

# 122140184\_Worksheet2

September 14, 2025

## 1 Tugas Analisis Multimedia: Audio, Gambar, Video

**Mata Kuliah:** Sistem & Teknologi Multimedia

**Nama:** Dina Rahma Dita

**NIM:** 122140184

---

### 1.1 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.

---

### 1.2 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.

```
[117]: # 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

```

## 1.3 Petunjuk Umum Pengerjaan

### 1.3.1 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

### 1.3.2 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

### 1.3.3 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

### 1.3.4 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

### 1.3.5 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

## 1.4 Checklist Kelengkapan (Centang saat selesai)

### 1.4.1 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

### 1.4.2 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

### 1.4.3 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

### 1.4.4 Analisis & Dokumentasi

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

## 2 Pendahuluan

### 2.1 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

## 2.2 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

## 2.3 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

# 3 Bagian A — Audio

### 3.0.1 A1. Deskripsi Data

**TODO:** Jelaskan audio yang akan Anda analisis: - Jenis audio: Suara Piano - Sumber: freesound.org (dataset publik) - Format file: wav - Alasan pemilihan: suara pianonya terdengar jernih dan ritmenya tenang bukan seperti sound horeg

**Path file:** data/suara\_piano.wav

---

### 3.0.2 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.

*(Tambahkan sel kode di sini)*

```
[118]: import librosa
import soundfile as sf

# Path ke file audio
audio_path = "data/suara_piano.wav"

# Muat audio (tanpa mengubah sample rate asli, dan biarkan stereo jika ada)
y, sr = librosa.load(audio_path, sr=None, mono=False)

# Metadata dasar
duration = len(y) / sr if y.ndim == 1 else y.shape[1] / sr
channels = 1 if y.ndim == 1 else y.shape[0]
```

```

print("Sample rate:", sr, "Hz")
print("Durasi:", round(duration, 2), "detik")
print("Jumlah kanal:", "Mono" if channels == 1 else "Stereo")
print("Jumlah total sampel:", len(y) if channels == 1 else y.shape[1])

```

Sample rate: 44100 Hz  
 Durasi: 4.2 detik  
 Jumlah kanal: Stereo  
 Jumlah total sampel: 185115

### 3.0.3 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.)

Jika dilihat dari pola naik turunnya amplitudo seiring waktu maka ketika amplitudo tinggi itu artinya bunyi piano keras sedangkan jika amplitudonya rendah itu artinya bunyi piano lebih pelan/hening. Dari pola tersebut juga terlihat cukup teratur bentuknya bukan pola yang kasar dan itu menunjukkan bahwa suara piano yang dihasilkan cukup tenang bukan suara piano yang “berisik”.

*(Tambahkan sel kode di sini)*

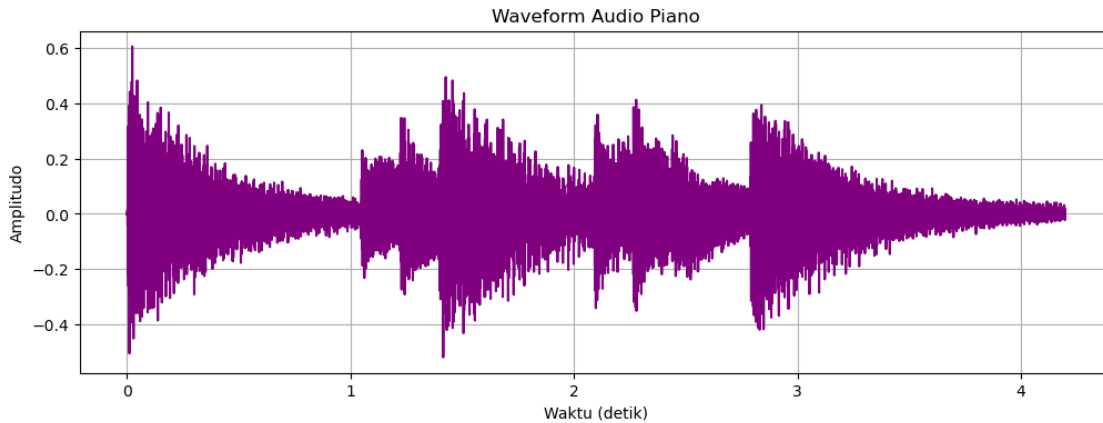
```

[119]: import matplotlib.pyplot as plt
import numpy as np

# Hitung waktu untuk sumbu X
if y.ndim == 1: # mono
    time = np.linspace(0, len(y) / sr, num=len(y))
else: # stereo → ambil channel pertama
    time = np.linspace(0, y.shape[1] / sr, num=y.shape[1])
    y = y[0] # pakai channel kiri biar lebih sederhana

# Plot waveform
plt.figure(figsize=(12, 4))
plt.plot(time, y, color='purple')
plt.title("Waveform Audio Piano")
plt.xlabel("Waktu (detik)")
plt.ylabel("Amplitudo")
plt.grid(True)
plt.show()

