

# Tugas Analisis Multimedia: Audio, Gambar, Video

**Mata Kuliah:** Sistem & Teknologi Multimedia

**Nama:** Eric Arwido Damanik

**NIM:** 122140157

---

## 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.pyplot as plt
import librosa
import matplotlib
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: 2.2.5  
 Matplotlib: 3.10.1  
 Librosa: 0.11.0  
 OpenCV: 4.10.0

## 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: Lalu Lintas Sibuk India
- Sumber: [Youtube-IncredibleIndianTraffic](#)
- Format file: wav
- Alasan pemilihan: \_\_\_\_\_

**Path file:** `data/audio_lalu_lintas.wav` (isi nama file Anda nanti di kode)

### 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]: # Menentukan Audio Path
audio_path = 'data/audio_lalu_lintas.wav'

if not os.path.exists(audio_path):
    print(f"Audio file not found at {audio_path}. Please ensure the file exists.")
else:
    # Baca metadata audio
    info = sf.info(audio_path)
    sr_meta = info.samplerate
    duration_meta = info.duration
    channels_meta = info.channels
    total_samples_meta = info.frames

    # Menentukan label channel (mono/stereo)
    channel_label = 'Mono' if channels_meta == 1 else 'Stereo'

    # Memuat file audio menggunakan Librosa
    y, sr = librosa.load(audio_path, sr=None, mono=False)

    # Menampilkan semua informasi metadata
    print(f"Sample Rate: {sr_meta} Hz")
    print(f"Duration: {duration_meta:.2f} seconds")
    print(f"Channels: {channels_meta} ({channel_label})")
    print(f"Total Samples: {total_samples_meta}")
```

Sample Rate: 48000 Hz  
Duration: 60.00 seconds  
Channels: 2 (Stereo)  
Total Samples: 2880000

## 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.)

### Analisis

- Sepanjang rekaman, suara terus ramai tanpa ada bagian yang benar-benar hening. Ini menggambarkan suasana lalu lintas yang sibuk dan bising tanpa henti.
- Bentuk gelombangnya acak, tidak berirama seperti musik. Hal ini cocok dengan sifat kebisingan jalanan yang sulit diprediksi.
- Sesekali muncul lonjakan suara yang lebih keras dan singkat, kemungkinan berasal dari klakson kendaraan atau suara mesin yang tiba-tiba meninggi.

```
In [4]: # Menyiapkan sumbu waktu (dalam detik)
samples = y.shape[-1]
time = np.arange(samples) / sr

# Membuat plot waveform
```

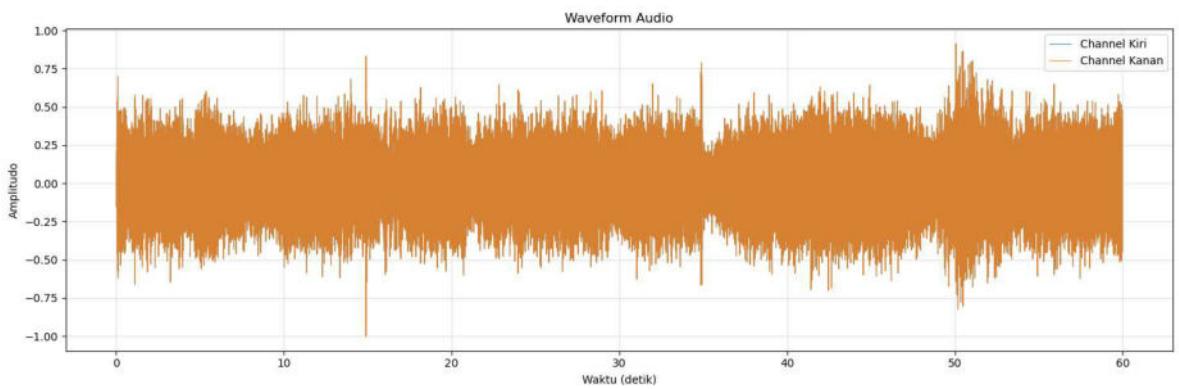
```

plt.figure(figsize=(15, 5))

