

Tugas Analisis Multimedia: Audio, Gambar, Video

Mata Kuliah: Sistem & Teknologi Multimedia

Nama: Elma Nurul Fatika

NIM: 122140069

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
import matplotlib
```

```
# 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: Suara kucing marah
- Sumber: Youtube
- Format file: WAV
- Alasan pemilihan: karena suara kucing marah memiliki suara seperti teriak dan rendah seperti menggeram (bervariasi), dan kucing saya di rumah juga sering berantem

Path file: data/____.wav (isi nama file Anda nanti di kode)

A2. TODO: Muat & Metadata

Instruksi: Tulis kode untuk memuat file audio dan menampilkan metadata dasar:

- Sample rate (Hz)
- Durasi (detik)
- Jumlah kanal (mono/stereo)

- Jumlah total sampel

Catatan: Jika file MP3 bermasalah saat loading, gunakan format WAV sebagai alternatif.

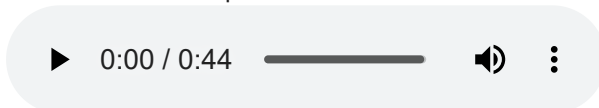
```
In [3]: audio_path = 'data\suara-kucing-marah.wav' #path audio
audio_data, sample_rate = sf.read(audio_path) #membaca file audio
duration = len(audio_data) / sample_rate #menghitung durasi audio
channels = audio_data.shape[1] if audio_data.ndim > 1 else 1 #menghitung jumlah
total_samples = audio_data.shape[0] #menghitung jumlah total sampel

from IPython.display import Audio

# Menampilkan informasi audio
print(f"Sample rate: {sample_rate} Hz")
print(f"Durasi: {duration:.2f} detik")
print(f"Jumlah kanal: {channels} ({'Stereo' if channels == 2 else 'Mono'})")
print(f"Jumlah total sampel: {total_samples}")
Audio(audio_path)
```

Sample rate: 44100 Hz
 Durasi: 44.98 detik
 Jumlah kanal: 2 (Stereo)
 Jumlah total sampel: 1983488

Out[3]:



A3. TODO: Waveform

Instruksi: Plot waveform audio dengan:

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

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

Analisis : yang saya lihat dari wavefrom yang telah dibuat pola amplitudonya seperti berulang tapi dalam rentan waktu yang berbeda, setelah suara kucing keras maka dia akan sedikit pelan dan begitu seterusnya. jadi kaya suara kucingnya teriak dan menggeram tapi terputus putus dalam jeda waktu yang berbeda-beda. seperti di detik 8-10 dia meggeram rendah, kemudian di detik 10-11 dia jeda sebentar, kemudian bersuara lagi pada detik 12-13, kemudian hening sampai detik 16 dan bersuara lagi keras pelan keras pelan lagi. sampai pada di akhir suara dia sedikit mengeluarkan suara seperti menggeram sedikit keras dan sedikit konsisten.

```
In [4]: # Plot waveform audio
time_axis = np.linspace(0, duration, total_samples) #membuat sumbu waktu, dari 0

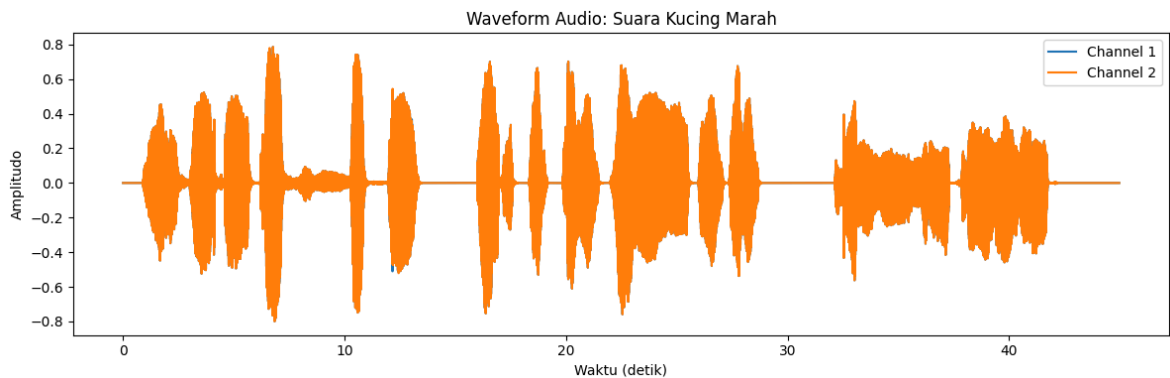
plt.figure(figsize=(12, 4))

for ch in range(channels):
```

