

# Tugas Analisis Multimedia: Audio, Gambar, Video

**Mata Kuliah:** Sistem & Teknologi Multimedia

**Nama:** Zefanya Danovanta Tarigan

**NIM:** 122140101

---

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

# Set matplotlib untuk menampilkan plot inline
%matplotlib inline

# Tampilkan versi Library untuk dokumentasi
print("Library versions:")
print(f"NumPy: {np.__version__}")
print(f"Matplotlib: {matplotlib.__version__}")
print(f"Librosa: {librosa.__version__}")
```

```
print(f"OpenCV: {cv2.__version__}")

# Tambahkan import Lain jika diperlukan saat mengerjakan tugas
import pandas as pd

print(f"Pandas: {pd.__version__}")

Library versions:
NumPy: 2.2.0
Matplotlib: 3.10.6
Librosa: 0.11.0
OpenCV: 4.12.0
Pandas: 2.3.2
```

## Petunjuk Umum Pengerajan

### 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 Sirine
- Sumber: Pixabay [Link Sumber](#)
- Format file: MP3
- Alasan pemilihan: Ingin mengetahui informasi spesifik dari audio sirine

**Path file:** `data\audio_suara_sirine.mp3`

---

### 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 [66]: PATH_AUDIO = os.path.join(os.getcwd(), 'data', 'audio_suara_sirine.mp3')
y = None
Kanal = None

y, sr = librosa.load(PATH_AUDIO, sr=None, mono =None)

if y.ndim == 1:
    total_sample = len(y)
    Kanal = "Mono (1)"

else:
    total_sample = y.shape[1]
    Kanal = "Stereo (2)"
```

```

print(f"Sample Rate : {sr} Hz")
print(f"Durasi : {y.shape[1]/sr:.2f} Detik")
print(f"Jumlah Kanal : {Kanal} ")
print(f"Jumlah Total Sample : {y.shape[1] * y.shape[0]}")

```

Sample Rate : 44100 Hz  
 Durasi : 15.05 Detik  
 Jumlah Kanal : Stereo (2)  
 Jumlah Total Sample : 1327104

---

### A3. TODO: Waveform

**Instruksi:** Plot waveform audio dengan:

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

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

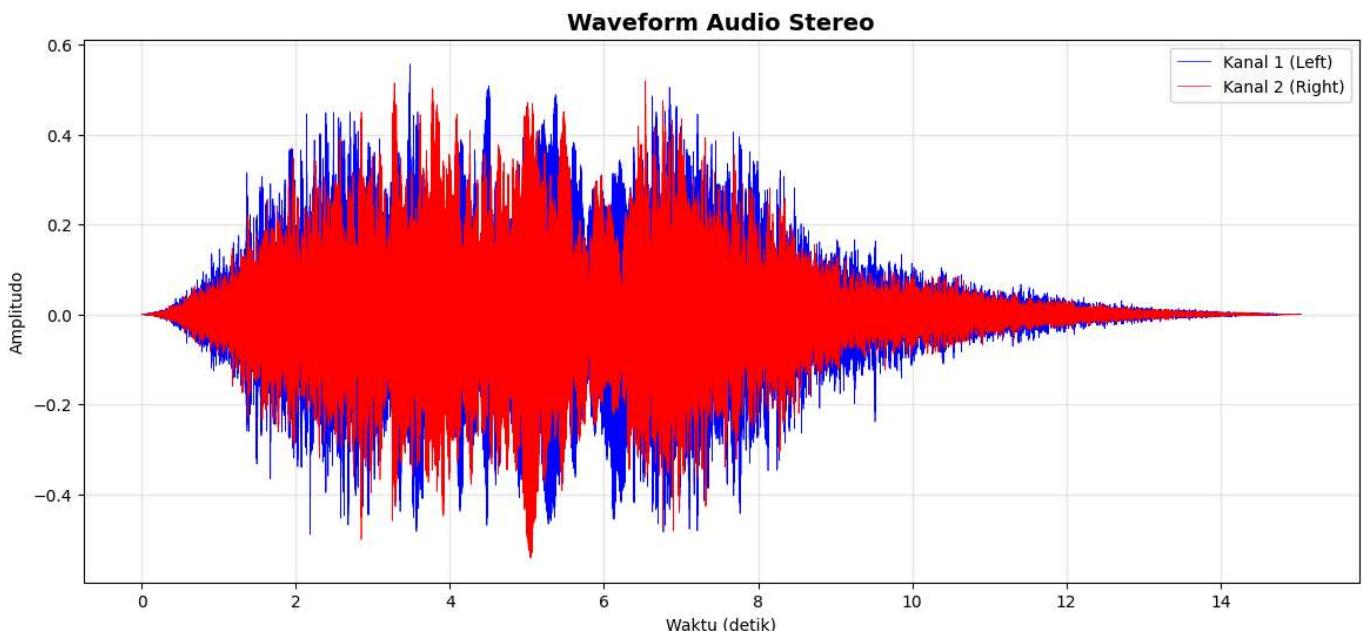
In [67]:

```

t = np.linspace(0, y.shape[1]/sr, y.shape[1])

fig= plt.figure(figsize=(14, 6))
plt.plot(t, y[0], color='blue', linewidth=0.5, label="Kanal 1 (Left)")
plt.plot(t, y[1], color='red', linewidth=0.5, label="Kanal 2 (Right)")
plt.legend()
plt.title('Waveform Audio Stereo', fontsize=14, fontweight='bold')
plt.xlabel('Waktu (detik)')
plt.ylabel('Amplitudo')
plt.grid(True, alpha=0.3)
plt.show()