```



### 3.0.4 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 dibandingkan waveform.

Spectrogram menunjukkan sebaran energi suara pada berbagai frekuensi sepanjang waktu, semakin tinggi energi suara maka nadanya akan terdengar kuat dan warna yang ditampilkan akan semakin terang sedangkan energi yang rendah menampilkan warna yang lebih gelap dan nada yang dihasilkan juga terdengar pelan. Spectrogram lebih jelas memperlihatkan nada piano bila dibandingkan dengan waveform hal tersebut terlihat dari nada dengan frekuensi rendah muncul pada bagian bawah sedangkan nada dengan frekuensi tinggi ditampilkan di bagian atas.

(Tambahkan sel kode di sini)

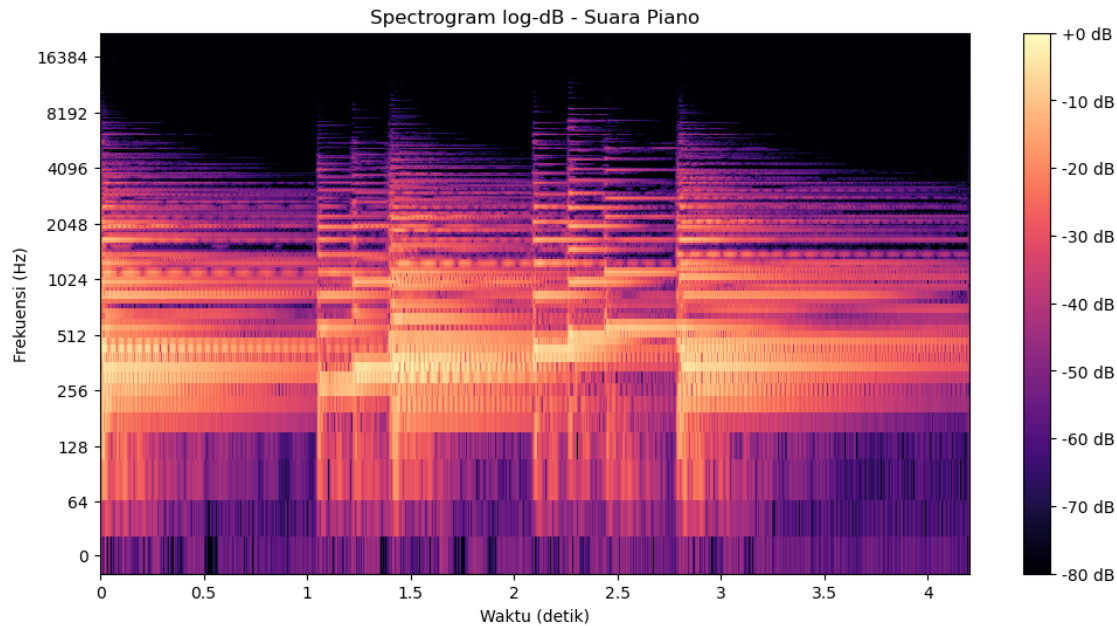
```
[120]: import librosa.display

# Hitung STFT (Short-Time Fourier Transform)
n_fft = 1024
hop_length = 256
stft = librosa.stft(y, n_fft=n_fft, hop_length=hop_length)
spectrogram = np.abs(stft)

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

# Plot spectrogram
plt.figure(figsize=(12, 6))
librosa.display.specshow(
    spectrogram_db, sr=sr, hop_length=hop_length,
    x_axis="time", y_axis="log", cmap="magma"
```

```
)
plt.colorbar(format="%+2.0f dB")
plt.title("Spectrogram log-dB - Suara Piano")
plt.xlabel("Waktu (detik)")
plt.ylabel("Frekuensi (Hz)")
plt.show()
```



