

# Tugas Analisis Multimedia: Audio, Gambar, Video

**Mata Kuliah:** Sistem & Teknologi Multimedia

**Nama:** Tawakkal Rabbani Muhammad

**NIM:** 122140029

---

## 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 [1]: # Import Library (Satu-satunya sel kode dalam template ini)
import numpy as np
import matplotlib # tambahkan ini untuk mendapatkan versi matplotlib
import matplotlib.pyplot as plt
import librosa
import soundfile as sf
from PIL import Image
import cv2
from IPython.display import Audio, HTML, display
import os
```

```
# Set matplotlib untuk menampilkan plot inline
%matplotlib inline

# Tampilkan versi library untuk dokumentasi
print("Library versions:")
print(f"NumPy: {np.__version__}")
print(f"Matplotlib: {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: 3.10.5  
Librosa: 0.11.0  
OpenCV: 4.12.0

## Petunjuk Umum Pengerjaan

### Cara Menggunakan Template

- Gunakan notebook ini sebagai kerangka kerja utama
- Tulis penjelasan (markdown) **SEBELUM** menaruh kode agar maksud dan tujuan jelas
- Tambahkan sel kode di tempat yang sudah disediakan (tandai dengan TODO)
- Semua plot/gambar harus diberi judul, label sumbu, dan keterangan singkat

### Standar Visualisasi

- Setiap plot harus memiliki judul yang deskriptif
- Label sumbu X dan Y harus jelas
- Gunakan colorbar untuk plot yang memerlukan skala warna
- Berikan interpretasi singkat setelah setiap visualisasi

### Struktur Data yang Direkomendasikan

- Buat folder `data/` di direktori yang sama dengan notebook
- Gunakan nama file yang deskriptif (contoh: `audio_musik_piano.wav`, `gambar_pemandangan_gunung.jpg`)
- Dokumentasikan sumber data jika menggunakan dataset publik

### Larangan

- **Jangan** menaruh seluruh pekerjaan dalam satu sel kode yang sangat panjang
- **Jangan** menempel hasil output tanpa interpretasi atau analisis
- **Jangan** bergantung sepenuhnya pada AI - pahami dan kuasai kode Anda






### Persiapan Presentasi Acak

- Pastikan Anda memahami setiap baris kode yang ditulis
- Latih menjelaskan logika dan alur pemikiran Anda





- Siapkan penjelasan untuk setiap visualisasi dan interpretasinya

## Checklist Kelengkapan (Centang saat selesai)





### Bagian Audio

- [] Muat audio dan tampilkan metadata (durasi, sample rate, jumlah kanal)
- [] Tampilkan waveform dengan label sumbu yang jelas
- [] Tampilkan spectrogram dalam skala log-dB dengan colorbar
- [] Tampilkan MFCC (minimal 13 koefisien) sebagai heatmap
- [] Berikan interpretasi dan analisis untuk setiap visualisasi audio





### Bagian Gambar

- [] Tampilkan gambar dengan benar dalam format RGB
- [] Tampilkan informasi dasar (dimensi, jumlah kanal, dtype)
- [] Tampilkan histogram warna untuk channel R, G, B
- [] Berikan analisis hubungan histogram dengan kesan visual gambar

### Bagian Video

- [] Tampilkan metadata video (resolusi, fps, frame count, durasi)
- [] Tampilkan 3 frame representatif (awal, tengah, akhir)
- [] Konversi BGR ke RGB dengan benar untuk visualisasi
- [] Analisis kesesuaian parameter video dengan use case

### Analisis & Dokumentasi

- [] Setiap bagian memiliki interpretasi dan analisis ringkas
- [] Perbandingan representasi ketiga jenis media
- [] Kesimpulan pembelajaran dan refleksi
- [] Semua sumber data dan referensi dicantumkan

## Pendahuluan

### Apa itu Data Multimedia?

Data multimedia adalah informasi yang dikodekan dalam berbagai format untuk merepresentasikan dunia nyata:

- **Audio (1D):** Sinyal satu dimensi yang berubah terhadap waktu
  - Contoh: musik, suara, speech
  - Representasi: amplitudo vs waktu
- **Gambar (2D):** Matriks nilai intensitas dalam ruang dua dimensi

- Contoh: foto, ilustrasi, grafik
- Representasi: intensitas pixel pada koordinat (x,y)
- **Video (2D + Waktu):** Rangkaian frame (gambar) yang ditampilkan berurutan
  - Contoh: film, rekaman, animasi
  - Representasi: frame berubah terhadap waktu dengan frame rate tertentu

## Tujuan Tugas

Memahami representasi dasar dan teknik visualisasi fundamental untuk setiap jenis media multimedia, termasuk:

- Cara memuat dan membaca file multimedia
- Ekstraksi informasi metadata yang penting
- Visualisasi yang informatif dan mudah dipahami
- Interpretasi hasil analisis secara kontekstual

