

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

**Mata Kuliah:** Sistem & Teknologi Multimedia

**Nama:** M. Raihan Athalah Ilham (isi dengan nama lengkap Anda)

**NIM:** 122140022 (isi dengan NIM Anda)

### 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.

```
# 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
print("Library versions:")
print(f"NumPy: {np.__version__}")
print(f"Matplotlib: {plt.__version__}")
print(f"Librosa: {librosa.__version__}")
print(f"OpenCV: {cv2.__version__}")
```

```
# Tambahkan import lain jika diperlukan saat mengerjakan tugas
```

```
Library versions:
NumPy: 2.0.2
```

```
-----
AttributeError                                Traceback (most recent call last)
/tmp/ipython-input-1566465522.py in <cell line: 0>()
    15 print("Library versions:")
    16 print(f"NumPy: {np.__version__}")
--> 17 print(f"Matplotlib: {plt.__version__}")
    18 print(f"Librosa: {librosa.__version__}")
    19 print(f"OpenCV: {cv2.__version__}")
```

```
AttributeError: module 'matplotlib.pyplot' has no attribute '__version__'
```

### 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 Alam (musik, pidato, suara alam, dll.)
- Sumber: Dari Youtube (rekaman sendiri, dataset publik, dll.)
- Format file: MP3 (WAV, MP3, dll.)
- Alasan pemilihan: Karena memiliki frekuensi beragam jadi lebih menarik.

**Path file:** `/media/1 menit relaksasi Suara Alam indah.mp3` (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.

*(Tambahkan sel kode di sini)*

---

### A3. TODO: Waveform

**Instruksi:** Plot waveform audio dengan:

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

**Analisis yang diperlukan:** Waveform menunjukkan suara alam relaksasi yang stabil dan berkesinambungan, tanpa variasi besar antara bagian keras dan pelan. Hal ini sesuai dengan tujuan audio untuk menenangkan, karena tidak ada perubahan mendadak yang mengejutkan pendengar.

*(Tambahkan sel kode di sini)*

---

### 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:** Waveform dan spectrogram menyajikan informasi audio dengan cara yang berbeda. Waveform hanya menunjukkan perubahan amplitudo sinyal terhadap waktu, sehingga lebih cocok untuk melihat intensitas atau energi keseluruhan dari suara. Namun, waveform tidak memberikan detail mengenai distribusi frekuensi. Sebaliknya, spectrogram menggambarkan bagaimana energi sinyal tersebar pada berbagai frekuensi sepanjang waktu. Melalui spectrogram, kita bisa melihat pola frekuensi rendah, sedang, hingga tinggi, serta intensitasnya dalam bentuk gradasi warna. Dengan demikian, spectrogram memberikan wawasan yang lebih kaya mengenai karakteristik spektral suara dibandingkan dengan waveform.

(Tambahkan sel kode di sini)

## 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? Pola MFCC pada audio dapat menunjukkan apakah karakteristik suara relatif konsisten atau mengalami banyak variasi. Jika pola MFCC terlihat stabil, hal ini mengindikasikan bahwa sifat akustik suara cenderung konstan, misalnya bunyi latar atau suara dengan nada monoton. Sebaliknya, jika pola MFCC berubah-ubah, berarti terdapat variasi spektral yang signifikan, yang biasanya muncul pada ucapan, musik, atau suara dengan dinamika tinggi. Perubahan tersebut dapat menjadi penanda adanya informasi penting, seperti transisi antar fonem dalam ucapan atau perubahan nada pada musik. (Tambahkan sel kode di sini)

## A6. Analisis Ringkas (Wajib)

**Jawab pertanyaan berikut:**

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

*Jawaban Anda:* Waveform memberi gambaran intensitas suara seiring waktu, sementara spectrogram memperlihatkan bagaimana energi suara tersebar pada frekuensi tertentu sepanjang durasi audio.

