

## ✓ Tugas Analisis Multimedia: Audio, Gambar, Video

**Mata Kuliah:** Sistem & Teknologi Multimedia

**Nama:** Rafki Haykhal Alif

**NIM:** 122140035

### 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.

```
# Import Library (Satu-satunya sel kode dalam template ini)
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
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__}")
print(f"Librosa: {librosa.__version__}")
print(f"OpenCV: {cv2.__version__}")

# Tambahkan import lain jika diperlukan saat mengerjakan tugas

Library versions:
NumPy: 2.0.2
Matplotlib: 3.10.0
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: Suara Mobil Ferrari
- Sumber: <https://pixabay.com/id/sound-effects/search/mobil/>
- Format file: WAV
- Alasan pemilihan: Suara mobil ferrari memiliki karakteristik suara yang unik dibandingkan dengan suara mobil lain.

**Path file:** `data/ferrari-testarossa.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.

```
# Path file audio
audio_path = '/data/ferrari-testarossa-sounds-384731.mp3'

# Muat file audio dengan librosa
y, sr = librosa.load(audio_path)

# Ekstrak metadata
duration = librosa.get_duration(y=y, sr=sr)
num_channels = y.ndim
total_samples = len(y)

# Output metadata
print("--- Metadata Audio ---")
print(f"Sample Rate: {sr} Hz")
print(f"Durasi: {duration:.2f} detik")
print(f"Jumlah Kanal: {'Mono' if num_channels == 1 else 'Stereo'}")
print(f"Jumlah Total Sampel: {total_samples}")

# Play audio
display(Audio(data=y, rate=sr))

--- Metadata Audio ---
Sample Rate: 22050 Hz
Durasi: 51.41 detik
Jumlah Kanal: Mono
Jumlah Total Sampel: 1133547

0:03 / 0:51
```

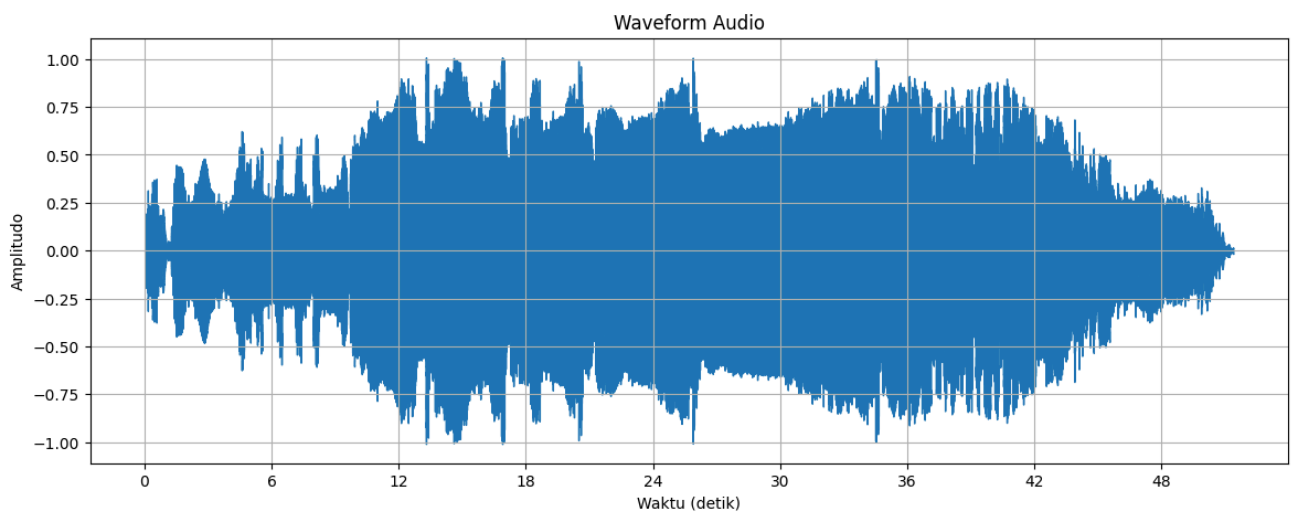
### ✓ A3. TODO: Waveform

**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.)