## Cara Kerja

1. Isi setiap bagian sesuai instruksi yang diberikan
2. Tambahkan sel kode di tempat yang ditandai dengan "TODO"
3. Berikan interpretasi dan analisis setelah setiap visualisasi
4. Pastikan semua plot memiliki judul, label, dan keterangan yang jelas

## Bagian A — Audio

### A1. Deskripsi Data

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

- Jenis audio: engine sound (Porsche 991 GT3 RS)
- Sumber: Youtube
- Format file: WAV
- Alasan pemilihan: karena saya suka dengan suara engine dari Porsche 991 GT3 RS

**Path file:** `data/audio_engine_porsche911.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 [2]: # Simpan path file audio menggunakan os.path.join dan os.getcwd()
audio_path = os.path.join(os.getcwd(), "data", "audio_engine_porsche911.wav")

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

# Baca metadata file audio menggunakan soundfile
with sf.SoundFile(audio_path) as f:
    samplerate = f.samplerate          # Sample rate (Hz)
    channels = f.channels              # Jumlah kanal (1 = mono, 2 = stereo)
    frames = len(f)                   # Jumlah total sampel (per kanal)
    duration = len(f) / f.samplerate  # Durasi audio dalam detik

# Cetak metadata dasar
print("Path file      :", audio_path)
print("Sample Rate   :", samplerate, "Hz")
print("Durasi        :", round(duration, 2), "detik")
print("Jumlah Kanal   :", "Mono" if channels == 1 else "Stereo")
print("Total Sampel   :", frames)
```

```
Path file      : c:\Users\tawak\OneDrive\Documents\takamuhammad\Kuliah\SMT7\Mulmet
\Tugas\Ming-2\data\audio_engine_porsche911.wav
Sample Rate    : 44100 Hz
Durasi         : 30.65 detik
Jumlah Kanal   : Stereo
Total Sampel   : 1351680
```

### 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 [3]: # Memuat ulang audio agar aman, tetap menggunakan path yang sudah dibuat sebelum
y, sr = librosa.load(audio_path, sr=None, mono=True)

# Membuat array waktu dalam detik berdasarkan jumlah sampel dan sample rate
time = np.linspace(0, len(y) / sr, num=len(y))

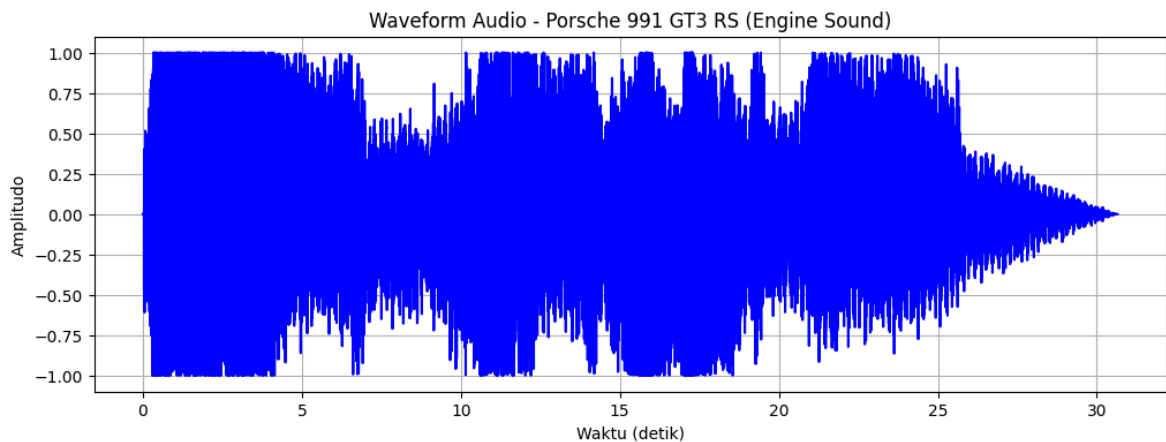
# Mulai membuat plot
plt.figure(figsize=(12, 4)) # pilih ukuran 12x4 agar lebih lebar dan jelas

# Plot waveform: waktu di sumbu X, amplitudo di sumbu Y
plt.plot(time, y, color="blue")

# Beri judul dan label agar lebih informatif
plt.title("Waveform Audio - Porsche 991 GT3 RS (Engine Sound)")
plt.xlabel("Waktu (detik)")
plt.ylabel("Amplitudo")
```

```
# Tambahkan grid supaya lebih mudah membaca grafik
plt.grid(True)

# Tampilkan hasil plot
plt.show()
```



**Hasil Analisis:** Berdasarkan waveform audio yang dibuat, pola gelombang di detik 0-5 amplitudonya sangat tinggi yang berarti suara yang dihasilkan keras, lalu dari detik 5-10 amplitudonya mulai menurun yang berarti suara yang dihasilkan pelan, kemudian naik lagi dari detik 10-25 dan menurun dari detik 25-30 yang dimana di sini menunjukkan bahwa suaranya mulai menghilang.

