

Resume Hasil Percobaan Audio

Kelompok 4 – Kelas C

Nama Anggota:

1. Hartarto
2. Muhammad Akbar F.
3. Damar Muaziz L.
4. Ahmad Ridwan F.

1. Audio Tanpa Efek

Gelombang terlihat alami sesuai rekaman, puncak (amplitudo) masih acak sesuai dinamika suara asli. Memperlihatkan distribusi frekuensi lengkap (rendah, menengah, tinggi) tanpa ada bagian yang terpotong. Suara terdengar natural, tapi bisa ada noise di frekuensi tertentu.

2. Audio dengan Efek EQ

Amplitudo pada bagian tertentu bisa naik/turun, tergantung frekuensi mana yang dinaikkan/dipotong. Ada bagian frekuensi yang lebih tebal (ditekan naik) atau lebih tipis (dipotong). Misalnya, bass lebih menonjol atau treble lebih cerah. Grafik menunjukkan distribusi frekuensi berubah sesuai setelan EQ, hasilnya bisa lebih seimbang atau lebih condong ke bass/treble.

3. Audio dengan Efek High Pass Filter

Bagian gelombang rendah (bass besar) terlihat mengecil/berkurang. Frekuensi rendah (misalnya <100 Hz atau <200 Hz, tergantung setting) menjadi hilang/terpotong. Frekuensi tinggi tetap ada. Suara terdengar lebih “tipis” atau “cempreng” karena bass dibuang. Cocok untuk mengurangi suara dengung atau noise frekuensi rendah.

4. Audio dengan Efek Low Pass Filter

Bagian gelombang tinggi menjadi lebih halus/berkurang detailnya. Frekuensi tinggi (misalnya >4 kHz atau >8 kHz, tergantung setting) menghilang, hanya frekuensi rendah dan menengah yang masih ada. Suara terdengar lebih “boomy” atau “bassy”, mirip seperti suara yang teredam.

Kesimpulan Perbandingan Grafik

- Tanpa efek: grafik penuh, semua frekuensi ada.
- EQ: grafik berubah sesuai boost/cut, ada penekanan/penipisan frekuensi tertentu.
- High Pass: grafik memotong frekuensi rendah → suara lebih tipis.
- Low Pass: grafik memotong frekuensi tinggi → suara lebih berat/boomy.

Program Gelombang Kotak

```
import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import square


amp = 1

freq = 5

dur = 1

fs = 1000


t = np.arange(0, dur, 1/fs)

sq = amp * square(2 * np.pi * freq * t)


plt.figure(figsize=(8,4))

plt.plot(t, sq, color='blue')

plt.title('Square Wave')

plt.xlabel('Time (s)')

plt.ylabel('Amplitude')

plt.grid(True)

plt.show()
```

Program Gelombang Sawtooth

```
import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import sawtooth


amp = 1

freq = 5

dur = 1

fs = 1000


t = np.arange(0, dur, 1/fs)

sw = amp * sawtooth(2 * np.pi * freq * t)


plt.figure(figsize=(8,4))

plt.plot(t, sw, color='red')

plt.title('Sawtooth Wave')

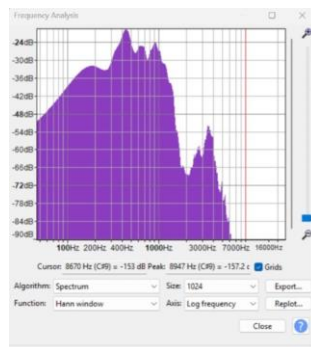
plt.xlabel('Time (s)')

plt.ylabel('Amplitude')

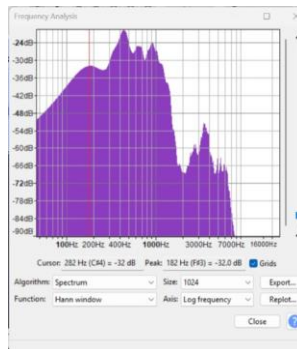
plt.grid(True)

plt.show()
```