```
# Plot waveform
plt.figure(figsize=(14, 5))
librosa.display.waveshow(y, sr=sr)
plt.title('Waveform Audio')
plt.xlabel('Waktu (detik)')
plt.ylabel('Amplitudo')
plt.grid(True)
plt.show()
```



#### Analisis Waveform :

Dari waveform suara mobil yang dihasilkan bisa dilihat bahwa pada 10 detik awal nilai amplitudo yang dihasilkan kecil yang menandakan volume suara nya kecil dan di detik 12 keatas amplitudonya bernilai besar yang menandakan mobil berkecepatan tinggi pada detik 12 keatas karena menghasilkan suara yang besar. Nilai amplitudo mengecil lagi pada detik 42 keatas yang menandakan mobil sudah berhenti.

### ✓ A4. TODO: Spectrogram log-dB

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

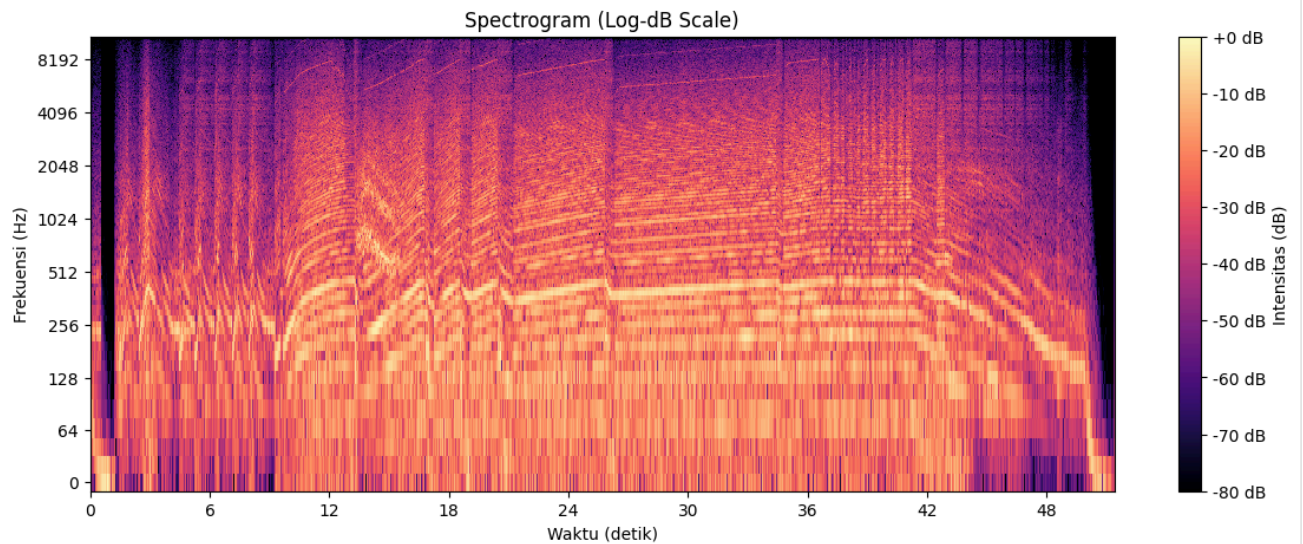
- Gunakan parameter standar ( $n_{\text{fft}}=1024$ ,  $\text{hop\_length}=256$ )
- Tampilkan dengan colorbar
- Label sumbu: waktu (detik) dan frekuensi (Hz)

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

```
# Hitung STFT
D = librosa.stft(y, n_fft=1024, hop_length=256)