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

*Jawaban Anda:* Dari MFCC terlihat pola yang stabil, menunjukkan suara alam ini konsisten tanpa banyak perubahan signifikan. Ini berarti MFCC dapat dipakai untuk mengenali ciri khas suara hujan/relaksasi secara andal.

```
# A2 TODO: Muat & Metadata

import librosa
import soundfile as sf

# Path ke file audio (ganti sesuai nama file Anda)
file_path = "/media/1 menit relaksasi Suara Alam indah.mp3"

# Load audio
y, sr = librosa.load(file_path, sr=None, mono=False) # sr=None agar pakai sample rate asli
info = sf.info(file_path)

# Metadata dasar
sample_rate = sr
duration = librosa.get_duration(y=y, sr=sr)
channels = 1 if y.ndim == 1 else y.shape[0]
total_samples = y.shape[-1]

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

=== Metadata Audio ===
Sample rate      : 48000 Hz
Durasi          : 60.05 detik
Jumlah kanal     : Stereo (2)
Total sampel     : 2882304
```

```
#A3 TODO: Waveform

import librosa
import soundfile as sf
import matplotlib.pyplot as plt
import numpy as np
```

```
# Path ke file audio (ganti sesuai nama file Anda)
file_path = "/media/1 menit relaksasi Suara Alam indah.mp3"

# Load audio
y, sr = librosa.load(file_path, sr=None, mono=False) # sr=None agar pakai sample rate asli
info = sf.info(file_path)

# Metadata dasar
n_samples = y.shape[-1]
n_channels = 1 if y.ndim == 1 else y.shape[0]
duration = librosa.get_duration(y=y, sr=sr)

