

Tugas Analisis Multimedia: Audio, Gambar, Video

Mata Kuliah: Sistem & Teknologi Multimedia

Nama: Muhammad Fauzi Azizi

NIM: 122140106

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.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: {plt.__version__}")
print(f"Librosa: {librosa.__version__}")
print(f"OpenCV: {cv2.__version__}")

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

Petunjuk Umum Pengerajan

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: Percakapan orang main ML
- Sumber: GOOD AI STUDIO
- Format file: WAV
- Alasan pemilihan: Mudah di dapat

Path file: suara_percakapan.wav

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.

(Tambahkan sel kode di sini)

```
In [1]: import librosa
import soundfile as sf
import numpy as np
import scipy
import matplotlib.pyplot as plt
from IPython.display import Audio
import os

# Memuat Audio dengan Sistem Fallback
PATH_AUDIO = os.path.join(os.getcwd(), 'suara_percakapan.wav')

# Variabel untuk menyimpan hasil
y = None
sr = None
source_info = ""

# Opsi 1: Coba muat file lokal
try:
    if os.path.exists(PATH_AUDIO):
        y, sr = librosa.load(PATH_AUDIO, sr=None) # sr=None untuk mempertahankan s
        source_info = f"✅ Berhasil memuat: {PATH_AUDIO}"
    else:
        raise FileNotFoundError("File tidak ditemukan")
except:
    print(f"⚠️ Tidak dapat memuat {PATH_AUDIO}")

# Opsi 2: Gunakan contoh bawaan Librosa
try:
    y, sr = librosa.load(librosa.ex('trumpet'), sr=None)
    source_info = "✅ Menggunakan contoh audio bawaan librosa (trumpet)"
except:
    print("⚠️ Contoh librosa tidak tersedia")
```

```
In [2]: ### METADATA

print("📄 METADATA AUDIO")
print("=" * 40)
print(f"⌚ {len(y)} Jumlah sampel: {len(y):,}")
print(f"🎵 Sample rate: {sr:,} Hz")
print(f"⌚ Durasi: {len(y)/sr:.2f} detik")
print(f"🎧 Jumlah kanal: {'Mono (1 kanal)' if y.ndim == 1 else f'Multi-kanal ({y.n
```

```

METADATA AUDIO
=====
12 Jumlah sampel: 641,543
🎵 Sample rate: 24,000 Hz
⌚ Durasi: 26.73 detik
🖨️ Jumlah kanal: Mono (1 kanal)

```

A3. TODO: Waveform

Instruksi: Plot waveform audio dengan:

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

Analisis yang diperlukan:

1. rekaman percakapan yang dilakukan 2 orang tersebut berlangsung selama 26 detik
2. pada gambar waveform terdapat - 0 + pada amplitudonya dimana ketika gelombang pada titik 0 maka percakapan yang berlangsung sedang diam ketika percakapan sampe 30000 maka orang yang berbicara sedang tegas dan ketika percakapan sampe -30000 maka orang berbicara sedang melemah atau suara semakin mengecil.

(Tambahkan sel kode di sini)