# Konversi ke skala log-dB
S_db = librosa.amplitude_to_db(np.abs(D), ref=np.max)

# Tampilkan spectrogram
plt.figure(figsize=(14, 5))
librosa.display.specshow(S_db, sr=sr, x_axis='time', y_axis='log', hop_length=256)
plt.colorbar(format='%+2.0f dB', label='Intensitas (dB)')
plt.title('Spectrogram (Log-dB Scale)')
plt.xlabel('Waktu (detik)')
plt.ylabel('Frekuensi (Hz)')
plt.show()
```



#### Analisis Perbedaan waveform dan spectrogram :

Perbedaan antara waveform dan spectrogram pada suara mobil terlihat jelas dalam cara informasi ditampilkan. Waveform hanya menunjukkan perubahan amplitudo dari waktu ke waktu, sehingga bisa melihat kapan suara mobil terdengar pelan, semakin keras ketika berakselerasi, lalu mengecil kembali saat berhenti. Namun, waveform tidak memperlihatkan detail frekuensi suara mesin mobil. Sebaliknya, spectrogram menampilkan distribusi frekuensi terhadap waktu, sehingga pola getaran khas mesin mobil (frekuensi rendah yang dominan saat idle, lalu muncul komponen frekuensi lebih tinggi saat akselerasi) bisa terlihat dengan jelas. Jadi, waveform membantu memahami dinamika volume suara mobil, sementara spectrogram memberikan gambaran lebih rinci tentang karakteristik frekuensinya.

#### ✓ A5. TODO: MFCC

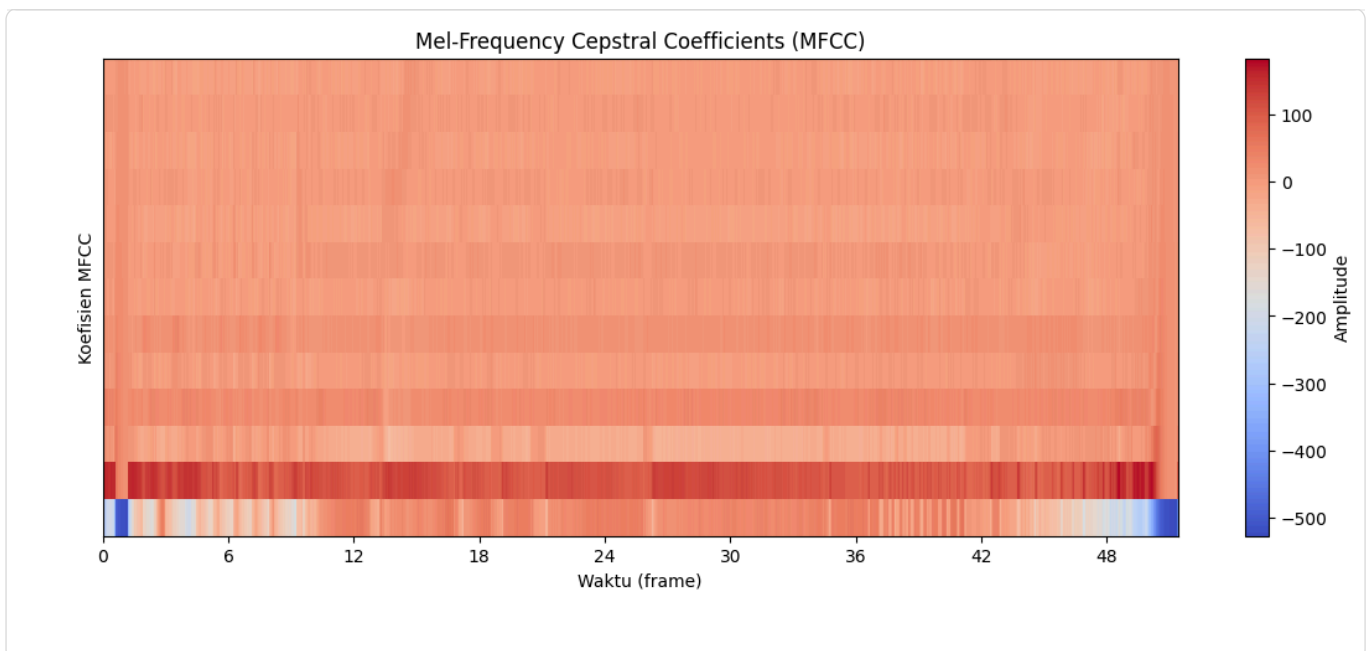
**Instruksi:** Hitung dan tampilkan minimal 13 koefisien MFCC sebagai heatmap:

- 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?

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

# Tampilkan MFCC sebagai heatmap
plt.figure(figsize=(14, 5))
librosa.display.specshow(mfccs, x_axis='time')
plt.colorbar(label='Amplitude')
plt.title('Mel-Frequency Cepstral Coefficients (MFCC)')
plt.xlabel('Waktu (frame)')
plt.ylabel('Koefisien MFCC')
plt.show()
```