## 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.

```
In [4]: # Hitung STFT (Short-Time Fourier Transform) dari sinyal audio
D = librosa.stft(y, n_fft=1024, hop_length=256)

# Ubah hasil STFT menjadi skala magnitude (abs) lalu konversi ke dB (logaritmik)
S_db = librosa.amplitude_to_db(abs(D), ref=np.max)

# Mulai membuat plot spectrogram
plt.figure(figsize=(12, 6)) # saya pilih ukuran 12x6 agar detailnya lebih jelas

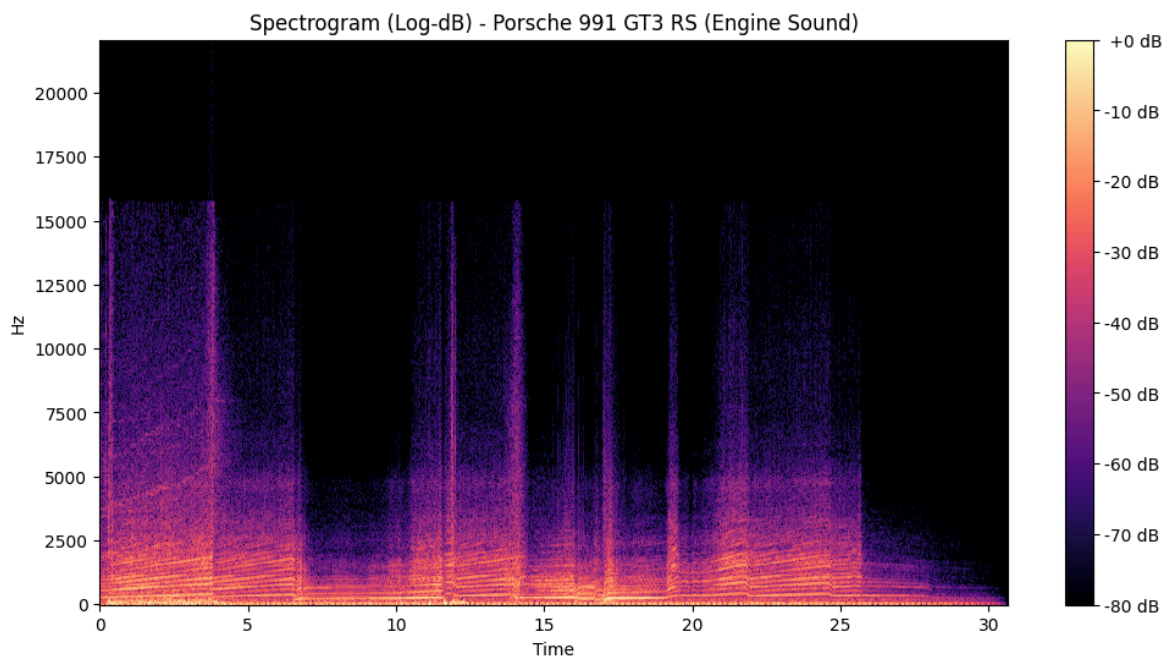
# Gunakan librosa.display.specshow untuk menampilkan spectrogram
librosa.display.specshow(
    S_db, sr=sr, hop_length=256, x_axis="time", y_axis="hz", cmap="magma"
)

# Beri judul dan label agar lebih informatif
plt.title("Spectrogram (Log-dB) - Porsche 991 GT3 RS (Engine Sound)")

# Tambahkan colorbar agar bisa melihat skala intensitas (dB)
```

```
plt.colorbar(format="%+3.0f dB")

# Tampilkan hasil spectrogram
plt.show()
```



**Hasil Analisis:** Informasi yang didapatkan dari Spectrogram jauh lebih banyak dibandingkan Waveform. Contohnya di Spectrogram selain dari suara mesin terlihat juga frekuensi dari komponen suara lain seperti letupan knalpot dan putaran mesin, sedangkan di Waveform hanya menampilkan seberapa keras dan rendah suara mesin pada tiap detiknya.