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

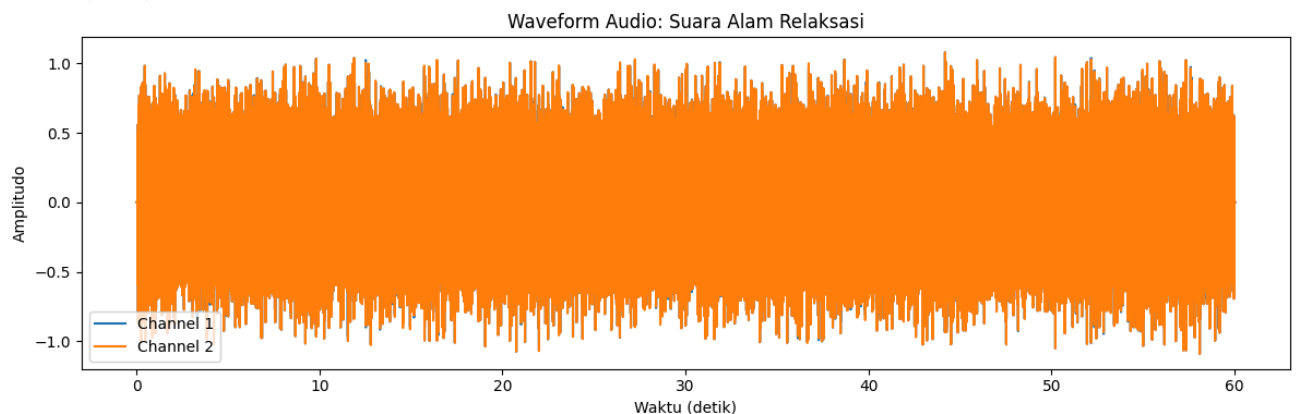
# Plot waveform
plt.figure(figsize=(12, 4))

times = np.arange(n_samples) / sr

if n_channels == 1:
    # Mono
    plt.plot(times, y, color='b')
else:
    # Stereo/multi-channel (y.shape = [channels, samples])
    for ch in range(n_channels):
        plt.plot(times, y[ch, :], label=f'Channel {ch+1}')
    plt.legend()

plt.title("Waveform Audio: Suara Alam Relaksasi")
plt.xlabel("Waktu (detik)")
plt.ylabel("Amplitudo")
plt.tight_layout()
plt.show()
```

```
=== Metadata Audio ===
Sample rate      : 48000 Hz
Durasi           : 60.05 detik
Jumlah kanal     : Stereo (2)
Total sampel     : 2882304
/tmp/ipython-input-2305325703.py:41: UserWarning: Creating legend with loc="best" can be slow with large amounts of data.
plt.tight_layout()
```



```
#A4 TODO: Spectrogram Log-dB
import librosa.display

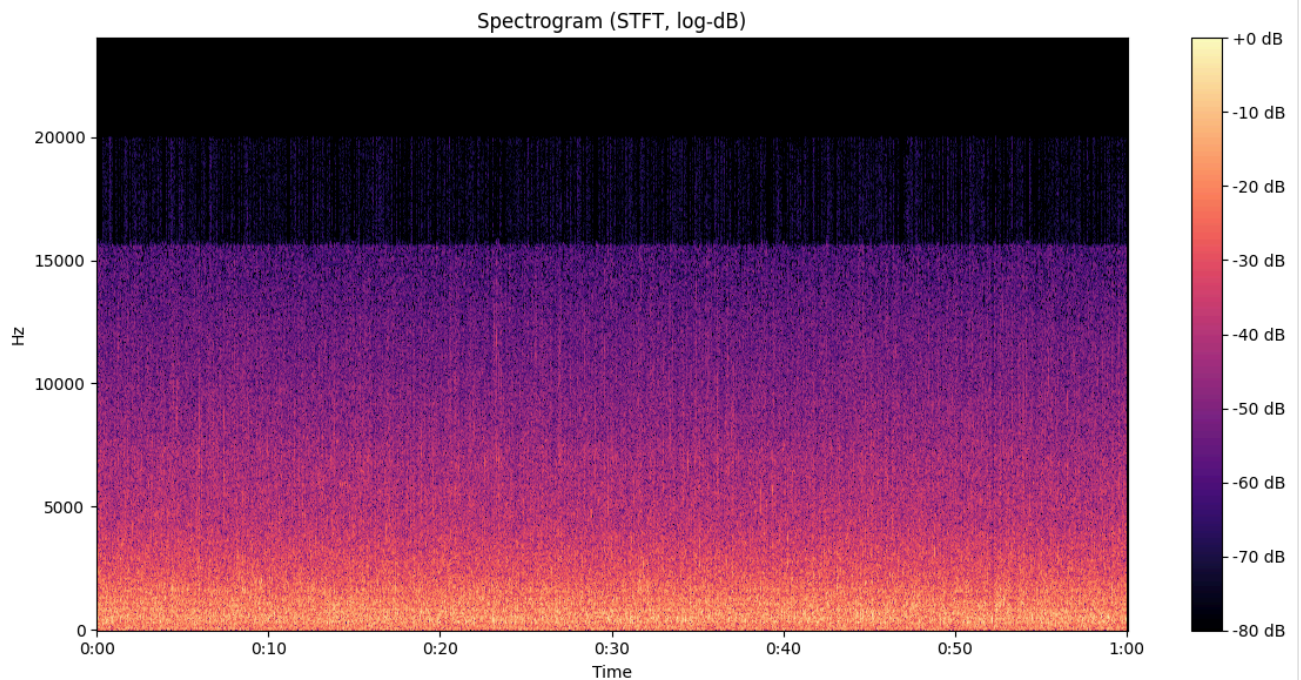
# Hitung STFT
n_fft = 1024
hop_length = 256
D = librosa.stft(y if n_channels == 1 else y[0], n_fft=n_fft, hop_length=hop_length) # ambil channel 0 jika stereo
S_db = librosa.amplitude_to_db(np.abs(D), ref=np.max)

# Plot spectrogram
plt.figure(figsize=(12, 6))
librosa.display.specshow(
    S_db,
    sr=sr,
```

```

        hop_length=hop_length,
        x_axis="time",
        y_axis="hz",
        cmap="magma"
    )
    plt.colorbar(format="%+2.1f dB")
    plt.title("Spectrogram (STFT, log-dB)")
    plt.tight_layout()
    plt.show()