#### Analisis MFCC :

Dari hasil visualisasi MFCC, terlihat bahwa pola koefisien relatif stabil dengan perubahan yang tidak terlalu drastis sepanjang waktu, meskipun ada sedikit variasi pada beberapa bagian. Stabilitas ini menunjukkan bahwa karakteristik suara mobil cenderung konsisten, sesuai dengan suara mesin yang menghasilkan bunyi berulang dan teratur. Namun, variasi kecil pada pola MFCC menandakan adanya perubahan kondisi, misalnya saat mobil berakselerasi, melambat, atau berhenti.

### A6. Analisis Ringkas (Wajib)

#### Jawab pertanyaan berikut:

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

*Jawaban Anda:* Waveform menampilkan perubahan amplitudo terhadap waktu sehingga lebih menekankan pada dinamika volume suara mobil, misalnya kapan suara terdengar pelan, keras, atau mereda. Sementara itu, spectrogram memberikan informasi tambahan berupa distribusi frekuensi terhadap waktu dengan intensitas tertentu. Dari spectrogram dapat terlihat bagaimana frekuensi rendah hingga tinggi muncul selama mobil berakselerasi maupun saat berhenti, sehingga memberikan gambaran yang lebih detail tentang karakteristik suara mesin dibandingkan waveform.

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

*Jawaban Anda:* Visualisasi MFCC menunjukkan pola koefisien yang relatif stabil dengan sedikit variasi, menandakan bahwa suara mobil memiliki karakteristik spektral yang konsisten sesuai dengan bunyi mesin yang ritmis. Variasi kecil pada MFCC menggambarkan perubahan kondisi seperti akselerasi atau perlambatan. Hal ini menunjukkan bahwa MFCC efektif digunakan untuk menangkap ciri khas suara mesin mobil yang bisa dijadikan fitur dalam analisis lebih lanjut, misalnya untuk identifikasi atau klasifikasi suara kendaraan.

## ✓ Bagian B — Gambar

### B1. Deskripsi Data

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

- Jenis gambar: Pemandangan sunset di tepi laut.
- Sumber: Foto sendiri
- Format file: jpg
- Alasan pemilihan: Pemandangan sunset memiliki kombinasi warna antara terang dan gelap.

**Path file:** `data/sunset.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).

```
# Path file gambar
image_path = '/data/sunset.jpg'

# Baca gambar dengan OpenCV
img_bgr = cv2.imread(image_path)

# Periksa apakah gambar berhasil dibaca sebelum konversi
if img_bgr is None:
    print(f"Error: Gagal membaca gambar dari {image_path}. Pastikan file ada dan path benar.")
else:
    # Konversi dari BGR ke RGB
    img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

    # Tampilkan gambar
    plt.figure(figsize=(8, 6))
    plt.imshow(img_rgb)
    plt.title('Gambar Pemandangan Sunset di Tepi Pantai')
    plt.axis('off') # Hilangkan sumbu x dan y
    plt.show()
```

Gambar Pemandangan Sunset di Tepi Pantai



#### Analisi Gambar :

Gambar pemandangan sunset di tepi pantai ini menunjukkan bahwa objek dominan dalam foto adalah matahari yang hampir tenggelam di ujung laut dengan siluet perahu di bagian bawah gambar, memberikan fokus utama pada suasana sore hari. Kondisi pencahayaan terlihat alami dengan cahaya matahari keemasan yang menyinari langit, menciptakan efek siluet pada objek di depannya. Komposisi warna didominasi oleh gradasi oranye, kuning, dan emas di sekitar matahari, berpadu dengan biru keabu-abuan pada langit serta hitam gelap pada perahu dan garis pantai.