```



#### Analisis Hasil Waveform :

Suara Sirine Tersebut memiliki durasi 15 Detik, Suara tersebut memiliki jumlah kanal 2 atau disebut dengan stereo. Audio tersebut memiliki rentang amplitudo -0.6 hingga 0.6

1. Pada detik 0 hingga 1, amplitudo-nya masih rendah karna suara sirine di awal belum terlalu bising
  2. Pada detik 2 hingga 6, amplitudo-nya sudah tinggi karna titik tertinggi dari suara sirine
  3. Pada detik 8 hingga selesai, suara amplitudo terlihat menurun karna suara sirine tersebut sudah mengecil sampai sunyi
- 

## A4. TODO: Spectrogram log-dB

**Instruksi:** Hitung STFT dan tampilkan spectrogram dalam skala log-dB:

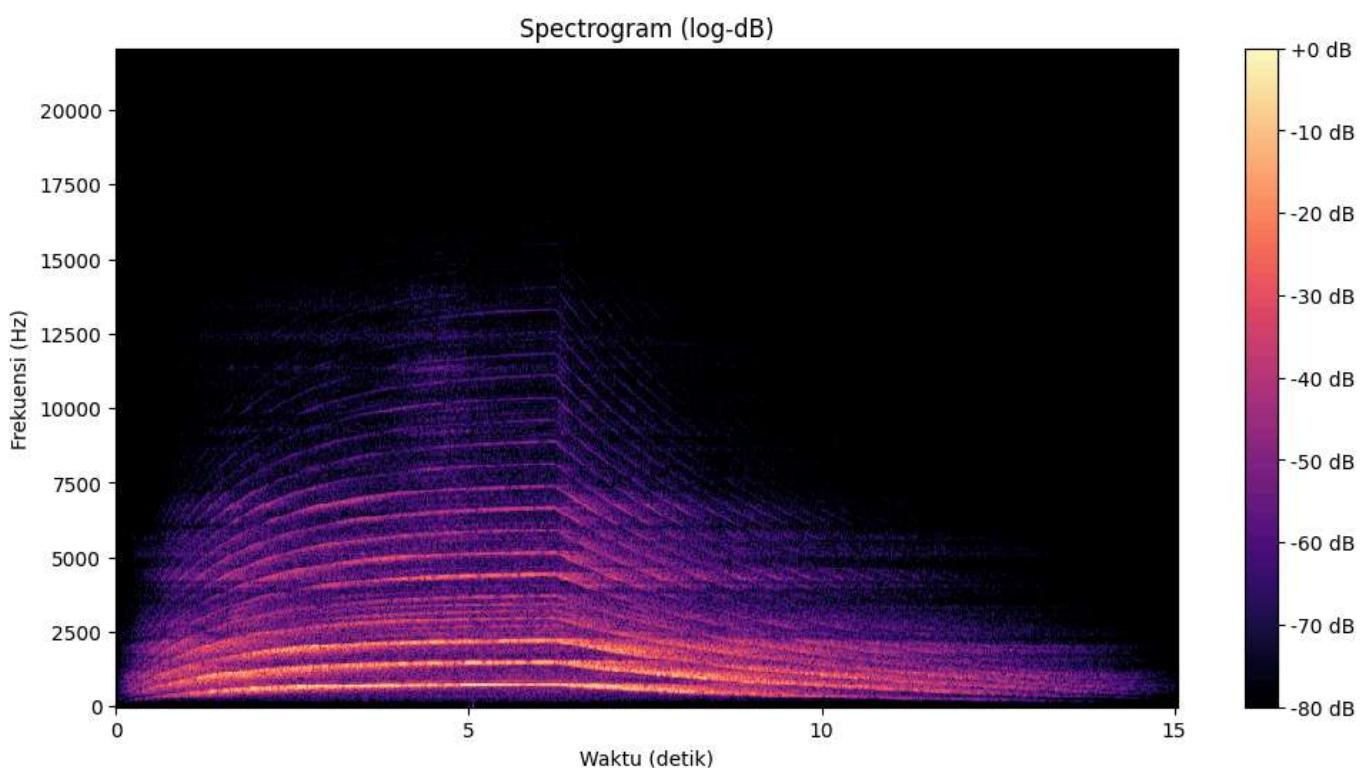
- Gunakan parameter standar (`n_fft=1024, hop_length=256`)
- Tampilkan dengan colorbar
- Label sumbu: waktu (detik) dan frekuensi (Hz)