In [3]:

```

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

# Reload the audio file
with wave.open(PATH_AUDIO, 'rb') as wav_file:
    sample_rate = wav_file.getframerate()
    num_frames = wav_file.getnframes()
    audio_data = wav_file.readframes(num_frames)
    num_channels = wav_file.getnchannels()
    sampwidth = wav_file.getsampwidth()

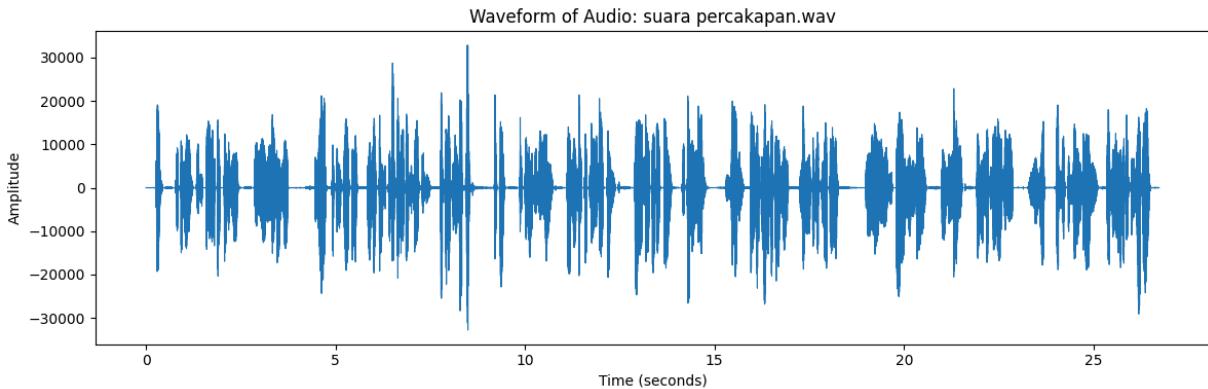
# Convert audio bytes to numpy array
audio_array = np.frombuffer(audio_data, dtype=np.int16)

# Create time axis
time_axis = np.linspace(0, num_frames / sample_rate, num=num_frames)

# Plot waveform
plt.figure(figsize=(12, 4))
plt.plot(time_axis, audio_array, linewidth=0.8)
plt.title("Waveform of Audio: suara_percakapan.wav")
plt.xlabel("Time (seconds)")
plt.ylabel("Amplitude")

```

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



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: Spectrogram ini menggambarkan sinyal audio dengan energi dominan di frekuensi rendah–menengah (0–4 kHz), dengan harmonik sampai 12 kHz. Sinyal cenderung kontinu dan berulang (terlihat dari pita vertikal berulang). dengan warna kuning kuat , hijau sedang dan biru lemah.

(Tambahkan sel kode di sini)

```
In [4]: import matplotlib.pyplot as plt
import numpy as np
import wave
from scipy.signal import stft

# Load audio dengan wave
with wave.open(PATH_AUDIO, 'rb') as wav_file:
    sample_rate = wav_file.getframerate()
    num_frames = wav_file.getnframes()