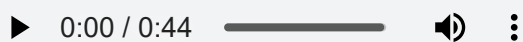
```
plt.plot(time_axis, audio_data[:, ch], label=f'Channel {ch+1}')

plt.title('Waveform Audio: Suara Kucing Marah')
plt.xlabel('Waktu (detik)') #sumbu x adalah waktu dalam detik
plt.ylabel('Amplitudo') #sumbu y adalah amplitudo
plt.legend()
plt.tight_layout()
plt.show()

from IPython.display import Audio
Audio(audio_path)
```



Out[4]:



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.

Analisis : jika di wavefrom mnurut saya hanya bisa melihat pola suara (di bagian mana saja) kucingnya dengan menyesuaikan dari hasil wavefromnya, tapi jika di spectrogram saya bisa melihat suara kucing yang paling keras di bagian warna yang terangnya (kuning/oren).

```
In [ ]: audio_data, sample_rate = librosa.load('data\suara-kucing-marah.wav', sr=None, m

# Jika stereo ambil channel kiri (0); jika mono langsung audio_data
if audio_data.ndim > 1:
    y = audio_data[0, :] # channel kiri
else:
    y = audio_data

# Parameter STFT
n_fft = 1024
hop_length = 256
```

```

# Hitung STFT
stft = librosa.stft(y, n_fft=n_fft, hop_length=hop_length)
spectrogram_db = librosa.amplitude_to_db(np.abs(stft), ref=np.max)

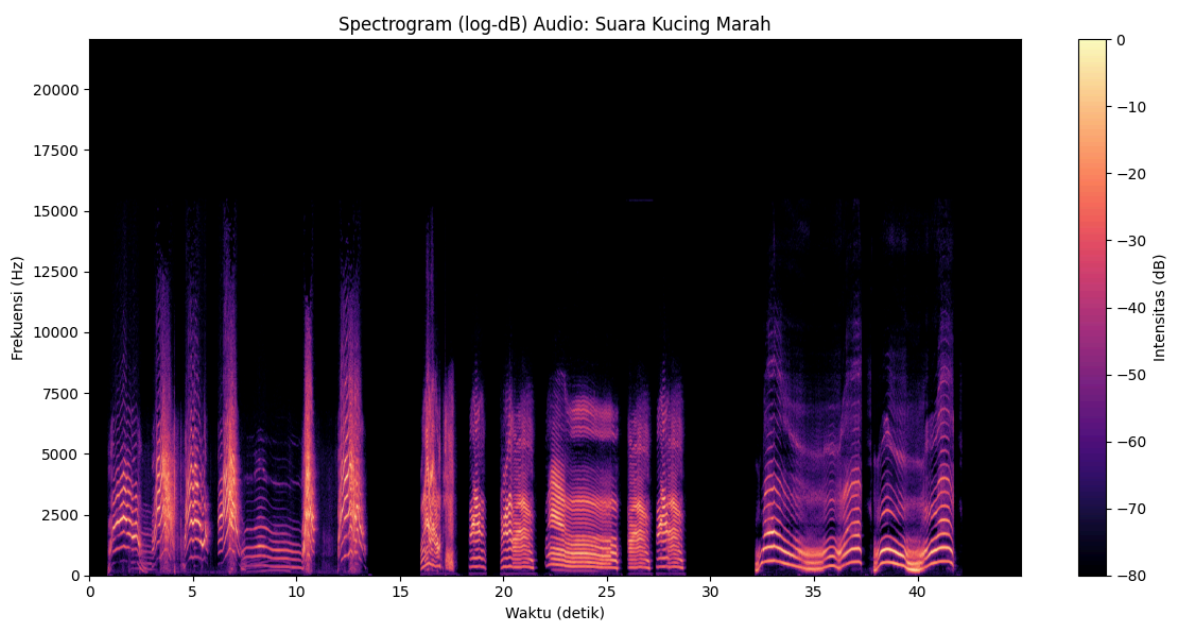
# Buat axis waktu dan frekuensi
times = librosa.frames_to_time(
    np.arange(spectrogram_db.shape[1]),
    sr=sample_rate,
    hop_length=hop_length
)
freqs = librosa.fft_frequencies(sr=sample_rate, n_fft=n_fft)

# Plot
plt.figure(figsize=(12, 6))

# extent = [xmin, xmax, ymin, ymax]
plt.imshow(
    spectrogram_db,
    aspect='auto',
    origin='lower',
    extent=[times[0], times[-1], freqs[0], freqs[-1]],
    cmap='magma'
)

plt.title('Spectrogram (log-dB) Audio: Suara Kucing Marah')
plt.xlabel('Waktu (detik)')
plt.ylabel('Frekuensi (Hz)')
plt.colorbar(label='Intensitas (dB)')
plt.tight_layout()
plt.show()