### 3.0.5 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?

Jika dilihat dari pola MFCC-nya, warnanya cenderung stabil dari awal sampai akhir, tidak banyak perubahan besar. Itu artinya suara piano yang dianalisis lebih tenang dan nada yang dimainkan konsisten. Jika polanya berubah-ubah drastis biasanya tandanya nadanya banyak berubah atau dimainkan dengan cepat. Jadi berdasarkan pola yang ditunjukkan pada heatmap dapat disimpulkan piano memiliki nada yang stabil dan tempo yang pelan.

*(Tambahkan sel kode di sini)*

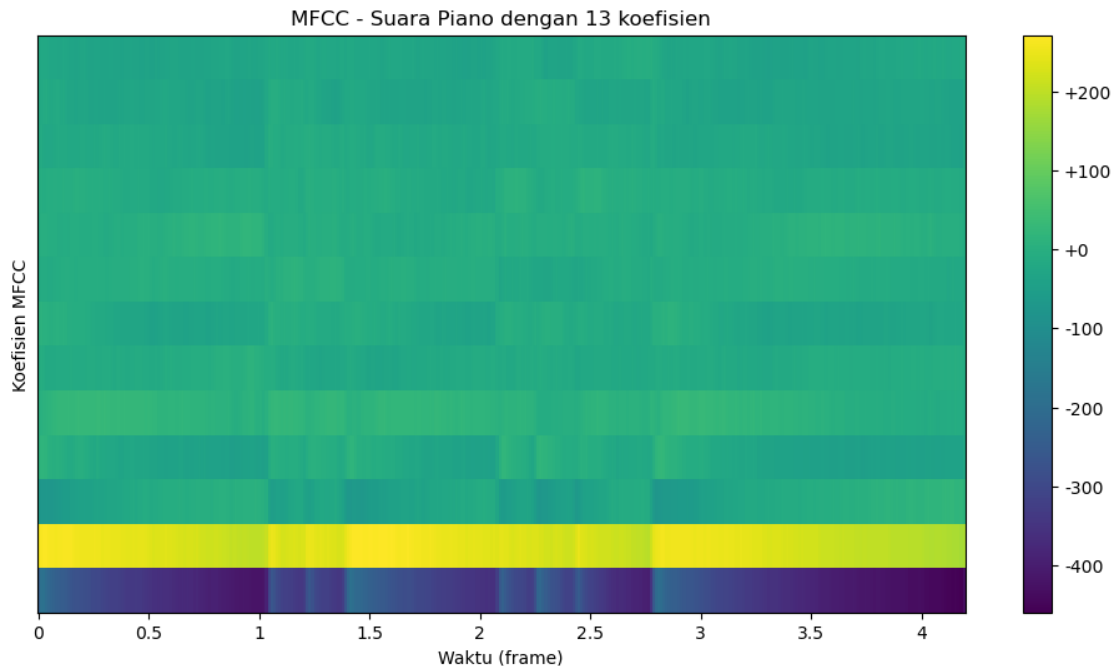
```
[122]: # Hitung MFCC (13 koefisien)
mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)

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