**Analisis yang diperlukan:** Jelaskan perbedaan informasi yang didapat dari spectrogram dibanding waveform.

```
In [68]: audio_data, sample_rate = librosa.load(PATH_AUDIO, sr=None)

# Hitung STFT
n_fft = 1024
hop_length = 256
stft = librosa.stft(audio_data, n_fft=n_fft, hop_length=hop_length)
stft_db = librosa.amplitude_to_db(np.abs(stft), ref=np.max)

# Plot spectrogram
plt.figure(figsize=(12, 6))
librosa.display.specshow(stft_db, sr=sample_rate, hop_length=hop_length, x_axis='time', y_ax
plt.colorbar(format='%+2.0f dB')
plt.title('Spectrogram (log-dB)')
plt.xlabel('Waktu (detik)')
plt.ylabel('Frekuensi (Hz)')
plt.show()
```



## Perbedaan Dari Waveform dengan Spectogram yang dihasilkan

### Waveform

1. Waveform menunjukkan alur bising nya audio tersebut dapat dilihat dari rentang amplitudo tersebut
2. Waveform menunjukkan jumlah dari jenis kanal yang dihasilkan dari audio tersebut

### Spectrogram

1. Spectrogram menunjukkan semakin gelap warnanya maka semakin kecil bunyi nya (Db nya)
2. Spectrogram menunjukkan tingginya frekuensi di setiap detik nya

