

Tugas Analisis Multimedia: Audio, Gambar, Video

Mata Kuliah: Sistem & Teknologi Multimedia

Nama: Muhammad Daffa Rafif Wibowo

NIM: 122140036

Deskripsi Tugas

Tugas ini bertujuan untuk memahami representasi dasar data multimedia (audio, gambar, dan video) melalui praktik langsung memuat data, visualisasi, dan ekstraksi informasi fundamental. Anda akan bekerja dengan tiga jenis media berbeda untuk menganalisis karakteristik temporal (audio), spasial (gambar), dan spatio-temporal (video).

Fokus tugas adalah pada pemahaman konsep dasar representasi multimedia dan kemampuan interpretasi hasil visualisasi, **bukan** pada manipulasi atau transformasi lanjutan data multimedia.



CATATAN PENTING: PRESENTASI ACAK & KEJUJURAN AKADEMIK

Sebagian mahasiswa akan dipilih secara ACAK untuk presentasi singkat (5-10 menit) menjelaskan kode dan interpretasi hasil mereka. Jika Anda:

- Tidak mampu menjelaskan kode yang Anda kumpulkan
- Hanya menyalin-tempel tanpa pemahaman
- Bergantung sepenuhnya pada AI tanpa memahami konsep

Maka nilai tugas Anda akan diberikan 0 (nol).

Gunakan referensi dan AI sebagai alat bantu pembelajaran, tetapi pastikan Anda memahami setiap baris kode dan dapat menjelaskan logika di baliknya.

```
In [ ]: # Import Library (Satu-satunya sel kode dalam template ini)
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import librosa
import soundfile as sf
from PIL import Image
import cv2
from IPython.display import Audio, HTML, display
import os

# Set matplotlib untuk menampilkan plot inline
%matplotlib inline

# Tampilkan versi library untuk dokumentasi
print("Library versions:")
print(f"NumPy: {np.__version__}")
print(f"Matplotlib: {matplotlib.__version__}") # Perbaiki bagian ini
print(f"Librosa: {librosa.__version__}")
print(f"OpenCV: {cv2.__version__}")

# Tambahkan import lain jika diperlukan saat mengerjakan tugas
```

```
Library versions:
NumPy: 2.2.6
Matplotlib: 3.10.6
Librosa: 0.11.0
OpenCV: 4.12.0
```

Petunjuk Umum Pengerjaan



Cara Menggunakan Template

- Gunakan notebook ini sebagai kerangka kerja utama
- Tulis penjelasan (markdown) **SEBELUM** menaruh kode agar maksud dan tujuan jelas
- Tambahkan sel kode di tempat yang sudah disediakan (tanda dengan TODO)
- Semua plot/gambar harus diberi judul, label sumbu, dan keterangan singkat

Standar Visualisasi

- Setiap plot harus memiliki judul yang deskriptif
- Label sumbu X dan Y harus jelas
- Gunakan colorbar untuk plot yang memerlukan skala warna
- Berikan interpretasi singkat setelah setiap visualisasi

Struktur Data yang Direkomendasikan

- Buat folder `data/` di direktori yang sama dengan notebook
- Gunakan nama file yang deskriptif (contoh: `audio_musik_piano.wav` , `gambar_pemandangan_gunung.jpg`)
- Dokumentasikan sumber data jika menggunakan dataset publik

Larangan




- **Jangan** menaruh seluruh pekerjaan dalam satu sel kode yang sangat panjang
- **Jangan** menempel hasil output tanpa interpretasi atau analisis
- **Jangan** bergantung sepenuhnya pada AI - pahami dan kuasai kode Anda

Persiapan Presentasi Acak

- Pastikan Anda memahami setiap baris kode yang ditulis
- Latih menjelaskan logika dan alur pemikiran Anda
- Siapkan penjelasan untuk setiap visualisasi dan interpretasinya

Checklist Kelengkapan (Centang saat selesai)

Bagian Audio

- [] Muat audio dan tampilkan metadata (durasi, sample rate, jumlah kanal)
- [] Tampilkan waveform dengan label sumbu yang jelas
- [] Tampilkan spectrogram dalam skala log-dB dengan colorbar

- [✓] Tampilkan MFCC (minimal 13 koefisien) sebagai heatmap
- [✓] Berikan interpretasi dan analisis untuk setiap visualisasi audio

Bagian Gambar

- [✓] Tampilkan gambar dengan benar dalam format RGB
- [✓] Tampilkan informasi dasar (dimensi, jumlah kanal, dtype)
- [✓] Tampilkan histogram warna untuk channel R, G, B
- [✓] Berikan analisis hubungan histogram dengan kesan visual gambar

Bagian Video

- [✓] Tampilkan metadata video (resolusi, fps, frame count, durasi)
- [✓] Tampilkan 3 frame representatif (awal, tengah, akhir)
- [✓] Konversi BGR ke RGB dengan benar untuk visualisasi
- [✓] Analisis kesesuaian parameter video dengan use case

Analisis & Dokumentasi

- [✓] Setiap bagian memiliki interpretasi dan analisis ringkas
- [✓] Perbandingan representasi ketiga jenis media
- [✓] Kesimpulan pembelajaran dan refleksi
- [✓] Semua sumber data dan referensi dicantumkan

Pendahuluan

Apa itu Data Multimedia?

Data multimedia adalah informasi yang dikodekan dalam berbagai format untuk merepresentasikan dunia nyata:

- **Audio (1D):** Sinyal satu dimensi yang berubah terhadap waktu

- Contoh: musik, suara, speech
- Representasi: amplitudo vs waktu
- **Gambar (2D):** Matriks nilai intensitas dalam ruang dua dimensi
 - Contoh: foto, ilustrasi, grafik
 - Representasi: intensitas pixel pada koordinat (x,y)
- **Video (2D + Waktu):** Rangkaian frame (gambar) yang ditampilkan berurutan
 - Contoh: film, rekaman, animasi
 - Representasi: frame berubah terhadap waktu dengan frame rate tertentu

Tujuan Tugas

Memahami representasi dasar dan teknik visualisasi fundamental untuk setiap jenis media multimedia, termasuk:

- Cara memuat dan membaca file multimedia
- Ekstraksi informasi metadata yang penting
- Visualisasi yang informatif dan mudah dipahami
- Interpretasi hasil analisis secara kontekstual

Cara Kerja

1. Isi setiap bagian sesuai instruksi yang diberikan
2. Tambahkan sel kode di tempat yang ditandai dengan "TODO"
3. Berikan interpretasi dan analisis setelah setiap visualisasi
4. Pastikan semua plot memiliki judul, label, dan keterangan yang jelas

Bagian A — Audio

A1. Deskripsi Data

TODO: Jelaskan audio yang akan Anda analisis:

- Jenis audio: suara hutan alam
- Sumber: youtube
- Format file: MP3
- Alasan pemilihan: saya ingin tahu bagaimana bentuk diagram dari suara hutan alam.