```

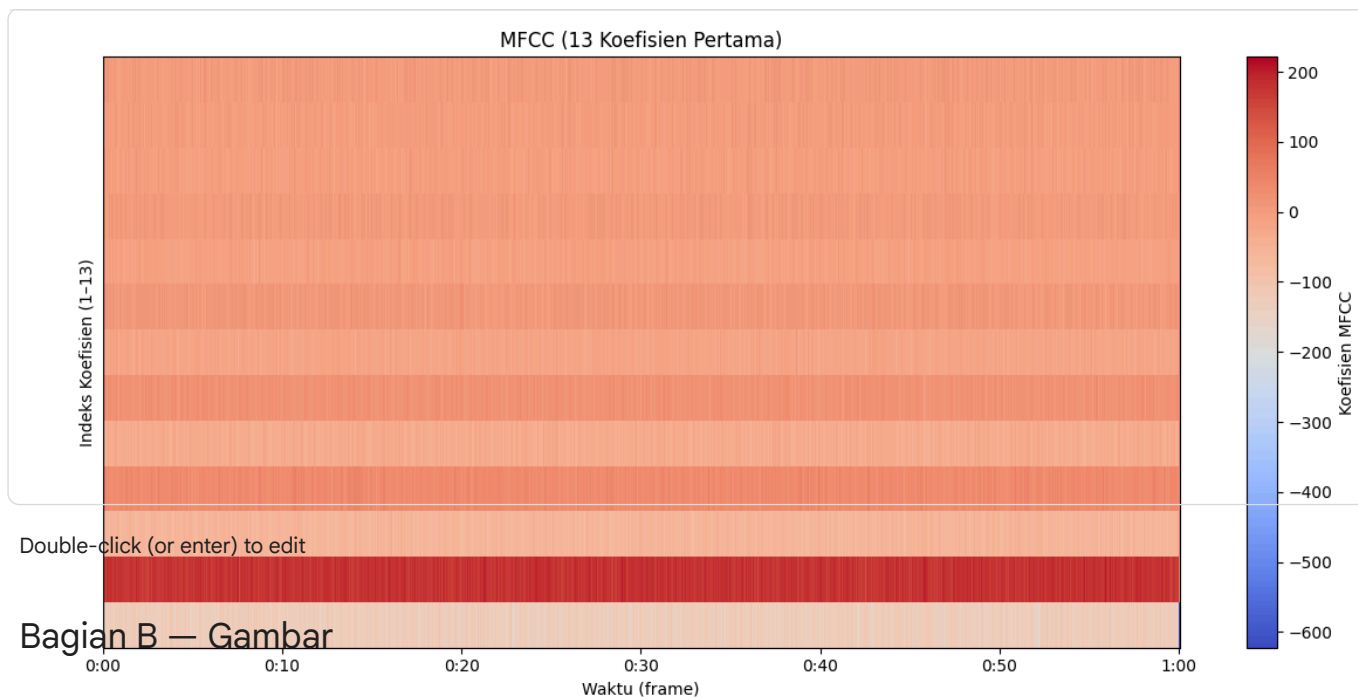


```

#A5 TODO: MFCC
# Hitung MFCC (13 koefisien)
mfccs = librosa.feature.mfcc(
    y=y if n_channels == 1 else y[0],
    sr=sr,
    n_mfcc=13,
    n_fft=1024,
    hop_length=256
)

# Plot MFCC
plt.figure(figsize=(12, 6))
librosa.display.specshow(
    mfccs,
    x_axis="time",
    sr=sr,
    hop_length=256,
    cmap="coolwarm"
)
plt.colorbar(label="Koefisien MFCC")
plt.title("MFCC (13 Koefisien Pertama)")
plt.ylabel("Indeks Koefisien (1-13)")
plt.xlabel("Waktu (frame)")
plt.tight_layout()
plt.show()

```



## Bagian B — Gambar

### B1. Deskripsi Data

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

- Jenis gambar: Foto (foto, ilustrasi, pemandangan, dll.)
- Sumber: Foto Sendiri (foto sendiri, dataset publik, dll.)
- Format file: JPG (JPG, PNG, BMP, dll.)
- Alasan pemilihan: Karena fotonya cukup banyak warnanya jadi lebih menarik untuk dijadikan objek eksperimen.