# Plot setiap channel (untuk audio stereo)
if y.ndim == 1: # Kasus audio mono
    plt.plot(time, y, linewidth=0.8, label='Mono Channel')
else: # Kasus audio stereo atau lebih
    channel_labels = ['Channel Kiri', 'Channel Kanan'] if y.shape[0] == 2 else [
        for i in range(y.shape[0]):
            plt.plot(time, y[i], linewidth=0.8, alpha=0.8, label=channel_labels[i] i

# Tambahkan elemen visualisasi
plt.title('Waveform Audio')
plt.xlabel('Waktu (detik)')
plt.ylabel('Amplitudo')
plt.legend()
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

```



Karena channel kanan dan kiri saling timpa jadi tidak keliatan, maka dibuat 2 plot vertikal untuk membandingkan channel kanan dan kiri

```

In [5]: # Membuat 2 plot vertikal (2 baris, 1 kolom)
# sharex=True artinya kedua plot akan punya sumbu-x (waktu) yang sama
fig, axes = plt.subplots(2, 1, figsize=(15, 7), sharex=True, sharey=True)

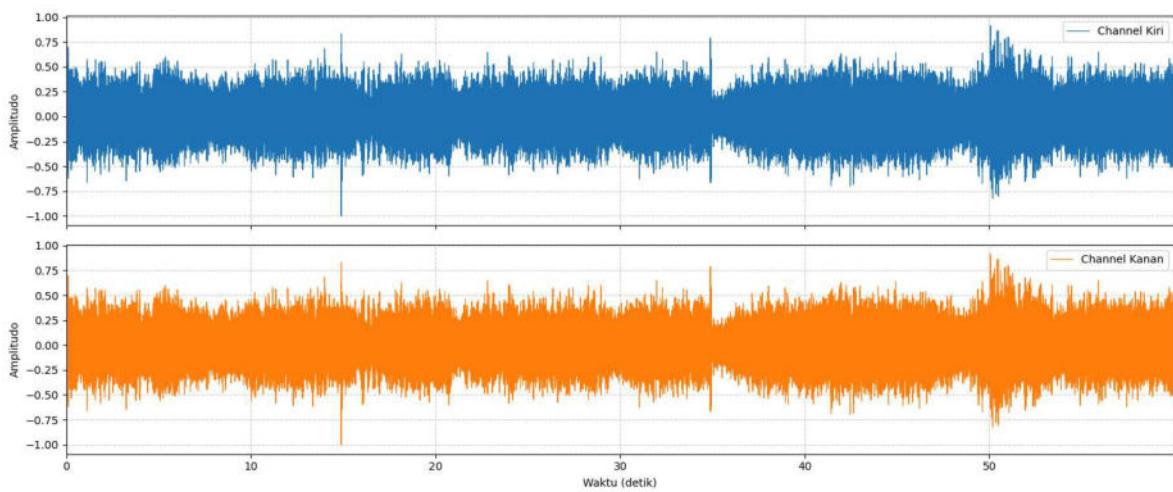
# Judul utama untuk keseluruhan gambar
fig.suptitle('Waveform Audio (Channel Kiri vs Kanan)', fontsize=16)

# Channel Kiri (atas)
axes[0].plot(time, y[0], linewidth=0.8, color='tab:blue', label='Channel Kiri')
axes[0].set_ylabel("Amplitudo")
axes[0].grid(True, linestyle='--', alpha=0.6)
axes[0].legend(loc='upper right')

# Channel Kanan (bawah)
axes[1].plot(time, y[1], linewidth=0.8, color='tab:orange', label='Channel Kanan')
axes[1].set_xlabel("Waktu (detik)")
axes[1].set_ylabel("Amplitudo")
axes[1].grid(True, linestyle='--', alpha=0.6)
axes[1].legend(loc='upper right')

# Atur batas sumbu-x agar sama
plt.xlim(0, time[-1])
plt.tight_layout(rect=[0, 0, 1, 0.96])
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:** Jelaskan perbedaan informasi yang didapat dari spectrogram dibanding waveform.