```

librosa.display.specshow(mfcc, x_axis="time", sr=sr, cmap="viridis")
plt.colorbar(format="+2.0f")
plt.title("MFCC - Suara Piano dengan 13 koefisien")
plt.xlabel("Waktu (frame)")
plt.ylabel("Koefisien MFCC")
plt.show()

```



### 3.0.6 A6. Analisis Ringkas (Wajib)

Jawab pertanyaan berikut:

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

*Jawaban Anda:* Jika hanya ingin tau bagian mana suara keras, pelan, atau jeda antar nada, kita dapat melihat informasinya melalui waveform karena waveform pada dasarnya hanya menampilkan perubahan amplitudo terhadap waktu tapi jika ingin melihat detail dari nada atau frekuensi yang digunakan maka kita dapat melihat informasinya dari spectrogram karena spectrogram memperlihatkan sebaran energi di setiap frekuensi sepanjang waktu. Nada piano yang tinggi ataupun rendah serta kapan nada dimainkan kita bisa mengetahuinya dari spectrogram sehingga lebih membantu dalam analisis musik.

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

*Jawaban Anda:* Berdasarkan visualisasi MFCC terlihat piano memiliki nada yang konsisten dan ritme yang tenang berdasarkan pola warna yang cukup stabil, perubahan warna yang ditampilkan tidak terlalu drastis sehingga dapat disimpulkan bahwa piano dimainkan dengan



tenang, santai, tidak cepat. Dari visualisasi tersebut, kita dapat memahami karakter suara seperti pada saat kapan piano dimainkan dengan nada yang stabil atau mengalami perubahan nada sehingga suara piano dapat dipakai untuk identifikasi musik.

## 4 Bagian B — Gambar

### 4.0.1 B1. Deskripsi Data

**TODO:** Jelaskan gambar yang akan Anda analisis: - Jenis gambar: Foto bunga Anggrek - Sumber: Foto Sendiri - Format file: JPG - Alasan pemilihan: Foto bunga anggrek tersebut memiliki warna yang cerah seperti warna ungu dan hijau, cocok untuk dianalisis histogram warnanya.

**Path file:** data/bunga.jpg

---

### 4.0.2 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).

Gambar menampilkan bunga anggrek ungu sebagai objek utama. Pencahayaan cukup terang sehingga detail kelopak terlihat jelas. Warna ungu bunga berpadu dengan hijau pada batang dan daun, memberi kontras dengan latar belakang batang pohon dan struktur rumah.

*(Tambahkan sel kode di sini)*

```
[124]: from PIL import Image
import cv2

# Path ke file gambar
img_path = "data/bunga.jpg"

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

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

# Tampilkan gambar
plt.figure(figsize=(6,6))
plt.imshow(img_rgb)
plt.title("Foto Bunga Anggrek Ungu - Format RGB")
plt.axis("off")
plt.show()
```

Foto Bunga Anggrek Ungu - Format RGB



#### 4.0.3 B3. TODO: Informasi Dasar

**Instruksi:** Tampilkan informasi metadata gambar: - Dimensi (Height  $\times$  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.

- Dimensi gambar memberi tahu seberapa besar resolusinya. Gambar besar biasanya lebih detail, tapi butuh memori lebih banyak sehingga tahap preprocessing perlu dilakukan agar tidak terlalu memakan kapasitas memori dan memiliki ukuran yang seragam.
- Jumlah kanal = 3 (RGB), artinya gambar berwarna penuh sehingga dalam proses analisis pun dapat dilakukan dengan maksimal, sebagai contoh apabila jumlah kanal hanya 1 (grayscale) maka pasti akan ada informasi yang terlewatkan.
- Tipe data (misalnya uint8) menunjukkan nilai pixel dari 0–255. Dari piksel tersebut perlu di normalisasi agar tidak memiliki angka yang besar dan memperlambat proses training.
- Ukuran file penting untuk pertimbangan efisiensi karena ukuran file dapat mempengaruhi waktu training dan kebutuhan memori.

(Tambahkan sel kode di sini)