    audio_data = wav_file.readframes(num_frames)
    num_channels = wav_file.getnchannels()

# Konversi audio ke numpy array
signal = np.frombuffer(audio_data, dtype=np.int16)

# Jika stereo, ambil hanya 1 channel
if num_channels > 1:
    signal = signal[:, 0]

# Parameter STFT
n_fft = 1024          # Ukuran FFT window
```

```

hop_length = 256      # Langkah antar frame
window = 'hann'       # Window function

# Hitung STFT dengan scipy
f, t, Zxx = stft(signal, fs=sample_rate, window=window, nperseg=n_fft, noverlap=n_f
magnitude = np.abs(Zxx)
magnitude_db = 20 * np.log10(magnitude + 1e-10) # konversi ke dB, tambah epsilon b

# Plot Spectrogram
plt.figure(figsize=(12, 6))
plt.pcolormesh(t, f, magnitude_db, shading='gouraud', cmap='viridis')
plt.title("▣ Spectrogram - Magnitude (dB)", fontsize=14, fontweight='bold')
plt.xlabel("Waktu (detik)")
plt.ylabel("Frekuensi (Hz)")
cbar = plt.colorbar(format='%+2.0f dB')
cbar.set_label("Magnitudo (dB)")
plt.tight_layout()
plt.show()

# Informasi STFT
print("▣ INFORMASI STFT")
print("=" * 40)
print(f"▣ Ukuran FFT: {n_fft}")
print(f"▣ Hop length: {hop_length}")
print(f"▣ Window function: {window}")
print(f"▣ Shape magnitude: {magnitude.shape}")
print(f"▣ Resolusi waktu: {hop_length/sample_rate*1000:.1f} ms per frame")
print(f"▣ Resolusi frekuensi: {sample_rate/n_fft:.1f} Hz per bin")
print(f"▣ Range magnitude (dB): {magnitude_db.min():.1f} - {magnitude_db.max():.1f} dB")

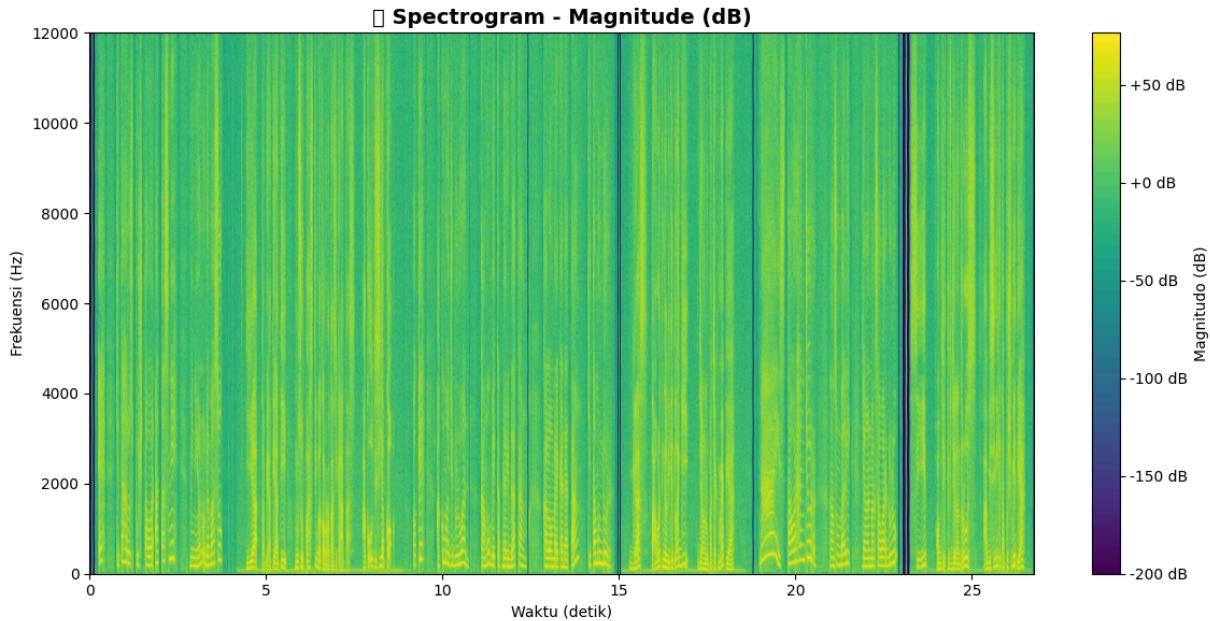
```