### Analisis:

- Perbedaan informasi:
  - Dalam waveform informasi yang didapat ada amplitudo.
  - Dalam spectrogram informasi yang didapat ada frekuensi dan intensitas suara.

```
In [6]: # Konversi audio ke mono jika aslinya stereo
if y.ndim > 1:
    y_mono = librosa.to_mono(y)
else:
    y_mono = y

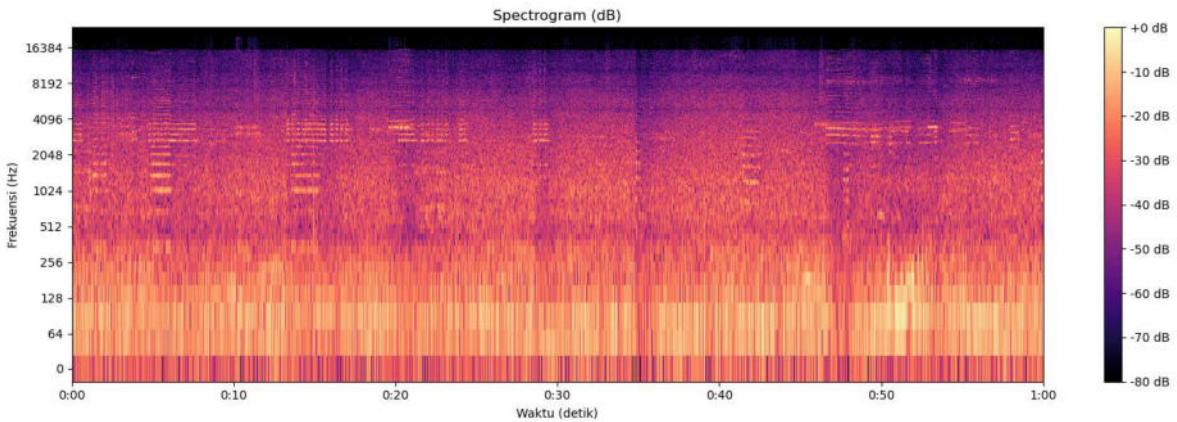
# Hitung Short-Time Fourier Transform (STFT)
D = librosa.stft(y_mono, n_fft=1024, hop_length=256)

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

# Membuat plot spectrogram
plt.figure(figsize=(15, 5))
librosa.display.specshow(S_db, sr=sr, hop_length=256, x_axis='time', y_axis='log')

# Menambahkan color bar, judul, dan Label
plt.colorbar(format='%.2f dB')
plt.title('Spectrogram (dB)')
plt.xlabel('Waktu (detik)')
plt.ylabel('Frekuensi (Hz)')
```

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



## A5. TODO: MFCC

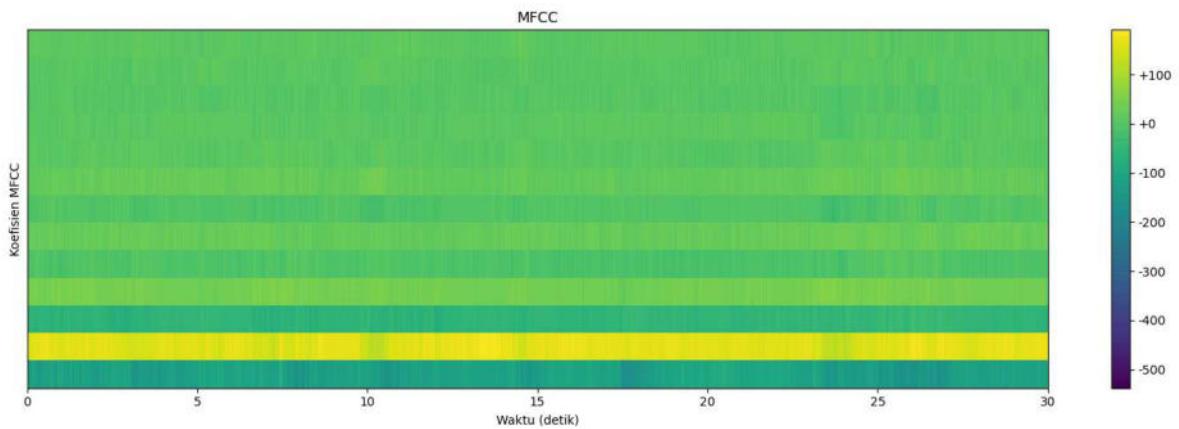
**Instruksi:** Hitung dan tampilkan minimal 13 koefisien MFCC sebagai heatmap:

- 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?