## 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 [5]: # Hitung MFCC dari audio, dengan jumlah koefisien = 13
mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)

# Mulai membuat plot heatmap untuk MFCC
plt.figure(figsize=(12, 6)) # saya pilih ukuran 12x6 agar hasilnya lebih jelas

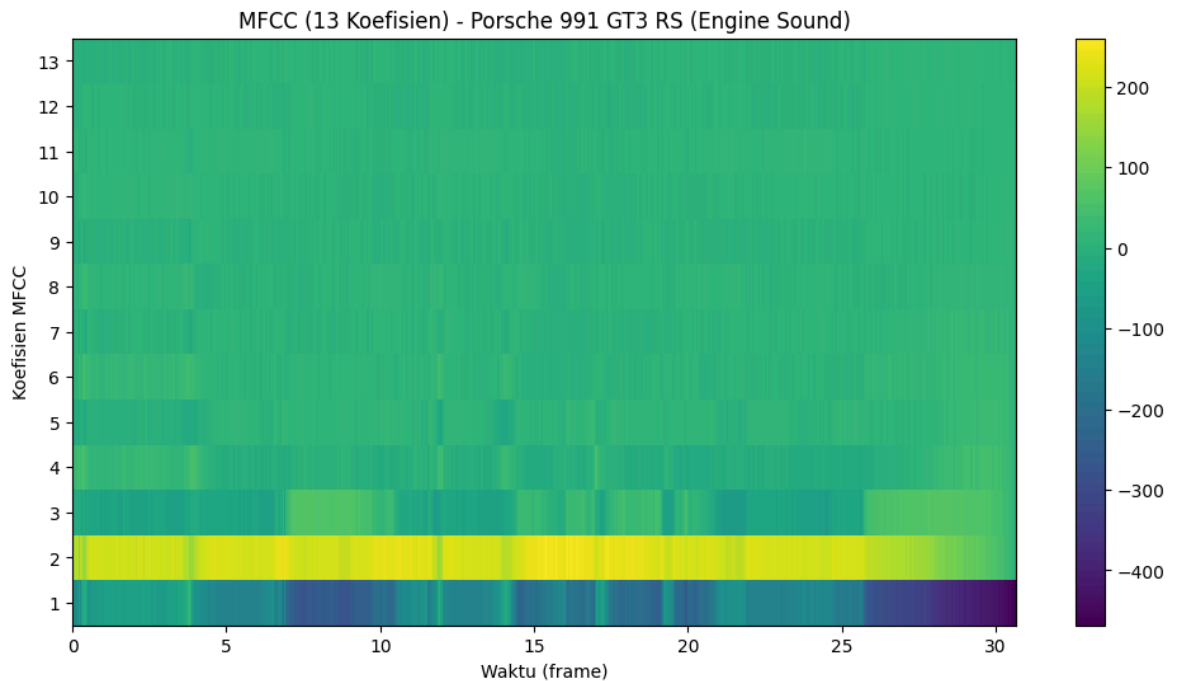
# Tampilkan MFCC dalam bentuk heatmap (koefisien vs waktu)
librosa.display.specshow(mfccs, x_axis="time", sr=sr, cmap="viridis")

# Beri judul dan label sumbu agar lebih informatif
plt.title("MFCC (13 Koefisien) - Porsche 991 GT3 RS (Engine Sound)")
plt.xlabel("Waktu (frame)")
plt.ylabel("Koefisien MFCC")
```

```
# Memberikan label pada sumbu y
plt.yticks(np.arange(0, 13), np.arange(1, 14)) # Label 1-13 for MFCC

# Tambahkan colorbar supaya terlihat skala nilai MFCC
plt.colorbar()

# Tampilkan hasil heatmap
plt.show()
```



**Hasil Analisis:** Dari hasil visualisasi MFCC tersebut terlihat bahwa pola koefisien relatif stabil, dengan beberapa variasi kecil pada rentang waktu tertentu. Stabilitas pola MFCC ini menunjukkan bahwa suara mesin Porsche 991 GT3 RS cenderung konsisten dalam frekuensi dan karakteristik akustiknya, yang bisa menandakan mesin bekerja pada putaran yang relatif konstan. Variasi kecil yang muncul mencerminkan adanya perubahan intensitas atau akselerasi singkat, yang secara akustik memberi informasi tentang dinamika kerja mesin.

## A6. Analisis Ringkas (Wajib)

**Jawab pertanyaan berikut:**

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

*Jawaban Anda:* Waveform hanya memperlihatkan amplitudo terhadap waktu, sehingga kita bisa melihat kapan suara mesin Porsche 991 GT3 RS lebih keras atau pelan, tetapi tidak memberi tahu isi frekuensinya. Spectrogram jauh lebih informatif karena menunjukkan distribusi frekuensi mesin, misalnya bunyi dominan di frekuensi rendah dan pola harmonik khas mesin sport yang berubah saat putaran naik atau turun, selain itu juga bunyi letupan knalpot dan putaran mesin.

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



*Jawaban Anda:* Dari visualisasi MFCC terlihat bahwa pola koefisien relatif stabil dengan sedikit variasi. Hal ini menunjukkan bahwa karakteristik akustik mesin Porsche 991 GT3 RS cukup konsisten, menandakan putaran mesin yang stabil. Variasi kecil pada MFCC memberi gambaran adanya dinamika seperti perubahan intensitas atau akselerasi singkat. Dengan kata lain, MFCC membantu menangkap ciri khas atau "fingerprint" suara mesin yang tidak terlihat dari waveform maupun spectrogram biasa.

## Bagian B — Gambar

### B1. Deskripsi Data

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

- Jenis gambar: lukisan
- Sumber: Wikipedia
- Format file: JPG
- Alasan pemilihan: karena saya suka dengan lukisan ini, di mana menggambarkan bagaimana Pangeran Diponegoro ditangkap

**Path file:** `data/gambar_lukisan_diponegoro.jpg`

---

### B2. TODO: Baca & Tampilkan (RGB)

**Instruksi:** Baca gambar dan tampilkan dengan benar dalam format RGB:

- Pastikan konversi warna benar (ingat perbedaan BGR vs RGB di OpenCV)
- Berikan judul yang deskriptif
- Hilangkan axis untuk tampilan yang bersih

**Analisis yang diperlukan:** Jelaskan gambar secara ringkas (objek dominan, kondisi pencahayaan, komposisi warna).