### ✓ 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.

```
# Ekstrak informasi dasar
height, width, channels = img_rgb.shape
dtype = img_rgb.dtype
size_in_memory = img_rgb.nbytes / 1024 # dalam KB
```

```
# Tampilkan informasi
print("--- Informasi Dasar Gambar ---")
print(f"Dimensi: {height} x {width} piksel")
print(f"Jumlah Kanal: {channels}")
print(f"Tipe Data (dtype): {dtype}")
print(f"Mode Warna: RGB")
print(f"Ukuran di Memori: {size_in_memory:.2f} KB")
```

```
--- Informasi Dasar Gambar ---
Dimensi: 4032 x 3024 piksel
Jumlah Kanal: 3
Tipe Data (dtype): uint8
Mode Warna: RGB
Ukuran di Memori: 35721.00 KB
```

#### Analisis informasi dasar gambar :

Informasi dasar gambar seperti dimensi ( $4032 \times 3024$  piksel), jumlah kanal (3 untuk RGB), tipe data (uint8), serta ukuran memori sangat penting pada tahap preprocessing maupun analisis lanjutan. Dengan mengetahui dimensi, sehingga dapat menyesuaikan ukuran gambar (resize) agar lebih efisien untuk pemrosesan. Jumlah kanal menunjukkan mode warna yang dipakai, sehingga mengetahui apakah gambar perlu dikonversi (misalnya ke grayscale). Tipe data menentukan bagaimana nilai piksel diolah, apakah perlu normalisasi atau konversi tipe agar sesuai dengan algoritma. Sementara ukuran memori membantu memperkirakan kebutuhan komputasi dan penyimpanan, terutama saat menangani dataset gambar dalam jumlah besar. Semua informasi ini menjadi dasar dalam menentukan langkah preprocessing yang optimal sebelum analisis lebih lanjut.

## ✓ 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?

Klik dua kali (atau tekan Enter) untuk mengedit

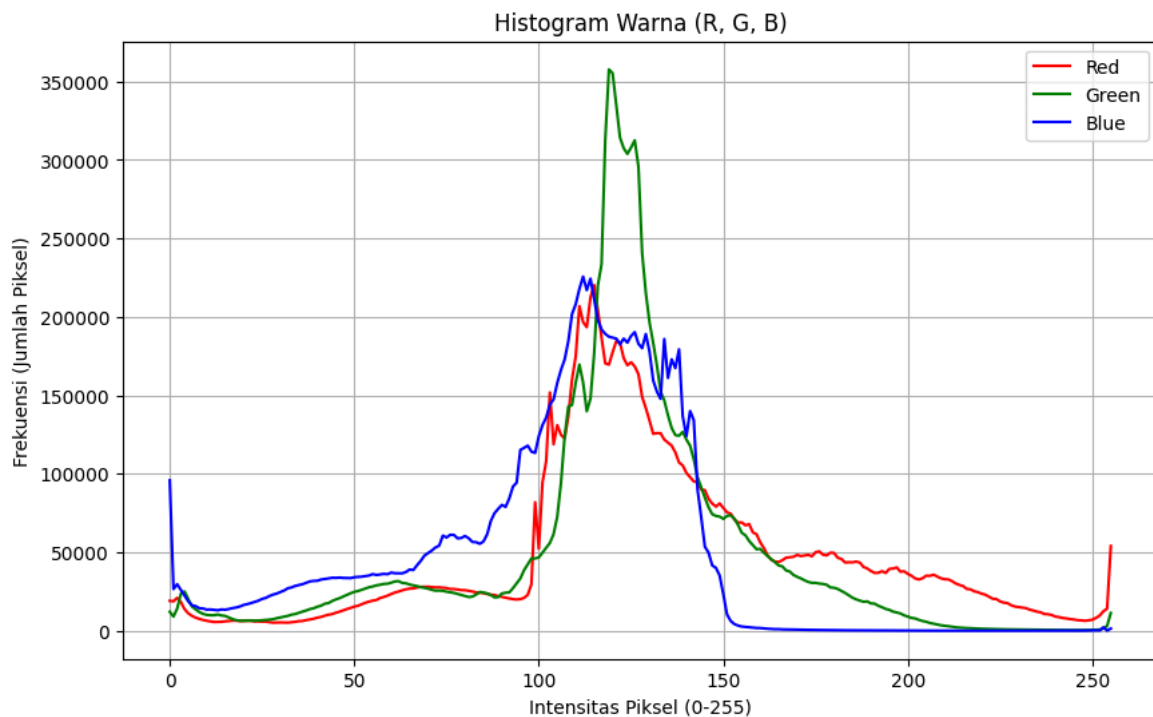
```
# Siapkan warna dan label untuk plot
colors = ('r', 'g', 'b')
labels = ('Red', 'Green', 'Blue')

plt.figure(figsize=(10, 6))
plt.title('Histogram Warna (R, G, B)')
plt.xlabel('Intensitas Piksel (0-255)')
plt.ylabel('Frekuensi (Jumlah Piksel)')

# Hitung dan plot histogram untuk setiap channel
for i, color in enumerate(colors):
    # cv2.calcHist([image], [channel_index], mask, [bins], [range])
    hist = cv2.calcHist([img_rgb], [i], None, [256], [0, 256])
    plt.plot(hist, color=color, label=labels[i])

plt.legend()
plt.grid(True)
plt.show()
```