C:\Users\USER\AppData\Local\Temp\ipykernel_30956\430409983.py:38: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.

 plt.tight_layout()

c:\Users\USER\AppData\Local\Programs\Python\Python313\Lib\site-packages\IPython\core\pylabtools.py:170: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.

 fig.canvas.print_figure(bytes_io, **kw)



INFORMASI STFT

```

 Ukuran FFT: 1024
 Hop length: 256
 Window function: hann
 Shape magnitude: (513, 2508)
 Resolusi waktu: 10.7 ms per frame
 Resolusi frekuensi: 23.4 Hz per bin
 Range magnitude (dB): -200.0 – 76.5

```

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:

1. pada sumbu x = waktu yang berdurasi 26detik dan sumbu y = koefisien yang berentang 1 sampai 13
2. pada gambar terlihat berubah rubah karena semua intonasi kata dari orang berbicara berbeda yang membuat warna yang berbeda
3. setiap baris mempresentasikan spektrum suara yang berbeda-beda : pada koefesien 1-3 : menangkap informasi dan format pelafalan huruf. pada koefesien 10-13 : lebih menangkap nada frekuensi dari percakapan 2 orang

(Tambahkan sel kode di sini)

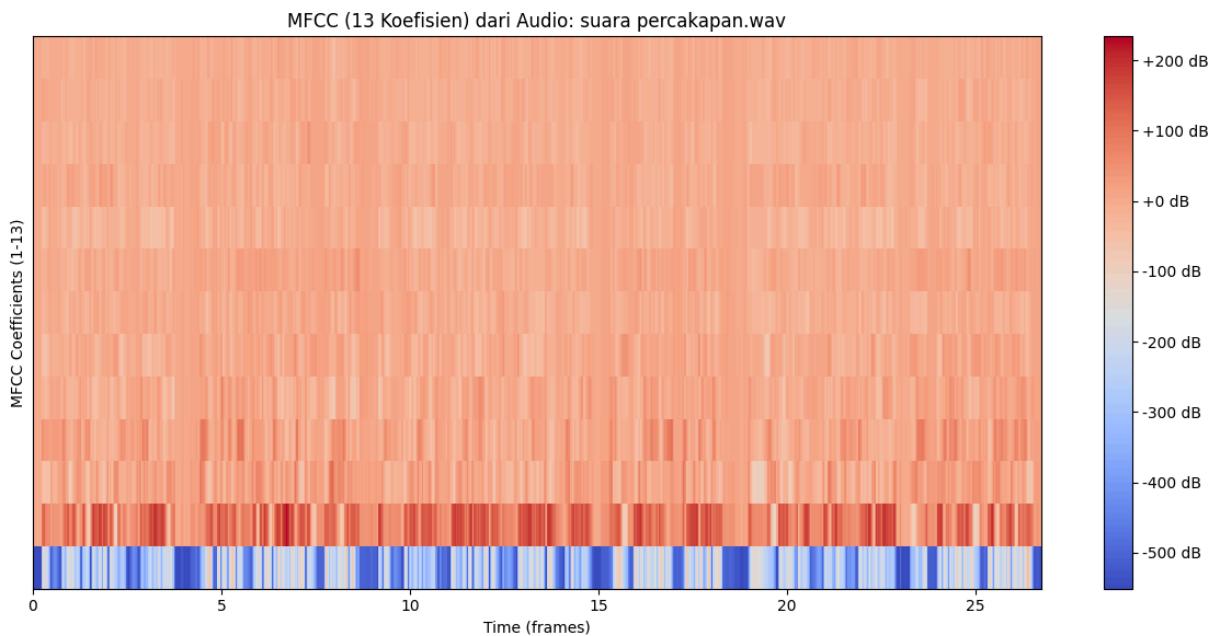
```
In [5]: import matplotlib.pyplot as plt
import numpy as np
import librosa
import librosa.display

# Load audio dengan Librosa
y, sr = librosa.load(PATH_AUDIO, sr=None)

# Hitung MFCC (minimal 13 koefisien)
mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)

# Plot heatmap MFCC
plt.figure(figsize=(12, 6))
librosa.display.specshow(mfccs, x_axis='time', sr=sr)
plt.colorbar(format="%+2.0f dB")
plt.title("MFCC (13 Koefisien) dari Audio: suara percakapan.wav")
plt.xlabel("Time (frames)")
plt.ylabel("MFCC Coefficients (1-13)")
plt.tight_layout()
plt.show()
```

```
mfccs.shape
```



```
Out[5]: (13, 1254)
```

A6. Analisis Ringkas (Wajib)

Jawab pertanyaan berikut:

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

Jawaban Anda: dari waveform kita hanya bisa tahu seberapa keras dan kecil suara yang dikeluarkan sedangkan dari spectrogram kita tahu suara dengan frequency apa yang muncul atau yang dominan dalam sebuah percakapan dan spectrogram lebih kompleks dari waveform.

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

Jawaban Anda: pada mfcc sekarang saya tahu setiap audio koefisien dari yang berbeda" dan setiap koefisien memiliki maknanya masing-masing dimana terdapat beberapa makna yaitu memberikan informasi intonasi kata atau huruf sampai suara keras frequency yang dihasilkan

Bagian B — Gambar

B1. Deskripsi Data

TODO: Jelaskan gambar yang akan Anda analisis:

- Jenis gambar: Foto satwa dengan latar pemandangan
- Sumber: foto saya sendiri
- Format file: JPG
- Alasan pemilihan: Bagus

Path file: `saya.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:

1. Objek dominan: Seseorang duduk di batang pohon kering di gunung, menghadap ke arah danau yang terlihat di bawah.
2. Kondisi pencahayaan: Cahaya alami siang hari, cukup terang dengan sinar matahari langsung. Kontras jelas antara objek manusia, pepohonan, dan latar belakang.
3. Komposisi warna: Didominasi oleh biru (langit dan danau), hijau (vegetasi di sekitar), serta putih (awan). Warna pakaian hitam dan abu-abu pada objek manusia memberikan kontras kuat dengan latar alam yang cerah.

