LAMPIRAN

- Koding Python untuk menampilkan gelombang sinusoidal:

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

# Membuat waktu sampling
dt = 1/8 #Frekuensi Sampling
time = np.arange(0, 1, dt)

# Membuat gelombang sinus dengan frekuensi 1 hertz
frequency = 1
waveform = np.sin(2 * np.pi * frequency * time)

# Menampilkan gelombang sinus
plt.stem(time, waveform)
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.title('Gelombang Sinyal Sinusoidal')
plt.show()
```

- Koding untuk Mengubah File Audio ke Domain Frekuensi Menggunakan FFT:

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

#Memproses file audio original
base_dir = "D:\Audio-Signal-Processing-master\Audio-Signal-Processing-
master\Data Set\Maju"
Sound = "Maju (4).wav"
samples, sampling_rate = librosa.load(os.path.join(base_dir, Sound), sr=None,
mono=True, offset=0.0, duration=None)
len(samples), sampling_rate)
print("Panjang Sampel:", len(samples))
print("Sampling Rate:", sampling_rate)

#Menghitung FFT pada sinyal audio original
FourierTransformation = fft(samples)
```

```
Frequency = np.linspace(0, sampling_rate, len(magnitude_spectrum))
num_frequency_bins = int(len(Frequency) * 0.5)
mag_spec = np.abs(FourierTransformation)
ref_value = np.max(mag_spec)
meg_spec_db = 20 * np.log10(mag_spec/ref_value)

Nyquist_limit = int(len(Frequency1)*0.5)

fig,axs = plt.subplots(3,1)

plt.sca(axs[0])
librosa.display.waveshow(y = samples, sr = sampling_rate)
plt.xlabel("Time (second)-->")
plt.ylabel("Amplitude")
plt.title("Gelombang Sinyal Suara Original")

plt.sca(axs[1])
plt.plot(frequency[:Nyquist_limit], magnitude_spectrum[:Nyquist_limit])
plt.xlabel("Frequency (Hz)-->")
plt.ylabel("Amplitude")
plt.title("Hasil FFT Sinyal Suara Original")

plt.sca(axs[2])
plt.plot(frequency[:Nyquist_limit], mag_spec_db[:Nyquist_limit])
plt.xlabel("Frequency (Hz)-->")
plt.ylabel("Magnitude(dB)")

plt.show()
```

- Koding Python untuk Memfilter File Audio

```
import scipy
import librosa
import os
from scipy import signal
import matplotlib.pyplot as plt

#Memproses file audio original
base_dir = "D:\Audio-Signal-Processing-master\Audio-Signal-Processing-
master"
```

```
Sound1 = "Maju (4).wav"
samplesl, sampling ratel = librosa.load(os.path.join(base_dir, Sound1), sr=None, mono=True, offset=0.0, duration=none)
#menfilter file audio dengan filter butterworth
b, a = signal.butter (5, 1000/(sampling_rate/2), btype='highpass) #(orde filter, fc/fs, tipe filter)
filteredsignal = signal.ifilter (b, a, samples)
```

- Koding Python untuk menampilkan gelombang yang sudah di filter dan belum di filter

```
import librosa
import os
import matplotlib.pyplot as plt

#Memproses file audio original
base_dir = "D:\Audio-Signal-Processing-master\Audio-Signal-Processing-master" Sound1 = "Maju (4).wav"
samplesl, sampling ratel = librosa.load(os.path.join(base_dir, Sound1), srtione, mono-True, offset=0.0, duration None)

#Memproses file audio yang sudah di filter
base_dir = "D:\Audio-Signal-Processing-master\Audio-Signal-Processing-master"
Sound2 = "Maju (4)-Filter.way
samples2, sampling rate2 = librosa.load(os.path.join(base_dir, Sound2), sr-one, mono-True, offset-0.0, duration=None)

fig.axs plt.subplots(2,1)

#Menampilkan gelombang sinyal original
plt.sca (axs[0])
plt.plot(samples)
plt.title('Gelombang Sinyal Suara Original")
plt.xlabel("Frequency (Hz)-->') plt.ylabel("Amplitude")
#Menampilkan gelombang sinyal yang sudah di filter
plt.sca (axs[1])
plt.plot(samples2)
plt.title('Gelombang Sinyal Suara Yang Sudah di Filter')
plt.xlabel("Frequency (Hz)-->')
plt.ylabel("Amplitude")
```

```
plt.show()
```

- Koding Python untuk menampilkan gelombang dalam domain frekuensi

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

#Memproses file audio original
base_dir = "D:\Audio-Signal-Processing-master\Audio-Signal-Processing-
master\Data Set\Maju"
Sound = "Maju (4).wav"
samples, sampling_rate = librosa.load(os.path.join(base_dir, Sound), sr=None,
mono=True, offset=0.0, duration=None)
len(samples), sampling_rate)
print("Panjang Sampel:", len(samples))
print("Sampling Rate:", sampling_rate)

#Menghitung FFT pada sinyal audio original
FourierTransformation = fft(samples)
Frequency = np.linspace(0, sampling_rate, len(magnitude_spectrum))
num_frequency_bins = int(len(Frequency) * 0.5)
mag_spec = np.abs(FourierTransformation)
ref_value = np.max(mag_spec)
meg_spec_db = 20 * np.log10(mag_spec/ref_value)

Nyquist_limit = int(len(Frequency1)*0.5)

fig,axs = plt.subplots(3,1)

#Menampilkan FFT Sinyal Original
plt.sca(axs[0])
librosa.display.waveshow(y = samples, sr = sampling_rate)
plt.xlabel("Time (second)-->")
plt.ylabel("Amplitude")
plt.title("Gelombang Sinyal Suara Original")

plt.sca(axs[1])
plt.plot(frequency[:Nyquist_limit], magnitude_spectrum[:Nyquist_limit])
plt.xlabel("Frequency (Hz)-->")
```

```
plt.ylabel("Amplitude")
plt.title("Hasil FFT Sinyal Suara Original")

#Menampilkan FFT Sinyal yang sudah di filter
plt.sca(axs[2])
plt.plot(frequency[:Nyquist_limit], mag_spec_db[:Nyquist_limit])
plt.xlabel("Frequency (Hz)-->")
plt.ylabel("Magnitude(dB)")

plt.show()
```