

# Tugas Analisis Multimedia: Audio, Gambar, Video

**Mata Kuliah:** Sistem & Teknologi Multimedia

**Nama:** M.Arief Rahman Hakim

**NIM:** 122140083

---

## 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.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
import matplotlib
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
- Sumber: youtube
- Format file: WAV
- Alasan pemilihan: lagi relate sama kehidupan

**Path file:** data/Yiruma (이루마) - River Flows in You [TubeRipper.cc].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]: # Memuat Audio
audio_path = os.path.join(os.getcwd(), "data", "riverflowsinyou.wav")

y, sr = librosa.load(audio_path, sr=None) # sr=None untuk mempertahankan sample rate

# Metadata dasar dengan soundfile
with sf.SoundFile(audio_path) as f:
    sample_rate = f.samplerate
    n_samples = len(f)
    duration = len(f) / f.samplerate
    channels = f.channels

print("== Metadata Dsar Audio ==")
print(f"Sample rate : {sample_rate} Hz")
print(f"Durasi : {duration:.2f} detik")
print(f"Jumlah kanal: {channels} ({'Mono' if channels==1 else 'Stereo'})")
print(f"Total sampel: {n_samples}")

== Metadata Dsar Audio ==
Sample rate : 48000 Hz
Durasi : 209.59 detik
Jumlah kanal: 2 (Stereo)
Total sampel: 10060292
```

### A3. TODO: Waveform

**Instruksi:** Plot waveform audio dengan:

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

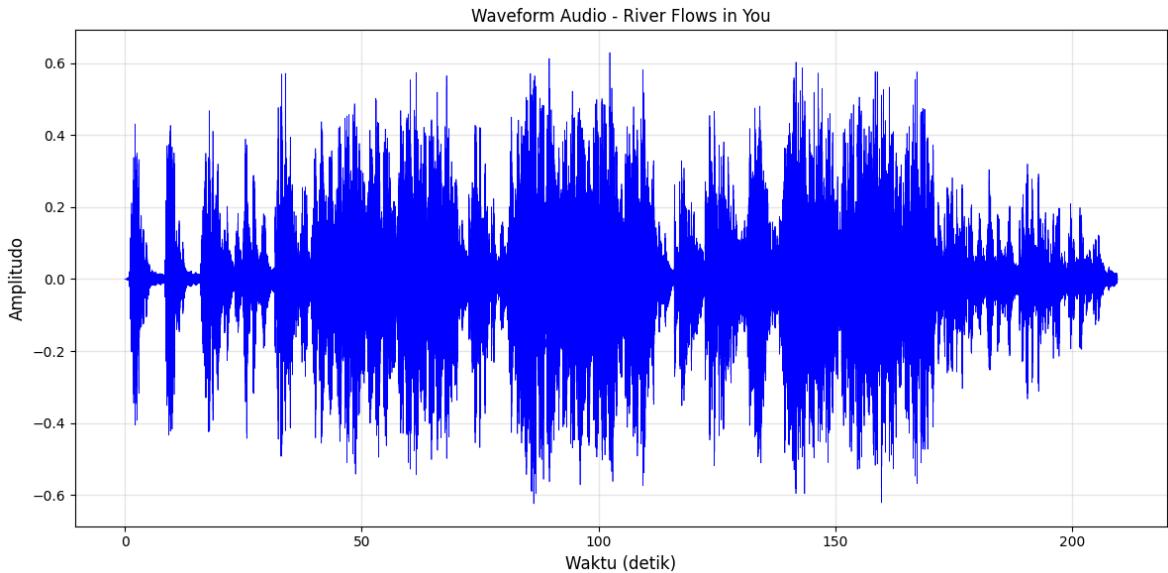
**Analisis yang diperlukan:** Jelaskan apa yang Anda lihat dari waveform (pola amplitudo, bagian keras/pelan, dll.)