```
[125]: import os

# Baca lagi gambar dengan PIL untuk metadata tambahan
img_pil = Image.open(img_path)

# Informasi dasar
height, width = img_pil.size
channels = len(img_pil.getbands()) # jumlah kanal warna
dtype = img_bgr.dtype             # tipe data pixel
file_size = os.path.getsize(img_path) / 1024 # ukuran file dalam KB

print("Dimensi gambar (Width x Height):", img_pil.size)
print("Jumlah kanal warna:", channels)
print("Tipe data pixel:", dtype)
print("Ukuran file di memori:", round(file_size, 2), "KB")
```

Dimensi gambar (Width x Height): (959, 1280)

Jumlah kanal warna: 3

Tipe data pixel: uint8

Ukuran file di memori: 113.64 KB

#### 4.0.4 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?

Channel yang dominan : - Channel R (Merah) memiliki intensitas paling tinggi artinya warna merah cukup dominan di gambar. - Channel B (Biru) juga memiliki intensitas kedua paling tinggi, sehingga warna merah dan biru dipadukan dan menghasilkan warna ungu - Channel G (Hijau) tidak setinggi merah, tapi juga tidak serendah biru pada bagian tertentu.

Kontras gambar : - Gambar memiliki kontras yang baik, tidak terlalu terang dan tidak terlalu gelap berdasarkan intensitas pixel antara 0 - 250

Sebaran intensitas : - Sebaran intensitas paling banyak ada ditengah sekitar 80 - 180 pixel, sedangkan untuk bagian gelap berada pada pixel 0 - 50 dan bagian terang berada pada sekitar pixel 200 - 255. Jadi intensitas sedang mendominasi gambar ini sehingga gambar yang dihasilkan pun terlihat lembut dan jelas.

(Tambahkan sel kode di sini)

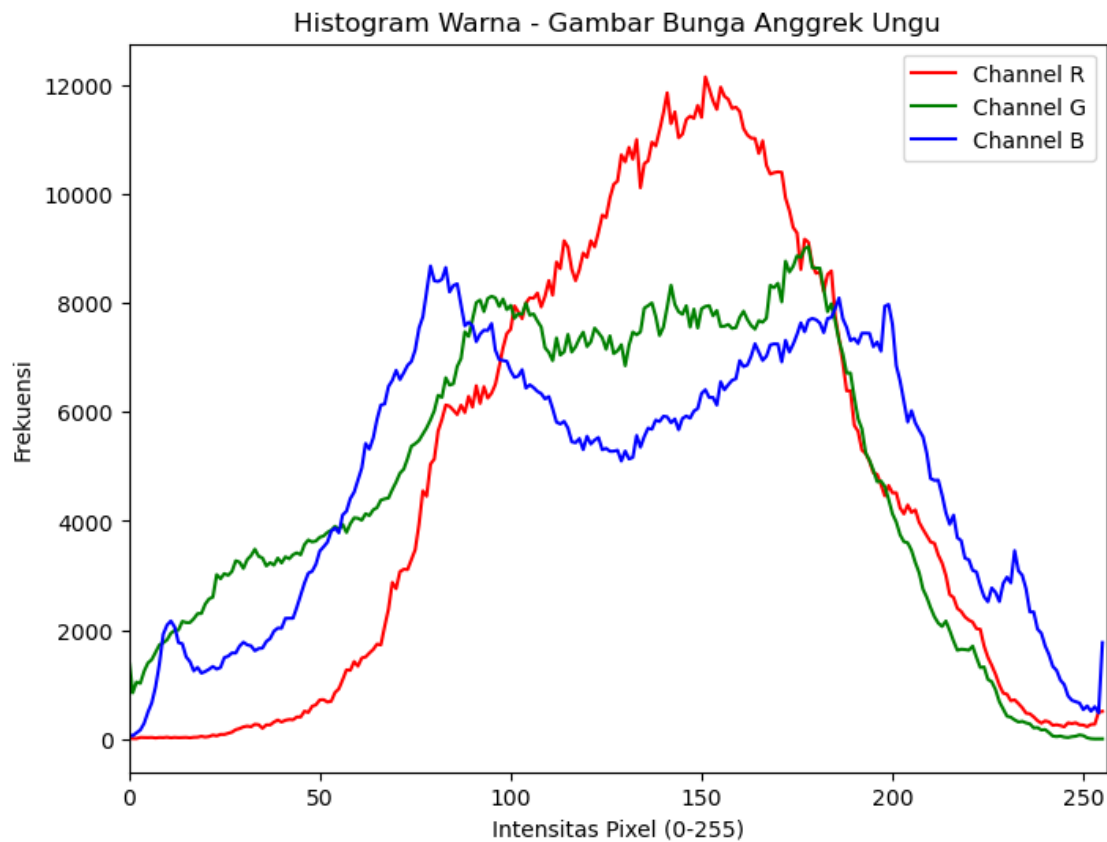
```
[126]: # Hitung histogram untuk tiap channel warna
colors = ('r', 'g', 'b')
plt.figure(figsize=(8,6))
```