**Path file:** `/media/Networking dinner telkom.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:** Gambar menunjukkan objek dominan berupa orang-orang dalam acara networking dinner. Kondisi pencahayaan tampak berasal dari lampu ruangan dalam, sehingga menghasilkan nuansa hangat dengan dominasi warna kuning dan oranye. Warna pakaian para peserta dan elemen dekorasi juga menambah variasi warna, namun suasana keseluruhan tetap berpusat pada cahaya ruangan yang cukup terang, membuat objek utama terlihat jelas.

*(Tambahkan sel kode di sini)*

### 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:** Informasi metadata penting karena membantu dalam tahap preprocessing sebelum analisis lanjutan. Misalnya, dimensi menentukan apakah gambar perlu di-resize agar seragam dalam dataset, jumlah kanal memastikan apakah gambar diproses dalam format RGB atau grayscale, dan tipe data berpengaruh pada presisi komputasi serta kebutuhan memori. Ukuran file juga relevan untuk efisiensi penyimpanan dan kecepatan pemrosesan.

*(Tambahkan sel kode di sini)*

### 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: 1.Channel dominan → dilihat dari kurva mana yang memiliki frekuensi lebih tinggi pada rentang intensitas tertentu.

2.Kontras gambar → jika histogram menyebar luas (0–255), gambar punya kontras tinggi; kalau terkumpul di tengah atau sisi tertentu, berarti kontras rendah/terbatas.

3.Sebaran intensitas → apakah merata, menumpuk di area gelap (0–100), menengah (100–180), atau terang (180–255). Ini memberi gambaran apakah gambar cenderung gelap, terang, atau seimbang.

(Tambahkan sel kode di sini)

## B5. Analisis Ringkas (Wajib)

**Jawab pertanyaan berikut:**

**Relasi histogram dengan kesan visual:** Pola histogram memiliki hubungan langsung dengan kesan visual gambar. Jika histogram lebih banyak terkonsentrasi di sisi kanan (nilai tinggi), gambar terlihat cenderung terang, sedangkan jika dominan di sisi kiri (nilai rendah), gambar tampak gelap. Puncak yang menonjol pada salah satu kanal RGB menunjukkan warna dominan yang memberi nuansa utama pada gambar, misalnya kanal merah tinggi menandakan kesan hangat. Sebaran histogram yang lebar menandakan kontras tinggi karena terdapat rentang luas antara area gelap dan terang, sementara sebaran sempit mencerminkan kontras rendah sehingga gambar tampak datar. Dengan demikian, histogram menjadi representasi kuantitatif dari apa yang secara visual kita rasakan pada gambar.

Jawaban Anda: \_\_\_\_\_

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

import cv2
import matplotlib.pyplot as plt

# Path file
img_path = "/media/Networking dinner telkom.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("Networking Dinner Telkom (Format RGB)")
plt.axis("off")
plt.show()
```

Networking Dinner Telkom (Format RGB)



```
#B3 TODO: Informasi Dasar

import cv2
import os

# Path file
img_path = "/media/Networking dinner telkom.jpg"
```



```
# Baca gambar
img = cv2.imread(img_path)

# Ambil informasi metadata
height, width, channels = img.shape
dtype = img.dtype
file_size = os.path.getsize(img_path) / 1024 # ukuran KB

print(f"Dimensi: {height} x {width}")
print(f"Jumlah kanal: {channels}")
print(f"Type data (dtype): {dtype}")
print(f"Ukuran file: {file_size:.2f} KB")
```

```
Dimensi: 1200 x 1600
Jumlah kanal: 3
Type data (dtype): uint8
Ukuran file: 376.28 KB
```