```
In [3]: # Plot waveform audio
plt.figure(figsize=(12, 6))

# Konversi waktu ke detik untuk sumbu X
time_seconds = np.arange(len(y)) / sr

plt.plot(time_seconds, y, color='blue', linewidth=0.5)
plt.title('Waveform Audio - River Flows in You')
plt.xlabel('Waktu (detik)', fontsize=12)
plt.ylabel('Amplitudo', fontsize=12)
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

# Analisis waveform
print("\n== Analisis Waveform ==")
print(f"Amplitudo maksimum: {np.max(y):.6f}")
print(f"Amplitudo minimum: {np.min(y):.6f}")
```



```
==== Analisis Waveform ====
Amplitudo maksimum: 0.628738
Amplitudo minimum: -0.623642
```

Analisis Pribadi dari Waveform diatas

- Waveform di atas merupakan representasi visual dari audio River Flows in You. Bentuk gelombang ini mengikuti irama musik, di mana terlihat adanya bagian dengan amplitudo kecil maupun amplitudo besar. Perbedaan amplitudo ini dipengaruhi oleh alunan musik yang dimainkan dengan keras ataupun lembut. Jika diperhatikan, terdapat pola yang hampir serupa pada tiga gelombang utama yang berukuran besar. Di sela-sela gelombang tersebut muncul nada dengan amplitudo rendah yang merepresentasikan bagian musik dengan volume lebih kecil, seperti yang terlihat pada rentang menit ke-50 hingga ke-100 serta menit ke-100 hingga ke-150.

## 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 [4]: # Hitung STFT dan spectrogram dalam skala Log-dB
# Parameter standar: n_fft=1024, hop_length=256
stft = librosa.stft(y, n_fft=1024, hop_length=256)

# Konversi ke magnitude dan kemudian ke skala Log-dB
magnitude = np.abs(stft)
spectrogram_db = librosa.amplitude_to_db(magnitude, ref=np.max)

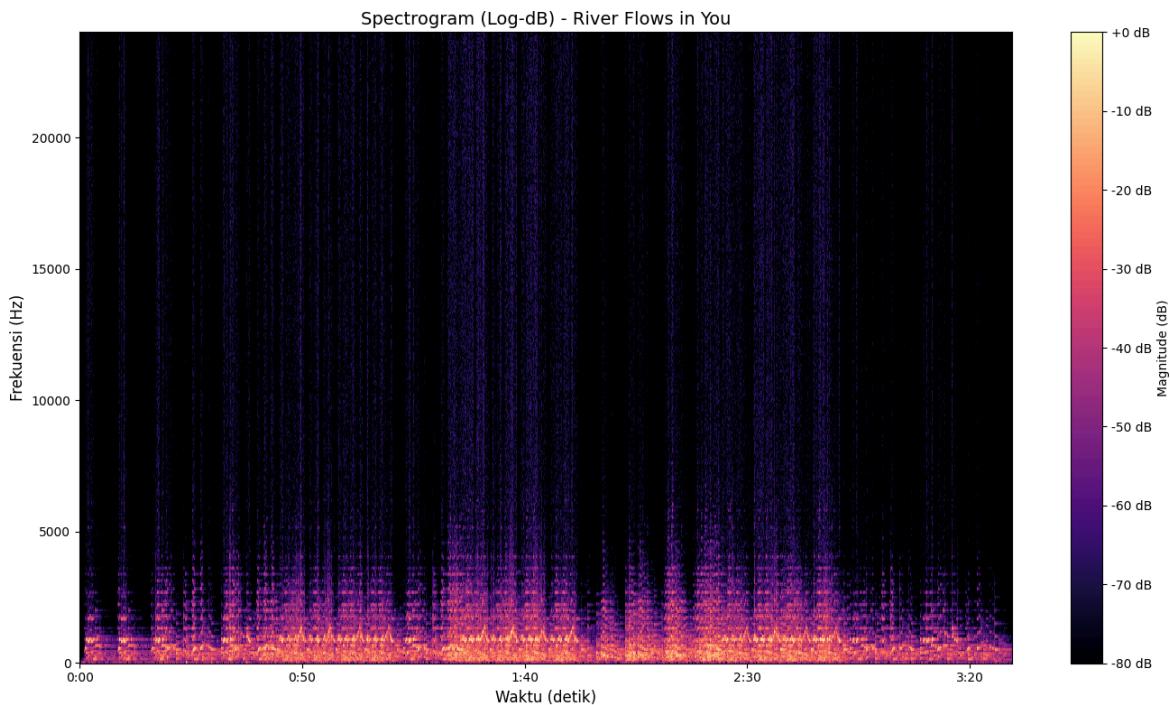
# Plot spectrogram
plt.figure(figsize=(14, 8))
```

```

librosa.display.specshow(
    spectrogram_db,
    sr=sr,
    hop_length=256,
    x_axis='time',
    y_axis='hz',
    cmap='magma'
)

plt.colorbar(format='%+2.0f dB', label='Magnitude (dB)')
plt.title('Spectrogram (Log-dB) - River Flows in You', fontsize=14)
plt.xlabel('Waktu (detik)', fontsize=12)
plt.ylabel('Frekuensi (Hz)', fontsize=12)
plt.tight_layout()
plt.show()

