

# Tugas Analisis Multimedia: Audio, Gambar, Video

**Mata Kuliah:** Sistem & Teknologi Multimedia

**Nama:** Dzaki Gastidirrijal **NIM:** 122140030

---

## Deskripsi Tugas

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

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

---

## CATATAN PENTING: PRESENTASI ACAK & KEJUJURAN AKADEMIK

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

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

**Maka nilai tugas Anda akan diberikan 0 (nol).**

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

```
In [ ]: # Import Library (Satu-satunya sel kode dalam template ini)
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import librosa
import librosa.display
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: 2.2.6  
 Matplotlib: 3.10.6  
 Librosa: 0.11.0  
 OpenCV: 4.12.0

## Petunjuk Umum Pengerjaan

### Cara Menggunakan Template

- Gunakan notebook ini sebagai kerangka kerja utama
- Tulis penjelasan (markdown) **SEBELUM** menaruh kode agar maksud dan tujuan jelas
- Tambahkan sel kode di tempat yang sudah disediakan (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: Musik (musik, pidato, suara alam, dll.)
- Sumber: Youtube (rekaman sendiri, dataset publik, dll.)
- Format file: MP3 (WAV, MP3, dll.)
- Alasan pemilihan: Ada bass dan vokal yang jelas

**Path file:** data/audio\_musik\_freak\_parody.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.

```
In [ ]: # Path file audio
file_path = os.path.join(os.getcwd(), 'data', 'audio_musik_freak_parody.wav')

# Load audio (librosa default: mono=True → jadi 1 channel)
y, sr = librosa.load(file_path, sr=None, mono=False)

# Hitung metadata
durasi = librosa.get_duration(y=y, sr=sr)
total_sampel = y.shape[-1] # ambil sumbu terakhir
kanal = 1 if y.ndim == 1 else y.shape[0] # kalau 1D berarti mono, kalau 2D berarti stereo

# Cetak hasil metadata
print("Metadata Audio")
print(f"Sample rate   : {sr} Hz")
print(f"Durasi         : {durasi:.2f} detik")
print(f"Jumlah kanal    : {'Mono' if kanal == 1 else 'Stereo'}")
print(f"Total sampel    : {total_sampel}")
```

```
Metadata Audio
Sample rate   : 44100 Hz
Durasi        : 145.19 detik
Jumlah kanal   : Stereo
Total sampel   : 6403072
```

---

## A3. TODO: Waveform

**Instruksi:** Plot waveform audio dengan:

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

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

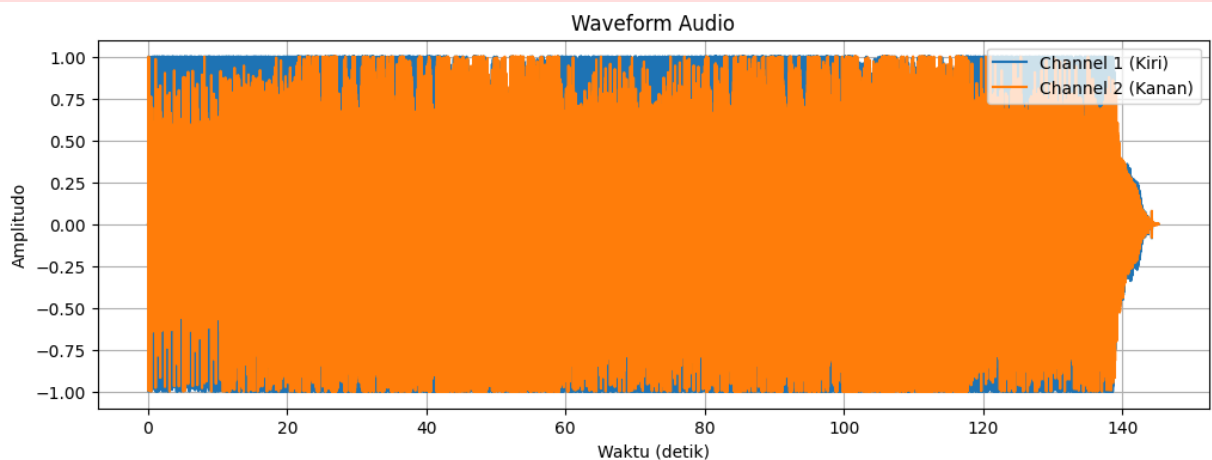
```
In [ ]: # Buat sumbu waktu (dalam detik)
waktu = np.arange(total_sampel) / sr