```
In [6]: # Tentukan path gambar dengan cara yang aman (supaya tidak hardcode)
img_path = os.path.join(os.getcwd(), "data", "gambar_lukisan_diponegoro.jpg")

# Baca gambar dengan OpenCV (default-nya dalam format BGR)
img_bgr = cv2.imread(img_path)

# Konversi gambar dari BGR ke RGB agar warnanya tampil benar
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

# Tampilkan gambar dengan matplotlib
plt.imshow(img_rgb) # tampilkan gambar dalam format RGB
plt.title("Lukisan Diponegoro (Raden Saleh - 1857)") # memberikan judul
plt.axis("off") # hilangkan axis agar tampilan bersih
plt.show() # tampilkan hasilnya
```

## Lukisan Diponegoro (Raden Saleh - 1857)



**Hasil Analisis:** Dari gambar yang ditampilkan, bisa terlihat objek dominannya adalah Pangeran Diponegoro yang berada di tengah kerumunan dengan ekspresi tegas menatap Letnan Jenderal Hendrik Merkus de Kock dan dikelilingi perwira Belanda dan orang Jawa. Kondisi pencahayaan difokuskan ke objek dominan yaitu dengan cahaya lembut yang menerangi Diponegoro. Komposisi warna didominasi oleh perpaduan warna bumi, seperti coklat, merah bata dan hijau yang berpadu dengan biru seragam tentara.

### 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 [7]: # Baca gambar dengan OpenCV (default BGR)
img = cv2.imread(img_path)

# Ambil dimensi gambar (tinggi, lebar, kanal warna)
height, width, channels = img.shape

# Ambil tipe data pixel (contoh: uint8)
dtype = img.dtype

# Mode warna saya anggap sebagai RGB (setelah konversi dari BGR)
mode_warna = "RGB" if channels == 3 else "Grayscale"
```

```
# Hitung ukuran file gambar di memori
ukuran_file_kb = img.nbytes / 1024          # dalam KB
ukuran_file_mb = img.nbytes / (1024 * 1024) # dalam MB

# Cetak hasil informasi dasar
print(f"Dimensi Gambar      : {height} x {width}")
print(f"Jumlah Kanal        : {channels}")
print(f"Tipe Data            : {dtype}")
print(f"Mode Warna            : {mode_warna}")
print(f"Ukuran Memori         : {ukuran_file_mb:.4f} MB")
```

```
Dimensi Gambar      : 3307 x 5255
Jumlah Kanal        : 3
Tipe Data           : uint8
Mode Warna          : RGB
Ukuran Memori       : 49.7197 MB
```

**Hasil Analisis:** Informasi ukuran file, dimensi, jumlah kanal, tipe data, dan ukuran memori penting untuk tahap preprocessing karena menentukan bagaimana data akan diproses, disimpan, dan dianalisis. Dengan mengetahui detail ini, saya bisa memastikan gambar memiliki format yang sesuai (misalnya RGB), ukuran yang konsisten untuk model, serta memperkirakan kebutuhan memori dan komputasi saat analisis lanjutan.