(Tambahkan sel kode di sini)

In [6]:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PIL import Image
import cv2
import numpy as np
import os

# Load Gambar dengan Sistem Fallback
PATH_IMAGE = os.path.join(os.getcwd(), 'saya.jpg') # Ganti dengan path gambar Anda

# Variabel untuk menyimpan hasil
img_pil = None
source_info = ""

# Opsi 1: Coba muat file lokal dengan PIL
try:
    if os.path.exists(PATH_IMAGE):
        img_pil = Image.open(PATH_IMAGE)

        # Pastikan dalam mode RGB (bukan RGBA atau mode lain)
```

```

        if img_pil.mode != 'RGB':
            img_pil = img_pil.convert('RGB')
            source_info = f"✅ Berhasil memuat: {PATH_IMAGE}"
        else:
            raise FileNotFoundError("File tidak ditemukan")
    except Exception as e:
        print(f"⚠️ Tidak dapat memuat {PATH_IMAGE}: {e}")

# Opsi 2: Gunakan gambar contoh dari matplotlib
try:
    # Membuat gambar contoh sederhana (gradient warna)
    height, width = 200, 300
    img_array = np.zeros((height, width, 3), dtype=np.uint8)

    # Buat gradient horizontal RGB
    for i in range(width):
        img_array[:, i, 0] = int(255 * i / width)      # Red channel
        img_array[:, i, 1] = int(255 * (1 - i / width)) # Green channel
        img_array[:, i, 2] = 128                         # Blue channel (konstan)

    img_pil = Image.fromarray(img_array)
    source_info = "✅ Menggunakan gambar contoh buatan (gradient RGB)"

except Exception as e:
    print(f"❌ Error membuat gambar contoh: {e}")
    # Buat gambar solid sederhana sebagai fallback terakhir
    img_array = np.full((100, 100, 3), [255, 128, 0], dtype=np.uint8) # Orange
    img_pil = Image.fromarray(img_array)
    source_info = "✅ Menggunakan gambar fallback (solid orange)"

# 🖼️ Tampilkan informasi
print(source_info)
print(f"📊 Ukuran gambar: {img_pil.size} (lebar x tinggi)")
print(f"🎨 Mode warna: {img_pil.mode}")

# ✅ Tambahkan: Tampilkan gambar
plt.imshow(img_pil)
plt.axis('off') # Hilangkan sumbu
plt.title("Tampilan Gambar: saya.jpg" if "saya.jpg" in source_info else "Tampilan G")
plt.show()

```

- ✅ Berhasil memuat: d:\Mulmed\Worksheet2\saya.jpg
- 📊 Ukuran gambar: (3024, 4032) (lebar x tinggi)
- 🎨 Mode warna: RGB

Tampilan Gambar: saya.jpg



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: Penting karena untuk melanjutkan ketahap processing kita harus tau berapa jumlah kanal 1 channel apa 3 channel , tipe data yang digunakan apa , mode warna yang digunakan apa dan ukuran file yang digunakan besar apa kecil agar kita bisa transfer data cepat.

(Tambahkan sel kode di sini)

In [7]:

```
# Buka gambar dengan PIL
img = Image.open(PATH_IMAGE)
img_array = np.array(img)

# Ambil metadata
height, width = img.size[1], img.size[0]
channels = img_array.shape[2] if len(img_array.shape) == 3 else 1
dtype = img_array.dtype
```

```
mode = img.mode
file_size = os.path.getsize(PATH_IMAGE) / 1024 # KB

{
    "Dimensi (Height × Width)": f"{height} × {width}",
    "Jumlah Kanal": channels,
    "Tipe Data (dtype)": str(dtype),
    "Mode Warna": mode,
    "Ukuran File": f"{file_size:.2f} KB"
}
```