---

## A5. TODO: MFCC

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

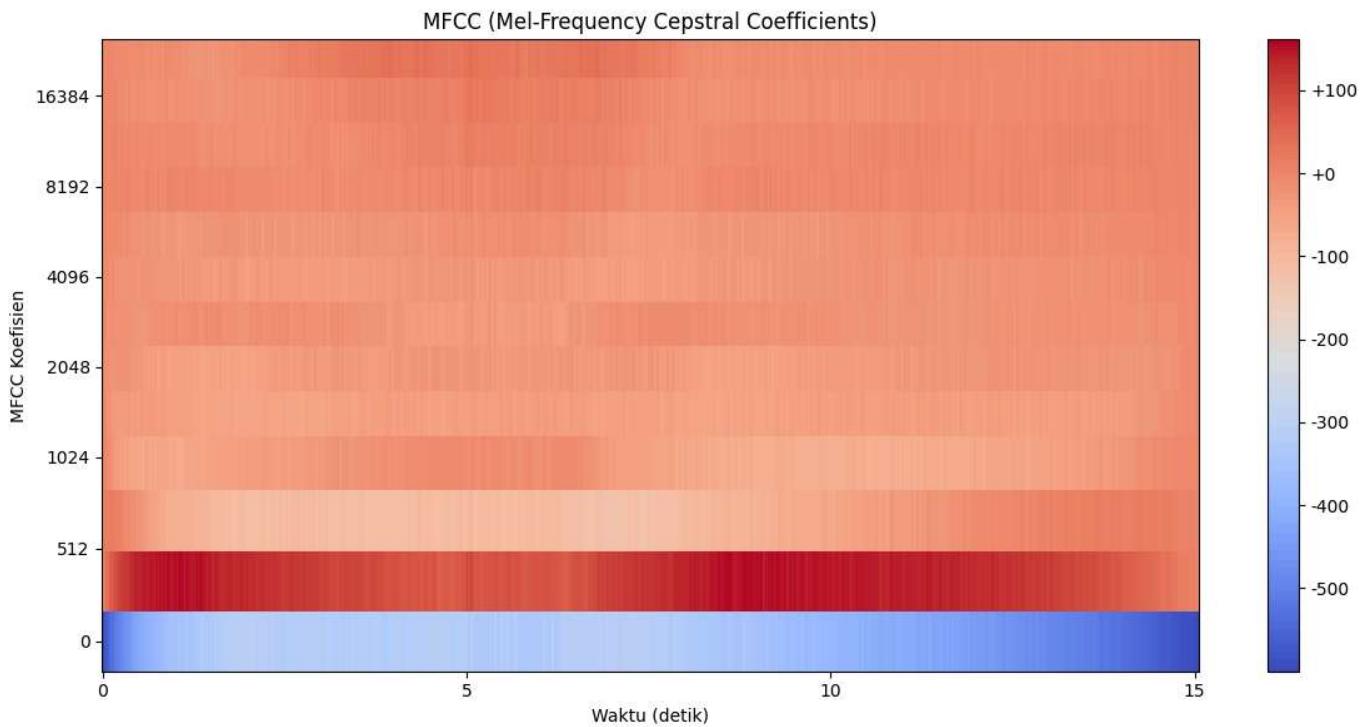
- Sumbu X: waktu (frame)
- Sumbu Y: koefisien MFCC (1-13)
- Gunakan colorbar dan judul yang jelas

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

```
In [69]: audio_data, sample_rate = librosa.load(PATH_AUDIO, sr=None)
mfccs = librosa.feature.mfcc(y=audio_data, sr=sample_rate, n_mfcc=13)

plt.figure(figsize=(12, 6))
img = librosa.display.specshow(mfccs, sr=sample_rate, x_axis='time', y_axis='mel')

plt.colorbar(format='%.2f')
plt.title('MFCC (Mel-Frequency Cepstral Coefficients)')
plt.xlabel('Waktu (detik)')
plt.ylabel('MFCC Koefisien')
plt.tight_layout()
plt.show()
```



**Analisis MFCC** Jika melihat grafik MFCC, didapatkan informasi warna terlihat cenderung stabil dan memiliki pola yang konsisten

---

## A6. Analisis Ringkas (Wajib)

**Jawab pertanyaan berikut:**

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

*Jawaban Anda:*

- Waveform dapat melihat rentang Amplitudo dan Alur kerasnya suara di tiap detik dari audio tersebut serta bisa melihat jumlah kanal nya. sedangkan jika Spectrogram kita hanya mendapatkan informasi Db dan tingginya frekuensi dari audio tersebut di tiap detiknya

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

*Jawaban Anda:* MFCC Menunjukkan lagu ini punya struktur yang konsisten.

# Bagian B — Gambar

## B1. Deskripsi Data

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

- Jenis gambar: pemandangan
- Sumber: Google [Link Sumber](#)
- Format file: JPG
- Alasan pemilihan: Ingin Melihat Komposisi dari pemandangan Lampung khususnya Siger

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

---

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

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

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

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

```
In [70]: PATH_IMAGE = os.path.join(os.getcwd(), 'data', 'gambar_pemandangan.jpg')

img = cv2.imread(PATH_IMAGE)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

plt.figure(figsize=(10, 8))
plt.imshow(img_rgb)
plt.title('Foto Pemandangan Siger')
plt.axis('off')
plt.tight_layout()
plt.show()
```

Foto Pemandangan Siger



### Analisis Dataset Gambar

1. **Objek Dominan :** Memiliki Objek yang dominan lautan, langit dan tumbuhan
2. **Kondisi Pencahayaan :** Memiliki Kondisi pencahayaan yang terang karna di ambil saat kondisi cerah dan Alami
3. **Komposisi Warna :** Memiliki warna dominan Hijau dan biru karena terdapat objek laut, langit dan tumbuhan