Path file: data/suarahutanalam.mp3

A2. TODO: Muat & Metadata

Instruksi: Tulis kode untuk memuat file audio dan menampilkan metadata dasar:

- Sample rate (Hz)
- Durasi (detik)
- Jumlah kanal (mono/stereo)
- Jumlah total sampel

Catatan: Jika file MP3 bermasalah saat loading, gunakan format WAV sebagai alternatif.

```
In [ ]: # A2. Muat file audio dan tampilkan metadata dasar
audio_path = "data/suarahutanalam.mp3" # Ganti dengan path file Anda
try:
    # Coba load dengan soundfile (bisa WAV/OGG/FLAC, kadang MP3 error)
    data, sr = sf.read(audio_path)
    if data.ndim == 1:
        n_channels = 1
        n_samples = data.shape[0]
    else:
        n_channels = data.shape[1]
        n_samples = data.shape[0]
    duration = n_samples / sr
except RuntimeError:
    # Jika gagal (misal MP3), fallback ke librosa (otomatis mono)
    data, sr = librosa.load(audio_path, sr=None, mono=False)
```

```

if data.ndim == 1:
    n_channels = 1
    n_samples = data.shape[0]
else:
    n_channels = data.shape[0]
    n_samples = data.shape[1]
duration = n_samples / sr

print("=== Metadata Audio ===")
print(f"Sample rate      : {sr} Hz")
print(f"Durasi           : {duration:.2f} detik")
print(f"Jumlah kanal       : {'Mono' if n_channels==1 else 'Stereo' if n_channels==2 else n_channels}")
print(f"Jumlah sampel      : {n_samples}")

```

```

=== Metadata Audio ===
Sample rate      : 44100 Hz
Durasi           : 15.81 detik
Jumlah kanal     : Stereo
Jumlah sampel    : 697344

```

A3. TODO: Waveform

Instruksi: Plot waveform audio dengan:

- Sumbu X: waktu (detik)
- Sumbu Y: amplitudo
- Judul dan label sumbu yang jelas

Analisis yang diperlukan: Jelaskan apa yang Anda lihat dari waveform (pola amplitudo, bagian keras/pelan, dll.)

```

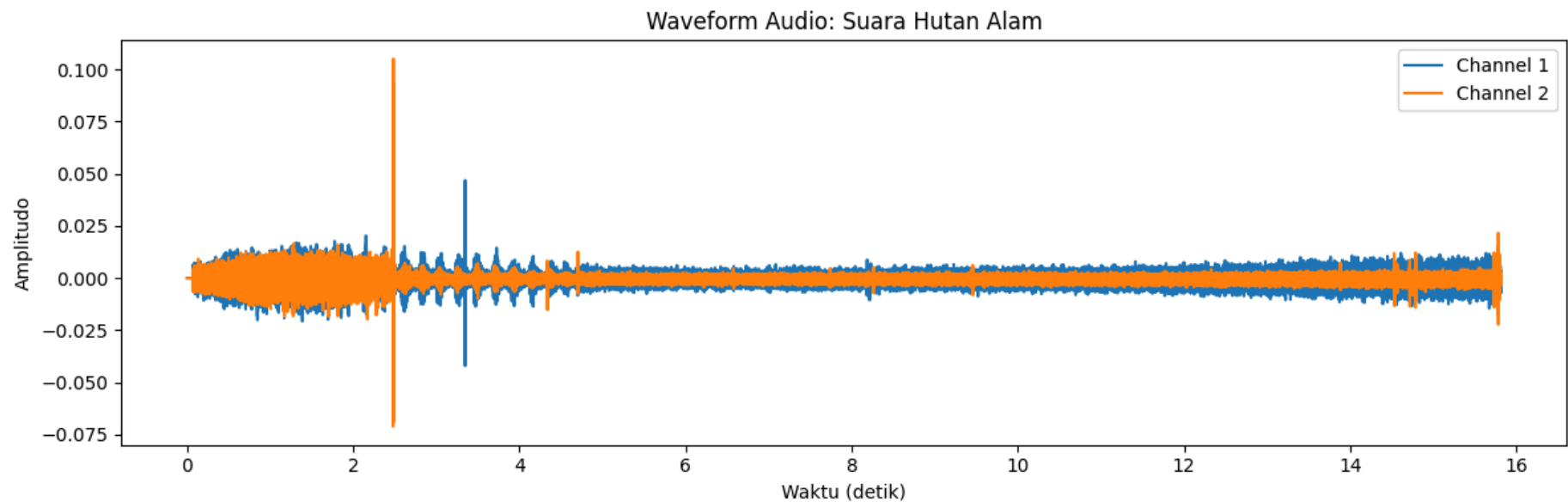
In [ ]: # Plot waveform audio
plt.figure(figsize=(12, 4))
if n_channels == 1:
    # Mono
    times = np.arange(n_samples) / sr
    plt.plot(times, data, color='b')
else:
    # Stereo atau multi-channel: plot tiap kanal

```

```

times = np.arange(n_samples) / sr
for ch in range(n_channels):
    plt.plot(times, data[:, ch], label=f'Channel {ch+1}')
plt.legend()
plt.title("Waveform Audio: Suara Hutan Alam")
plt.xlabel("Waktu (detik)")
plt.ylabel("Amplitudo")
plt.tight_layout()
plt.show()

```



Analisis: Berdasarkan waveform audio suara hutan alam yang dihasilkan, terlihat adanya variasi amplitudo sepanjang waktu. Pada bagian awal (misal detik 0–3) dan beberapa bagian lain yang amplitudonya besar, kemungkinan menandakan adanya suara keras seperti dentuman petir atau suara hewan yang dominan. Sementara pada bagian lain dengan pola amplitudo yang lebih stabil dan kecil, kemungkinan merupakan suara latar seperti gemericik air hujan, angin, atau suara serangga. Pola amplitudo yang naik-turun menunjukkan dinamika lingkungan hutan yang alami, di mana suara keras dan pelan silih berganti sesuai aktivitas di alam.