```
Out[7]: {'Dimensi (Height × Width)': '4032 × 3024',
         'Jumlah Kanal': 3,
         'Tipe Data (dtype)': 'uint8',
         'Mode Warna': 'RGB',
         'Ukuran File': '2737.80 KB'}
```

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 Dominan:

1. Biru (B) jelas dominan dengan puncak tinggi di intensitas 240 ini berasal dari langit cerah dan air danau.
2. Hijau (G) juga cukup tinggi di area intensitas menengah 210 ini berasal dari vegetasi dan pepohonan.
3. Merah (R) relatif lebih rendah, hanya menyumbang dari tanah, batang pohon, dan pakaian orang.

Sebaran Intensitas:

1. Histogram melebar dari intensitas rendah hingga tinggi menunjukkan gambar punya kontras yang baik.
2. Ada puncak signifikan di area tinggi cerah untuk channel biru sesuai dengan kondisi pencahayaan terang siang hari dengan langit cerah.
3. Distribusi hijau dan merah lebih rata, tapi tetap cenderung ke arah menengah-tinggi.

Kontras Gambar: Karena ada distribusi yang luas dari gelap (0–50) hingga terang (200–255), gambar memiliki dinamika cahaya yang kaya. Hal ini membuat detail bayangan, vegetasi, dan langit tetap terlihat jelas tanpa overexposure.

(Tambahkan sel kode di sini)

```
In [8]: # Baca gambar dengan OpenCV (BGR)
img_bgr = cv2.imread(PATH_IMAGE)
# Konversi ke RGB
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

# Hitung histogram untuk setiap channel
colors = ('r', 'g', 'b')
plt.figure(figsize=(10, 5))

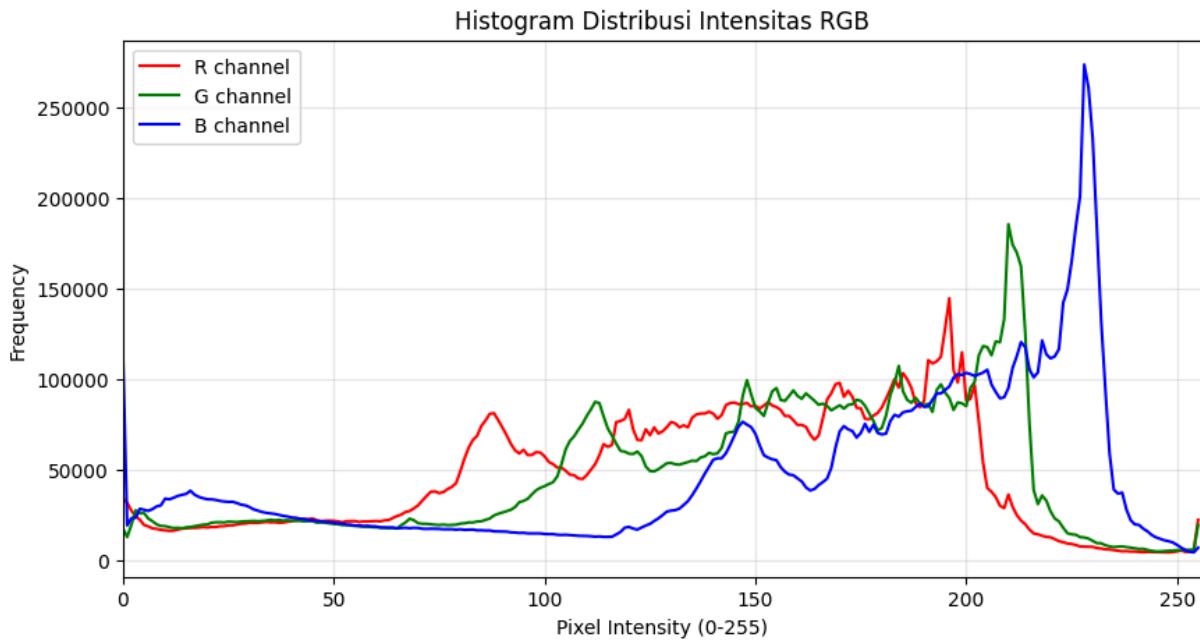
for i, col in enumerate(colors):
    hist = cv2.calcHist([img_rgb], [i], None, [256], [0, 256])
    plt.plot(hist, color=col, label=f'{col.upper()} channel')
    plt.xlim([0, 256])