```
#B4 TODO: Histogram Warna

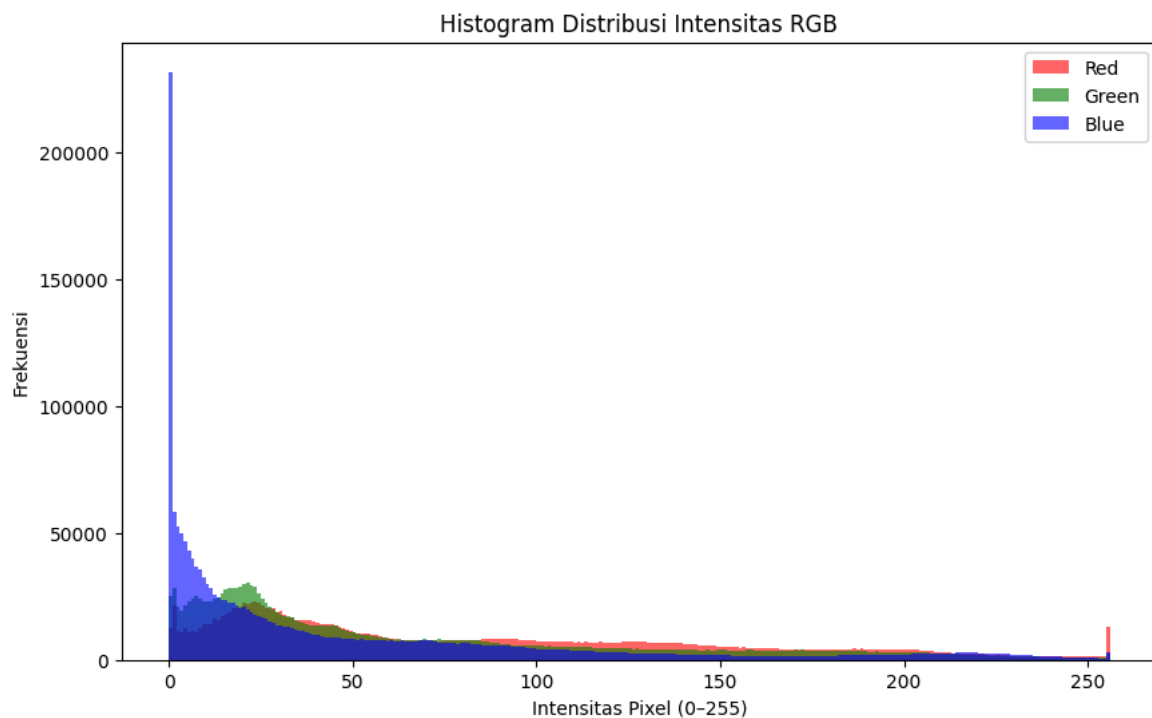
import cv2
import matplotlib.pyplot as plt

# Path gambar
img_path = "/media/Networking dinner telkom.jpg"

# Baca gambar (OpenCV default: BGR)
img = cv2.imread(img_path)
b, g, r = cv2.split(img)

# Plot histogram
plt.figure(figsize=(10,6))
plt.hist(r.ravel(), bins=256, range=(0,256), color='r', alpha=0.6, label='Red')
plt.hist(g.ravel(), bins=256, range=(0,256), color='g', alpha=0.6, label='Green')
plt.hist(b.ravel(), bins=256, range=(0,256), color='b', alpha=0.6, label='Blue')