A4. TODO: Spectrogram log-dB

Instruksi: Hitung STFT dan tampilkan spectrogram dalam skala log-dB:

- Gunakan parameter standar ($n_{\text{fft}}=1024$, $\text{hop_length}=256$)
- Tampilkan dengan colorbar
- Label sumbu: waktu (detik) dan frekuensi (Hz)

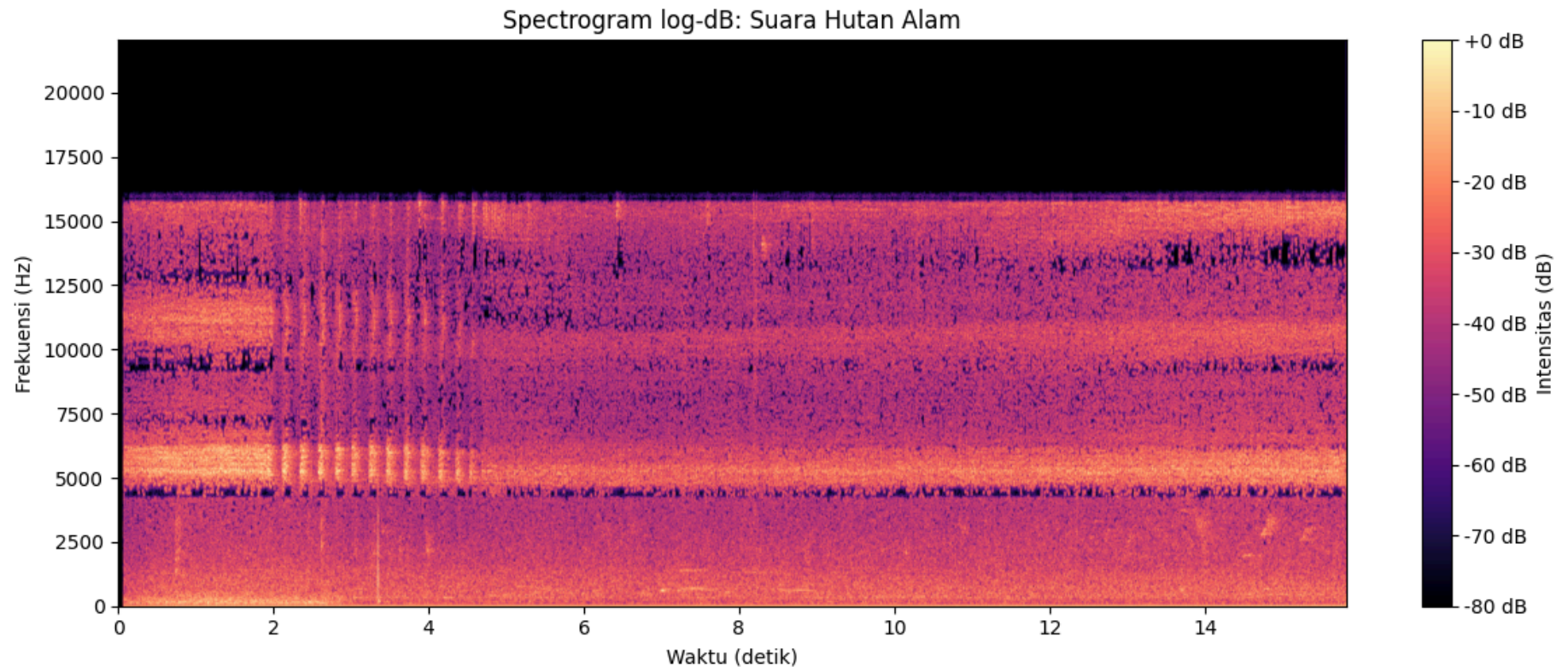
Analisis yang diperlukan: Jelaskan perbedaan informasi yang didapat dari spectrogram dibanding waveform.

```
In [ ]: # Hitung dan tampilkan spectrogram log-dB
n_fft = 1024
hop_length = 256

# Jika data stereo, gunakan channel pertama saja untuk visualisasi spektrum
if n_channels > 1:
    y = data[:, 0]
else:
    y = data

# Hitung STFT
D = librosa.stft(y, n_fft=n_fft, hop_length=hop_length)
S_db = librosa.amplitude_to_db(np.abs(D), ref=np.max)

plt.figure(figsize=(12, 5))
img = plt.imshow(
    S_db,
    aspect='auto',
    origin='lower',
    cmap='magma',
    extent=[
        0, S_db.shape[1] * hop_length / sr,
        0, sr / 2
    ]
)
plt.colorbar(img, format="%+2.0f dB", label="Intensitas (dB)")
plt.title("Spectrogram log-dB: Suara Hutan Alam")
plt.xlabel("Waktu (detik)")
plt.ylabel("Frekuensi (Hz)")
plt.tight_layout()
plt.show()
```



Analisis: Spectrogram log-dB dari audio suara hutan alam memperlihatkan distribusi energi suara pada berbagai frekuensi sepanjang waktu. Pada bagian bawah (frekuensi rendah), biasanya terdapat suara latar seperti angin atau gemuruh jauh yang tampak sebagai area warna terang stabil. Sementara itu, pola-pola vertikal atau area warna terang di frekuensi menengah hingga tinggi menandakan adanya suara mendadak seperti kicauan burung, serangga, atau suara hewan lain. Intensitas warna yang berubah-ubah menunjukkan dinamika lingkungan hutan yang alami, di mana suara keras dan pelan muncul secara acak. Dengan spectrogram, kita dapat mengidentifikasi kapan dan pada frekuensi berapa suara tertentu muncul, yang tidak bisa didapatkan hanya dari waveform.