# Label dan Legend
```

```

plt.title('Histogram Distribusi Intensitas RGB')
plt.xlabel('Pixel Intensity (0-255)')
plt.ylabel('Frequency')
plt.legend()
plt.grid(alpha=0.3)
plt.show()

```



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: Dari data atau pola histogram yang saya lihat, saya dapat tahu bahwa warna yang lebih dominan itu apa , dan kontraktsnya baik

Bagian C — Video

C1. Deskripsi Data

TODO: Jelaskan video yang akan Anda analisis:

- Jenis video: Pemandangan
- Sumber: Rekaman sendiri
- Durasi target: 15 detik
- Alasan pemilihan: Pemandangan yang bagus

Path file: pemandangan.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:

1. Resolusi (Width × Height): Menentukan tingkat detail visual. Resolusi tinggi (seperti 4K) penting untuk aplikasi computer vision presisi tinggi (misalnya deteksi objek kecil, surveilans, atau penginderaan lingkungan).
2. Frame Rate (fps): Mempengaruhi kelancaran gerakan. 30 fps: cukup untuk video standar. 60 fps (seperti video ini): ideal untuk menangkap gerakan cepat dengan detail lebih halus (gaming, olahraga, drone, dll.).
3. Jumlah Frame & Durasi: Berguna untuk segmentasi temporal, misalnya jika ingin ekstraksi potongan adegan, analisis gerakan, atau sinkronisasi dengan audio.
4. Klasifikasi Resolusi: Membantu standarisasi. Misalnya: HD/Full HD: cukup untuk streaming biasa. 4K: lebih cocok untuk konten sinematik atau analisis detail.

Memilih resolusi juga berhubungan langsung dengan kebutuhan memori dan bandwidth.

(Tambahkan sel kode di sini)

In [9]:

```
import cv2
import os

# Path video
PATH_VIDEO = os.path.join(os.getcwd(), 'pemandangan.MOV')

# Baca video dengan OpenCV
cap = cv2.VideoCapture(PATH_VIDEO)

# 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)
frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
duration = frame_count / fps if fps > 0 else 0

# Klasifikasi resolusi
if width == 1280 and height == 720:
    res_class = "HD (720p)"
elif width == 1920 and height == 1080:
    res_class = "Full HD (1080p)"
```

```

    elif width == 3840 and height == 2160:
        res_class = "4K (2160p)"
    elif width == 7680 and height == 4320:
        res_class = "8K (4320p)"
    else:
        res_class = f"Custom ({width}x{height})"

cap.release()

{
    "Resolusi (Width x Height)": f"{width} x {height}",
    "Frame Rate (fps)": fps,
    "Jumlah Total Frame": frame_count,
    "Durasi (detik)": duration,
    "Klasifikasi Resolusi": res_class
}

```

Out[9]: {'Resolusi (Width x Height)': '3840 x 2160',
 'Frame Rate (fps)': 59.94088669950739,
 'Jumlah Total Frame': 1014,
 'Durasi (detik)': 16.916666666666668,
 'Klasifikasi Resolusi': '4K (2160p)'}

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

Instruksi: Ambil dan tampilkan 3 frame representatif:

- Frame pertama (index 0)
- Frame tengah (index ~total_frame/2)
- Frame terakhir (index total_frame-1)
- **Konversi BGR→RGB** sebelum ditampilkan
- Subplot dengan judul frame dan timestamp

Analisis yang diperlukan:

1. Pada frame 0 warna daun masih terlihat karena masih ada cahaya
2. pada frame 507 warna daun sudah mulai tidak terlihat karena sudah kurang pencahayaan
3. pada frame 1013 warna daun sudah hitam karena pencahayaan sudah tidak ada

{}

(Tambahkan sel kode di sini)

In [10]: # Buka video dengan OpenCV
cap = cv2.VideoCapture(PATH_VIDEO)

Ambil metadata
frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
fps = cap.get(cv2.CAP_PROP_FPS)

Tentukan indeks frame yang diambil

```

frame_indices = [0, frame_count // 2, frame_count - 1]
frames = []
timestamps = []

for idx in frame_indices:
    cap.set(cv2.CAP_PROP_POS_FRAMES, idx)
    ret, frame = cap.read()
    if ret:
        # Konversi BGR → RGB
        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        frames.append(frame_rgb)
        timestamps.append(idx / fps)

cap.release()

