filename, sr=None, mono=True)
# 'sr=None' = keep original sampling rate
# 'mono=True' = convert to mono

# 2. Normalize
data /= np.max(np.abs(data))

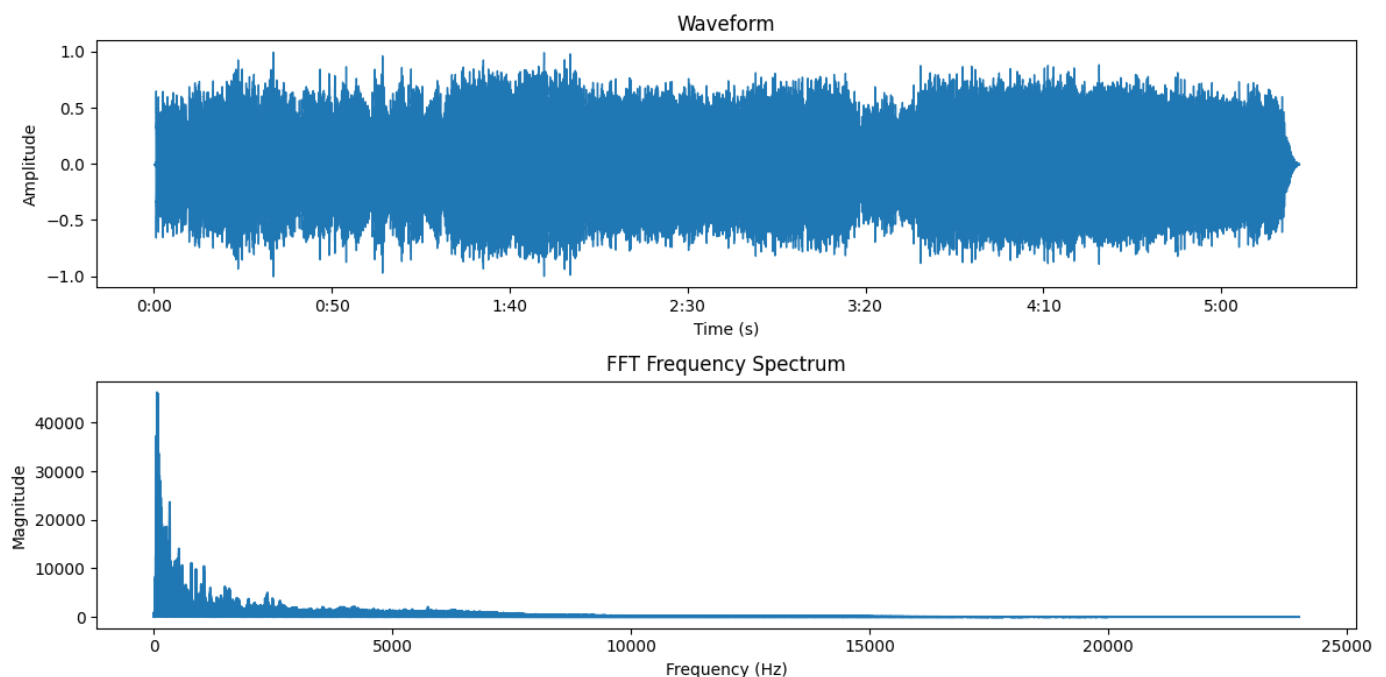
# 3. Fast fourier transform
N = len(data)
fft_data = np.fft.fft(data)
freqs = np.fft.fftfreq(N, 1/fs)

# Set the figure size
plt.figure(figsize=(12,6))

# 4. Plot waveform
plt.subplot(2,1,1)
librosa.display.waveshow(data, sr=fs)
plt.title("Waveform")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")

# 5. Plot magnitude spectrum (only positive frequencies)
plt.subplot(2,1,2)
plt.plot(freqs[:N//2], np.abs(fft_data[:N//2]))
plt.title("FFT Frequency Spectrum")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Magnitude")

# Set the window title after the figure is created
plt.gcf().canvas.manager.set_window_title("FFT 1st ver.py")
plt.tight_layout()
plt.show()
```



Source Code 2<sup>nd</sup> VER:

```
import numpy as np
import matplotlib.pyplot as plt
import soundfile as sf

# === 1. Load audio ===
filename = "Dewa - Dua Sedjoli.mp3" # <<<< input file here
data, fs = sf.read(filename, dtype='float32')