plt.figure(figsize=(12, 4))
if kanal == 1:
    plt.plot(waktu, y, label="Mono")
else:
    plt.plot(waktu, y[0], label="Channel 1 (Kiri)")
    plt.plot(waktu, y[1], label="Channel 2 (Kanan)")

# Judul dan Label
plt.title("Waveform Audio")
plt.xlabel("Waktu (detik)")
plt.ylabel("Amplitudo")
plt.legend()
plt.grid(True)
plt.show()
```

d:\00 kuliah s7\multimedia\worksheet 2\.venv\Lib\site-packages\IPython\core\pylabtools.py:170: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.

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



Audio diatas yang digunakan merupakan audio lagu karena dapat dilihat pola yang berulang dengan amplitudo tinggi karena bila menuju rendah maka akan mendekati amplitudo 0, dimana amplitudo rendah ada di outro/akhir audio lagu. Bisa termasuk musik juga karena variasi cepat yang menyebabkan suara lebih kompleks. Adanya 2 channel yang muncul karena jenis audio adalah Stereo.

## A4. TODO: Spectrogram log-dB

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

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

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

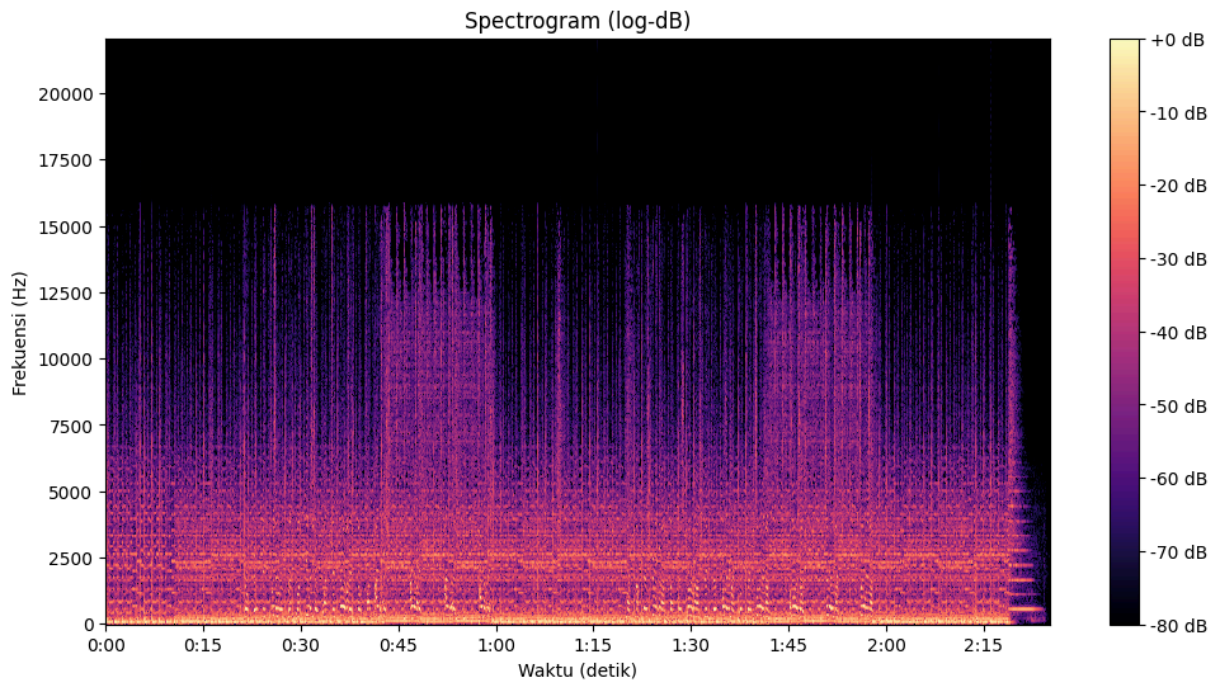
```
In [ ]: # Short-Time Fourier Transform
# Kalau stereo → ambil channel kiri (y[0])
# Kalau mono → langsung pakai y
# untuk memastikan ambil 1 sinyal 1D untuk STFT
data = y[0] if y.ndim > 1 else y