---

## B3. TODO: Informasi Dasar

**Instruksi:** Tampilkan informasi metadata gambar:

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

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

```
In [71]: height, width , channel = img.shape  
data_type= img.dtype  
  
size = height * width * channel / 1024  
  
print(f"Dimensi : {height} x {width} px")  
print(f"Jumlah Kanal : {channel} channel")  
print(f"Tipe Data : {data_type}")  
print(f"Ukuran File Dalam Memori : {size:.2f} KB")
```

```
Dimensi : 789 x 1178 px  
Jumlah Kanal : 3 channel  
Tipe Data : uint8  
Ukuran File Dalam Memori : 2722.97 KB
```

### Analisis Informasi

- Informasi ini diperlukan untuk mengetahui detail dari foto tersebut secara spesifik, mulai dari dimensi foto tersebut dapat mengetahui panjang dan lebar foto tersebut, ukuran file dari gambar yang dimiliki dan tipe data yang dimiliki

---

## B4. TODO: Histogram Warna

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

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

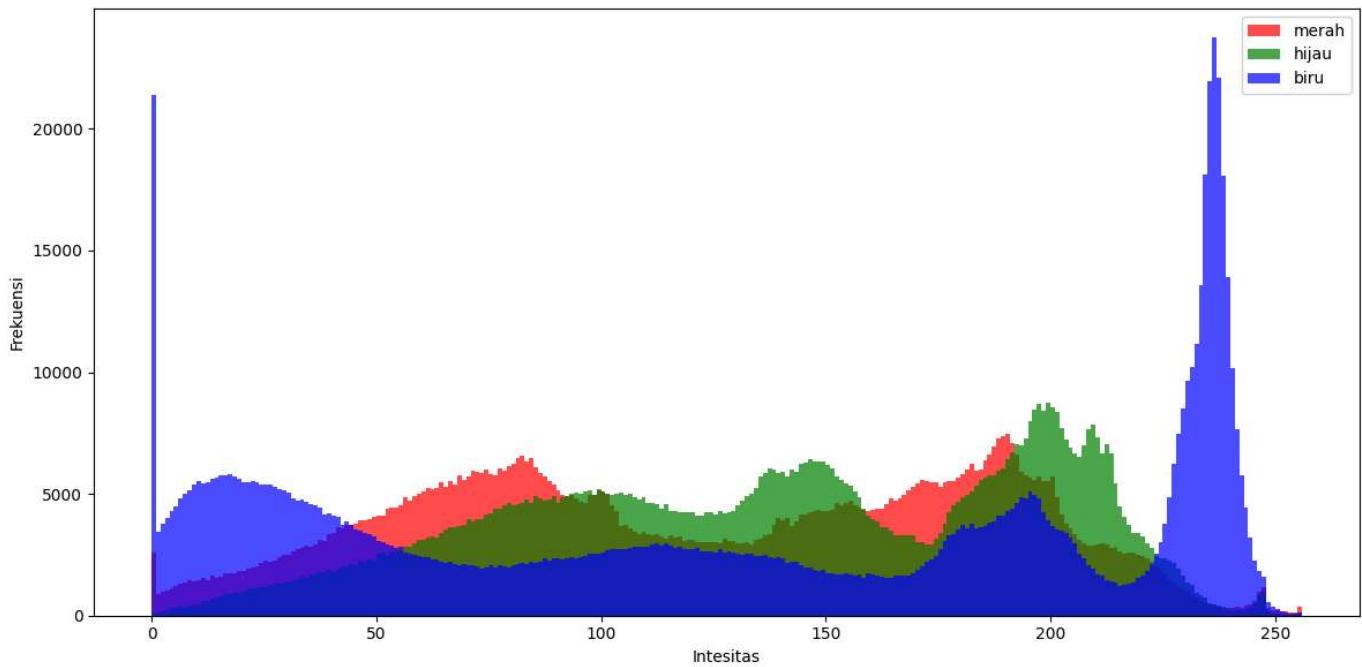
**Analisis yang diperlukan:** Analisis: channel mana yang dominan? Bagaimana kontras gambar? Seperti apa sebaran intensitasnya?

```
In [72]: r, g, b = cv2.split(img_rgb)  
  
plt.figure(figsize=(12,6))  
  
plt.hist(r.ravel(), bins=256, range=[0, 256], color='red', alpha=0.7, label="merah")  
plt.hist(g.ravel(), bins=256, range=[0, 256], color='green', alpha=0.7, label="hijau")  
plt.hist(b.ravel(), bins=256, range=[0, 256], color='blue', alpha=0.7, label="biru")
```

```

plt.xlabel("Intesitas")
plt.ylabel("Frekuensi")
plt.legend()
plt.tight_layout()
plt.show()

