

# Tugas Analisis Multimedia: Audio, Gambar, Video

**Mata Kuliah:** Sistem & Teknologi Multimedia

**Nama:** Adin Adry Tjindarbumi **NIM:** 122140024

---

## 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 [2]: # Import Library (Satu-satunya sel kode dalam template ini)
import numpy as np
import matplotlib as mpl
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: {mpl.__version__}")
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: 3.10.5
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 (tandai 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: suarah hujan
- Sumber: youtube
- Format file: MP3
- Alasan pemilihan: Saya ingin tahu bagaimana diagram dari suara hujan

**Path file:** data/suara\_hujan.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 [3]: # A2. Muat file audio dan tampilkan metadata dasar

audio_path = "data/suara_hujan.mp3"

try:
    # Coba Load dengan soundfile (bisa WAV/OGG/FLAC, kadang MP3 error)
    data, sr = sf.read(audio_path)
    # Jika data 1D, berarti mono. Jika 2D, shape[1] = jumlah kanal
    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}")
print(f"Jumlah sampel     : {n_samples}")

=== Metadata Audio ===
Sample rate      : 48000 Hz
Durasi           : 63.24 detik
Jumlah kanal     : Stereo
Jumlah sampel    : 3035520
```

---

### A3. TODO: Waveform

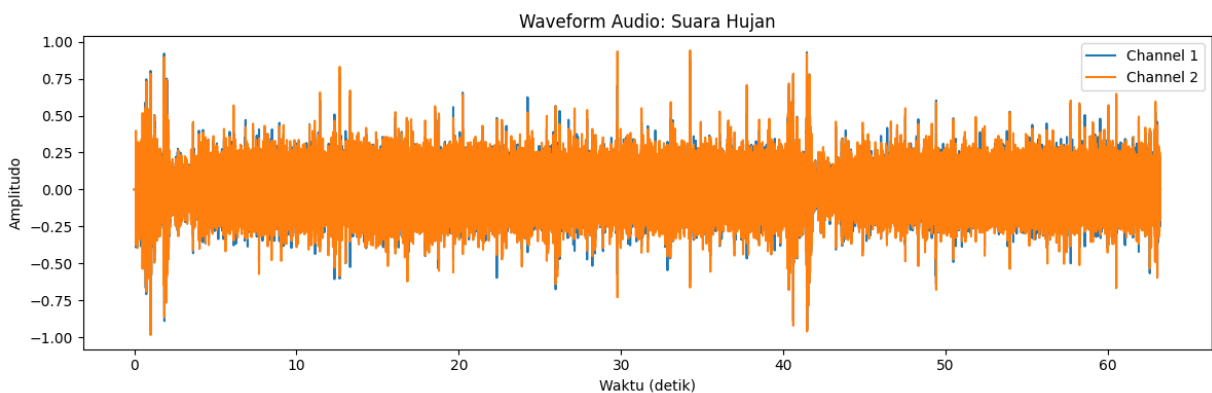
**Instruksi:** Plot waveform audio dengan:

- 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 [4]: # 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 Hujan")
plt.xlabel("Waktu (detik)")
plt.ylabel("Amplitudo")
plt.tight_layout()
plt.show()
```



**Analisis:** Berdasarkan waveform audio suara hujan yang dihasilkan, yang saya tangkap adalah bahwa bagian yang memiliki amplitudo yang besar seperti yang ditunjukkan pada detik 0 - 3 dan 40 - 43 menandakan adanya suara dentuman petir, sedangkan pada detik lainnya yang memiliki pola amplitudo yang stabil menandakan bunyi gemericik air hujan.

### A4. TODO: Spectrogram log-dB

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

- Gunakan parameter standar (n\_fft=1024, hop\_length=256)
- Tampilkan dengan colorbar
- Label sumbu: waktu (detik) dan frekuensi (Hz)