## 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?

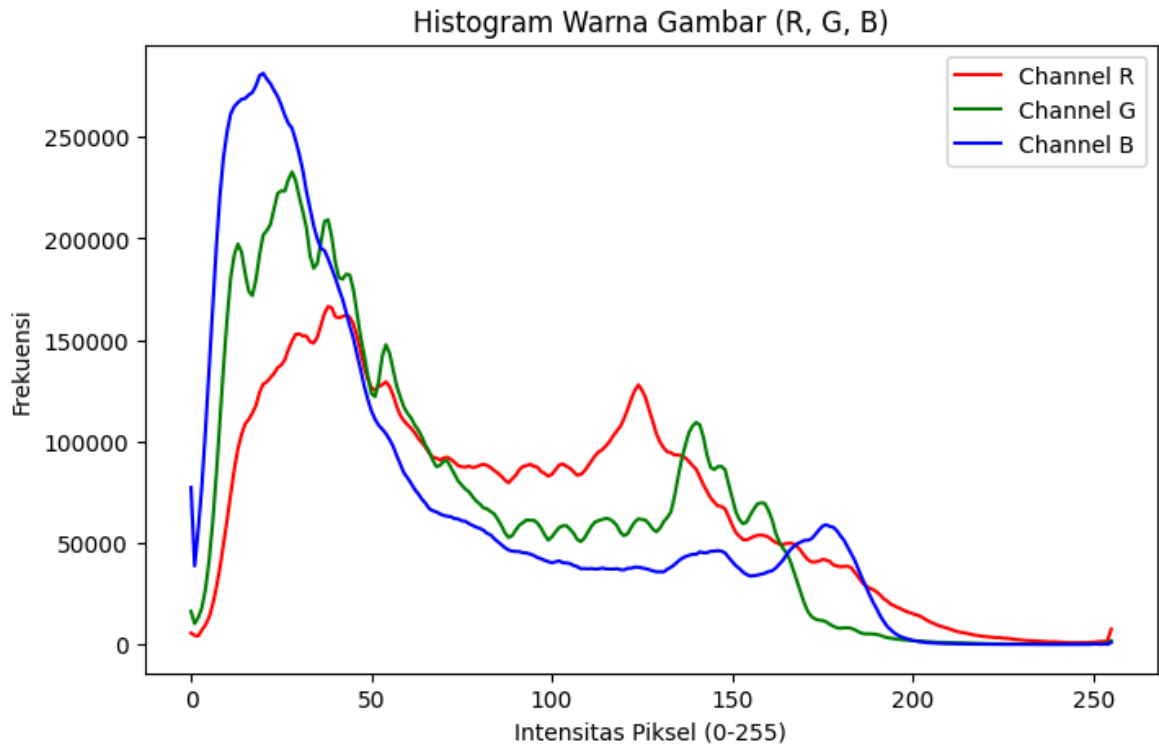
```
In [8]: # Hitung histogram untuk tiap channel warna (R, G, B)
colors = ('r', 'g', 'b') # Warna sesuai channel
plt.figure(figsize=(8, 5)) # Ukuran figure agar lebih jelas

for i, col in enumerate(colors):
    # Hitung distribusi intensitas piksel dari 0-255 untuk channel ke-i
    hist = cv2.calcHist([img_rgb], [i], None, [256], [0, 256])
    plt.plot(hist, color=col, label=f'Channel {col.upper()}') # Plot histogram

# Tambahkan Label sumbu agar pembaca paham
plt.xlabel("Intensitas Piksel (0-255)")
plt.ylabel("Frekuensi")

# Tambahkan judul dan Legenda agar informatif
plt.title("Histogram Warna Gambar (R, G, B)")
plt.legend()

# Tampilkan hasil histogram
plt.show()
```



**Hasil Analisis:** Dari histogram terlihat bahwa channel biru (B) lebih dominan karena memiliki frekuensi tertinggi pada rentang intensitas rendah–sedang dibanding R dan G. Kontras gambar sedang, karena distribusi intensitas tidak hanya menumpuk di satu sisi, melainkan menyebar dari gelap hingga terang. Sebaran intensitasnya lebih banyak terkonsentrasi di rentang rendah hingga menengah (0–150), sedangkan pada rentang tinggi (200–255) jumlah piksel relatif sedikit, sehingga gambar terlihat lebih bernuansa warna gelap dengan sedikit area terang.

## B5. Analisis Ringkas (Wajib)

**Jawab pertanyaan berikut:**

**Relasi histogram dengan kesan visual:** Apa hubungan antara pola histogram yang Anda lihat dengan kesan visual gambar (terang/gelap, warna dominan, kontras)?

*Jawaban Anda:* Pola histogram menunjukkan bahwa sebagian besar intensitas berada pada rentang rendah hingga menengah (0–150), sehingga memberi kesan visual gambar lebih gelap dan teduh. Channel biru yang dominan membuat gambar terasa memiliki nuansa dingin dan agak kebiruan. Kontras terlihat sedang, karena meskipun ada distribusi hingga ke intensitas tinggi, jumlahnya lebih sedikit, sehingga bagian terang hanya muncul di area tertentu. Dengan demikian, hubungan histogram dengan kesan visual gambar adalah: histogram yang terkonsentrasi di area gelap–menengah dan dominasi biru menciptakan suasana gambar yang serius, teduh, dan tidak terlalu cerah.