# === 1. Hitung STFT ===
n_fft = 1024 # ukuran window FFT (1024 sampel ≈ 23 ms pada 44.1kHz)
hop_length = 256 # pergeseran antar window (≈ 6 ms)
stft = librosa.stft(data, n_fft=n_fft, hop_length=hop_length)

# === 2. Magnitude Spectrogram ===
spectrogram = np.abs(stft)

# === 3. Konversi ke Log-dB ===
spectrogram_db = librosa.amplitude_to_db(spectrogram, ref=np.max)

# === 4. Plot ===
plt.figure(figsize=(12, 6))
librosa.display.specshow(
    spectrogram_db,
    sr=sr,
    hop_length=hop_length,
    x_axis='time',
    y_axis='hz',
    cmap='magma'
)
plt.colorbar(format='%+2.0f dB')
plt.title("Spectrogram (log-dB)")
plt.xlabel("Waktu (detik)")
plt.ylabel("Frekuensi (Hz)")
plt.show()
```



Sumbu Horizontal menunjukkan durasi waktu dimana audio sekitar 2 menit 20 detik. Sumbu vertikal menunjukkan rentang frekuensi dari yang rendah(bass) di bagian bawah hingga yang tinggi(pitch) di bagian atas. Warna yang lebih terang (kuning) adalah suara keras, warna yang lebih gelap (ungu/hitam) berarti suara lebih pelan. Suara tinggi di frekuensi rendah menunjukkan bahwa adanya suara bass/drum yang konsisten. Vokal bisa dilihat di frekuensi lebih dari 5000Hz dengan warna ungu yang menandakan tidak terlalu tinggi volumenya.

### **ANALISIS perbedaan spectrogram dan waveform**

Waveform menunjukkan Bagian mana yang keras atau pelan (volume/energi), juga Pola amplitude kasar (misalnya ada hentakan, kediaman, atau ritme). Tidak menunjukkan frekuensi apa yang sedang dimainkan dan Dua suara berbeda (misalnya gitar vs. piano) bisa terlihat mirip di waveform.

Spectrogram menunjukkan informasi tentang Distribusi frekuensi seiring waktu, misalnya bass (frekuensi rendah) atau vokal (frekuensi menengah-tinggi). Perubahan warna menunjukkan bagian keras/pelan pada frekuensi tertentu, juga Memudahkan identifikasi pola musik, harmoni, atau bahkan jenis instrumen.

---

## **A5. TODO: MFCC**

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

- Sumbu X: waktu (frame)
- Sumbu Y: koefisien MFCC (1-13)
- Gunakan colorbar dan judul yang jelas

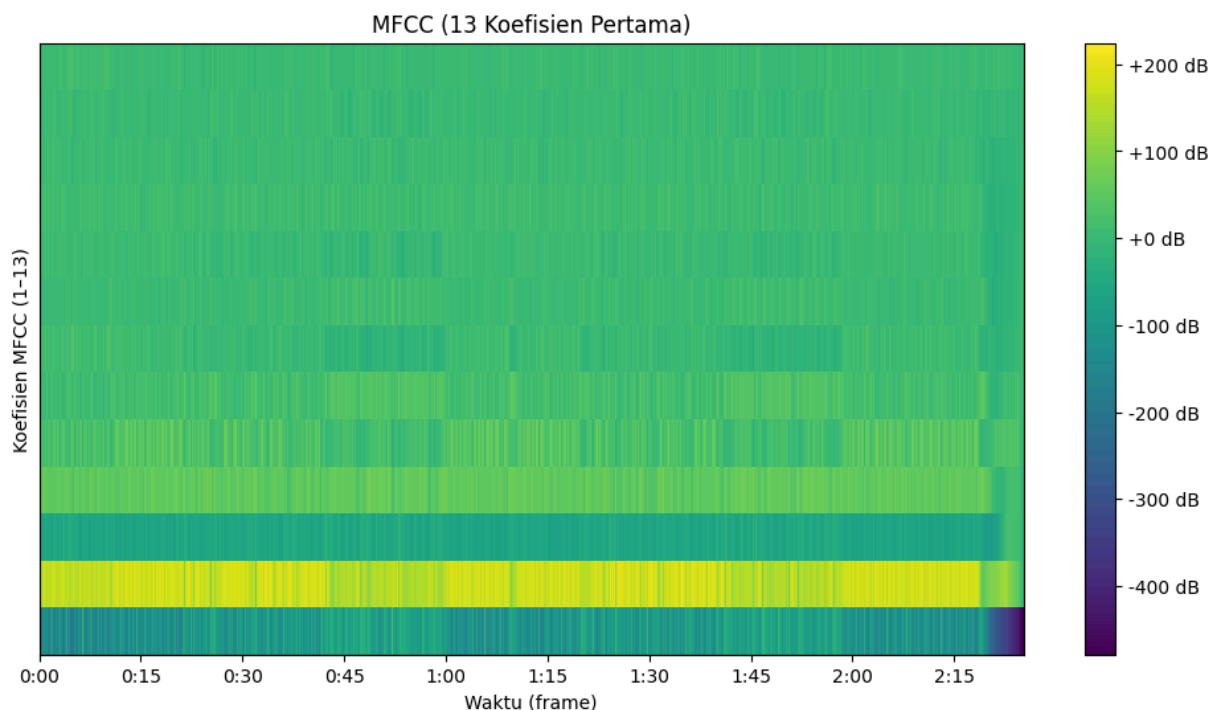


**Analisis yang diperlukan:** Interpretasi sederhana: apakah pola MFCC stabil atau berubah-ubah? Apa potensi maknanya?