A5. TODO: MFCC

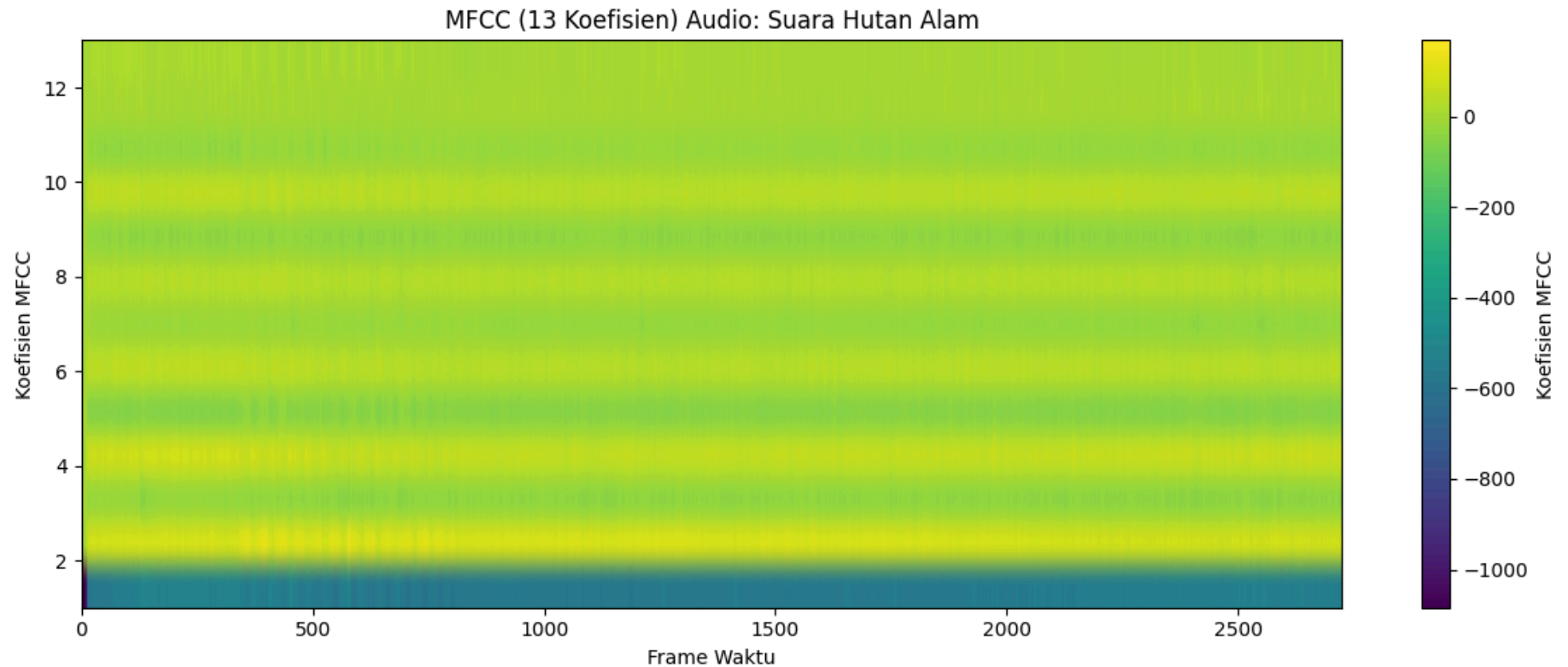
Instruksi: Hitung dan tampilkan minimal 13 koefisien MFCC sebagai heatmap:

- Sumbu X: waktu (frame)
- Sumbu Y: koefisien MFCC (1-13)

- Gunakan colorbar dan judul yang jelas

Analisis yang diperlukan: Interpretasi sederhana: apakah pola MFCC stabil atau berubah-ubah? Apa potensi maknanya?

```
In [ ]: # Hitung dan tampilkan minimal 13 koefisien MFCC sebagai heatmap
n_mfcc = 13
# Gunakan channel pertama jika stereo
if n_channels > 1:
    y = data[:, 0]
else:
    y = data
mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=n_mfcc, n_fft=1024, hop_length=256)
plt.figure(figsize=(12, 5))
img = plt.imshow(
    mfccs,
    aspect='auto',
    origin='lower',
    cmap='viridis',
    extent=[0, mfccs.shape[1], 1, n_mfcc]
)
plt.colorbar(img, label="Koefisien MFCC")
plt.title("MFCC (13 Koefisien) Audio: Suara Hutan Alam")
plt.xlabel("Frame Waktu")
plt.ylabel("Koefisien MFCC")
plt.tight_layout()
plt.show()
```



Analisis Pola MFCC dari audio suara hutan alam menunjukkan adanya bagian yang stabil dan bagian yang berubah-ubah. Bagian MFCC yang stabil biasanya merepresentasikan suara latar yang konstan, seperti hujan atau angin di hutan. Sementara itu, terdapat beberapa bagian pada heatmap MFCC yang tampak tidak stabil atau berubah drastis, yang kemungkinan besar menandakan adanya suara mendadak seperti gemuruh petir atau suara hewan. Secara umum, pola MFCC yang bervariasi ini mencerminkan dinamika lingkungan hutan alam, di mana suara latar yang konstan sesekali diselingi oleh suara mendadak dari alam sekitar.

A6. Analisis Ringkas (Wajib)

Jawab pertanyaan berikut:

1. **Perbedaan insight:** Apa perbedaan informasi yang didapat dari waveform versus spectrogram?

Jawaban: Waveform hanya menunjukkan perubahan amplitudo sinyal audio terhadap waktu, sehingga kita bisa melihat pola keras dan pelan, tetapi tidak mengetahui detail frekuensi atau jenis suara yang muncul. Sedangkan spectrogram memberikan informasi distribusi energi pada berbagai frekuensi sepanjang waktu, sehingga kita bisa mengetahui kapan dan pada frekuensi berapa suara tertentu muncul, misalnya suara hewan, hujan, atau petir.

2. **Pembelajaran dari MFCC:** Apa yang Anda pelajari dari visualisasi MFCC audio ini?

Jawaban: Dari visualisasi MFCC, saya belajar bahwa pola MFCC yang stabil biasanya menunjukkan suara latar yang konstan seperti hujan atau angin, sedangkan pola yang berubah-ubah atau tiba-tiba menandakan adanya suara mendadak seperti petir atau suara hewan. MFCC membantu mengidentifikasi karakteristik dan dinamika suara dalam lingkungan hutan secara lebih detail daripada hanya melihat waveform atau spectrogram saja.