## Bagian C — Video

### C1. Deskripsi Data

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

- Jenis video: olahraga basket
- Sumber: Youtube Shorts
- Durasi target: 20 detik
- Alasan pemilihan: gerakan layup dari Kyrie Irving pada video ini sangat keren.

**Path file:** data/video\_kyrie\_layup.mp4

---

## C2. TODO: Baca & Metadata

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

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

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

```
In [9]: # Baca video menggunakan OpenCV
video_path = os.path.join(os.getcwd(), "data/video_kyrie_layup.mp4")
cap = cv2.VideoCapture(video_path)

# Ambil metadata dasar dari video
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)) # lebar frame
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)) # tinggi frame
fps = cap.get(cv2.CAP_PROP_FPS) # frame per detik
total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT)) # total jumlah frame

# Hitung durasi video dalam detik
duration = total_frames / fps

# Buat klasifikasi resolusi sederhana berdasarkan ukuran
if width >= 3840 and height >= 2160:
    resolution_class = "4K"
elif width >= 1920 and height >= 1080:
    resolution_class = "Full HD"
elif width >= 1280 and height >= 720:
    resolution_class = "HD"
else:
    resolution_class = "SD"

# Tampilkan hasil metadata agar lebih mudah dipahami
print(f"Resolusi: {width} x {height}")
print(f"Frame rate: {fps:.2f} fps")
print(f"Jumlah total frame: {total_frames}")
print(f"Durasi: {duration:.2f} detik")
print(f"Klasifikasi resolusi: {resolution_class}")

# Tutup video setelah selesai membaca metadata
cap.release()
```

Resolusi: 1080 x 1920  
Frame rate: 60.00 fps  
Jumlah total frame: 1217  
Durasi: 20.28 detik  
Klasifikasi resolusi: SD

**Hasil Analisis:** Parameter metadata video seperti resolusi, frame rate, jumlah frame, durasi, dan klasifikasi resolusi sangat penting karena memengaruhi kualitas visual, kebutuhan komputasi, serta kecocokan dengan aplikasi tertentu. Resolusi menentukan tingkat detail gambar, frame rate berpengaruh pada kelancaran pergerakan, jumlah frame berguna untuk analisis frame-by-frame, durasi menunjukkan panjang video, sedangkan klasifikasi resolusi membantu menstandarkan kualitas. Dengan memahami parameter ini, analisis video dapat dilakukan lebih optimal sesuai tujuan, misalnya untuk deteksi objek, pelacakan gerakan, hingga kebutuhan streaming.

### 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 [10]: # Saya baca video menggunakan OpenCV
video_path = os.path.join(os.getcwd(), "data/video_kyrie_layup.mp4")
cap = cv2.VideoCapture(video_path)

# Ambil metadata dasar untuk menghitung frame tengah dan timestamp
fps = cap.get(cv2.CAP_PROP_FPS) # frame per detik
total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT)) # jumlah total frame

# Tentukan index untuk 3 frame representatif
frame_indices = [0, total_frames // 2, total_frames - 1]

# Definisikan label untuk tiap frame (awal, tengah, akhir)
frame_labels = ["Awal", "Tengah", "Akhir"]

# Buat figure untuk menampilkan 3 frame secara berdampingan
plt.figure(figsize=(15, 5))

for i, frame_idx in enumerate(frame_indices):
    # Pindahkan posisi video ke frame tertentu
    cap.set(cv2.CAP_PROP_POS_FRAMES, frame_idx)
    ret, frame = cap.read() # baca frame

    if ret:
        # Konversi dari BGR (OpenCV) ke RGB (matplotlib)
        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
```

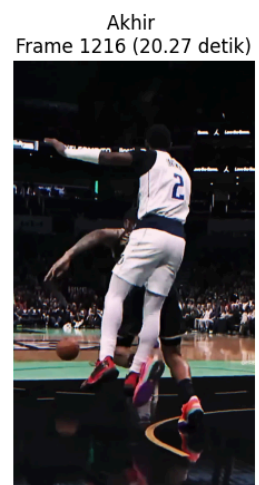
```
# Hitung timestamp frame dalam detik
timestamp = frame_idx / fps

# Tampilkan frame dalam subplot
plt.subplot(1, 3, i+1)
plt.imshow(frame_rgb)
# Tambahkan label (Awal/Tengah/Akhir) di judul
plt.title(f"{frame_labels[i]} \nFrame {frame_idx} ({timestamp:.2f} detik)
plt.axis("off") # Hilangkan axis agar Lebih bersih