```
In [ ]: # Mel-Frequency Cepstral Coefficients

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

# Plot heatmap MFCC
plt.figure(figsize=(12, 6))
librosa.display.specshow(
    mfcc,
    x_axis="time",
    sr=sr,
    cmap="viridis"
)
plt.colorbar(format="%+2.0f dB")
plt.title("MFCC (13 Koefisien Pertama)")
plt.xlabel("Waktu (frame)")
plt.ylabel("Koefisien MFCC (1-13)")
plt.show()
```



**apakah pola MFCC stabil atau berubah-ubah? Apa potensi maknanya?**

Pola yang stabil menurut yang saya lihat ada di bagian bawah yang menunjukkan bahwa sumber utama suara kemungkinan besar konsisten dan tidak berubah. Ini merupakan karakteristik dasar dari suara tersebut. Bisa jadi yang ada adalah Instrumen musik yang sama yang dimainkan.

Untuk yang Berubah-ubah (Hijau/Biru) Pola yang dinamis ini menunjukkan konten atau informasi yang berubah dari sumber suara tersebut. perubahan ini adalah karena vokal suara

yang dinyanyikan. Jika sumbernya adalah instrumen musik, perubahan ini adalah melodi, not, atau harmoni yang berbeda yang dimainkan.

---

## A6. Analisis Ringkas (Wajib)

**Jawab pertanyaan berikut:**

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

*Jawaban Anda:*

Secara sederhana, Waveform hanya menunjukkan keras atau pelannya volume suara dari waktu ke waktu, tetapi tidak bisa membedakan nada atau jenis instrumen.

Sebaliknya, Spectrogram menunjukkan informasi yang lebih lengkap dengan menampilkan frekuensi (nada tinggi/rendah) sekaligus volume, sehingga memudahkan untuk mengenali pola musik, harmoni, dan jenis suara.

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

*Jawaban Anda:*

MFCC menampilkan pola koefisien yang berbeda-beda untuk setiap karakteristik frekuensi, dengan beberapa koefisien menunjukkan pola yang stabil dan berulang, sementara yang lain menunjukkan variasi yang lebih dinamis.

Pola koefisien yang dihasilkan masih tidak stabil, karena karakteristik nada masih bervariasi dan belum membentuk ritme yang jelas. Namun ketika memasuki pertengahan lagu, pola koefisien mulai stabil sehingga membentuk ritme yang berulang.

## Bagian B — Gambar

### B1. Deskripsi Data

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

- Jenis gambar: foto (foto, ilustrasi, pemandangan, dll.)
- Sumber: pinterest (foto sendiri, dataset publik, dll.)
- Format file: JPG (JPG, PNG, BMP, dll.)
- Alasan pemilihan: Gambar yang dipilih adalah gambar musisi yang saya suka dengan editan

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