#### Analisis Histogram Warna :

Dari histogram warna terlihat bahwa channel hijau (Green) lebih dominan dibandingkan channel merah (Red) dan biru (Blue), terutama dengan puncak intensitas tinggi di sekitar nilai 120–130. Hal ini sesuai dengan gambar sunset yang banyak memuat warna langit dan awan bercampur dengan nuansa keemasan serta gradasi alami. Channel merah juga cukup kuat, terutama pada rentang intensitas tinggi, sehingga mendukung warna hangat dari cahaya matahari. Channel biru relatif lebih menyebar dengan distribusi signifikan di intensitas rendah hingga menengah, menggambarkan area langit biru dan bayangan.

Kontras gambar cukup baik karena distribusi intensitas tidak hanya terkonsentrasi di satu titik, tetapi menyebar dari gelap (rendah) hingga terang (tinggi). Sebaran intensitas menunjukkan adanya variasi piksel pada area gelap (siluet perahu, awan tebal), area sedang (langit), hingga area terang (pantulan cahaya matahari), sehingga menghasilkan komposisi visual yang seimbang dan dramatis.

#### 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 warna hijau dan merah lebih dominan dibanding biru, sehingga mendukung kesan hangat pada gambar sunset. Sebaran intensitas yang merata dari gelap hingga terang menghasilkan kontras yang baik, sesuai dengan kesan visual gambar yang menampilkan siluet gelap, langit berwarna keemasan, dan area terang dari cahaya matahari.

## Bagian C — Video

### C1. Deskripsi Data

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

- Jenis video: Pemandangan
- Sumber: Rekaman Sendiri
- Durasi target: 11 detik
- Alasan pemilihan: Video Pemandangan dengan kualitas yang cukup bagus.

**Path file:** `data/Pemandangan.mov`

### 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.

```
# Path file video
video_path = '/data/Pemandangan.mov'

# Buka video dengan OpenCV
cap = cv2.VideoCapture(video_path)

# Periksa apakah video berhasil dibuka
if not cap.isOpened():
    print(f"Error: Gagal membuka file video dari {video_path}. Pastikan file ada dan path benar.")
else:
    # Ekstrak 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))

    # Hitung durasi hanya jika fps > 0
    duration = 0
    if fps > 0:
        duration = frame_count / fps
    else:
        print("Warning: Frame rate (fps) adalah 0 atau tidak valid, durasi tidak dapat dihitung.")

    # Klasifikasi resolusi sederhana
    if height >= 2160:
        resolution_class = "4K"
    elif height >= 1080:
        resolution_class = "Full HD"
    elif height >= 720:
        resolution_class = "HD"
    else:
        resolution_class = "SD"

    # Tampilkan metadata
    print("--- Metadata Video ---")
    print(f"Resolusi: {width} x {height}")
    print(f"Frame Rate (FPS): {fps:.2f}")
    print(f"Jumlah Total Frame: {frame_count}")
    print(f"Durasi: {duration:.2f} detik")
    print(f"Klasifikasi Resolusi: {resolution_class}")

    cap.release()
```

```
--- Metadata Video ---
Resolusi: 1080 x 1920
Frame Rate (FPS): 59.96
Jumlah Total Frame: 651
Durasi: 10.86 detik
Klasifikasi Resolusi: Full HD
```