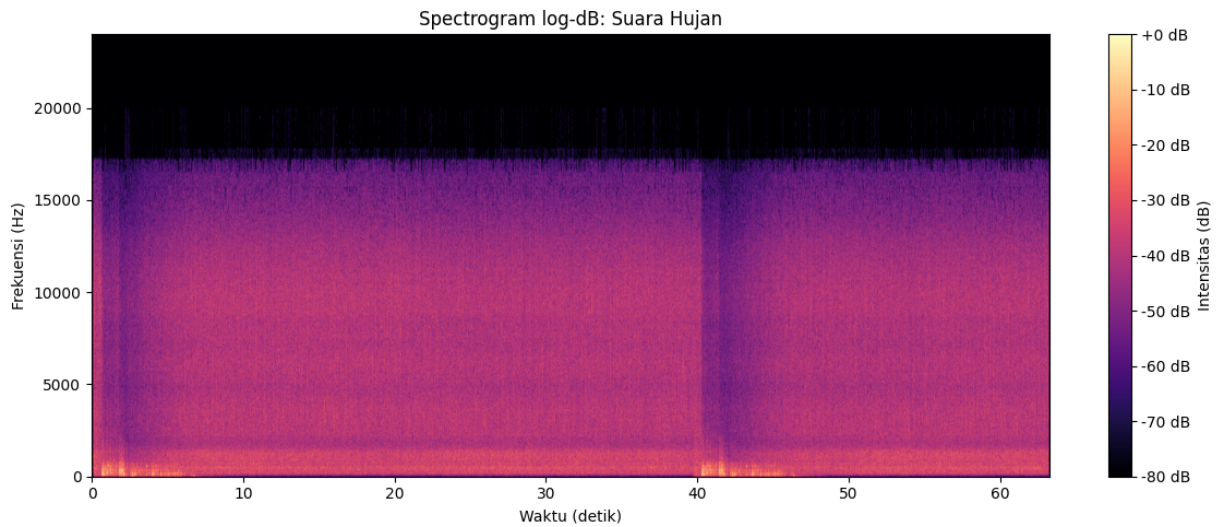
**Analisis yang diperlukan:** Jelaskan perbedaan informasi yang didapat dari spectrogram dibanding waveform.

```
In [5]: # A4. 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 Hujan")
plt.xlabel("Waktu (detik)")
plt.ylabel("Frekuensi (Hz)")
plt.tight_layout()
plt.show()
```



**Analisis:** Spectrogram memberikan visualisasi dari audio suara hujan berdasarkan frekuensi (hz) terhadap waktu dengan menandakan intensitas suara, sedangkan waveform memberikan visualisasi audio suara hujan berdasarkan amplitudo terhadap waktu nya.

## 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 [6]: # A5. Hitung dan tampilkan minimal 13 koefisien MFCC sebagai heatmap
n_mfcc = 13

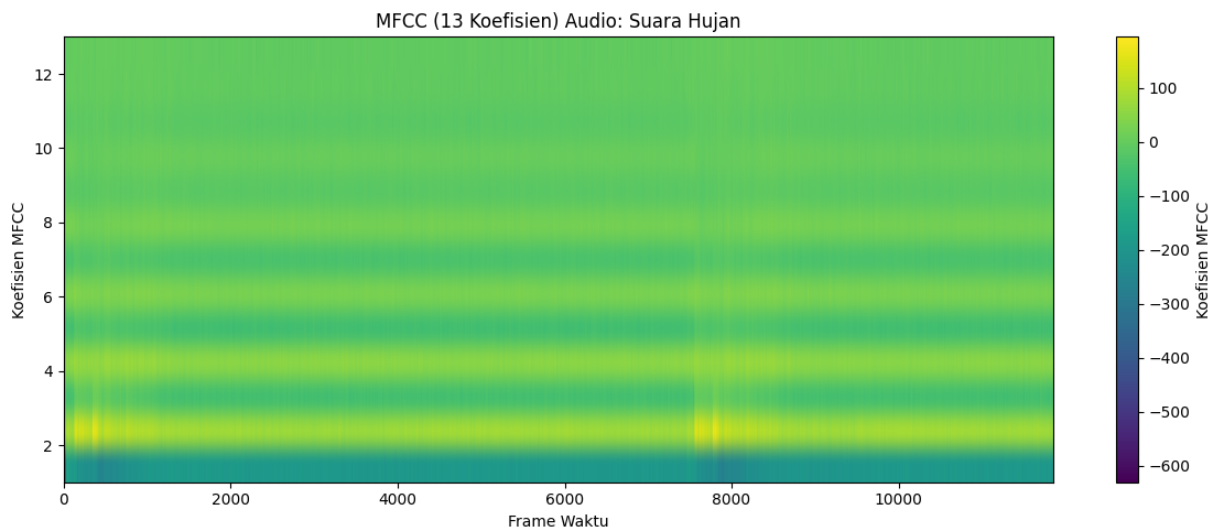
# Gunakan channel pertama jika stereo
if n_channels > 1:
    y = data[:, 0]
else:
    y = data