Bagian B — Gambar

B1. Deskripsi Data

TODO: Jelaskan gambar yang akan Anda analisis:

- Jenis gambar: Gambar igabakar
- Sumber: Foto sendiri
- Format file: JPG
- Alasan pemilihan: Karena menurut saya gambar ini memiliki perpaduan warna yang lezat, sehingga menarik untuk dipandang dan diteliti lebih dalam.

Path file: data/igabakar.jpg

B2. TODO: Baca & Tampilkan (RGB)

Instruksi: Baca gambar dan tampilkan dengan benar dalam format RGB:

- Pastikan konversi warna benar (ingat perbedaan BGR vs RGB di OpenCV)
- Berikan judul yang deskriptif

- Hilangkan axis untuk tampilan yang bersih

Analisis yang diperlukan: Jelaskan gambar secara ringkas (objek dominan, kondisi pencahayaan, komposisi warna).

```
In [ ]: # B2. Baca gambar dan tampilkan dalam format RGB
img_path = "data/igabakar.jpg" # Ganti dengan path gambar Anda
# Baca gambar dengan OpenCV (hasilnya BGR)
img_bgr = cv2.imread(img_path)
if img_bgr is None:
    raise FileNotFoundError(f"Gambar tidak ditemukan: {img_path}")
# Konversi BGR ke RGB
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)
plt.figure(figsize=(8, 6))
plt.imshow(img_rgb)
plt.title("Gambar Igabakar (RGB)")
plt.axis('off') # Hilangkan axis
plt.show()
```

Gambar Igabakar (RGB)



Analisis: Gambar yang dipilih menampilkan hidangan iga bakar sebagai objek dominan. Pencahayaan pada gambar tampak terang dan alami, sehingga detail tekstur dan warna makanan terlihat jelas. Komposisi warna didominasi oleh warna coklat keemasan dari daging iga yang terbakar, disertai warna merah dari saus atau bumbu, serta hijau dari garnish atau sayuran pelengkap. Perpaduan warna-warna ini memberikan kesan makanan yang menggugah selera dan menarik untuk dipandang. Tekstur permukaan iga yang terlihat jelas juga menambah daya tarik visual pada gambar tersebut.

B3. TODO: Informasi Dasar

Instruksi: Tampilkan informasi metadata gambar:

- Dimensi (Height × Width)
- Jumlah kanal
- Tipe data (dtype)
- Mode warna (jika relevan)
- Ukuran file dalam memori

Analisis yang diperlukan: Jelaskan mengapa informasi ini penting untuk tahap preprocessing atau analisis lanjutan.

```
In [ ]: # B3. Tampilkan informasi metadata gambar
height, width, channels = img_rgb.shape
dtype = img_rgb.dtype
mode = "RGB"
mem_bytes = img_rgb.nbytes
mem_kb = mem_bytes / 1024
print("=== Metadata Gambar ===")
print(f"Dimensi (Height x Width): {height} x {width}")
print(f"Jumlah kanal          : {channels}")
print(f"Tipe data (dtype)      : {dtype}")
print(f"Mode warna             : {mode}")
print(f"Ukuran file di memori   : {mem_kb:.2f} KB")
```

```
=== Metadata Gambar ===
Dimensi (Height x Width): 1280 x 720
Jumlah kanal          : 3
Tipe data (dtype)      : uint8
Mode warna             : RGB
Ukuran file di memori : 2700.00 KB
```

Analisis: Informasi metadata gambar sangat penting untuk tahap preprocessing dan analisis lanjutan. Dengan mengetahui dimensi (tinggi x lebar), kita bisa menentukan apakah perlu melakukan resize agar gambar sesuai dengan input model atau aplikasi. Jumlah kanal dan mode warna memastikan gambar sudah dalam format yang diinginkan (misal RGB, bukan BGR atau grayscale). Tipe data (dtype) penting untuk proses normalisasi dan efisiensi memori. Ukuran file di memori membantu memperkirakan kebutuhan storage dan kecepatan pemrosesan. Semua informasi ini memastikan proses analisis atau pelatihan model berjalan optimal dan sesuai spesifikasi teknis yang dibutuhkan.

B4. TODO: Histogram Warna

Instruksi: Tampilkan histogram distribusi intensitas untuk channel R, G, B:

- Range: 0-255
- Plot terpisah atau overlay dengan warna sesuai channel
- Label sumbu: intensitas pixel dan frekuensi
- Legend yang jelas

Analisis yang diperlukan: Analisis: channel mana yang dominan? Bagaimana kontras gambar? Seperti apa sebaran intensitasnya?