```

for i, col in enumerate(colors):
    hist = cv2.calcHist([img_rgb], [i], None, [256], [0,256])
    plt.plot(hist, color=col, label=f'Channel {col.upper()}')
    plt.xlim([0,256])

plt.title("Histogram Warna - Gambar Bunga Anggrek Ungu")
plt.xlabel("Intensitas Pixel (0-255)")
plt.ylabel("Frekuensi")
plt.legend()
plt.show()

```



#### 4.0.5 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:* Histogram menunjukkan jika kanal merah dan biru lebih tinggi dibanding hijau. Ini sesuai dengan tampilan bunga anggrek yang berwarna ungu, karena ungu terbentuk dari campuran merah dan biru. Kanal hijau tetap ada, tapi lebih banyak berasal dari daun dan tangkai. Sebaran

nilai piksel yang sampai ke intensitas tinggi menandakan gambar cukup terang. Perbedaan antara puncak merah, biru, dan hijau juga membuat kontras antara bunga dan elemen lain terlihat jelas.

## 5 Bagian C — Video

### 5.0.1 C1. Deskripsi Data

**TODO:** Jelaskan video yang akan Anda analisis: - Jenis video: Video rangkaian buket bunga berwarna ungu yang cantik - Sumber: youtube.com ([https://www.youtube.com/watch?v=KWVK0w\\_X0VQ](https://www.youtube.com/watch?v=KWVK0w_X0VQ)) - Durasi target: 28 detik - Alasan pemilihan: video dipilih karena memiliki warna yang menarik dan gerakan yang dinamis.

**Path file:** data/video1.mp4

---

### 5.0.2 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.

- Resolusi menunjukkan kualitas detail video, semakin tinggi resolusi berarti kualitas video semakin tajam dan detail.
- FPS (frame per second) menentukan kelancaran gerakan animasi, semakin tinggi fps, gerakan video makin mulus tidak patah - patah.
- Jumlah total frame menunjukkan total unit gambar. Semakin banyak jumlah frame artinya informasi yang diberikan lebih banyak tetapi juga membutuhkan lebih banyak penyimpanan.
- Durasi menentukan panjang video.
- Klasifikasi resolusi memberi gambaran apakah video cocok untuk platform tertentu. Video dengan kualitas HD sudah cukup efisien untuk analisis atau streaming.

*(Tambahkan sel kode di sini)*

```
[127]: # Path ke file video
video_path = "data/video1.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)
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:
    resolusi = "4K"
elif width >= 1920 and height >= 1080:
    resolusi = "Full HD"
elif width >= 1280 and height >= 720:
    resolusi = "HD"
else:
    resolusi = "SD"

print("Resolusi video:", f"{width}x{height}")
print("Frame rate (fps):", fps)
print("Jumlah total frame:", frame_count)
print("Durasi video:", round(duration, 2), "detik")
print("Klasifikasi resolusi:", resolusi)