```



### Analisis Histogram Warna

1. **Warna Dominan :** Memiliki dominan warna biru dan Hijau karna dari gambar tersebut memiliki objek dominan langit, laut dan tumbuhan
2. **Sebaran Warna :**
  - Biru : Warna paling dominan karna gambar tersebut dalam kondisi terang / cerah
  - Hijau : Warna Terbanyak kedua karna terdapat objek pepohonan pada gambar tersebut
  - Merah : tidak terlalu dominan tapi menyebar rata
3. **Kontras Gambar :** Memiliki warna yang sangat kontras karna dominan warna biru

## B5. Analisis Ringkas (Wajib)

**Jawab pertanyaan berikut:**

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

*Jawaban Anda:* Pola Histogram menunjukkan warna biru dominan karena memiliki objek dominan seperti laut dan langit dan komposisi warnanya cerah dan warna terbanyak selanjutnya warna hijau karna terdapat pepohonan di gambar tersebut, lalu untuk merah sedikit karna gambar tersebut diambil dalam kondisi cerah dan tidak gelap

## Bagian C — Video

### C1. Deskripsi Data

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

- Jenis video: Hewan Kucing
- Sumber: Tiktok [Link Sumber](#)
- Durasi target: 27 Detik
- Alasan pemilihan: mudah di dapatkan dan durasinya tidak terlalu panjang

**Path file:** data/vidio\_kucing.mp4

---

## C2. TODO: Baca & Metadata

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

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

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

```
In [73]: PATH_VIDEO = os.path.join(os.getcwd(), 'data', 'vidio_kucing.mp4')

vid = cv2.VideoCapture(PATH_VIDEO)
lebar = int(vid.get(cv2.CAP_PROP_FRAME_WIDTH))
panjang = int(vid.get(cv2.CAP_PROP_FRAME_HEIGHT))
fps = vid.get(cv2.CAP_PROP_FPS)
frame_count = int(vid.get(cv2.CAP_PROP_FRAME_COUNT))
durasi = frame_count / fps

if lebar >= 3840 and panjang >= 2160:
    fps_class= "4K Ultra HD"
elif lebar >= 2560 and panjang >= 1440:
    fps_class= "2K Quad HD"
elif lebar >= 1920 and panjang >= 1080:
    fps_class= "Full HD (1080p)"
elif lebar >= 1280 and panjang >= 720:
    fps_class= "HD (720p)"
else:
    fps_class= "Standard Definition"

print(f"Resolusi : {lebar} x {panjang} px")
print(f"Frame rate : {fps:.2f} fps")
print(f"Jumlah Total Frame : {frame_count}")
print(f"Durasi : {durasi:.2f} detik")
print(f"Klasifikasi resolusi : {fps_class}")
```

```
Resolusi : 576 x 1024 px
Frame rate : 28.26 fps
Jumlah Total Frame : 765
Durasi : 27.07 detik
Klasifikasi resolusi : Standard Definition
```

### Analisis Parameter Vidio

1. Resolusi

- Menunjukkan detail kualitas dari video tersebut, semakin tinggi resolusi video nya maka video yang dihasilkan semakin tajam gambarnya

### 2. Frame Rate

- Menunjukkan Jumlah gambar per detik, semakin besar fps yang dihasilkan maka semakin halus video tersebut

### 3. Durasi

- Menunjukkan berapa lama video tersebut, semakin lama durasi video maka semakin besar kapasitas memori yang dibutuhkan

### 4. Klasifikasi Resolusi

- Setiap resolusi punya level masing-masing dan kegunaannya berbeda, contohnya :
  - HD (720p): nonton di HP
  - Full HD (1080p): untuk nonton di laptop/TV.