# Plot dalam subplot
plt.figure(figsize=(18, 6))
for i, (frame, ts, idx) in enumerate(zip(frames, timestamps, frame_indices)):
    plt.subplot(1, 3, i+1)
    plt.imshow(frame)
    plt.axis('off')
    plt.title(f"Frame {idx} (t = {ts:.2f}s)")

plt.suptitle("Sampel Frame dari Video: pemandangan.MOV", fontsize=16, fontweight="bold")
plt.tight_layout()
plt.show()

```

Sampel Frame dari Video: pemandangan.MOV



(Tambahkan sel kode di sini)

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: Tidak sesuai karena tidak semua platform bisa mengupload 4k , ketika saya mencoba upload ke media sosial seperti WhatsApp maka di compress videonya menjadi kecil yang mengakibatkan videonya menjadi jelek atau bahasa gaulnya buruk

Perbandingan & Kesimpulan

Perbandingan Representasi Media

TODO: Bandingkan secara ringkas representasi dan visualisasi ketiga media:

Audio (1D - Temporal)

- Representasi: nilai untuk setiap amplitudo terhadap waktu yang sering berubah rubah
- Visualisasi utama: waveform dan spectrogram bisa digunakan untuk analisis
- Informasi yang diperoleh: bisa mendapatkan informasi tentang intensitas suara, pola temporal , dan frequency yang digunakan untuk analisis nada suara atau percakapan orang

Gambar (2D - Spasial)

- Representasi: Matriks piksel dua dimensi biasanya dalam format RGB, di mana setiap nilai piksel merepresentasikan intensitas cahaya atau warna.
- Visualisasi utama:foto image display yang ditampilkan di layar sebagai gambar statis.
- Informasi yang diperoleh:bisa mendapatkan informasi spasial seperti bentuk warna, tekstur, pola dan struktur objek yang digunakan untuk pengenalan objek

Video (2D + Waktu - Spatio-temporal)

- Representasi:Urutan frame dua dimensi yang ditampilkan secara berurutan terhadap waktu. biasanya direpresentasikan sebagai tensor 3D atau 4D
- Visualisasi utama: klip video
- Informasi yang diperoleh: Informasi spasial bentuk, warna, objek dan temporal gerakan, perubahan, aktivitas digunakan untuk analisis gerakan

Refleksi Pembelajaran

3 Poin yang Saya Pelajari:

1. bisa menampilkan audio , gambar , vidio dalam bentuk kode
2. bisa menganalisis waveform dan spectrogram dari audio yang ada
3. bisa menganalisis RBG dari gambar yang ada

2 Hal yang Masih Membingungkan/Ingin Diperdalam:

1. pengen mendalami gimana cara membuat smooth video dari kode
2. pengen mendalami gimana caranya awalnya video itu terbentuk

Sumber Data & Referensi

TODO: Cantumkan semua sumber data dan referensi yang digunakan:

- **Audio:** GOOGLE AI STUDIO
- **Gambar:** Foto saya sendiri
- **Video:** Rekaman saya sendiri
- **Referensi teknis:** GPT link (<https://chatgpt.com/share/68c71f21-7558-8006-9bdc-c352dbbe9391>) dan modul mulmed

Rubrik Penilaian

Distribusi Bobot Penilaian

Aspek Penilaian	Bobot	Deskripsi
Kelengkapan	35%	Semua langkah inti dikerjakan sesuai checklist
Kualitas Visualisasi	20%	Judul, label sumbu, colorbar, legend, keterbacaan plot
Analisis & Interpretasi	30%	Kemampuan interpretasi hasil, bukan sekadar output mentah
Kerapuhan & Struktur	10%	Markdown jelas, kode modular, dokumentasi baik
Orientalitas & 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, terkompromimen, dan mudah dibaca
- Dokumentasi yang memadai
- Flow logical dari satu bagian ke bagian lain

Originalitas & 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: $0.8 * \text{penilaian dosen} + \text{nilai waktu pengumpulan}$

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