```
In [ ]: # Path file gambar
img_path = os.path.join(os.getcwd(), 'data', 'gambar_joji.jpg')

# Baca gambar dengan OpenCV (default BGR)
img_bgr = cv2.imread(img_path)

# Konversi ke RGB
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

# Tampilkan gambar
plt.figure(figsize=(6,6))
plt.imshow(img_rgb)
plt.title("Gambar Joji (RGB, Tampilan Benar)")
plt.axis("off") # Hilangkan axis
plt.show()
```

## Gambar Joji (RGB, Tampilan Benar)



Objek dominan adalah orang yang ada di tengah gambar yang sedang menghadap ke kamera dengan kondisi pencahayaan backlight dan komposisi warna dominan kuning.

---

### B3. TODO: Informasi Dasar

**Instruksi:** Tampilkan informasi metadata gambar:

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

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

```
In [ ]: # Ambil metadata dasar
height, width, channels = img_bgr.shape
dtype = img_bgr.dtype
mode = "Grayscale" if channels == 1 else "RGB" if channels == 3 else f"{channels}-c"
size_in_bytes = img_bgr.nbytes # ukuran file di memori (array numpy)
```

```
# Ukuran file di disk
file_size = os.path.getsize(img_path)

# Cetak hasil metadata
print("Metadata Gambar")
print(f"Dimensi (HxW) : {height} x {width}")
print(f"Jumlah kanal : {channels} ({mode})")
print(f"Tipe data : {dtype}")
print(f"Ukuran di mem : {size_in_bytes/1024:.2f} KB")
print(f"Ukuran file : {file_size/1024:.2f} KB (di disk)")
```

```
Metadata Gambar
Dimensi (HxW) : 809 x 828
Jumlah kanal : 3 (RGB)
Tipe data : uint8
Ukuran di mem : 1962.46 KB
Ukuran file : 28.11 KB (di disk)
```

Menurut saya, bila kita tidak tahu informasi metadata ini seperti ukuran, kanal, tipe data, maka yang kemungkinan akan terjadi adalah ketidaksesuaian antara data dan ekspektasi algoritma/model.

## B4. TODO: Histogram Warna

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

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

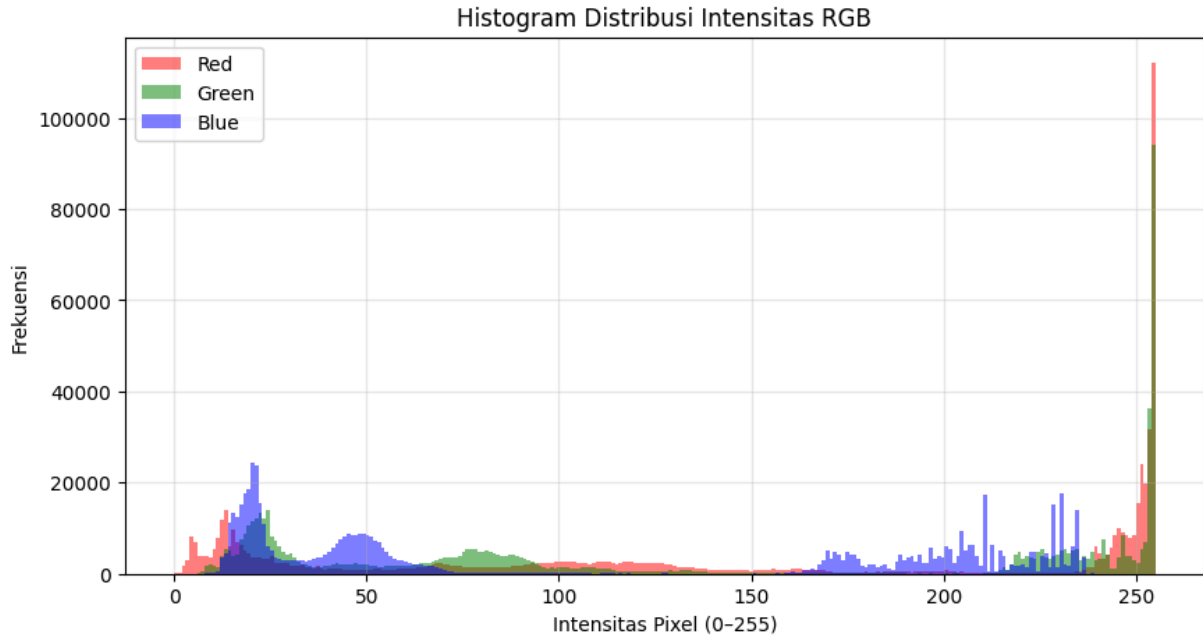
**Analisis yang diperlukan:** Analisis: channel mana yang dominan? Bagaimana kontras gambar? Seperti apa sebaran intensitasnya?