```



Dari spectrogram ini kita bisa liat bagian mana ketika audio menujukkan frekuensi yang tinggi. Jika kita sesuaikan dengan lagunya memang di beberapa detik tertentu suara musik berjalan dengan slow atau datar, namun ada banyak titik dimana musik langsung tinggi (instan) yang menyebabkan frekuensinya terlihat tinggi.

## 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 dengan minimal 13 koefisien  
mfcc = librosa.feature.mfcc(y=y, sr=sr, n\_mfcc=13)

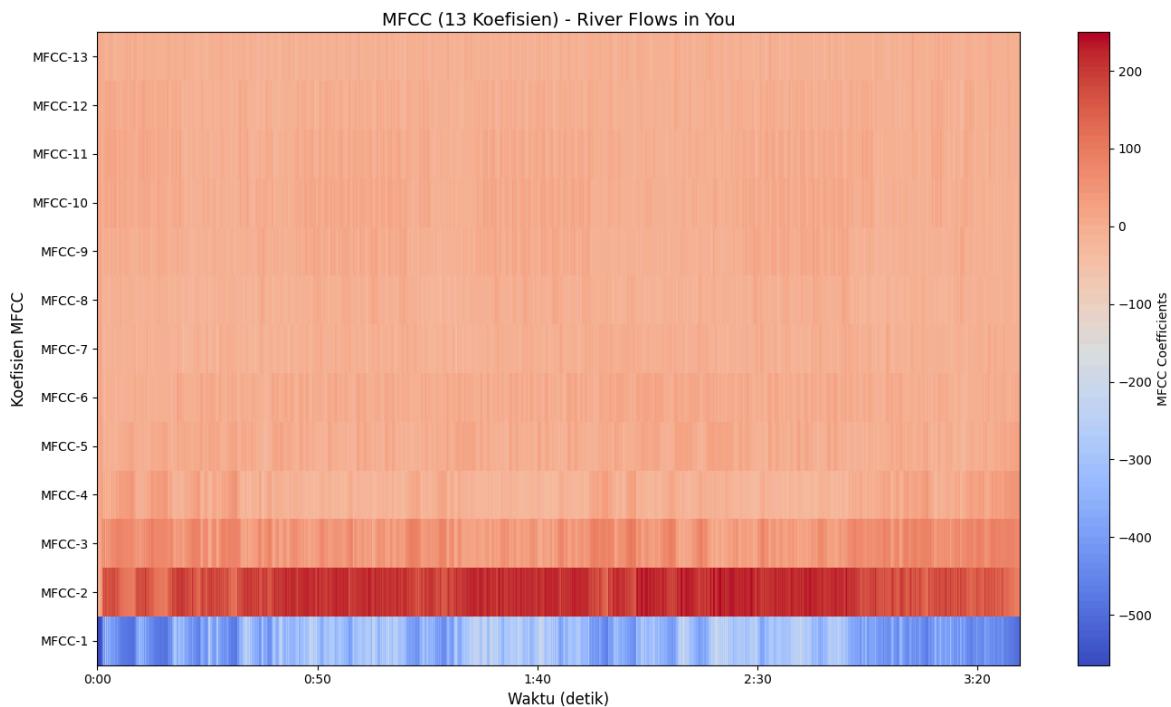
```

# Plot MFCC sebagai heatmap
plt.figure(figsize=(14, 8))
librosa.display.specshow(
    mfcc,
    sr=sr,
    x_axis='time',
    cmap='coolwarm'
)

plt.colorbar(label='MFCC Coefficients')
plt.title('MFCC (13 Koefisien) - River Flows in You', fontsize=14)
plt.xlabel('Waktu (detik)', fontsize=12)
plt.ylabel('Koefisien MFCC', fontsize=12)
plt.yticks(range(13), [f'MFCC-{i+1}' for i in range(13)])
plt.tight_layout()
plt.show()

# Analisis MFCC
print("\n==== Analisis MFCC ===")
print(f"Jumlah koefisien MFCC: {mfcc.shape[0]}")
print(f"Jumlah frame waktu: {mfcc.shape[1]}")
print(f"Range nilai MFCC: {np.min(mfcc):.2f} hingga {np.max(mfcc):.2f}")