**Analisis:** Pola suara yang padat tetapi stabil dengan banyak sekali kejadian akustik yang acak dan cepat (berubah-ubah). Potensi terbesarnya adalah untuk klasifikasi suara.

```
In [7]: # Hitung MFCC dari sinyal mono  
mfccs = librosa.feature.mfcc(y=y_mono, sr=sr, n_mfcc=13)  
  
# Membuat plot heatmap untuk MFCC  
plt.figure(figsize=(15, 5))  
librosa.display.specshow(mfccs, x_axis='time', sr=sr, hop_length=256, cmap='viridis')  
  
# Menambahkan color bar, judul, dan Label  
plt.colorbar(format='%+2.0f')  
plt.title('MFCC')  
plt.xlabel('Waktu (detik)')  
plt.ylabel('Koefisien MFCC')  
plt.tight_layout()  
plt.show()
```



## A6. Analisis Ringkas (Wajib)

**Jawab pertanyaan berikut:**

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

*Jawaban Anda:* Waveform hanya menunjukkan keras-lemahnya suara seiring waktu. Sedangkan Spectrogram memecah suara menjadi frekuensi, sehingga kita tahu suara apa yang dominan.

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

*Jawaban Anda:* Visualisasinya menangkap pola khas dari audionya melalui frekuensi.

## Bagian B — Gambar

### B1. Deskripsi Data

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

- Jenis gambar: foto patung binatang di Zoo
- Sumber: foto sendiri
- Format file: JPG
- Alasan pemilihan: Warna Beragam

**Path file:** `data/image_patung_zoo.jpg` (isi nama file Anda nanti di kode)

### 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).

## Analisis

- Objek utama dalam gambar adalah patung-patung hewan. Di bagian depan (foreground), terdapat dua patung harimau di atas bebatuan artifisial. Di belakangnya, ada dua patung jerapah yang berdiri di balik pagar putih. Latar belakang didominasi oleh pepohonan hijau yang rimbun dan langit biru.
- Pencahayaan pada gambar ini sangat terang dan merata.
- Gambar ini kaya akan warna yang kontras. Warna oranye dan kuning dari patung harimau dan jerapah sangat menonjol. Warna ini kontras dengan warna hijau dari dedaunan dan biru dari langit. Warna coklat dan abu-abu pada bebatuan memberikan dasar yang netral.

```
In [9]: # Path gambar
image_path = 'data/image_patung_zoo.jpg'

if not os.path.exists(image_path):
    print(f"File tidak ditemukan di: {image_path}")
else:
    # Baca gambar menggunakan OpenCV. Hasilnya adalah format BGR.
    image_bgr = cv2.imread(image_path)

    # Konversi dari BGR ke RGB agar warnanya benar saat ditampilkan Matplotlib
    image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)

    # Tampilkan gambar
    plt.figure(figsize=(10, 7))
    plt.imshow(image_rgb)

    # Beri judul dan hilangkan sumbu agar rapi
    plt.title('Gambar yang Dimuat dengan OpenCV', fontsize=16)
    plt.axis('off')
    plt.show()
```

## Gambar yang Dimuat dengan OpenCV



### 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.

#### Analisis

- Dimensi memberikan detail dari gambarnya.
- Kanal, Agar tau apakah 3 Channel(RGB) atau 1 Channel(Hitam-Putih).
- Tipe Data, Memberi tau nilai pixelnya, 0-255, penting untuk akurasi perhitungan.
- Ukuran file dalam memori, seberapa banyak RAM yang digunakan, jika untuk 1 gambar tidak terlalu berat tetapi jika ribuan (dataset) mungkin akan membebani