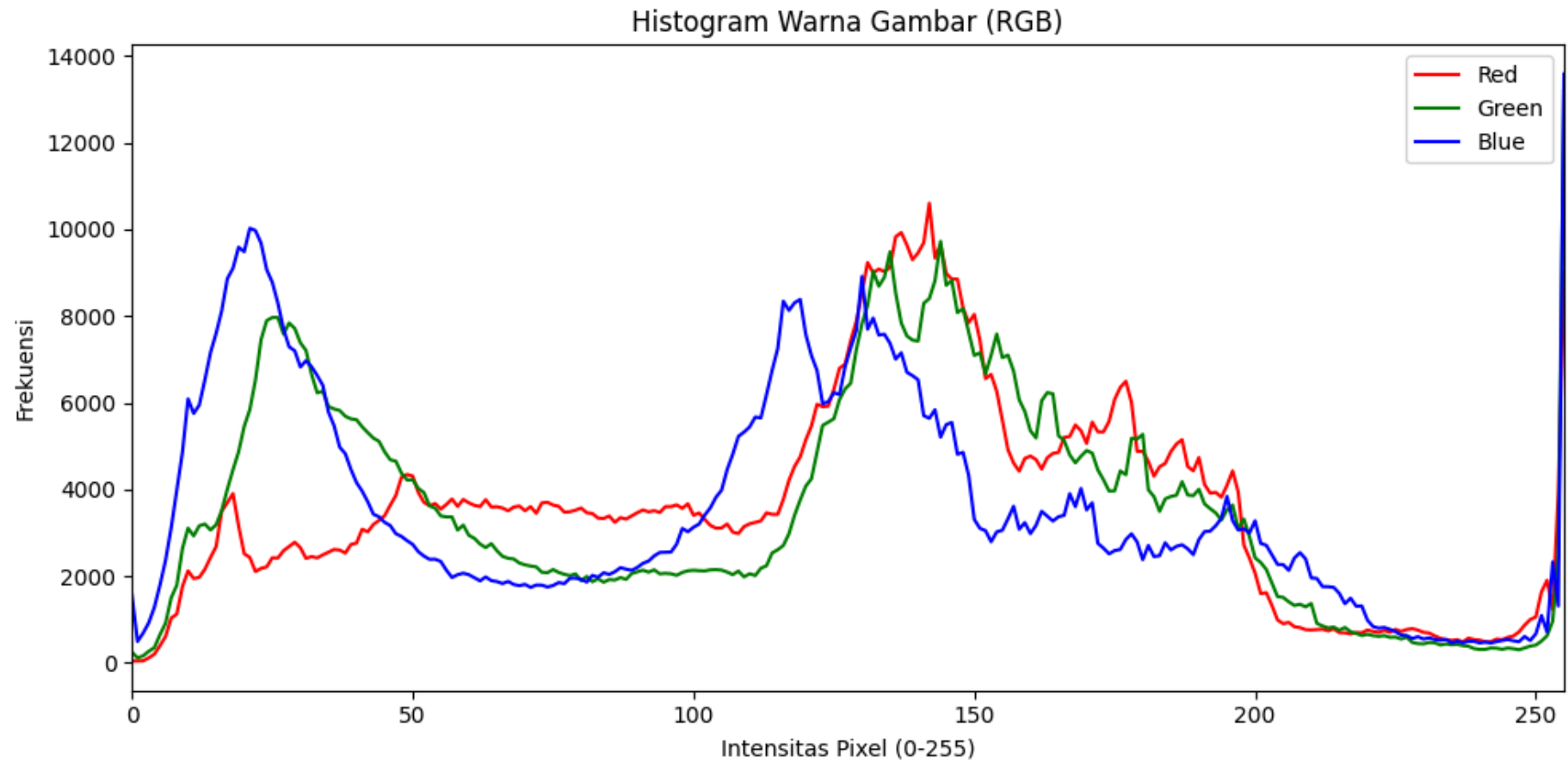
```
In [ ]: import cv2
import matplotlib.pyplot as plt

# B4. Tampilkan histogram distribusi intensitas untuk channel R, G, B
colors = ('r', 'g', 'b')
channel_names = ('Red', 'Green', 'Blue')

plt.figure(figsize=(10, 5))

for i, color in enumerate(colors):
    hist = cv2.calcHist([img_rgb], [i], None, [256], [0, 256])
    plt.plot(hist, color=color, label=channel_names[i])

plt.xlim([0, 255])
plt.title("Histogram Warna Gambar (RGB)")
plt.xlabel("Intensitas Pixel (0-255)")
plt.ylabel("Frekuensi")
plt.legend()
plt.tight_layout()
plt.show()
```



Analisis: Histogram menunjukkan bahwa channel biru lebih dominan, terutama pada intensitas rendah (0–50), sedangkan hijau dan merah lebih menonjol pada rentang menengah (100–180). Sebaran intensitas cukup luas dari 0 hingga 255, menandakan gambar memiliki kontras yang baik dengan area gelap dan terang yang sama-sama signifikan. Secara keseluruhan, distribusi intensitas bergelombang sehingga mencerminkan gambar memiliki variasi warna dan tekstur yang kaya.

B5. Analisis Ringkas (Wajib)

Jawab pertanyaan berikut:

Relasi histogram dengan kesan visual: Apa hubungan antara pola histogram yang Anda lihat dengan kesan visual gambar (terang/gelap, warna dominan, kontras)?

Pola histogram menunjukkan sebaran intensitas yang luas dari gelap hingga terang, sehingga gambar memiliki kontras yang baik. Puncak yang tinggi pada channel biru di intensitas rendah memberi kesan gambar agak dingin atau kebiruan, sementara hijau dan merah pada intensitas menengah membuat warna tampak lebih seimbang. Dengan demikian, histogram ini berkaitan erat dengan kesan visual gambar yang tidak terlalu gelap maupun terlalu terang, tetapi cenderung memiliki nuansa biru dominan dengan detail kontras yang jelas.

Bagian C — Video

C1. Deskripsi Data

TODO: Jelaskan video yang akan Anda analisis:

- Jenis video: Pemandangan UMKM martabak
- Sumber: rekaman sendiri
- Durasi target: 15 Detik
- Alasan pemilihan: Dikarenakan suasana gelap dan menggugah selera serta kamera seadanya membuat saya ingin meneliti video ini.

Path file: `data/pemandanganumkm.mp4`

C2. TODO: Baca & Metadata

Instruksi: Baca video dengan OpenCV dan tampilkan metadata:

- Resolusi (Width × Height)
- Frame rate (fps)
- Jumlah total frame
- Durasi (detik)
- Klasifikasi resolusi (HD, Full HD, 4K, dll.)

Analisis yang diperlukan: Jelaskan pentingnya parameter-parameter tersebut untuk analisis video atau aplikasi tertentu.

```
In [ ]: import cv2

# C2. Baca video dengan OpenCV dan tampilkan metadata
video_path = "data/pemandanganumkm.mp4"
cap = cv2.VideoCapture(video_path)

if not cap.isOpened():
    raise FileNotFoundError(f"Video tidak ditemukan: {video_path}")

# Ambil metadata
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
fps = cap.get(cv2.CAP_PROP_FPS)
total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
duration = total_frames / fps if fps > 0 else 0

# Klasifikasi resolusi
if width >= 3840 and height >= 2160:
    res_class = "4K"
elif width >= 1920 and height >= 1080:
    res_class = "Full HD"
elif width >= 1280 and height >= 720:
    res_class = "HD"
else:
    res_class = "SD/Low"

# Tampilkan metadata
print("=== Metadata Video ===")
print(f"Resolusi (Width x Height): {width} x {height}")
print(f"Frame rate (fps)           : {fps:.2f}")
print(f"Jumlah total frame           : {total_frames}")
print(f"Durasi (detik)               : {duration:.2f}")
print(f"Klasifikasi resolusi         : {res_class}")

cap.release()
```

=== Metadata Video ===

Resolusi (Width x Height): 478 x 850
Frame rate (fps) : 29.90
Jumlah total frame : 463
Durasi (detik) : 15.48
Klasifikasi resolusi : SD/Low