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 Hujan")
plt.xlabel("Frame Waktu")
plt.ylabel("Koefisien MFCC")
plt.tight_layout()
plt.show()
```



**Analisis:** Pola MFCC yang dihasilkan dari audio suara hujan, memiliki 2 bagian yang tidak stabil menandakan potensi bunyi gemuruh petir dengan sisa pola nya stabil yang menandakan potensi bunyi hujan yang konstan

## A6. Analisis Ringkas (Wajib)

**Jawab pertanyaan berikut:**

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

*Jawaban Anda:* Waveform menunjukkan amplitudo terhadap waktu sehingga kita bisa tahu pola keras/pelan audio tersebut secara umum tetapi tidak memberitahukan , sedangkan spectrogram memberikan informasi detail dari besar frekuensi tiap waktunya sehingga kita dapat melihat karakteristik dari audio tersebut

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

*Jawaban Anda:* Berdasarkan yang saya lihat di MFCC, saya bisa melihat pola koefisien yang stabil atau berubah-ubah sepanjang waktu. Pola MFCC yang stabil menandakan suara yang konsisten yaitu suara hujan yang konstan, sedangkan perubahan mendadak pada MFCC bisa menunjukkan adanya kejadian khusus seperti petir atau suara lain yang berbeda karakter. MFCC sangat berguna untuk membedakan jenis suara atau mengenali pola audio secara otomatis.

# Bagian B — Gambar

## B1. Deskripsi Data

**TODO:** Jelaskan gambar yang akan Anda analisis:

- Jenis gambar: Pemandangan Pantai
- Sumber: Foto Sendiri
- Format file: JPG
- Alasan pemilihan: Karena menurut saya gambar ini memiliki perpaduan warna yang bagus, sehingga menarik untuk diteliti

**Path file:** data/pantai\_krui.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 [7]: # B2. Baca gambar dan tampilkan dalam format RGB
img_path = "data/pantai_krui.jpg"

# 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("Pemandangan Pantai Krui (RGB)")
plt.axis('off') # Hilangkan axis
plt.show()
```

## Pemandangan Pantai Krui (RGB)



**Analisis:** Gambar yang dipilih menampilkan pemandangan pantai dengan objek dominan berupa laut, langit, kapal nelayan dan garis pantai. Kondisi pencahayaan tampak terang alami, dengan komposisi warna biru (laut & langit) yang kuat, serta warna pasir dan hijau dari vegetasi di area lain.

---

### 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 [8]: # 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): 3472 x 4624
Jumlah kanal : 3
Tipe data (dtype) : uint8
Mode warna : RGB
Ukuran file di memori : 47034.75 KB
```

**Analisis:** Informasi ini penting untuk menentukan langkah preprocessing (resize, normalisasi, konversi warna), serta memastikan format data sesuai kebutuhan model atau aplikasi yang digunakan.

## B4. TODO: Histogram Warna

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

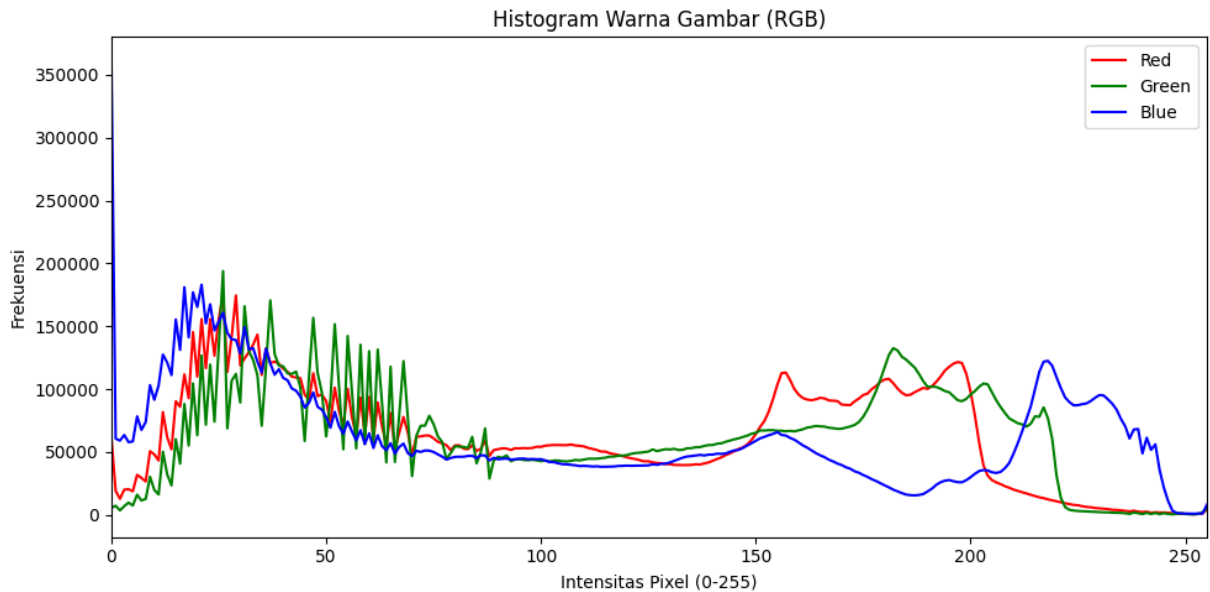
- Range: 0-255
- Plot terpisah atau overlay dengan warna sesuai channel
- Label sumbu: intensitas pixel dan frekuensi
- Legend yang jelas

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