```



A5. TODO: MFCC

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

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

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

Analisis : pola MFCC dari audio suara kucing marah ini menurut saya berubah-ubah, bias dilihat dari nilai MFCC nya, dimana pada koefisien 1 sampai 4 dia banyak variasi warnanya karena menangkap suara vokalnya yang masih kasar (jadi seperti suara aslinya atau masih mengidentifikasi suaranya), kemudian di koefisien 5 sampai seterusnya dia sudah menangkap detail suaranya yang halus. potensi maknanya yaitu suara dari kucing marah ini berubah ubah, jadi dia ada suara mendesis, mengeong keras, dan suara lainnya. kemudian di waktu yang diam (tidak ada suara) dia tidak ada variasi warna berarti tidak ada variasi suaranya.

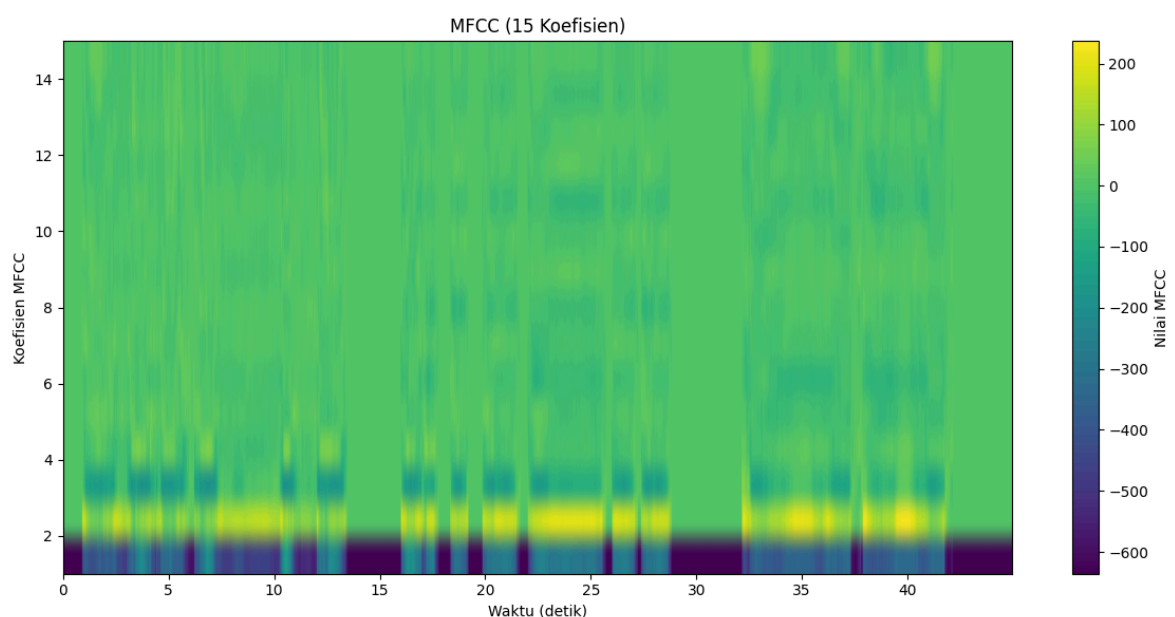
```
In [17]: if audio_data.ndim > 1:
          y = audio_data[0, :] # ambil channel pertama
        else:
          y = audio_data

        n_fft = 1024
        hop_length = 256

        mfcc_15 = librosa.feature.mfcc(y=y, sr=sample_rate, n_mfcc=15,
                                         n_fft=n_fft, hop_length=hop_length)

        times = librosa.frames_to_time(np.arange(mfcc_15.shape[1]), sr=sample_rate, hop_

        plt.figure(figsize=(12, 6))
        plt.imshow(mfcc_15, aspect='auto', origin='lower',
                   extent=[times[0], times[-1], 1, 15], cmap='viridis')
        plt.title('MFCC (15 Koefisien)')
        plt.xlabel('Waktu (detik)')
        plt.ylabel('Koefisien MFCC')
        plt.colorbar(label='Nilai MFCC')
        plt.tight_layout()
        plt.show()
```