cap.release()

```

```

Resolusi video: 1280x720
Frame rate (fps): 29.97
Jumlah total frame: 813
Durasi video: 27.13 detik
Klasifikasi resolusi: HD

```

### 5.0.3 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.

Deskripsi per frame: - Frame 0 ( $t=0.00\text{s}$ ): Menampilkan buket bunga anggrek ungu dari awal sehingga warna ungu langsung menjadi fokus utama, terlihat diletakkan di atas meja dengan kertas pembungkus putih. - Frame 406 ( $t=13.55\text{s}$ ): Berada di tengah video, detail kelopak bunga terlihat lebih jelas karena sudut pengambilan gambar lebih dekat. - Frame 811 ( $t=27.06\text{s}$ ): Di akhir video, tampilannya lebih luas, terlihat bukan hanya ada buket anggrek ungu tapi ada buket bunga lain juga (kuning dan biru).

Perbedaan visual utama: - Awal video hanya terlihat pada satu buket, di tengah video menampilkan detail kelopak, dan akhir memperlihatkan bahwa bukan hanya ada buket anggrek tapi ada buket bunga lain. - Transisi yang digunakan juga memberikan variasi sudut pandang.

Pelajaran dari sampel frame: - Berdasarkan sampel frame awal, tengah, dan akhir, kita bisa memahami alur visual video: dari pengenalan objek, penekanan detail, hingga penutup dengan tampilan lebih luas.

Tambahan : terdapat kendala pada saat melakukan eksekusi perintah ini, karena pada saat pengambilan sampel frame - 1 tidak tampil gambar sampelnya sehingga diubah menjadi frame - 2 supaya tampil ketiga sampel frame. Sudah beberapa video saya coba tapi hasilnya tetap sama.

(Tambahkan sel kode di sini)

```
[128]: import cv2
import matplotlib.pyplot as plt

# Baca ulang video
cap = cv2.VideoCapture(video_path)

# Metadata
frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
fps = cap.get(cv2.CAP_PROP_FPS)
mid_frame = frame_count // 2

frames = []

# Fungsi untuk ambil frame dengan timestamp
def get_frame(cap, idx, fps):
    cap.set(cv2.CAP_PROP_POS_FRAMES, idx)
    ret, frame = cap.read()
    if ret and frame is not None:
        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        timestamp = idx / fps if fps > 0 else 0
        return (idx, frame_rgb, timestamp)
    return None

# Frame awal (0)
first_frame = get_frame(cap, 0, fps)
if first_frame: frames.append(first_frame)

# Frame tengah
middle_frame = get_frame(cap, mid_frame, fps)
if middle_frame: frames.append(middle_frame)

# Frame terakhir (coba -1, fallback ke -2 kalau gagal)
last_idx = frame_count - 1
last_frame = get_frame(cap, last_idx, fps)
if not last_frame: # fallback
    print(f"Gagal ambil frame {last_idx}, coba fallback ke {frame_count-2}")
    last_idx = frame_count - 2
    last_frame = get_frame(cap, last_idx, fps)
if last_frame: frames.append(last_frame)

cap.release()