```



```

==== Analisis MFCC ===
Jumlah koefisien MFCC: 13
Jumlah frame waktu: 19650
Range nilai MFCC: -564.87 hingga 249.86

```

## A6. Analisis Ringkas (Wajib)

**Jawab pertanyaan berikut:**

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

*Jawaban Anda:* Spectogram memberikan gambaran frekuensi terhadap waktu, sehingga kita bisa melihat pola nada, harmoni, dan pergeseran dari spektrum.

Waveform fokus memberikan gambaran tentang perubahan amplitudo sinyal audio terhadap waktu, sehingga bisa melihat dinamika volume.

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

*Jawaban Anda:* Dari visualisasi MFCC ini terlihat bahwa koefisien rendah (MFCC 1–4) menunjukkan variasi yang cukup dinamis, mencerminkan perubahan energi dasar, pitch, dan konten melodi dalam musik. Sementara itu, koefisien tinggi (MFCC 5–13) relatif stabil, menunjukkan bahwa timbre instrumen dalam hal ini piano tetap konsisten sepanjang lagu. Hal ini mengajarkan bahwa MFCC efektif untuk memisahkan aspek dinamika melodi dari kestabilan warna suara instrumen, sehingga membantu analisis baik untuk musik maupun pengenalan suara.

## Bagian B — Gambar

### B1. Deskripsi Data

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

- Jenis gambar: Gambar pemandangan
- Sumber: Pngtree
- Format file: JPG
- Alasan pemilihan: Bagus gambarnya

**Path file:** data/fotolaut.jpeg

---

### 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]: # Baca dan tampilkan gambar dalam format RGB
image_path = os.path.join(os.getcwd(), "data", "fotolaut.jpg")

# Baca gambar menggunakan OpenCV (BGR format)
image_bgr = cv2.imread(image_path)

# Konversi dari BGR ke RGB untuk tampilan yang benar
image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)

# Tampilkan gambar
plt.figure(figsize=(12, 8))
```

```

plt.imshow(image_rgb)
plt.title('Seni Abstrak - Format RGB', fontsize=14)
plt.axis('off')
plt.tight_layout()
plt.show()

print(f"Gambar berhasil dimuat dari: {image_path}")

```

Seni Abstrak - Format RGB



Gambar berhasil dimuat dari: c:\DATA\KULIYEAH\2025\Multimedia\mg2\data\fotolaut.jpg

### B3. TODO: Informasi Dasar

**Instruksi:** Tampilkan informasi metadata gambar:

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

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

In [7]:

```

# Tampilkan informasi metadata gambar
print("== Informasi Metadata Gambar ==")
print(f"Dimensi (Height x Width): {image_rgb.shape[0]} x {image_rgb.shape[1]}")
print(f"Jumlah kanal: {image_rgb.shape[2]}")
print(f"Tipe data (dtype): {image_rgb.dtype}")
print(f"Range nilai pixel: {image_rgb.min()} - {image_rgb.max()}")