plt.title("Histogram Distribusi Intensitas RGB")
plt.xlabel("Intensitas Pixel (0-255)")
plt.ylabel("Frekuensi")
plt.legend()
plt.show()
```



## ✓ Bagian C — Video

### C1. Deskripsi Data

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

- Jenis video: Pemandangan (aktivitas, pemandangan, tutorial, dll.)
- Sumber: Rekaman Sendiri (rekaman sendiri, dataset publik, dll.)
- Durasi target: 9 detik (disarankan  $\leq 30$  detik untuk efisiensi)
- Alasan pemilihan: karena bagus pemandangannya.

**Path file:** [/media/Basecamp\\_Ratai.mp4](#) (isi nama file Anda nanti di kode)

## C2. TODO: Baca & Metadata

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

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

**Analisis yang diperlukan:** Parameter-parameter metadata video sangat penting untuk analisis maupun aplikasi. Resolusi berhubungan dengan detail visual yang bisa diproses (misalnya, deteksi objek akan lebih akurat pada resolusi tinggi). Frame rate (fps) menentukan kelancaran pergerakan; fps tinggi diperlukan untuk aplikasi real-time seperti motion tracking. Jumlah frame dan durasi membantu dalam estimasi panjang konten, sinkronisasi audio, serta kebutuhan penyimpanan. Sementara itu, klasifikasi resolusi (HD, Full HD, 4K) mempermudah dalam menentukan kualitas video yang sesuai dengan perangkat target atau batasan komputasi. Dengan mengetahui metadata ini, kita bisa menyesuaikan preprocessing, kompresi, atau pipeline analisis video secara lebih tepat.

*(Tambahkan sel kode di sini)*

## 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 $\rightarrow$ RGB** sebelum ditampilkan
- Subplot dengan judul frame dan timestamp

**Analisis yang diperlukan:** Ketiga frame ini memberikan gambaran umum isi video dari awal hingga akhir. Frame pertama biasanya menampilkan kondisi awal, misalnya suasana sebelum aktivitas utama berlangsung. Frame tengah merepresentasikan bagian inti video, di mana aktivitas utama biasanya terlihat jelas. Sementara frame terakhir memberikan informasi mengenai kondisi penutup atau akhir adegan. Dengan melihat perbedaan visual antar frame ini, kita bisa memahami dinamika pergerakan atau alur cerita dalam video tanpa harus menonton keseluruhan durasi. Sampel frame ini juga bermanfaat untuk video summarization atau pengecekan cepat konten video.

*(Tambahkan sel kode di sini)*

## 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:** Resolusi dan fps video ini sudah cukup sesuai untuk kebutuhan seperti media sosial atau dokumentasi kegiatan. Resolusi HD/Full HD memastikan detail visual tetap jelas, sedangkan fps yang stabil memungkinkan gerakan terlihat halus tanpa patah-patah. Untuk keperluan kuliah daring atau presentasi, kombinasi ini juga ideal karena menjaga keseimbangan antara kualitas gambar dan ukuran file, sehingga mudah diakses tanpa membutuhkan bandwidth terlalu besar.

```
#C2 TODO: Baca & Metadata
import cv2

# Path file video
video_path = "/media/Basecamp_Ratai.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 == 1280 and height == 720:
```

```

        resolution_type = "HD (720p)"
    elif width == 1920 and height == 1080:
        resolution_type = "Full HD (1080p)"
    elif width == 3840 and height == 2160:
        resolution_type = "4K (2160p)"
    else:
        resolution_type = f"Custom ({width}x{height})"

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

cap.release()

```

```

=== Metadata Video ===
Resolusi      : 720 x 1280
Frame rate    : 59.97 fps
Total frame   : 556
Durasi       : 9.27 detik
Klasifikasi   : Custom (720x1280)

```

```

#C3 TODO: Tampilkan 3 Frame (Awal-Tengah-Akhir)
import cv2
import matplotlib.pyplot as plt

# Path video
video_path = "/media/Basecamp_Ratai.mp4"
cap = cv2.VideoCapture(video_path)

# Metadata dasar
total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
fps = cap.get(cv2.CAP_PROP_FPS)




# Tentukan indeks frame
frame_indices = [0, total_frames // 2, total_frames - 1]
frames = []
timestamps = []

for idx in frame_indices:
    cap.set(cv2.CAP_PROP_POS_FRAMES, idx)
    ret, frame = cap.read()
    if ret:
        # Konversi BGR → RGB
        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        frames.append(frame_rgb)
        timestamps.append(idx / fps)

cap.release()