komputer.

```
In [10]: # Dapatkan informasi dari properti array NumPy
height, width, channels = image_rgb.shape
data_type = image_rgb.dtype
size_in_bytes = image_rgb.nbytes
size_in_mb = size_in_bytes / (1024 * 1024) # Konversi ke MB

# Tampilkan informasi gambar
print(f"Dimensi Gambar: {width} x {height} pixels")
print(f"Jumlah Channel: {channels} (3 = RGB)")
print(f"Tipe Data: {data_type}")
print(f"Ukuran File dalam Memori: {size_in_mb:.2f} MB")
```

```
Dimensi Gambar: 2976 x 3968 pixels
Jumlah Channel: 3 (3 = RGB)
Tipe Data: uint8
Ukuran File dalam Memori: 33.79 MB
```

## 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?

### Analisis

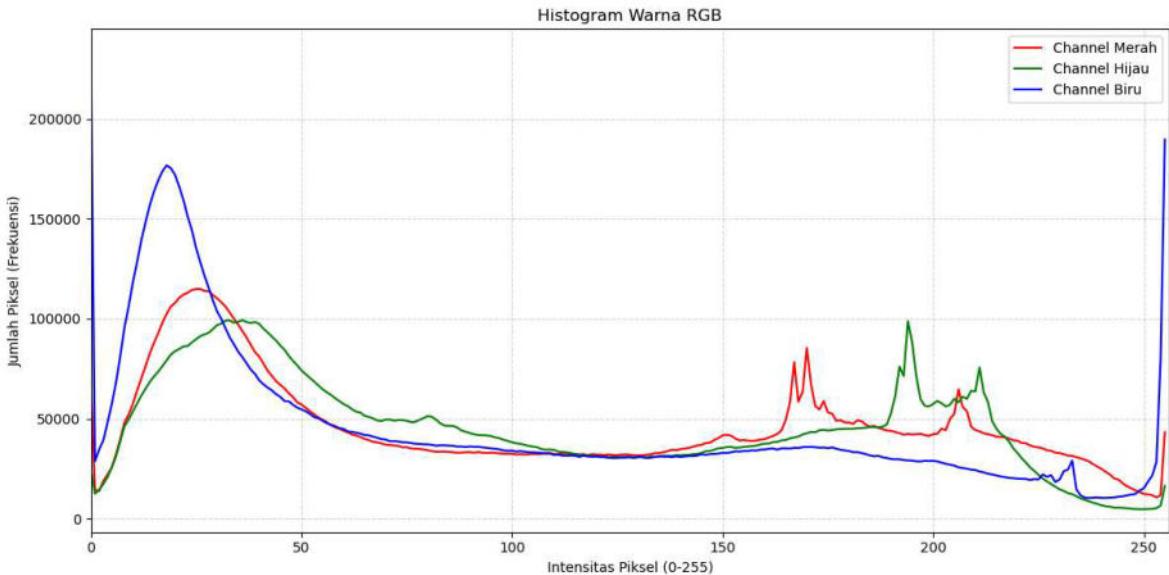
- Channel Dominan, Tidak ada channel yang dominan, tetapi pada areacahaya yang gelap Channel biru lebih dominan, sedangkan pada area cahaya yang terang Channel Merah dan Hijau lebih dominan.
- Kontras, Sebaran Pixelnya merata, tidak ada yang menumpuk ditengah.
- Sebaran intesitas, Gambar ini memiliki banyak piksel di area gelap dan juga banyak piksel di area terang , namun lebih sedikit di area tengah.

```
In [11]: # Siapkan warna dan Label untuk setiap channel
colors = ['red', 'green', 'blue']
labels = ['Channel Merah', 'Channel Hijau', 'Channel Biru']

# Siapkan plot
plt.figure(figsize=(12, 6))
plt.title('Histogram Warna RGB')
plt.xlabel('Intensitas Piksel (0-255)')
plt.ylabel('Jumlah Piksel (Frekuensi)')

# Hitung dan plot histogram untuk setiap channel
for i, color in enumerate(colors):
    hist, bins = np.histogram(image_rgb[:, :, i], bins=256, range=(0, 256))
    plt.plot(bins[:-1], hist, color=color, label=labels[i], linewidth=1.5)
```

```
# Tampilkan Legend dan rapikan plot
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)
plt.xlim([0, 256])
plt.tight_layout()
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:*

- Puncak histogram menandakan gambar tersebut terang atau gelap diliat dari kiri ke kanan (gelap ke terang). Jika diliat pada histogram puncak berada pada kanan, menandakan gambar terlihat terang.
- Channel warna yang grafiknya paling tinggi menunjukkan warna yang paling banyak muncul (dominan). Jika diliat dari histogram yang paling tinggi menunjukkan biru dan hijau, menandakan gambar terlihat langit dan pohon atau dedaunan.
- Semakin lebar sebaran histogram dari kiri ke kanan, semakin tinggi kontrasnya. Pada histogram melebar dari ujung kiri ke ujung kanan. Ini menunjukkan gambar memiliki rentang warna yang lengkap, sehingga menghasilkan kontras yang baik.

## Bagian C — Video

### C1. Deskripsi Data

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

- Jenis video: Rekaman suasana warung saat hujan deras.
- Sumber: Rekaman Sendiri.

- Durasi target: ~20 detik.
- Alasan pemilihan: Video ini memiliki elemen audio dan visual yang kaya dan dinamis (suara hujan, lalu lintas, kilatan petir) sehingga menarik untuk dianalisis.

**Path file:** data/video\_warung\_hujan.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.

**Analisis** Metadata video berisi info teknis penting:

- Resolusi menentukan detail visual (4K tajam tapi berat, HD lebih ringan).
- Frame rate (FPS) menentukan kehalusan gerakan (24–30 FPS natural, 60 FPS cocok aksi cepat/slow motion, FPS rendah bikin patah).
- Jumlah frame & durasi menunjukkan skala analisis (semakin panjang video, semakin besar beban komputasi).