# Plot hasil
plt.figure(figsize=(15,5))
for i, (idx, frame, timestamp) in enumerate(frames):
```

```
plt.subplot(1, 3, i+1)
plt.imshow(frame)
plt.title(f"Frame {idx}\n(t={timestamp:.2f}s)")
plt.axis("off")
plt.suptitle("Frame Awal - Tengah - Akhir Video")
plt.show()
```

Gagal ambil frame 812, coba fallback ke 811



#### 5.0.4 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:* Video ini memiliki resolusi 1280×720 (HD) dengan frame rate sekitar 29.97 fps. Spesifikasi ini sudah cocok untuk media sosial, kuliah daring, maupun presentasi

Alasan:

- Resolusi 720p cukup jelas untuk menampilkan detail di laptop atau proyektor, sekaligus tidak terlalu besar sehingga mudah dibagikan dan diputar online.
- Frame rate 29.97 fps merupakan standar umum, membuat gerakan terlihat halus tanpa terasa patah.
- Untuk media sosial dan kuliah daring, format ini ideal karena mudah diakses di hampir semua perangkat tanpa buffering yang berat.

## 6 Perbandingan & Kesimpulan

### 6.1 Perbandingan Representasi Media

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

#### 6.1.1 Audio (1D - Temporal)

- Representasi: Sinyal 1D berupa amplitudo terhadap waktu
- Visualisasi utama: Waveform, spectrogram, dan MFCC



- Informasi yang diperoleh: Pola keras–pelan (dynamics) berdasarkan waveform, tinggi–rendah nada (frekuensi) berdasarkan spectrogram, dan karakteristik timbre (warna nada) suara berdasarkan heatmap MFCC.

### 6.1.2 Gambar (2D - Spasial)

- Representasi: Matriks piksel dengan nilai warna
- Visualisasi utama: Tampilan gambar dengan warna RGB dan histogram warna
- Informasi yang diperoleh: Warna dominan pada gambar, distribusi intensitas warna, kontras yang digunakan, serta detail dari objek.

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

- Representasi: Rangkaian gambar (frame) berurutan sepanjang waktu (2D + waktu)
- Visualisasi utama: Cuplikan frame awal–tengah–akhir, metadata dasar video (Sample rate, durasi, jumlah kanal, jumlah total sampel).
- Informasi yang diperoleh: Perubahan visual antar waktu, gerakan objek, alur peristiwa, serta kualitas video.

---

## 6.2 Refleksi Pembelajaran

### 6.2.1 3 Poin yang Saya Pelajari:

1. Perbedaan mendasar bahwa setiap media (audio, gambar, video) memiliki sifat data sehingga cara representasi dan visualisasinya juga berbeda.
2. Visualisasi membantu memahami informasi yang berusaha disampaikan melalui media misalnya kapan nada tersebut bernada tinggi atau rendah, atau kapan nada tersebut terdengar keras atau pelan.
3. Resolusi, frame rate, dan durasi adalah parameter penting dalam menilai kualitas video dan menentukan kesesuaian media dengan konteks penggunaan.

### 6.2.2 2 Hal yang Masih Membingungkan/Ingin Diperdalam:

1. Bagaimana cara memahami pola MFCC secara lebih rinci, serta bagaimana koefisiennya dapat dimanfaatkan untuk klasifikasi suara, misalnya dalam membedakan jenis instrumen musik?
2. Bagaimana teknik kompresi audio maupun video (codec) berpengaruh terhadap keterbacaan frame terakhir serta kualitas data saat dianalisis dengan OpenCV atau librosa?

---

## 6.3 Sumber Data & Referensi

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

- **Audio:** (<https://freesound.org/s/219213/>)
- **Gambar:** Foto pribadi

- **Video:** ([https://www.youtube.com/watch?v=KWVK0w\\_X0VQ](https://www.youtube.com/watch?v=KWVK0w_X0VQ))
- **Referensi teknis:** (<https://chatgpt.com/share/68c634bc-bc8c-8008-b223-93ddf8a60ec7>)

## 7 Rubrik Penilaian

### 7.1 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

### 7.2 Detail Kriteria Penilaian

#### 7.2.1 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

#### 7.2.2 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

#### 7.2.3 Analisis & Interpretasi (30%)

- Interpretasi menunjukkan pemahaman konsep
- Analisis kontekstual, bukan sekadar deskripsi output
- Mampu menghubungkan hasil dengan teori
- Refleksi pembelajaran yang thoughtful

#### 7.2.4 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

### 7.2.5 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

### 7.3 Proporsi Penilaian Total

- Proporsi penilaian hanya 80%, 20% lagi akan didasarkan pada kecepatan pengumpulan tugas
- Sehingga:  $0.8 * \text{penilaian dosen} + \text{nilai waktu pengumpulan}$

## 8 Aturan Kejujuran Akademik

### 8.1 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

### 8.2 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

### 8.3 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

### 8.4 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

## 9 Panduan Pengumpulan

### 9.1 Berkas yang Harus Dikumpulkan

#### 9.1.1 Wajib:

1. **Notebook Jupyter** (.ipynb) dengan nama: `NIM_Nama_TugasMultimedia.ipynb`
    - Contoh: `123456789_JohnDoe_TugasMultimedia.ipynb`
  2. **PDF** hasil render dari notebook
- 

### 9.2 Informasi Pengumpulan

---

### 9.3 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
- 

#### 9.3.1 Export ke PDF:

- File → Save and Export Notebook As → HTML
- Buka HTML di browser -> Save as PDF