Analisis: Berdasarkan metadata video, resolusi 478 x 850 termasuk kategori SD/Low, sehingga detail gambar tidak terlalu tajam namun lebih ringan untuk disimpan dan diputar di perangkat dengan keterbatasan memori. Frame rate sekitar 29,90 fps sudah cukup untuk memberikan kesan gerakan yang halus, sehingga transisi antarframe tidak terlihat patah-patah. Jumlah total frame 463 dengan durasi 15,48 detik menunjukkan bahwa video ini relatif singkat dan tidak membutuhkan kapasitas penyimpanan yang besar. Dengan klasifikasi resolusi SD/Low, video ini lebih cocok digunakan untuk kebutuhan dokumentasi sederhana atau berbagi di platform dengan keterbatasan bandwidth dibandingkan untuk tampilan layar besar yang memerlukan detail tinggi.

C3. TODO: Tampilkan 3 Frame (Awal–Tengah–Akhir)

Instruksi: Ambil dan tampilkan 3 frame representatif:

- Frame pertama (index 0)
- Frame tengah (index $\sim \text{total_frame}/2$)
- Frame terakhir (index $\text{total_frame}-1$)
- **Konversi BGR→RGB** sebelum ditampilkan
- Subplot dengan judul frame dan timestamp

Analisis yang diperlukan: Deskripsikan perbedaan visual antar frame dan apa yang dapat dipelajari dari sampel frame ini.

```
In [ ]: # C3. Ambil dan tampilkan 3 frame representatif (awal, tengah, akhir)
video_path = "data/pemandanganumkm.mp4"
cap = cv2.VideoCapture(video_path)

if not cap.isOpened():
    raise FileNotFoundError(f"Video tidak ditemukan: {video_path}")

# Ambil metadata
total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
```

```
fps = cap.get(cv2.CAP_PROP_FPS)

# Tentukan frame awal, tengah, dan akhir
frame_idx = [0, total_frames // 2, total_frames - 1]
frames = []
timestamps = []

for idx in frame_idx:
    cap.set(cv2.CAP_PROP_POS_FRAMES, idx)
    ret, frame = cap.read()

    if not ret:
        frames.append(None)
        timestamps.append(None)
        continue

    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frames.append(frame_rgb)
    timestamps.append(idx / fps if fps > 0 else 0)

cap.release()

# Tampilkan frame
plt.figure(figsize=(16, 6))
titles = ["Frame Awal", "Frame Tengah", "Frame Akhir"]

for i, (img, ts) in enumerate(zip(frames, timestamps)):
    plt.subplot(1, 3, i + 1)
    if img is not None:
        plt.imshow(img)
        plt.title(f"{titles[i]}\nFrame {frame_idx[i]} (t={ts:.2f} detik)")
        plt.axis('off')
    else:
        plt.title(f"{titles[i]}\nFrame tidak ditemukan")
        plt.axis('off')

plt.tight_layout()
plt.show()
```

Frame Awal
Frame 0 (t=0.00 detik)



Frame Tengah
Frame 231 (t=7.73 detik)



Frame Akhir
Frame tidak ditemukan

Analisis Berdasarkan tiga frame representatif, terlihat perbedaan visual yang cukup jelas dari awal hingga tengah video. Frame awal memperlihatkan suasana awal proses memasak, dengan kondisi pencahayaan agak redup dan objek utama (penggorengan serta adonan) baru mulai terlihat. Pada frame tengah, posisi kamera lebih fokus ke aktivitas utama yaitu seseorang sedang menyiapkan martabak dengan pencahayaan yang lebih terang, sehingga detail objek dan aktivitas terlihat lebih jelas. Sementara itu, frame akhir tidak terbaca, kemungkinan karena keterbatasan pembacaan frame terakhir pada video. Dari sampel ini, kita bisa memahami adanya perubahan posisi objek, aktivitas manusia yang semakin jelas, serta variasi pencahayaan yang menggambarkan dinamika proses pembuatan makanan dalam video.

C4. Analisis Ringkas (Wajib)

Jawab pertanyaan berikut:

Kesesuaian parameter: Apakah fps dan resolusi video ini sesuai untuk use case pilihan Anda (misalnya: media sosial, kuliah daring, presentasi, dll.)? Jelaskan alasan singkat.

Dengan resolusi 478 x 850 (SD/Low) dan frame rate sekitar 29,9 fps, video ini masih sesuai untuk media sosial, khususnya platform berbasis mobile seperti Instagram atau TikTok yang mendukung format vertikal. Resolusi rendah membuat ukuran file lebih kecil dan mudah diunggah, sementara fps mendekati 30 sudah cukup untuk menghasilkan gerakan yang halus. Namun, untuk kebutuhan kuliah daring atau presentasi formal di layar besar, kualitas ini kurang ideal karena detail visual terbatas.

Perbandingan & Kesimpulan

Perbandingan Representasi Media

TODO: Bandingkan secara ringkas representasi dan visualisasi ketiga media:

Audio (1D - Temporal)

- Representasi: Sinyal 1 dimensi terhadap waktu.
- Visualisasi utama: Waveform dan spektrogram.
- Informasi yang diperoleh: Intensitas suara, frekuensi, durasi, serta pola temporal audio.

Gambar (2D - Spasial)

- Representasi: Matriks piksel dengan nilai intensitas (grayscale) atau kombinasi channel warna (RGB).
- Visualisasi utama: Histogram warna, tampilan citra.
- Informasi yang diperoleh: Distribusi warna, kontras, tekstur, dan detail spasial objek.

Video (2D + Waktu - Spatio-temporal)

- Representasi: Rangkaian gambar (frame) berurutan terhadap waktu.
 - Visualisasi utama: Frame representatif, metadata video (resolusi, fps, durasi).
 - Informasi yang diperoleh: Perubahan gerakan, dinamika objek, pencahayaan, serta narasi visual sepanjang waktu.
-

Refleksi Pembelajaran

3 Poin yang Saya Pelajari:

- 1. Perbedaan mendasar antara representasi audio, gambar, dan video berdasarkan dimensi data.
- 2. Teknik visualisasi sederhana seperti waveform, histogram, dan frame representatif dapat membantu analisis media.
- 3. Metadata (fps, resolusi, durasi) sangat penting untuk menilai kesesuaian video dengan kebutuhan penggunaan.