# Hitung ukuran file dalam memori (dalam bytes dan MB)
size_bytes = image_rgb.nbytes
size_mb = size_bytes / (1024 * 1024)

print(f"Ukuran dalam memori: {size_bytes:,} bytes ({size_mb:.2f} MB)")

```

```
== Informasi Metadata Gambar ==
Dimensi (Height x Width): 360 x 641
Jumlah kanal: 3
Tipe data (dtype): uint8
Range nilai pixel: 0 - 255
Ukuran dalam memori: 692,280 bytes (0.66 MB)
```

Analisis:

Mengetahui metadata gambar adalah hal yang perlu dilakukan karena dengan ini kita bisa melakukan tahap preprocessing atau analisis lanjutan dengan lebih tepat apa yang harus dilakukan

## 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?

## 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: \_\_\_\_\_*

```
In [8]: # Hitung dan tampilkan histogram untuk setiap channel (R, G, B)
plt.figure(figsize=(12, 8))

# Buat subplot untuk masing-masing channel
fig, axes = plt.subplots(2, 2, figsize=(15, 10))

# Plot overlay histogram (subplot pertama)
colors = ('r', 'g', 'b')
channel_names = ('Red', 'Green', 'Blue')

axes[0, 0].set_title('Histogram Overlay RGB')
for i, col in enumerate(colors):
    hist = cv2.calcHist([image_rgb], [i], None, [256], [0, 256])
    axes[0, 0].plot(hist, color=col, label=channel_names[i], alpha=0.7)
axes[0, 0].set_xlabel('Intensitas Pixel (0-255)')
axes[0, 0].set_ylabel('Frekuensi')
axes[0, 0].legend()
axes[0, 0].grid(True, alpha=0.3)

# Plot histogram individual untuk setiap channel
for i, (col, name) in enumerate(zip(colors, channel_names)):
```

```

hist = cv2.calcHist([image_rgb], [i], None, [256], [0, 256])
row = (i + 1) // 2
col_idx = (i + 1) % 2

axes[row, col_idx].hist(image_rgb[:, :, i].ravel(), bins=256, range=[0, 256]
                        color=col, alpha=0.7, edgecolor='black')
axes[row, col_idx].set_title(f'Histogram Channel {name}')
axes[row, col_idx].set_xlabel('Intensitas Pixel (0-255)')
axes[row, col_idx].set_ylabel('Frekuensi')
axes[row, col_idx].grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
print(f"Total piksel pada gambar: {image_rgb.shape[0] * image_rgb.shape[1]}")

# Analisis warna dominan dan statistik histogram
print("\n== Analisis Histogram RGB ==")

# Hitung statistik untuk setiap channel
for i, (color, name) in enumerate(zip(colors, channel_names)):
    channel_data = image_rgb[:, :, i].ravel()
    mean_val = np.mean(channel_data)
    std_val = np.std(channel_data)
    dominant_range = np.argmax(cv2.calcHist([image_rgb], [i], None, [256], [0, 256]))

    print(f"{name} Channel:")
    print(f" - Mean: {mean_val:.2f}")
    print(f" - Std: {std_val:.2f}")
    print(f" - Dominant intensity: {dominant_range}")
    print(f" - Range: {np.min(channel_data)} - {np.max(channel_data)}")

# Tentukan warna dominan berdasarkan mean values
channel_means = [np.mean(image_rgb[:, :, i]) for i in range(3)]
dominant_channel_idx = np.argmax(channel_means)
dominant_channel = channel_names[dominant_channel_idx]

print(f"\nWarna dominan: {dominant_channel} (mean: {channel_means[dominant_channel]})")