```
In [12]: # Path Video
video_path = 'data/video_warung_hujan.mp4'

if not os.path.exists(video_path):
    print(f"File tidak ditemukan di: {video_path}")
else:
    # Buka file video
    cap = cv2.VideoCapture(video_path)

    if not cap.isOpened():
        print("Error: Gagal membuka file video.")
    else:
        # Ambil semua properti dari video
        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))

        # Hitung durasi dan klasifikasi resolusi
        duration = total_frames / fps if fps > 0 else 0

        def classify_resolution(w, h):
            if w >= 3840 or h >= 2160:
                return "4K"
            elif w >= 2560 or h >= 1440:
                return "2K"
            elif w >= 1920 or h >= 1080:
                return "Full HD"
            else:
                return "HD"

        print(f"Properti video: Width={width}, Height={height}, FPS={fps}, Duration={duration}, Resolution={classify_resolution(width, height)}")
```

```

        return "Full HD"
    elif w >= 1280 or h >= 720:
        return "HD"
    elif w >= 854 or h >= 480:
        return "SD"
    else:
        return "Low Res"

resolution_class = classify_resolution(width, height)

# Tampilkan informasi video
print(f"Resolusi Video: {width} x {height} pixels ({resolution_class})")
print(f"Frame Rate: {fps:.2f} FPS")
print(f"Total Frame: {total_frames}")
print(f"Durasi Video: {duration:.2f} detik")

# Selalu lepaskan video setelah selesai
cap.release()

```

Resolusi Video: 3840 x 2160 pixels (4K)  
Frame Rate: 59.79 FPS  
Total Frame: 1220  
Durasi Video: 20.40 detik

### 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:** Deskripsikan perbedaan visual antar frame dan apa yang dapat dipelajari dari sampel frame ini.

#### Analisis

- Sebenarnya perbedaan visual dari ketiganya tidak terlalu keliatan, selain dari pergerakan orang berbaju puth dan orang yang sedang memakai mantel. Hujannya juga terlihat konstan. Meskipun dalam video ada beberapa visual yang berbeda seperti mobil dan motor yang sedang lewat, dan juga ada kilatan dari petir yang ada di frame lainnya.
- Tidak ada perubahan besar pada objek utama (warung, jalanan, gerobak) pada ketiga frame tersebut. Artinya, video ini lebih merekam suasana statis dengan elemen dinamis berupa hujan. Frame ini cocok untuk analisis citra terkait hujan, visibilitas, pencahayaan, atau pengaruh cuaca terhadap kondisi jalan dan aktivitas manusia.