# Tampilkan semua subplot
plt.suptitle("3 Frame Representatif (Awal - Tengah - Akhir)", fontsize=14, y=1.0)
plt.show()

# Tutup video setelah selesai digunakan
cap.release()
```

3 Frame Representatif (Awal - Tengah - Akhir)



Perbedaan visual antar frame adalah sebagai berikut:

1. Frame Awal -> Frame 0 (0.00 detik): Di frame ini Kyrie beserta tim nya sedang melakukan posisi bertahan.
2. Frame Tengah -> Frame 608 (10.13 detik): Di frame ini Kyrie akan melakukan gerakan layup di mana satu kaki sudah berada di atas.
3. Frame Akhir -> Frame 1216 (20.27 detik): Di frame ini Kyrie akan landing (mendarat) setelah melakukan layup

Dari tiga frame representatif (awal, tengah, akhir) dapat dipelajari bahwa pemilihan frame kunci mampu merangkum alur peristiwa dalam video secara ringkas. Frame awal menunjukkan konteks suasana lapangan, frame tengah menyorot aksi inti yaitu layup Kyrie, dan frame akhir memperlihatkan penyelesaian gerakan. Dengan cara ini, informasi penting dari video dapat diambil tanpa harus menonton keseluruhan durasi, sehingga bermanfaat untuk analisis momen penting, pembuatan highlight, maupun deteksi event tertentu.

## 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:* Dengan fps (60 fps) dan resolusi video (1080 x 1920) format vertikal, untuk use case media sosial sudah sangat sesuai, contohnya untuk kebutuhan konten olahraga maka dibutuhkan detail gerakan yang cepat, mulus dan tajam apalagi jika digunakan untuk melakukan slow motion pada video.

## Perbandingan & Kesimpulan

### Perbandingan Representasi Media

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

#### Audio (1D - Temporal)

- Representasi: Sinyal 1D berbentuk gelombang yang dihasilkan terhadap waktu
- Visualisasi utama: Waveform, Spectrogram (log-dB), MFCC
- Informasi yang diperoleh: Informasi yang didapatkan dari Spectrogram lebih detail (bisa melihat frekuensi suara rendah dan tinggi), sedangkan Waveform hanya informasi secara umumnya saja (keras atau rendahnya suara di tiap detik). Dengan MFCC bisa membantu menangkap ciri khas suara yang tidak terlihat dari Waveform ataupun Spectrogram

#### Gambar (2D - Spasial)

- Representasi: Matriks piksel dengan tiga kanal warna (R, G, B)
- Visualisasi utama: Tampilan RGB, metadata gambar, dan histogram warna.
- Informasi yang diperoleh: Dari metadata gambar bisa diperoleh informasi seperti ukuran file, dimensi, jumlah kanal, tipe data, dan ukuran memori. Kemudian dari Histogram warna kita bisa mendapatkan informasi mengenai dominasi warna, distribusi intensitas, serta karakter visual dari lukisan Diponegoro.

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

- Representasi: Rangkaian frame gambar dengan dimensi spasial dan urutan waktu
- Visualisasi utama: Metadata video dan cuplikan 3 frame representatif (awal-tengah-akhir)
- Informasi yang diperoleh: Dari metadata gambar diperoleh informasi seperti resolusi, frame rate, total frame, durasi, klasifikasi resolusi. Selain itu diperoleh kumpulan frame-frame dari pergerakan aksi layup Kyrie Irving mulai dari awal-tengah-akhir.

---

## Refleksi Pembelajaran



### 3 Poin yang Saya Pelajari:

1. Saya jadi mengetahui bagaimana cara membaca dan menampilkan file audio, gambar/citra dan video
2. Saya jadi mengetahui kegunaan beberapa library seperti librosa (Waveform & Spectrogram), OpenCV (Histogram warna, Menampilkan frame)
3. Informasi yang dihasilkan dari penggunaan beberapa library, bisa digunakan untuk menganalisis lebih lanjut atau memahami lebih dalam lagi makna-mana dari informasi yang didapat.

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

1. Saya masih bingung apa kegunaan MFCC dan bagaimana dari MFCC ini bisa menemukan ciri khas pada suatu audio.
2. Saya masih bingung mengapa video ditampilkan dalam bentuk frame dan dibagi menjadi 3 bagian, lalu bagaimana cara terbaik untuk menganalisis seluruh isi video .

## Sumber Data & Referensi

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

- **Audio:** [https://youtu.be/4YGgw9LX7\\_s?si=2ZL1pGcxJfLSCO6E](https://youtu.be/4YGgw9LX7_s?si=2ZL1pGcxJfLSCO6E)
- **Gambar:** [https://id.wikipedia.org/wiki/Penangkapan\\_Pangeran\\_Diponegoro](https://id.wikipedia.org/wiki/Penangkapan_Pangeran_Diponegoro)
- **Video:** [https://youtube.com/shorts/0pBkt6\\_8JZ8?si=mOd65iE1b3GP4fwp](https://youtube.com/shorts/0pBkt6_8JZ8?si=mOd65iE1b3GP4fwp)
- **Referensi teknis:** <https://chatgpt.com/share/68c66a12-b200-800e-a2b8-7b0beeb62918>





## 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