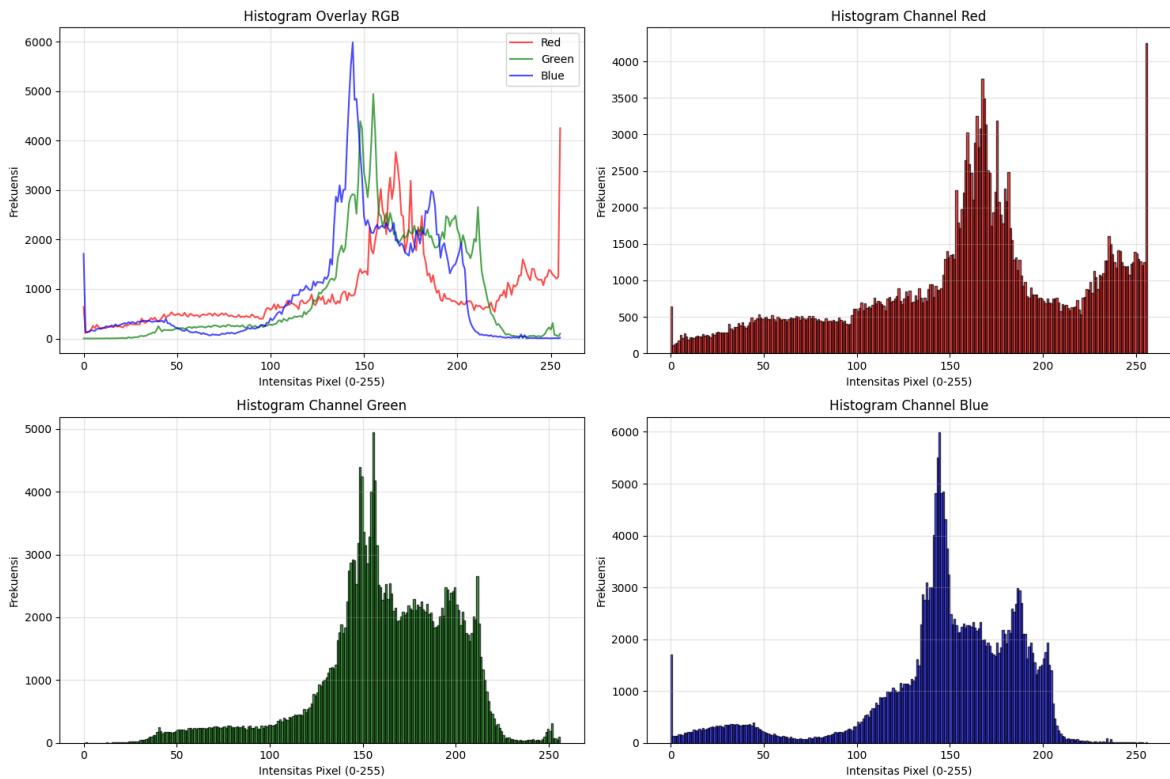
# Analisis kontras (berdasarkan standar deviasi)
overall_contrast = np.mean([np.std(image_rgb[:, :, i]) for i in range(3)])
print(f"Kontras keseluruhan (std rata-rata): {overall_contrast:.2f}")

if overall_contrast > 50:
    contrast_level = "Tinggi"
elif overall_contrast > 30:
    contrast_level = "Sedang"
else:
    contrast_level = "Rendah"