```
In [9]: # 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:** Berdasarkan histogram warna, channel biru tampak paling dominan, yang sesuai dengan visual gambar pantai dan langit. Channel hijau dan merah juga muncul, namun intensitas biru lebih tinggi pada sebagian besar rentang intensitas. Kontras gambar cukup baik karena sebaran intensitas menyebar dari rendah hingga tinggi, tidak cuma terkumpul di satu area. Sebaran intensitas yang lumayan merata menunjukkan gambar memiliki detail di area terang dan gelap, sehingga gambar tampak jelas dan tidak terlalu gelap atau terlalu terang.

## 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)?

*Jawaban Anda:* Pola histogram menunjukkan bahwa channel biru paling dominan, sesuai dengan visual gambar yang didominasi warna biru dari laut dan langit. Sebaran intensitas yang cukup merata dari rendah hingga tinggi menandakan gambar memiliki kontras yang baik, sehingga detail di area terang dan gelap tetap terlihat jelas. Jika histogram terkonsentrasi di intensitas tinggi, gambar akan tampak terang; jika di intensitas rendah, gambar akan tampak gelap. Pada gambar ini, distribusi yang seimbang membuat gambar tampak natural, tidak terlalu gelap atau terlalu terang, dan warna dominan biru memperkuat suasana pantai.

## Bagian C — Video

### C1. Deskripsi Data

**TODO:** Jelaskan video yang akan Anda analisis:

- Jenis video: Pemandangan Air Terjun
- Sumber: Rekaman Sendiri
- Durasi target: 3 Detik
- Alasan pemilihan:

**Path file:** data/video\_arter.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 [10]: # C2. Baca video dengan OpenCV dan tampilkan metadata
video_path = "data/video_arter.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"

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): 2160 x 3840
Frame rate (fps)          : 29.41
Jumlah total frame        : 102
Durasi (detik)            : 3.47
Klasifikasi resolusi      : Full HD
```

**Analisis:** Parameter seperti resolusi, frame rate (fps), jumlah total frame, dan durasi sangat penting untuk analisis video maupun aplikasi tertentu. Resolusi menentukan seberapa detail gambar yang bisa ditampilkan—semakin tinggi resolusi, semakin tajam dan jelas videonya, tapi juga membutuhkan lebih banyak memori. Frame rate mempengaruhi kelancaran gerakan; fps tinggi cocok untuk aksi cepat sedangkan fps rendah bisa menyebabkan gerakan tampak patah-patah. Jumlah total frame dan durasi penting untuk mengetahui panjang video dan kebutuhan penyimpanan. Klasifikasi resolusi (HD, Full HD, 4K) membantu menyesuaikan video dengan kebutuhan platform.

### 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 [11]: # C3. Ambil dan tampilkan 3 frame representatif (awal, tengah, akhir)
video_path = "data/video_arter.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)
frame_idxs = [0, total_frames // 2, total_frames - 1]
frames = []
timestamps = []

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



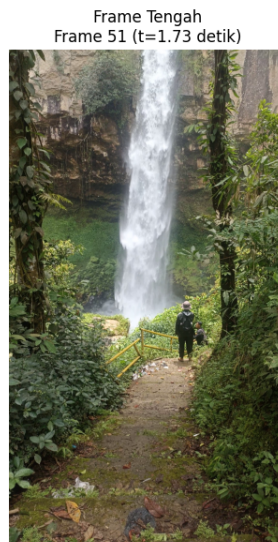
```