A6. Analisis Ringkas (Wajib)

Jawab pertanyaan berikut:

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

Jawaban Anda: perbedaan dari waveform vs spectrogram adalah pada waveform kita bisa melihat perubahan pola suara keras dan lembutnya ada di bagian mana saja, ini karena waveform merepresentasikan grafik amplitudo terhadap waktu. kemudian spectrogram kita bisa melihat dimana suara tinggi rendahnya melalui representasi dari grafik frekuensi terhadap waktu.

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

Jawaban Anda: menurut saya visualisasi MFCC ini saya mengetahui ketika di koefisien rendah ini biasanya menggambarkan karakteristik dari suaranya (suara dasar/umum), ketika koefisiennya tinggi maka menggambarkan perubahan halus suara.

Bagian B — Gambar

B1. Deskripsi Data

TODO: Jelaskan gambar yang akan Anda analisis:

- Jenis gambar: bunga
- Sumber: pinterest
- Format file: jpg
- Alasan pemilihan: karena bunganya warnanya cantik dan ada variasi warna di dalam foto tersebut, ada kuning, hijau, dan biru.

Path file: data/____.jpg (isi nama file Anda nanti di kode)

B2. TODO: Baca & Tampilkan (RGB)

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

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

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

Analisis : objek dominan dari gambar yang saya inputkan adalah bunga matahari yang menonjol, kondisi pencahayaannya termasuk terang karena fotonya diambil pada siang

hari dan komposisi warnanya kuning cerah pada kelopak bunganya, warna hijau pada daun dan batang, selain itu juga warna biru langit yang cerah.

```
In [18]: # Baca gambar dengan OpenCV (BGR)
img_path = 'data/bunga-matahari.jpg'
img_bgr = cv2.imread(img_path)

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

# Tampilkan gambar dalam format RGB
plt.figure(figsize=(8, 6))
plt.imshow(img_rgb)
plt.title('Gambar Bunga (Format RGB)')
plt.axis('off')
plt.show()
```

Gambar Bunga (Format RGB)



B3. TODO: Informasi Dasar

Instruksi: Tampilkan informasi metadata gambar:

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

- Ukuran file dalam memori

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

Analisis : informasi ini penting untuk tahap preprocessing atau analisis lanjutan karena kita bisa mengetahui identitas yang ada pada gambar seperti dimensinya berapa, ini penting untuk mempermudah penyesuaian resolusi yang akan digunakan, kemudian jumlah kanal ini bisa mempermudah untuk mengolah gambar agar warnanya tidak keliru, tipe data ini berguna untuk mencegah terjadinya eror perhitungan yang disebabkan karena beda jenis tipe data. mode warna, ini berguna untuk mengetahui formatnya yang dipakai apa, jadi ketika akan melakukan analisis tidak terjadi kesalahan, dan yang terakhir ukuran file, ini berguna untuk mengetahui berapa besar/banyak ram yang dipakai untuk memproses gambarnya. jadi informasi ini cukup penting karena memudahkan kita saat ingin mengolah dan menganalisis gambar dengan baik.

```
In [19]: # Ambil metadata gambar
height, width = img_rgb.shape[:2]
channels_img = img_rgb.shape[2] if img_rgb.ndim == 3 else 1
dtype_img = img_rgb.dtype
mode_warna = "RGB" if channels_img == 3 else "Grayscale"
ukuran_memori = img_rgb.nbytes / 1024 # dalam KB

print(f"Dimensi (Height x Width): {height} x {width}")
print(f"Jumlah kanal: {channels_img}")
print(f"Tipe data (dtype): {dtype_img}")
print(f"Mode warna: {mode_warna}")
print(f"Ukuran file dalam memori: {ukuran_memori:.2f} KB")
```

```
Dimensi (Height x Width): 1106 x 736
Jumlah kanal: 3
Tipe data (dtype): uint8
Mode warna: RGB
Ukuran file dalam memori: 2384.81 KB
```