print(f"Level kontras: {contrast_level}")

```

&lt;Figure size 1200x800 with 0 Axes&gt;



Total piksel pada gambar: 230760

#### ==== Analisis Histogram RGB ===

##### Red Channel:

- Mean: 159.32
- Std: 60.70
- Dominant intensity: 255
- Range: 0 - 255

##### Green Channel:

- Mean: 162.07
- Std: 37.00
- Dominant intensity: 155
- Range: 1 - 255

##### Blue Channel:

- Mean: 146.62
- Std: 43.45
- Dominant intensity: 144
- Range: 0 - 255

Warna dominan: Green (mean: 162.07)

Kontras keseluruhan (std rata-rata): 47.05

Level kontras: Sedang

## 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 distribusi intensitas warna yang sesuai dengan kesan visual gambar pantai saat matahari terbenam. Channel biru dan hijau memiliki puncak tinggi pada intensitas menengah (sekitar 120–180), mencerminkan dominasi warna laut dan

langit. Channel merah juga cukup tinggi pada intensitas menengah hingga tinggi (150–230), sesuai dengan cahaya matahari terbenam yang memberi warna keemasan. Secara keseluruhan, histogram menyebar cukup lebar hingga ke intensitas tinggi, sehingga gambar tampak terang dengan kontras yang seimbang, tidak terlalu gelap. Warna dominan yang muncul dari kombinasi histogram ini adalah biru dan hijau (laut dan vegetasi) dengan warna merah-kuning dari matahari.

## Bagian C — Video

### C1. Deskripsi Data

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

- Jenis video: Awards
- Sumber: youtube
- Durasi target: 26 detik
- Alasan pemilihan: Simple dan Keren

**Path file:** `data/ae-award.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 [14]: # Baca video dan tampilkan metadata
video_path = os.path.join(os.getcwd(), "data", "ae-award.mp4")

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

# Cek apakah video berhasil dibuka
if not cap.isOpened():
    print(f"Error: Tidak dapat membuka video dari {video_path}")
else:
    # Ambil metadata 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)
    frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    duration = frame_count / fps if fps > 0 else 0

    # Klasifikasi resolusi
```

```

if width >= 3840 and height >= 2160:
    resolution_class = "4K Ultra HD"
elif width >= 1920 and height >= 1080:
    resolution_class = "Full HD (1080p)"
elif width >= 1280 and height >= 720:
    resolution_class = "HD (720p)"
elif width >= 854 and height >= 480:
    resolution_class = "SD (480p)"
else:
    resolution_class = "Low Resolution"

print("== Metadata Video ==")
print(f"Resolusi (Width x Height): {width} x {height}")
print(f"Klasifikasi resolusi: {resolution_class}")
print(f"Frame rate (fps): {fps:.2f}")
print(f"Jumlah total frame: {frame_count}")
print(f"Durasi: {duration:.2f} detik ({duration/60:.2f} menit)")

# Tutup video capture
cap.release()

print(f"Video berhasil dimuat dari: {video_path}")

```

== Metadata Video ==  
 Resolusi (Width x Height): 1920 x 1080  
 Klasifikasi resolusi: Full HD (1080p)  
 Frame rate (fps): 30.00  
 Jumlah total frame: 781  
 Durasi: 26.03 detik (0.43 menit)  
 Video berhasil dimuat dari: c:\DATA\KULIYEAH\2025\Multimedia\mg2\data\ae-award.mp4

Dengan mengetahui metadata dari video tersebut, kita bisa melihat bahwa resolusi video yang kita miliki adalah FHD 1920 x 1080 dengan fps nya : 30, yang artinya disetiap 1 sec terdapat 30 frame. Dengan demikian, metadata ini tidak hanya berupa informasi teknis, tetapi juga menjadi dasar untuk menilai kualitas video, kebutuhan sumber daya penyimpanan, serta kesesuaian distribusi konten di berbagai platform.

### 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.

In [20]:

```

# Tampilkan 3 frame awal-tengah-akhir
video_path = os.path.join(os.getcwd(), "data", "ae-award.mp4")

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

```

```

if not cap.isOpened():
    print(f"Error: Tidak dapat membuka video dari {video_path}")
else:
    # Ambil metadata untuk menentukan frame indices
    fps = cap.get(cv2.CAP_PROP_FPS)
    frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))

    # Tentukan indeks frame yang akan diambil
    frame_indices = [0, frame_count // 2, frame_count - 1]
    frame_labels = ["Frame Awal", "Frame Tengah", "Frame Akhir"]

    # Buat subplot untuk menampilkan ketiga frame
    fig, axes = plt.subplots(1, 3, figsize=(18, 6))

    for i, (frame_idx, label) in enumerate(zip(frame_indices, frame_labels)):
        # Set posisi frame
        cap.set(cv2.CAP_PROP_POS_FRAMES, frame_idx)

        # Baca frame
        ret, frame = cap.read()

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

            # Hitung timestamp
            timestamp = frame_idx / fps

            # Tampilkan frame
            axes[i].imshow(frame_rgb)
            axes[i].set_title(f'{label}\n(Index: {frame_idx}, Timestamp: {timestamp})')
            axes[i].axis('off')
        else:
            axes[i].text(0.5, 0.5, f'Gagal membaca frame {frame_idx}', ha='center', va='center', transform=axes[i].transAxes)
            axes[i].set_title(f'{label} - Error')
            axes[i].axis('off')

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