```
In [13]: if 'total_frames' not in locals():
    print("Harap jalankan sel C2 terlebih dahulu untuk mendapatkan metadata video")
else:
```

```

cap = cv2.VideoCapture(video_path)

if not cap.isOpened():
    print("Error: Gagal membuka file video.")
else:
    # Tentukan indeks frame yang akan diambil
    frame_indices = {
        "Frame Awal (0.00s)": 0,
        "Frame Tengah (10.20s)": total_frames // 2,
        # Frame Akhir (~20.40s)": total_frames - 1
        # Mundur 2 frame dari akhir karena terjadi error saat dibaca
        "Frame Akhir (~20.40s)": total_frames - 2
    }

    frames = []
    for title, index in frame_indices.items():
        cap.set(cv2.CAP_PROP_POS_FRAMES, index)
        ret, frame = cap.read()
        if ret:
            frames.append((title, cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)))

cap.release()

# Tampilkan ketiga frame menggunakan subplot
fig, axes = plt.subplots(1, len(frames), figsize=(18, 6))
fig.suptitle('Frame Representatif dari Video', fontsize=16)

for i, (title, frame_rgb) in enumerate(frames):
    axes[i].imshow(frame_rgb)
    axes[i].set_title(title)
    axes[i].axis('off')

plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()

```

Frame Representatif dari 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:* Untuk ukuran 4K dan 60 FP terlalu overkill untuk penggunaan sehari-hari untuk medsos, kuliah daring tapi sangat cocok untuk konten Youtube.

Alasannya ukuran file akan terlalu besar dan boros kuota jika digunakan untuk medsos atau kuliah daring. Sedangkan untuk konten Youtube akan cocok untuk memberikan

kualitas yang tajam.

# Perbandingan & Kesimpulan

## Perbandingan Representasi Media

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

### Audio (1D - Temporal)

- Representasi: Audio adalah sinyal 1D yang nilai amplitudonya berubah seiring waktu.
- Visualisasi utama: Waveform dan Spectrogram. Waveform menunjukkan kapan suara berubah pelan atau kenceng (keras). Spectrogram menunjukkan frekuensi suara yang paling dominan pada waktu tertentu.
- Informasi yang diperoleh: Audio fokus pada bagaimana suara berubah seiring waktu.

### Gambar (2D - Spasial)

- Representasi: Gambar adalah data 2D dengan tinggi dan lebar yang memiliki 3 channel (RGB).
- Visualisasi utama: Tampilan gambar dan Histogram Warna. Tampilan gambar menampilkan bentuk, objek, dan komposisi. Histogram Warna menampilkan distribusi warna misalnya dominan hijau dan biru, gambar terang.
- Informasi yang diperoleh: Gambar fokus pada isi ruang dalam satu momen statis.

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

- Representasi: Video adalah gabungan gambar 2D dan waktu.
- Visualisasi utama: Perbandingan frame yang menunjukkan perubahan antarwaktu. Metadata yang menyimpan informasi kualitas gerakan dan detail visual.
- Informasi yang diperoleh: Video fokus pada isi ruang dan bagaimana semuanya berubah seiring waktu.

---

## Refleksi Pembelajaran

### 3 Poin yang Saya Pelajari:

1. Waveform menunjukkan kekuatan sinyal sedangkan Spectrogram menunjukkan karakteristik dari frekuensi.
2. Kesesuaian spesifikasi video seperti Resolusi dan FPS sangat bergantung pada penggunaannya untuk apa.
3. Metadata Audio, Image dan Video sangat penting.

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

## Sumber Data & Referensi

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

- **Audio:** [Youtube-IncredibleIndianTraffic](#)
- **Gambar:** Punya Sendiri.
- **Video:** Punya Sendiri.
- **Referensi teknis:** [GeminiAI](#)

## 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, terkompromen, 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