B4. TODO: Histogram Warna

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

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

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

Analisis : menurut saya chanel yang paling dominan itu yang biru, karena dia memiliki frekuensi paling tinggi diantara merah dan hijau yaitu sekitar 65000 frekuensi. kemudian kontras gambarnya cukup tinggi karena ada di beberapa titik yang berbeda antara merah(80-90 an), hijau (130-140an), biru (0-20an dan 210-225an). untuk sebaran

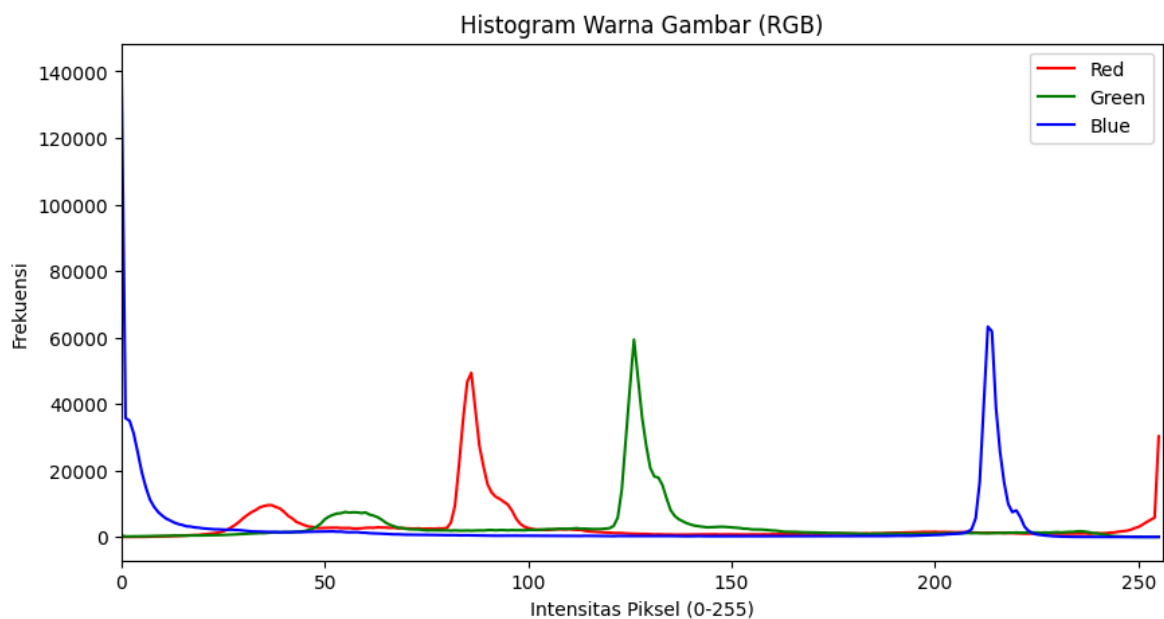
intensitasnya ada di beberapa titik yang rendah dan tinggi yang berragam jadi warnanya sangat bertabrakan (gradasinya kuat)

```
In [20]: # Baca gambar
img = cv2.imread("data\bunga-matahari.jpg", cv2.IMREAD_COLOR)

# Hitung histogram untuk tiap channel
colors = ('r', 'g', 'b')
labels = ('Red', 'Green', 'Blue')

plt.figure(figsize=(10, 5))
for i, color in enumerate(colors):
    hist = cv2.calcHist([img_rgb], [i], None, [256], [0, 256])
    plt.plot(hist, color=color, label=labels[i])
    plt.xlim([0, 256])

# Label dan Legend
plt.title("Histogram Warna Gambar (RGB)")
plt.xlabel("Intensitas Piksel (0-255)")
plt.ylabel("Frekuensi")
plt.legend()
plt.show()
```



B5. Analisis Ringkas (Wajib)