```

Frame Representatif Video



#### Analisis Frame:

- Dengan memuat frame kita dapat melihat bahwa setiap index dari sebuah frame akan menampilkan gambar yang berbeda. Bisa dilihat bahwa frame awal berbeda dari 2 frame lainnya. Jika kita coba bandingkan frame dengan index 0 dan 1 maka

mungkin akan terlihat sama, yang sebenarnya tidak. Dalam 1 sec frame dapat terbentuk 30. Sesuai dengan metadata dari vide yang saya muat

## C4. Analisis Ringkas (Wajib)

### Jawab pertanyaan berikut:

**Kesesuaian parameter:** Apakah fps dan resolusi video ini sesuai untuk use case pilihan Anda (kuliah daring)? Jelaskan alasan singkat.

*Jawaban Anda:* Sebagai media pembelajaran, pemilihan resolusi dan frame rate (fps) perlu menyesuaikan antara kualitas tampilan, kenyamanan menonton, serta efisiensi ukuran file: Umumnya, 720p (HD) sudah cukup untuk materi pembelajaran karena teks, grafik, dan wajah pengajar tetap terlihat jelas tanpa ukuran file yang terlalu besar. Namun, jika materi banyak mengandung detail halus seperti diagram teknis atau kode pemrograman, 1080p (Full HD) lebih dianjurkan agar tulisan dan objek kecil tetap terbaca dengan baik. Sedangkan fps 30 sudah sangat baik karena menghasilkan gerakan mulus dan alami.

# Perbandingan & Kesimpulan

## Perbandingan Representasi Media

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

### Audio (1D - Temporal)

- Representasi: Direpresentasikan dalam Sinyal 1D terhadap waktu (amplitudo vs waktu)
- Visualisasi utama: Waveform, spektogram, dan MFCC
- Informasi yang diperoleh:
  - Karakteristik Audio dari musik (keras/lemah, stabil/berubah)
  - Distribusi energi musik yang representasikan lewat frekuensi
  - Adanya Pola yang terbentuk hampir sama yang menunjukkan adanya variasi suara

### Gambar (2D - Spasial)

- Representasi: Direpresentasikan dalam Matriks 2D berisi nilai warna (R,G,B)
- Visualisasi utama: Menampilkan Gambar dalam format RGB dan melihat distribusi warna melalui histogram
- Informasi yang diperoleh: Metadata gambar, Warna Dominan (RGB) dan Level Kontras

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

- Representasi: Susunan frame gambar 2D yang berurutan sepanjang waktu.
  - Visualisasi utama: Frame Gambar (Awal, Tengah, Akhir)
  - Informasi yang diperoleh: Metadata dari Video seperti resolusi, fps, dan durasi. Serta dapat melihat perbedaan frame disetiap index nya.
- 

## Refleksi Pembelajaran

### 3 Poin yang Saya Pelajari:

1. Melakukan visualisasi audio ke dalam bentuk spektogram, waveform, dan MFCC untuk dilakukan analisis dari setiap karakteristik yang dihasilkan tanpa harus mendengarkannya lewat audio.
2. Melakukan visualisasi gambar ke representasi RGB dan melihat sebaran warna apa yang dominan dalam gambar tersebut.
3. Melakukan Visualisasi Video lewat bentuk frame awal, tengah, akhir. melihat setiap perubahan dari durasi dan index frame. Tujuannya untuk memahami bagaimana video yang bagus lewat dari parameter yang ada yaitu: resolusi, fps, dan memory.

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

1. Cara membaca MFCC untuk memahami pola audio
  2. Teknik lainnya dalam melakukan analisis Audio dan Gambar
- 

## Sumber Data & Referensi

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

- **Audio:** [Youtube](#)
- **Gambar:** [Pngtree](#)
- **Video:** [Youtube](#)
- **Referensi teknis:**
  - [Materi Kuliah](#)
  - [Github IF-40304](#)
  - [Percakapan AI](#)

## 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

Aspek Penilaian	Bobot	Deskripsi
<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, 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