#### Analisis Parameter Video :

Parameter metadata video seperti resolusi, frame rate (FPS), jumlah frame, durasi, dan klasifikasi resolusi sangat penting untuk analisis maupun aplikasi tertentu. Resolusi menentukan ketajaman detail visual, sedangkan FPS memengaruhi kelancaran gerakan. Jumlah frame dan durasi memberikan gambaran banyaknya data yang harus diproses, sehingga berhubungan langsung dengan kebutuhan komputasi. Klasifikasi resolusi memudahkan dalam standar kualitas, misalnya membedakan SD, HD, hingga 4K. Kombinasi parameter-parameter ini membantu menyesuaikan antara kualitas visual, kebutuhan penyimpanan, dan performa aplikasi seperti streaming, computer vision, maupun analisis video berbasis AI.

### ✓ 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.

```
# Path file video
video_path = '/data/Pemandangan.mov'

# Buka video
cap = cv2.VideoCapture(video_path)

if not cap.isOpened():
    print(f"Error: Gagal membuka file video dari {video_path}. Pastikan file ada dan path benar.")
else:
    # Dapatkan jumlah total frame
    frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))

    # Hitung indeks frame awal, tengah, dan akhir
    start_frame_idx = 0
    middle_frame_idx = frame_count // 2
    end_frame_idx = max(0, frame_count - 1)

    frame_indices = [start_frame_idx, middle_frame_idx, end_frame_idx]
    frames_to_display = [] # List untuk menyimpan frame yang berhasil dibaca
    timestamps_to_display = [] # List untuk menyimpan timestamp frame yang berhasil dibaca
    titles_to_display = [] # List untuk menyimpan judul frame

    for idx in frame_indices:
        # Atur posisi video ke frame yang diinginkan
        cap.set(cv2.CAP_PROP_POS_FRAMES, idx)
        ret, frame_bgr = cap.read()
        if ret:
            # Konversi BGR ke RGB dan simpan
            frame_rgb = cv2.cvtColor(frame_bgr, cv2.COLOR_BGR2RGB)
            frames_to_display.append(frame_rgb)
            timestamps_to_display.append(idx / fps)

            if idx == start_frame_idx:
                titles_to_display.append('Frame Awal')
            elif idx == middle_frame_idx:
                titles_to_display.append('Frame Tengah')
            elif idx == end_frame_idx:
                titles_to_display.append('Frame Akhir')

    cap.release()

    # Tampilkan frame menggunakan subplot
    if frames_to_display: # Cek apakah ada frame yang berhasil dibaca
        fig, axes = plt.subplots(1, len(frames_to_display), figsize=(6 * len(frames_to_display), 6))

        # Jika hanya ada 1 frame, axes bukan array, tangani kasus ini
        if len(frames_to_display) == 1:
            axes = [axes] # Ubah menjadi list agar loop bisa berjalan

        for i, ax in enumerate(axes):
            ax.imshow(frames_to_display[i])
            ax.set_title(f"{titles_to_display[i]} (t={timestamps_to_display[i]:.2f}s)")
            ax.axis('off')

        plt.tight_layout()
        plt.show()
    else:
        print("Tidak ada frame yang berhasil dibaca.")
```