Jawab pertanyaan berikut:

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

Jawaban Anda: hubungan antara pola histogram dengan kesan visual menurut saya pada intensitas 0-20an itu memiliki warna biru dengan frekuensi tinggi namun ada pada intensitas piksel yang rendah, jadi menurut saya disitu cenderung berwarna gelap (bayangan hitam di daun). kemudian pada intensitas piksel yang tinggi di angka 200-250an itu ada beberapa warna yang frekuensinya tinggi sehingga menghasilkan warna yang terang/cerah. untuk histogram yang berwarna merah yang berada pada intensitas

80-90an berada di frekuensi 50000an dan warna hijau di intensitas 130-140an berada di frekuensi 60000an sepertinya memiliki warna yang pas, tidak terlalu gelap dan tidak terlalu terang. kemudian untuk warna dominan yang paling tinggi ada pada warna biru yang memungkinkan bahwa gambar ini memiliki warna dominan di warna biru(langit), sedangkan warna lainnya juga tinggi namun bukan warna yang dominan (kelopak bunga, dan daun). untuk kontrasnya termasuk tinggi karena intensitas pikselnya dia menyebar dan tajam, tidak cenderung rata, jadi gambar ini termasuk ke dalam kontras yang tinggi.

Bagian C — Video

C1. Deskripsi Data

TODO: Jelaskan video yang akan Anda analisis:

- Jenis video: Pemandangan alam
- Sumber: youtube
- Durasi target: 28 detik
- Alasan pemilihan: karena videonya memiliki scene yang berbeda beda dan cinematicnya bagus

Path file: data/____.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.

Analisis :

parameter-parameter yang digunakan dalam menganalisis video ini yang pertama adalah:

1. resolusi, ini penting untuk menentukan ketajaman dan detail gambar yang ada di video, karena semakin tinggi resolusinya semakin jelas detail-detailnya tapi juga memakan banyak penyimpanan.
2. frame rate, ini berpengaruh terhadap kelancaran gerakan pada video, dan di video ini menggunakan 30 fps yang merupakan ukuran standar untuk video-video youtube dll.
3. jumlah total frame, ini digunakan untuk menghitung durasinya.

4. durasi, ini digunakan untuk estimasi ukuran output (ukuran video), bisa juga untuk memotong video.
5. klasifikasi resolusi, ini berguna untuk menentukan kualitas streamingnya.

```
In [21]: # Path video
video_path = 'data\pemandangan alam .mp4'

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

# Ambil 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))
duration = total_frames / fps if fps > 0 else 0

# Klasifikasi resolusi
if width >= 3840 and height >= 2160:
    resolusi_label = "4K (Ultra HD)"
elif width >= 1920 and height >= 1080:
    resolusi_label = "Full HD (1080p)"
elif width >= 1280 and height >= 720:
    resolusi_label = "HD (720p)"
else:
    resolusi_label = "SD (Standard Definition)"

# Tampilkan metadata
print(f"Resolusi: {width} x {height} px")
print(f"Frame rate (FPS): {fps:.2f}")
print(f"Jumlah total frame: {total_frames}")
print(f"Durasi: {duration:.2f} detik")
print(f"Klasifikasi resolusi: {resolusi_label}")

cap.release()
```

```
Resolusi: 1920 x 1080 px
Frame rate (FPS): 30.00
Jumlah total frame: 850
Durasi: 28.33 detik
Klasifikasi resolusi: Full HD (1080p)
```

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.