```
In [ ]: # Pisahkan channel
r_channel = img_rgb[:, :, 0].ravel()
g_channel = img_rgb[:, :, 1].ravel()
b_channel = img_rgb[:, :, 2].ravel()

# Plot histogram
plt.figure(figsize=(10,5))
plt.hist(r_channel, bins=256, range=(0,255), color='r', alpha=0.5, label='Red')
plt.hist(g_channel, bins=256, range=(0,255), color='g', alpha=0.5, label='Green')
plt.hist(b_channel, bins=256, range=(0,255), color='b', alpha=0.5, label='Blue')

# Label & Judul
plt.title("Histogram Distribusi Intensitas RGB")
plt.xlabel("Intensitas Pixel (0-255)")
plt.ylabel("Frekuensi")
plt.legend()
```

```
plt.grid(True, alpha=0.3)
plt.show()
```



### Channel dominan

Di area sangat terang (sisi kanan, intensitas 255), channel Merah (Red) dan Hijau (Green) sangat dominan. Ini menunjukkan sebagian besar area paling terang pada gambar berwarna putih atau kekuningan.

Di area gelap (sisi kiri, intensitas 10-60), channel Biru (Blue) memiliki frekuensi yang paling tinggi. Ini berarti area bayangan pada gambar memiliki rasa kebiruan.

### Kontras Gambar

Kontras gambar ini menurut saya adalah TINGGI karena sebaran piksel (intensitas) merata dari ujung kiri (area paling gelap, dekat 0) hingga ujung kanan (area paling terang, di 255). Gambar dengan kontras tinggi memiliki area hitam pekat dan putih terang yang jelas, dan histogram ini mengkonfirmasi hal tersebut.

### Sebaran Intensitasnya

- Banyak pixel di Area Gelap: Terdapat kumpulan pixel yang signifikan di rentang intensitas 0 hingga 75. Ini menandakan gambar memiliki banyak detail di bagian bayangan atau area gelap.
- Sedikit Piksel di Area Tengah (Mid-tones): Area antara 75 hingga 175 relatif kosong, yang berarti tidak banyak warna atau detail dengan tingkat kecerahan sedang.
- Sangat Banyak Piksel di Area Terang: Terdapat lonjakan piksel yang sangat besar di sisi kanan, terutama puncaknya yang ada di angka 255. Ini berarti sebagian besar area gambar sangat terang atau bahkan overexposed (detail hilang menjadi putih bersih).

---

## 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:*

Menurut saya hubungan antara pola histogram dan kesan visual adalah semacam representasi dalam numerik dari apa yang mata kita lihat atau yang dirasakan secara visual.

## Bagian C — Video

### C1. Deskripsi Data

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

- Jenis video: komedi (aktivitas, pemandangan, tutorial, dll.)
- Sumber: Youtube (rekaman sendiri, dataset publik, dll.)
- Durasi target: 18 detik (disarankan  $\leq 30$  detik untuk efisiensi)
- Alasan pemilihan: Merupakan salah satu komedi masterpiece menurut saya pada saat ini

**Path file:** `data/video_mambo.mp4` (isi nama file Anda nanti di kode)

---

### C2. TODO: Baca & Metadata

**Instruksi:** Baca video dengan OpenCV dan tampilkan metadata:

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

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