# Plot hasil
plt.figure(figsize=(15,5))
for i, (frame, ts, idx) in enumerate(zip(frames, timestamps, frame_indices)):
    plt.subplot(1, 3, i+1)
    plt.imshow(frame)
    plt.axis("off")
    plt.title(f"Frame {idx}\nTimestamp: {ts:.2f}s")
plt.suptitle("Frame Representatif Video", fontsize=14)
plt.show()

```

Frame 0 Timestamp: 0.00s	Frame Representative Video Frame 278 Timestamp: 4.64s	Frame 555 Timestamp: 9.25s
		

Double-click (or enter) to edit

## Perbandingan & Kesimpulan

## Perbandingan Representasi Media

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

Audio (1D - Temporal)

- Representasi: Sinyal 1 dimensi berupa amplitudo terhadap waktu.
- Visualisasi utama: Waveform, spectrogram, dan MFCC.
- Informasi yang diperoleh: Pola intensitas suara, variasi frekuensi, kestabilan nada, hingga karakteristik akustik.

### Gambar (2D - Spasial)

- Representasi: Matriks piksel dua dimensi dengan nilai intensitas warna (RGB).
- Visualisasi utama: Tampilan citra, histogram distribusi warna.
- Informasi yang diperoleh: Objek dominan, komposisi warna, pencahayaan, serta distribusi intensitas yang memengaruhi kesan visual.

Video (2D + Waktu - Spatio-temporal)

- Representasi: Urutan frame gambar 2D yang tersusun dalam dimensi waktu.
- Visualisasi utama: Sampel frame, metadata (fps, resolusi, durasi).
- Informasi yang diperoleh: Dinamika pergerakan, perubahan adegan dari awal hingga akhir, serta alur aktivitas yang tidak bisa ditangkap dari satu gambar statis.

## Refleksi Pembelajaran

3 Poin yang Saya Pelajari:

1. Saya belajar bagaimana cara memuat banyak media mulai dari audio, gambar, video.

2. Belajar memvisualisasikan berbagai data dari setiap media yang telah dimuat.
3. Belajar menganalisis setiap grafik yang ditampilkan dari setiap media.

## 2 Hal yang Masih Membingungkan/Ingin Diperdalam:

1. Saya masih bingung dalam membaca berbagai grafiknya seperti dalam membaca maksud dari spectrogram, waveform dan lain-lain.
2. Karena masih bingung cara membaca grafiknya jadi saya bingung juga untuk membuat kesimpulan analisisnya.

## Sumber Data & Referensi

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

- **Audio:** <https://youtu.be/iY93pTUU1RU?si=i-myn9VLMYgZHPUC> (Youtube Video)
- **Gambar:** Foto Sendiri
- **Video:** Video Sendiri
- **Referensi teknis:** ChatGPT.





## Rubrik Penilaian

### Distribusi Bobot Penilaian

Aspek Penilaian	Bobot	Deskripsi
Kelengkapan	35%	Semua langkah inti dikerjakan sesuai checklist
Kualitas Visualisasi	20%	Judul, label sumbu, colorbar, legend, keterbacaan plot
Analisis & Interpretasi	30%	Kemampuan interpretasi hasil, bukan sekadar output mentah
Kerapihan & Struktur	10%	Markdown jelas, kode modular, dokumentasi baik
Orisinalitas & Penguasaan	5%	Pemahaman saat presentasi acak

### Detail Kriteria Penilaian

#### Kelengkapan (35%)

-  Semua 4 visualisasi audio (metadata, waveform, spectrogram, MFCC)
-  Semua 3 visualisasi gambar (display RGB, metadata, histogram)
-  Semua 2 visualisasi video (metadata, frame extraction)
-  Analisis ringkas untuk setiap bagian

#### Kualitas Visualisasi (20%)

- Plot memiliki judul yang informatif dan deskriptif
- Label sumbu X dan Y jelas dan sesuai
- Colorbar/legend tersedia jika diperlukan
- Ukuran plot proporsional dan mudah dibaca

#### Analisis & Interpretasi (30%)

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

#### Kerapihan & Struktur (10%)

- Markdown terstruktur dengan heading yang konsisten
- Kode bersih, terkompartemen, dan mudah dibaca
- Dokumentasi yang memadai
- Flow logical dari satu bagian ke bagian lain

#### Orisinalitas & Penguasaan (5%)

- **PENTING:** Jika saat presentasi acak Anda tidak mampu menjelaskan kode yang Anda tulis atau menunjukkan ketergantungan buta pada AI/copy-paste, **nilai tugas akan dianggap 0.**
- Kemampuan menjelaskan logika dan alur pemikiran
- Pemahaman konsep di balik implementasi kode

### Proporsi Penilaian Total





- Proporsi penilaian hanya 80%, 20% lagi akan didasarkan pada kecepatan pengumpulan tugas

- Sehingga:  $0.8 \times \text{penilaian dosen} + \text{nilai waktu pengumpulan}$

## 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