2 Hal yang Masih Membingungkan/Ingin Diperdalam:

- 1. Bagaimana cara mengoptimalkan kualitas media dengan tetap menjaga ukuran file tetap kecil.
- 2. Kenapa frame terakhir tidak ada dalam video iga bakar.

Sumber Data & Referensi

TODO: Cantumkan semua sumber data dan referensi yang digunakan:

- **Audio:** (<https://youtube.com/shorts/EYuBgVSSuJE?si=oJ5w6RFpDOkRYf2u>)
- **Gambar:** Foto Sendiri
- **Video:** Video Sendiri
- **Referensi teknis:** (<https://chatgpt.com/share/68c82ae2-df3c-8008-ab74-8d1c291f219d>)

Rubrik Penilaian

Distribusi Bobot Penilaian

Aspek Penilaian	Bobot	Deskripsi
Kelengkapan	35%	Semua langkah inti dikerjakan sesuai checklist

Aspek Penilaian	Bobot	Deskripsi
Kualitas Visualisasi	20%	Judul, label sumbu, colorbar, legend, keterbacaan plot
Analisis & Interpretasi	30%	Kemampuan interpretasi hasil, bukan sekadar output mentah
Kerapihan & Struktur	10%	Markdown jelas, kode modular, dokumentasi baik
Orisinalitas & Penguasaan	5%	Pemahaman saat presentasi acak

Detail Kriteria Penilaian

Kelengkapan (35%)

- ✓ Semua 4 visualisasi audio (metadata, waveform, spectrogram, MFCC)
- ✓ Semua 3 visualisasi gambar (display RGB, metadata, histogram)
- ✓ Semua 2 visualisasi video (metadata, frame extraction)
- ✓ Analisis ringkas untuk setiap bagian

Kualitas Visualisasi (20%)

- Plot memiliki judul yang informatif dan deskriptif
- Label sumbu X dan Y jelas dan sesuai
- Colorbar/legend tersedia jika diperlukan
- Ukuran plot proporsional dan mudah dibaca

Analisis & Interpretasi (30%)

- Interpretasi menunjukkan pemahaman konsep
- Analisis kontekstual, bukan sekadar deskripsi output
- Mampu menghubungkan hasil dengan teori
- Refleksi pembelajaran yang thoughtful

Kerapihan & Struktur (10%)

- Markdown terstruktur dengan heading yang konsisten
- Kode bersih, terkompartemen, dan mudah dibaca
- Dokumentasi yang memadai
- Flow logical dari satu bagian ke bagian lain

Orisinalitas & Penguasaan (5%)

- **PENTING:** Jika saat presentasi acak Anda tidak mampu menjelaskan kode yang Anda tulis atau menunjukkan ketergantungan buta pada AI/copy-paste, **nilai tugas akan dianggap 0**.
- Kemampuan menjelaskan logika dan alur pemikiran
- Pemahaman konsep di balik implementasi kode





Proporsi Penilaian Total

- Proporsi penilaian hanya 80%, 20% lagi akan didasarkan pada kecepatan pengumpulan tugas
- Sehingga: $0.8 \times \text{penilaian dosen} + \text{nilai waktu pengumpulan}$

Aturan Kejujuran Akademik





Penggunaan Referensi & AI yang Diperbolehkan

Anda **BOLEH** menggunakan:

-  Dokumentasi resmi library (NumPy, Matplotlib, Librosa, OpenCV)
-  Tutorial dan contoh kode dari sumber terpercaya
-  AI tools (ChatGPT, GitHub Copilot, dll.) sebagai **alat bantu pembelajaran**
-  Diskusi dengan teman untuk pemahaman konsep

Syarat & Batasan WAJIB

Namun Anda **HARUS**:

-  **Memahami setiap baris kode** yang Anda masukkan ke notebook
-  **Menulis interpretasi dengan kata-kata sendiri**, bukan hasil copy-paste
-  **Mencantumkan sumber data dan referensi** yang digunakan, termasuk transkrip percakapan dengan AI dalam link atau teks
-  **Mampu menjelaskan logika dan alur pemikiran** saat presentasi acak

Pelanggaran yang Berakibat Nilai 0

- **Plagiarisme atau penyalinan buta** dari sumber manapun
- **Copy-paste kode tanpa pemahaman** dan tidak dapat menjelaskan
- **Menggunakan AI untuk mengerjakan seluruh tugas** tanpa pembelajaran personal
- **Tidak dapat menjawab pertanyaan dasar** tentang kode yang dikumpulkan
- **Menyalin pekerjaan teman** atau bekerjasama dalam pengerjaan individual

Persiapan Presentasi Acak

Kemungkinan pertanyaan yang akan ditanyakan:

- "Jelaskan mengapa Anda menggunakan parameter ini di STFT?"
- "Apa arti dari pola yang terlihat di MFCC?"
- "Mengapa perlu konversi BGR ke RGB?"
- "Interpretasikan hasil histogram yang Anda buat"
- "Bagaimana cara kerja spectrogram?"

Tips sukses:

- Pahami konsep dasar setiap teknik yang digunakan
- Latih menjelaskan dengan bahasa sederhana
- Siapkan justifikasi untuk setiap pilihan parameter

- Kuasai interpretasi setiap visualisasi yang dibuat

Panduan Pengumpulan

Berkas yang Harus Dikumpulkan

Wajib:

1. **Notebook Jupyter** (.ipynb) dengan nama: `NIM_Nama_TugasMultimedia.ipynb`
 - Contoh: `123456789_JohnDoe_TugasMultimedia.ipynb`
 2. **PDF hasil render dari notebook**
-



Informasi Pengumpulan



Checklist Sebelum Submit

- [☒] Semua cell sudah dijalankan dan menampilkan output
 - [☒] Nama file sesuai format: `NIM_Worksheet2.ipynb` dan `NIM_Worksheet2.pdf`
 - [☒] Semua TODO sudah diisi dengan lengkap
 - [☒] Analisis dan interpretasi sudah ditulis untuk setiap bagian
 - [☒] Sumber data dan referensi sudah dicantumkan
-

Export ke PDF:

- File → Save and Export Notebook As → HTML
- Buka HTML di browser -> Save as PDF