```
In [ ]: # Baca video
video_path = os.path.join(os.getcwd(), 'data', 'video_mambo.mp4')
cap = cv2.VideoCapture(video_path)

if not cap.isOpened():
    print("Gagal membuka video!")
```

```

else:
    # Metadata
    width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fps = cap.get(cv2.CAP_PROP_FPS)
    total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))

    # Hitung durasi (detik)
    duration = total_frames / fps if fps > 0 else 0

    # Klasifikasi resolusi
    if width == 1280 and height == 720:
        resolution_class = "HD (720p)"
    elif width == 1920 and height == 1080:
        resolution_class = "Full HD (1080p)"
    elif width == 3840 and height == 2160:
        resolution_class = "4K (2160p)"
    elif width == 7680 and height == 4320:
        resolution_class = "8K (4320p)"
    elif width < 1280 and height < 720:
        resolution_class = "SD"
    else:
        resolution_class = f"Custom ({width}x{height})"

    # Tampilkan hasil
    print("=== Metadata Video ===")
    print(f"Resolusi : {width} x {height}")
    print(f"Frame rate (fps) : {fps:.2f}")
    print(f"Jumlah frame : {total_frames}")
    print(f"Durasi (detik) : {duration:.2f}")
    print(f"Klasifikasi : {resolution_class}")

cap.release()

```

```

=== Metadata Video ===
Resolusi      : 854 x 480
Frame rate (fps) : 25.00
Jumlah frame   : 467
Durasi (detik) : 18.68
Klasifikasi    : SD

```

parameter-parameter yang dianalisis merupakan penting karena parameter ini adalah sebagai petunjuk untuk memberitahu seberapa besar, cepat, dan detail sebuah video yang nantinya bisa ditentukan strategi pemrosesan terbaik dari parameter tersebut.

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

**Instruksi:** Ambil dan tampilkan 3 frame representatif:

- Frame pertama (index 0)
- Frame tengah (index  $\sim \text{total\_frame}/2$ )
- Frame terakhir (index  $\text{total\_frame}-1$ )
- **Konversi BGR→RGB** sebelum ditampilkan



- Subplot dengan judul frame dan timestamp

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

```
In [ ]: # Buka video
cap = cv2.VideoCapture(video_path)

# Ambil metadata
total_frame = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
fps = cap.get(cv2.CAP_PROP_FPS) or 30.0 # fallback jika fps = 0
durasi = total_frame / fps if fps > 0 else 0
print(f"Total frame: {total_frame}, FPS: {fps:.2f}, Durasi: {durasi:.2f} detik")

# Fungsi ambil frame dengan validasi
def get_valid_frame(cap, index):
    cap.set(cv2.CAP_PROP_POS_FRAMES, index)
    ret, frame = cap.read()
    if ret:
        return cv2.cvtColor(frame, cv2.COLOR_BGR2RGB), index
    # kalau gagal, mundur 1 frame sampai ketemu valid
    for i in range(index-1, -1, -1):
        cap.set(cv2.CAP_PROP_POS_FRAMES, i)
        ret, frame = cap.read()
        if ret:
            return cv2.cvtColor(frame, cv2.COLOR_BGR2RGB), i
    return None, None