Analisis : perbedaan visual antar frame pertama kedua dan ketiga adalah untuk frame pertama dia menunjukkan gambar pemandangan pepohonan dari awalan video (bisa dibidang opening), kemudian frame kedua adalah gambar di tengah atau scene pertengahan dari video pemandangan, dan frame ketiga adalah gambar hitam dimana video akan berakhir (closing). jadi perbedaan ketiga frame ini menunjukkan opening, pertengahan scene dan closing dari video.

```
In [22]: # Buka video
video_path = 'data\pemandangan alam .mp4'
cap = cv2.VideoCapture(video_path)

# Hitung total frame & fps
total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
fps = cap.get(cv2.CAP_PROP_FPS) # frame per second

# Untuk menentukan index frame yang diambil (awal, tengah, akhir)
frame_indices = [0, total_frames // 2, total_frames - 1]

frames_rgb = []
timestamps = []

# Ambil frame representatif
for idx in frame_indices:
    cap.set(cv2.CAP_PROP_POS_FRAMES, idx)
    ret, frame = cap.read()
    if not ret:
        print(f"Gagal membaca frame ke-{idx}")
        continue

    # Konversi BGR ke RGB
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frames_rgb.append(frame_rgb)

    # Hitung timestamp (detik)
    time_sec = idx / fps
    timestamps.append(time_sec)

cap.release()

# Tampilkan di subplot
plt.figure(figsize=(15, 5))

for i, (frame_rgb, idx, t) in enumerate(zip(frames_rgb, frame_indices, timestamp)):
    plt.subplot(1, 3, i + 1)
    plt.imshow(frame_rgb)
    plt.title(f"Frame {idx}\nTimestamp: {t:.2f}s")
    plt.axis('off')

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



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 sudah sesuai, karena frame ratenya berada pada 30.00 fps dan resolusinya di 1920 x 1080 px (full HD) ini sesuai dengan use case yang saya pilih yaitu media sosial youtube. karena dengan frame 30 fps dan resolusi 1080 sudah menciptakan grafik yang baik (sudah tidak patah-patah dan pas untuk ukuran penyimpanan).

Perbandingan & Kesimpulan

Perbandingan Representasi Media

TODO: Bandingkan secara ringkas representasi dan visualisasi ketiga media:

Audio (1D - Temporal)

- Representasi: suara/audio stereo
- Visualisasi utama: format wav, waveform, spectrogram, dan MFCC
- Informasi yang diperoleh: dapat mengetahui sampel rate, durasi, jumlah kanal, dan total sampel, mengetahui keras dan lembutnya suara yang di inputkan melalui waveform, mengetahui frekuensi tinggi atau rendahnya dari suara melalui spectrogram, dan perbedaan suara (suara besar/tebas, suara melengking, dll) melalui MFCC.

Gambar (2D - Spasial)

- Representasi: data piksel 2 dimensi (x dan y) dengan tiap piksel memiliki warna/intensitas
- Visualisasi utama: gambar dan histogram
- Informasi yang diperoleh: dapat mengetahui dimensi, jumlah kanal, tipe data, mode warna, dan ukuran file yang digunakan dalam memori, mengetahui hubungan gambar asli dengan histogram yang memuat intensitas warna, frekuensinya, dan kontrasnya.

Video (2D + Waktu - Spatio-temporal)

- Representasi:
- Visualisasi utama: mengetahui perubahan dari 3 frame dari video (frame awal, tengah, dan akhir)
- Informasi yang diperoleh: mengetahui metadata dari video yang di analisis (resolusi, durasi, frame rate, total framerate, dll), kemudian mengetahui perubahan gambar dari tiap frame.

Refleksi Pembelajaran

3 Poin yang Saya Pelajari:

1. mengetahui metadata dari ketiga media (audio, gambar, video)
2. mengetahui cara untuk memvisualisasikan ketiga media tersebut menggunakan cara yang beragam (waveform, histogram, spectrogram)
3. mengetahui perbedaan dari data temporal, spatial, dan spatio-temporal.

2 Hal yang Masih Membingungkan/Ingin Diperdalam:

1. bagian MFCC, itu saya kurang paham dan masih bingung daripada lain lainnya (spectrogram, waveform, dll)
2. membaca histogram warna dengan jelas, karena saya merasa kaya kurang yakin, kadang bingung, tapi paham lagi, tapi bingung lagi.

Sumber Data & Referensi

TODO: Cantumkan semua sumber data dan referensi yang digunakan:

- **Audio:** youtube : https://youtu.be/ocqD0IFk_jY?si=18M3WvrgsnKWcBYs
- **Gambar:** pinterest : <https://pin.it/4vdcuMOdw>
- **Video:** youtube : <https://youtu.be/iY93pTUU1RU?si=o3Lb0YTggk3l2miD>
- **Referensi teknis:** <https://chatgpt.com/share/68c7ac7d-5274-8005-9035-1087880c2e72>

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

Aspek Penilaian	Bobot	Deskripsi
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: $0.8 \times \text{penilaian dosen} + \text{nilai waktu pengumpulan}$

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