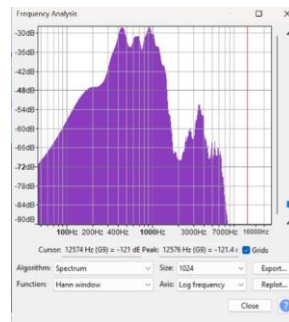
LAMPIRAN



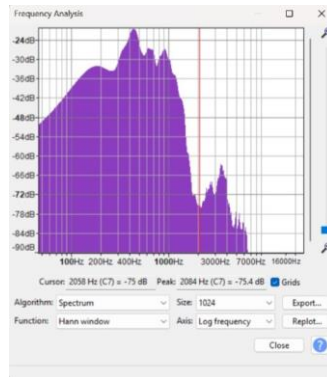
Gambar Grafik Audio 1 – EQ



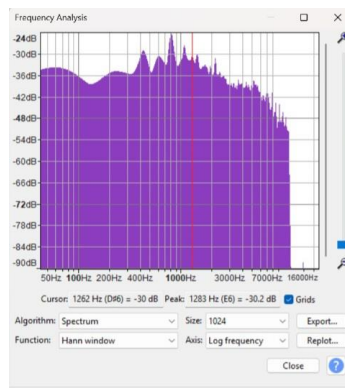
Gambar Grafik Audio 1 – Plot Spectrum



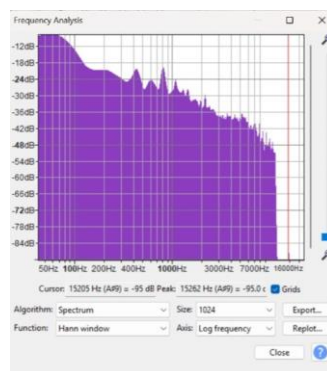
Gambar Grafik Audio 1 – High Pass



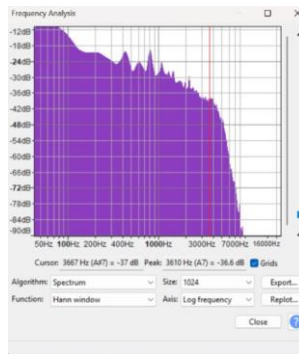
Gambar Grafik Audio 1 – Low Pass



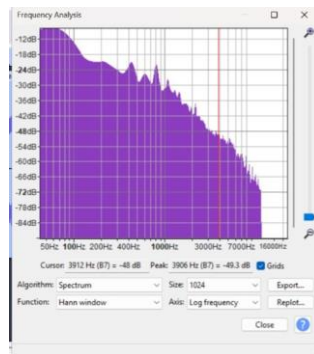
Gambar Grafik Audio 2 – EQ



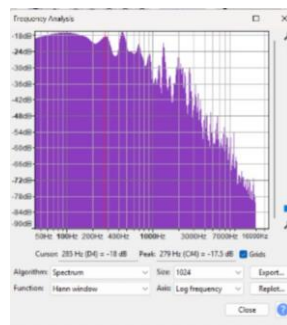
Gambar Grafik Audio 2 – Plot Spectrum



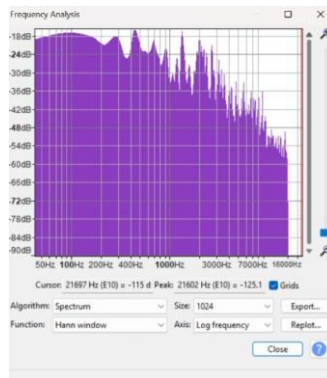
Gambar Grafik Audio 2 – High Pass



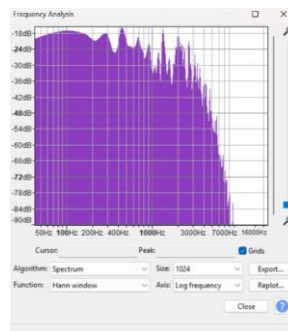
Gambar Grafik Audio 2 – Low Pass



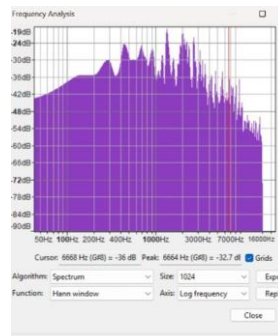
Gambar Grafik Audio 3 – EQ



Gambar Grafik Audio 3 – Plot Spectrum



Gambar Grafik Audio 3 – High Pass



Gambar Grafik Audio 3 – Low Pass