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()

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_idxs[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()

```



**Analisis:** Perbedaan visual antar frame terlihat dari perubahan posisi objek dan pencahayaan. Frame awal menampilkan kondisi awal video, yaitu air terjun dengan aliran air yang tidak utuh dan posisi ditangga paling atas. Pada frame tengah, terlihat perubahan posisi air, percikan, atau pergerakan objek lain di sekitar air terjun. Frame akhir bisa menunjukkan perubahan pemandangan air terjun terlihat utuh. Dari sampel tiga frame ini, kita bisa memahami dinamika gerakan, perubahan pencahayaan, dan mengetahui apakah ada objek atau aktivitas penting yang terjadi sepanjang 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.

*Jawaban Anda:* Fps dan resolusi video ini sudah sesuai untuk use case seperti media sosial. Resolusi yang cukup tinggi membuat detail gambar tetap jelas saat ditampilkan di layar besar maupun kecil. Frame rate yang stabil memastikan gerakan air terjun terlihat halus dan tidak patah-patah, sehingga pengalaman menonton tetap nyaman. Selain itu, ukuran file masih efisien untuk diunggah atau dibagikan secara online tanpa mengorbankan kualitas visual.

## Perbandingan & Kesimpulan

### Perbandingan Representasi Media

**TODO:** Bandingkan secara ringkas representasi dan visualisasi ketiga media:

#### Audio (1D - Temporal)

- Representasi: Sinyal satu dimensi berupa amplitudo terhadap waktu.
- Visualisasi utama: Waveform, spectrogram, dan MFCC.
- Informasi yang diperoleh: Pola amplitudo, intensitas suara, karakteristik frekuensi, serta perubahan suara sepanjang waktu.

#### Gambar (2D - Spasial)

- Representasi: Matriks dua dimensi berisi nilai intensitas pixel pada koordinat (x, y).
- Visualisasi utama: Tampilan gambar RGB, histogram warna.
- Informasi yang diperoleh: Komposisi warna, distribusi intensitas, detail spasial, dan objek dominan dalam gambar

#### Video (2D + Waktu - Spatio-temporal)

- Representasi: Rangkaian frame gambar 2D yang berubah terhadap waktu (array 3D/4D: frame, height, width, channel).
- Visualisasi utama: Ekstraksi frame, metadata (fps, resolusi), dan urutan gambar.
- Informasi yang diperoleh: Dinamika gerakan, perubahan visual antar waktu, serta detail spasial dan temporal dalam satu media.

---

## Refleksi Pembelajaran

### 3 Poin yang Saya Pelajari:

1. Cara membaca dan menampilkan metadata serta visualisasi dasar dari data audio, gambar, dan video menggunakan Python.
2. Pentingnya interpretasi visualisasi seperti waveform, histogram warna, dan ekstraksi frame untuk memahami karakteristik media.
3. Perbedaan representasi dan informasi yang bisa diperoleh dari masing-masing jenis data multimedia (audio-temporal, gambar-spasial, video-spatio-temporal).

### 2 Hal yang Masih Membingungkan/Ingin Diperdalam:

1. Cara kerja dari ekstraksi fitur lanjutan seperti MFCC dan bagaimana penggunaannya.
2. Cara memilih parameter optimal untuk ekstraksi fitur audio/gambar/video agar hasil analisis lebih akurat.

---

## Sumber Data & Referensi

**TODO:** Cantumkan semua sumber data dan referensi yang digunakan:

- **Audio:** <https://youtu.be/wjDgiF1AXJw?si=E6wl54iD24U0reNt>
- **Gambar:** Foto Sendiri
- **Video:** Video Sendiri
- **Referensi teknis:** <https://g.co/gemini/share/f4de24a525e8>





## Rubrik Penilaian

### Distribusi Bobot Penilaian

Aspek Penilaian	Bobot	Deskripsi
<b>Kelengkapan</b>	<b>35%</b>	Semua langkah inti dikerjakan sesuai checklist
<b>Kualitas Visualisasi</b>	<b>20%</b>	Judul, label sumbu, colorbar, legend, keterbacaan plot
<b>Analisis &amp; Interpretasi</b>	<b>30%</b>	Kemampuan interpretasi hasil, bukan sekadar output mentah
<b>Kerapihan &amp; Struktur</b>	<b>10%</b>	Markdown jelas, kode modular, dokumentasi baik
<b>Orisinalitas &amp; Penguasaan</b>	<b>5%</b>	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