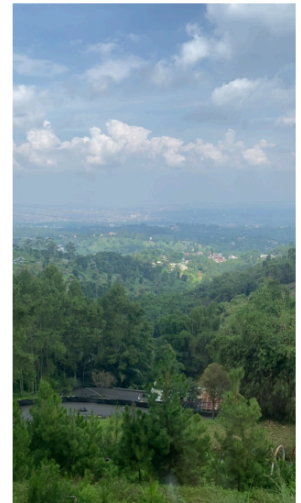
Frame Awal (t=0.00s)



Frame Tengah (t=5.42s)



Frame Akhir (t=10.84s)



#### Analisis dari 3 frame tersebut :

Ketiga frame tersebut memperlihatkan perubahan sudut pandang dan cakupan pemandangan sepanjang video. Pada frame awal (t=0.00s) lebih menyorot area dekat dengan detail jalan, tanaman, dan objek di sekitar depan kamera. Frame tengah (t=5.42s) beralih ke pemandangan yang lebih luas dengan fokus pada pepohonan dan langit, sehingga memberikan kesan transisi ke area yang lebih terbuka. Sedangkan frame akhir (t=10.84s) menampilkan panorama luas dengan kombinasi hutan, langit cerah, dan latar belakang yang semakin jauh terlihat. Dari sampel ini dapat dipelajari bagaimana video menangkap pergeseran perspektif, transisi visual dari detail ke panorama, serta variasi pencahayaan dan komposisi yang memperkaya informasi untuk analisis konten visual.

#### 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:* Resolusi video Full HD (1080p) dengan fps sekitar 59.96 sudah sangat sesuai untuk use case seperti media sosial, kuliah daring, maupun presentasi, karena kualitas visualnya tajam dan pergerakan video terlihat halus tanpa patah-patah. Kombinasi ini memberikan keseimbangan antara detail gambar dan kelancaran gerakan, sehingga orang yang melihat video dapat menerima informasi dengan jelas serta nyaman dilihat di berbagai perangkat.

## Perbandingan & Kesimpulan

### Perbandingan Representasi Media

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

#### Audio (1D - Temporal)

- Representasi: Sinyal 1 dimensi berbasis waktu.
- Visualisasi utama: Waveform, Spectrogram, dan MFCC.
- Informasi yang diperoleh: Amplitudo terhadap waktu, distribusi frekuensi, pola suara, dan ciri khas audio (contoh: suara mobil).

#### Gambar (2D - Spasial)

- Representasi: Matriks piksel dengan dimensi tinggi × lebar × kanal warna.
- Visualisasi utama: Histogram warna, dimensi gambar, distribusi intensitas piksel.
- Informasi yang diperoleh: Warna dominan, kontras, sebaran intensitas cahaya, serta ukuran file.

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

- Representasi: Rangkaian frame (gambar) berurutan dalam dimensi waktu.
- Visualisasi utama: Metadata (resolusi, fps, durasi), cuplikan frame kunci.
- Informasi yang diperoleh: Kualitas visual (resolusi), kelancaran gerakan (fps), perubahan visual antar frame, dan konteks temporal dari sebuah adegan.

## Refleksi Pembelajaran

### 3 Poin yang Saya Pelajari:

1. Setiap media (audio, gambar, video) memiliki cara representasi dan visualisasi berbeda sesuai dimensinya.
2. Analisis dasar seperti histogram, waveform, atau metadata sangat penting untuk preprocessing sebelum pemrosesan lebih lanjut.
3. Visualisasi membantu memahami karakteristik media yang tidak langsung terlihat, seperti frekuensi dominan pada audio atau distribusi warna pada gambar.

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

1. Bagaimana teknik lanjutan ekstraksi fitur dari audio (misalnya untuk klasifikasi suara).
2. Bagaimana cara yang paling optimal dalam penggunaan media dalam menyampaikan suatu informasi.

## Sumber Data & Referensi

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

- **Audio:** <https://pixabay.com/id/sound-effects/search/mobil/>
- **Gambar:** Gambar Sendiri
- **Video:** Rekaman Sendiri
- **Referensi teknis:** <https://g.co/gemini/share/5ada008f9820>





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

## 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?"