# Target frame: awal, tengah, akhir
target_frames = [0, total_frame // 2, total_frame - 1]

frames = []
for idx in target_frames:
    frame, valid_idx = get_valid_frame(cap, idx)
    if frame is not None:
        timestamp = valid_idx / fps
        frames.append((frame, valid_idx, timestamp))
    else:
        print(f"✗ Tidak bisa ambil frame {idx}")

cap.release()

# Plot hasil
plt.figure(figsize=(15, 5))
for i, (frame, idx, timestamp) in enumerate(frames):
    plt.subplot(1, 3, i+1)
    plt.imshow(frame)
    plt.axis("off")
    plt.title(f"Frame {idx}\n{timestamp:.2f} detik")

plt.suptitle("Tiga Frame Representatif dari Video", fontsize=14)
plt.tight_layout()
plt.show()
```

Total frame: 467, FPS: 25.00, Durasi: 18.68 detik



## Frame 0 (Awal)

- Terlihat pada frame pertama adalah awal mula video yang menunjukkan mambo sedang memutar kebelakang, meskipun sedang bergerak tetapi frame tidak blur atau pecah.

## Frame 233 (Tengah)

- Pada frame ini sama seperti sebelumnya, karakter mambo sedang bergerak tetapi dapat dilihat bahwa frame tersebut tidak blur dan tidak terlalu pecah dan dapat dilihat dengan cukup jelas.

## Frame 798 (Akhir)

- frame akhir menunjukkan mambo sedang melihat ke kamera dan mealkukan outro.

Yang dapat dipelajari adalah meskkipun dalam video karakter terlihat bergerak namun setiap frame tidak menampilkan blur tersebut. hal ini disebabkan karena Setiap frame dalam animasi digambar atau di-render oleh komputer sebagai satu gambar tunggal yang sempurna dan tajam. Tidak ada "blur" alami yang terjadi seperti saat kita merekam dengan kamera biasa.

## 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:*

Ya, parameter video ini, terutama 25 FPS, sudah sangat sesuai untuk sebagian besar kasus penggunaan umum seperti media sosial. Meskipun beberapa platform mendukung hingga 60 FPS untuk gerakan yang sangat halus, 25 FPS adalah frame rate yang sepenuhnya dapat

diterima dan umum digunakan. Dengan resolusi 480 dan 25 FPS yang bisa dibilang tidak mulus, justru disitulah letak lucu dalam video tersebut.

# Perbandingan & Kesimpulan

## Perbandingan Representasi Media

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

### Audio (1D - Temporal)

- Representasi: Gelombang suara dan frekuensinya
- Visualisasi utama: Waveform, Spectrogram, MFCC
- Informasi yang diperoleh: Durasi, frekuensi dominan, pola perubahan suara

### Gambar (2D - Spasial)

- Representasi: Pixel yang ada pada gambar dalam format RGB
- Visualisasi utama: Histogram warna
- Informasi yang diperoleh: Dimensi, jumlah kanal, channel dominan, contrast Gambar

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

- Representasi: Pixel video dalam format BGR
  - Visualisasi utama: Frame Representatif
  - Informasi yang diperoleh: Resolusi, fps, frame count, durasi
- 

## Refleksi Pembelajaran

### 3 Poin yang Saya Pelajari:

1. Dapat menganalisis audio dengan cara menampilkan waveform, spectrogram, dan MFCC
2. Dapat menganalisis video dengan cara menampilkan histogram warna dan analisis hubungan histogram dengan kesan visual gambar
3. Menampilkan metadata video dan analisis kesesuaian parameter video dengan use case

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

- 1. Logika yang digunakan pada kode perlu didalami lebih lanjut karena menurut saya masih abu abu tentang cara kerjanya
- 2. Pembacaan data stereo suara masih bingung dan beberapa data lainnya juga.

## Sumber Data & Referensi

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

- **Audio:** <https://www.youtube.com/watch?v=1goAp0XmhZQ>
- **Gambar:** <https://id.pinterest.com/pin/158892693096571234/>
- **Video:** <https://www.youtube.com/watch?v=M3hXWX5MdUM>
- **Referensi teknis:** GPT - <https://chatgpt.com/share/68c81264-3f54-8001-a35b-442382acda99>

## 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?"

**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