# If stereo, convert to mono
if data.ndim > 1:
    data = data.mean(axis=1)

# === Normalize ===
data /= np.max(np.abs(data))

# === 2. Time axis ===
N = len(data)
time = np.linspace(0, N/fs, N)

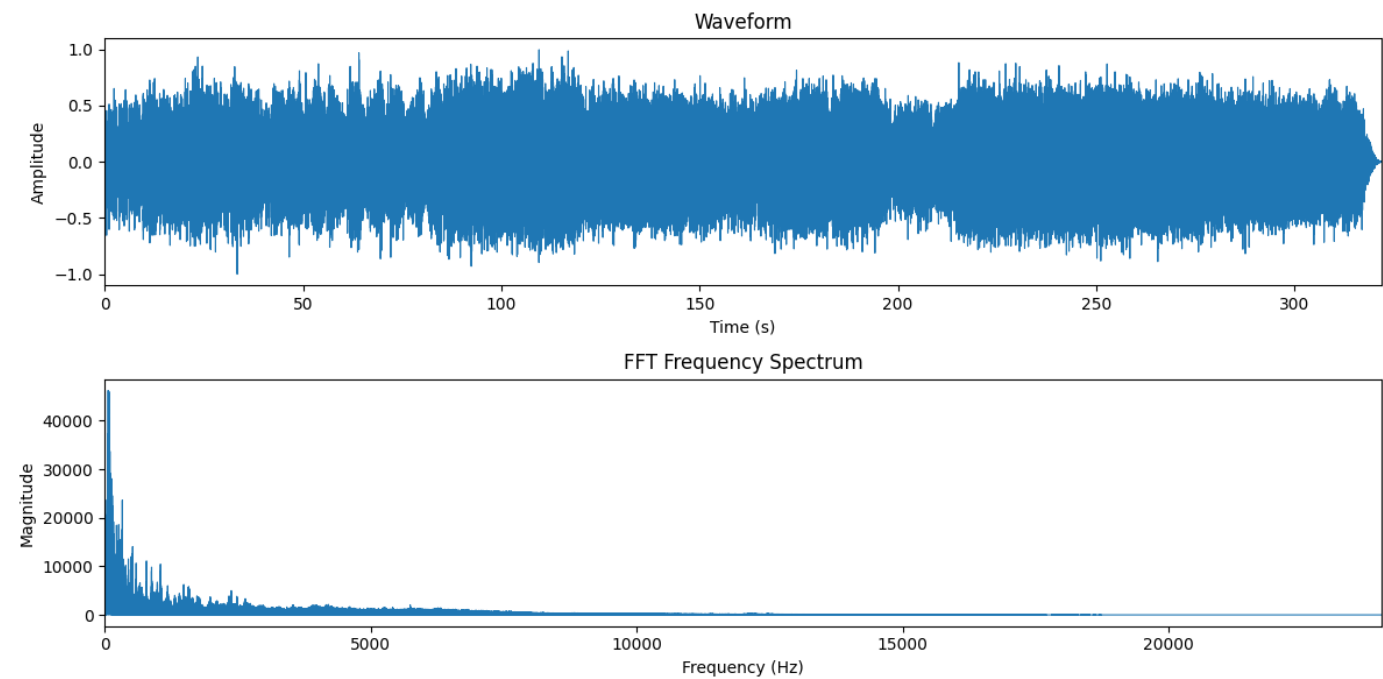
# === 3. FFT ===
fft_data = np.fft.fft(data)
freqs = np.fft.fftfreq(N, 1/fs)

# === Set figure size ===
plt.figure(figsize=(12, 6))

# === 4. Plot waveform ===
plt.subplot(2,1,1)
plt.plot(time, data, linewidth=0.8)
plt.title("Waveform")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.xlim(0, time[-1]) # start at 0, stop at end

# === 5. Plot FFT spectrum ===
plt.subplot(2,1,2)
plt.plot(freqs[:N//2], np.abs(fft_data[:N//2]), linewidth=0.8)
plt.title("FFT Frequency Spectrum")
plt.xlabel("Frequency (Hz)")
plt.ylabel("Magnitude")
plt.xlim(0, freqs[N//2 - 1]) # start at 0, stop at Nyquist

# === Set window title ===
plt.gcf().canvas.manager.set_window_title("FFT 2nd ver.py")
plt.tight_layout()
plt.show()
```



Source Code 3<sup>rd</sup> VER:

```

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import soundfile as sf

# Function: format time axis as 70s -> 1:10
def format_time(x, pos):
    minutes = int(x // 60)
    seconds = int(x % 60)
    return f"{minutes}:{seconds:02d}"

# Function: format frequency axis as 2000 -> 2k
def format_freq(x, pos):
    return f"{x/1000:.0f}k" if x >= 1000 or x == 0 else f"{int(x)}"

# === 1. Load audio ===
filename = "Dewa - Dua Sedjoli.mp3" # <<<< input file here
data, fs = sf.read(filename, dtype='float32')

# If stereo, convert to mono
if data.ndim > 1:
    data = data.mean(axis=1)

# === Normalize ===
data /= np.max(np.abs(data))

# === 2. Time & duration ===
N = len(data)
duration = N / fs
time = np.linspace(0, duration, N)

# === 3. FFT ===
fft_data = np.fft.fft(data)
freqs = np.fft.fftfreq(N, 1/fs)

# === Set figure size ===
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(12, 6))

# === 4. Plot waveform ===
ax1.plot(time, data, linewidth=0.8)
ax1.set_title("Waveform")
ax1.set_xlabel("Time (m:ss)")
ax1.set_ylabel("Amplitude")

# -- Margins for waveform --
margin = duration * 0.013
ax1.set_xlim(-margin, duration + margin)

# -- Ticks for waveform --
tick_interval = 1 if duration <= 10 else int(duration // 10)
ax1.xaxis.set_major_locator(ticker.MultipleLocator(tick_interval))
ax1.xaxis.set_major_formatter(ticker.FuncFormatter(format_time))
ax1.grid(True, linestyle='--', alpha=0.3)

# === 5. Plot frequency ===
ax2.plot(freqs[:N/2], np.abs(fft_data[:N/2]), linewidth=0.8)
ax2.set_title("FFT Frequency Spectrum")
ax2.set_xlabel("Frequency (Hz)")
ax2.set_ylabel("Magnitude")

# -- Margins for frequency --
freq_end = fs / 2
freq_margin = freq_end * 0.01
ax2.set_xlim(-freq_margin, freq_end + freq_margin)

# -- Ticks for frequency --
freq_tick = 1000 if fs <= 48000 else fs // 40
ax2.xaxis.set_major_locator(ticker.MultipleLocator(freq_tick))
ax2.xaxis.set_major_formatter(ticker.FuncFormatter(format_k))
ax2.grid(True, linestyle='--', alpha=0.3)

# Additional magnitude format
ax2.yaxis.set_major_formatter(ticker.FuncFormatter(format_k))

# === Set window title ===
plt.gcf().canvas.manager.set_window_title("FFT 3rd ver.py")
plt.tight_layout()
plt.show()

```