## C3. TODO: Tampilkan 3 Frame (Awal–Tengah–Akhir)

**Instruksi:** Ambil dan tampilkan 3 frame representatif:

- Frame pertama (index 0)
- Frame tengah (index ~total\_frame/2)
- Frame terakhir (index total\_frame-1)
- **Konversi BGR→RGB** sebelum ditampilkan
- Subplot dengan judul frame dan timestamp

**Analisis yang diperlukan:** Deskripsikan perbedaan visual antar frame dan apa yang dapat dipelajari dari sampel frame ini.

```
In [74]: import cv2
import matplotlib.pyplot as plt

total_frames = int(vid.get(cv2.CAP_PROP_FRAME_COUNT))
fps = int(vid.get(cv2.CAP_PROP_FPS))

# Simpan frame terakhir valid Lewat Loop
last_frame = None
vid.set(cv2.CAP_PROP_POS_FRAMES, 0)
idx = 0
while True:
    ret, frame = vid.read()
    if not ret:
        break
    last_frame = (idx, frame)
    idx += 1

last_frame_idx, last_frame_img = last_frame

frames_idx = [0, total_frames // 2]

frames = []
timestamps = []

# Ambil frame awal & tengah
for idx in frames_idx:
    vid.set(cv2.CAP_PROP_POS_FRAMES, idx)
    ret, frame = vid.read()
```

```

if ret:
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    frames.append(frame_rgb)
    timestamp = idx / fps
    timestamps.append(f"{{int(timestamp//60)}}:{int(timestamp%60):02d}")

frame_rgb = cv2.cvtColor(last_frame_img, cv2.COLOR_BGR2RGB)
frames.append(frame_rgb)
timestamp = last_frame_idx / fps
timestamps.append(f"{{int(timestamp//60)}}:{int(timestamp%60):02d}")
frames_idx.append(last_frame_idx)

plt.figure(figsize=(18, 6))
for i in range(len(frames)):
    plt.subplot(1, 3, i+1)
    plt.imshow(frames[i])
    plt.title(f"Frame {frames_idx[i]}\nTimestamp: {timestamps[i]}")
    plt.axis("off")

plt.tight_layout()
plt.show()

```



### Analisis Frame

Jumlah frame pada vidio tersebut : 764

#### 1. Frame Awal

- Diambil dari awal vidio pada durasi 0.00 detik

#### 2. Frame tengah

- Diambil dari durasi 0.13 Detik dan pada frame ke 382, terdapat perbedaan pada frame awal yaitu di bagian ekor kucing nya sedikit bergerak ke bawah

#### 3. Frame Akhir

- Diambil dari akhir vidio pada durasi 0.27 detik dan pada frame terakhir yaitu ke 764, terdapat perbedaan pada frame kedua yaitu wajah kucing tersebut sedikit bergerak ke arah kamera

## 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:* Masih tergolong ke dalam Standar Definition (SD), Vidio ini sudah cocok untuk media sosial dan tepat digunakan karna data vidio ini juga berasal dari sosial media yaitu Tiktok

# Perbandingan & Kesimpulan

## Perbandingan Representasi Media

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

### Audio (1D - Temporal)

- Representasi: Audio Hanya memiliki array 1 Dimensi namun bisa jumlah kanalnya bisa berbeda tiap audio nya
- Visualisasi utama: Waveform menunjukkan keras-lembut suara sepanjang waktu, spectrogram menunjukkan nada rendah–tinggi (frekuensi) seiring waktu kemudian dan MFCC untuk melihat ciri khas suara
- Informasi yang diperoleh: Dapat mengetahui pola audio yang dihasilkan sebelum mendengarkan audio tersebut, seperti tingginya frekuensi atau amplitudo nya

### Gambar (2D - Spasial)

- Representasi: Gambar memiliki array 2 dimensi dan memiliki informasi yaitu panjang, lebar dan warna. dan warna tersebut memiliki 3 channel yaitu RGB (Merah, Hijau, Biru)
- Visualisasi utama: RGB dan Histogram untuk melihat sebaran intensitas tiap channel nya
- Informasi yang diperoleh: warna terang di dapatkan jika histogramnya memiliki dominan warna biru

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

- Representasi: Video merupakan gabungan dari banyaknya gambar yang dihasilkan
- Visualisasi utama: 3 Frame (Awal, Tengah dan Akhir Pada Video)
- Informasi yang diperoleh: Dapat mengetahui perubahan visual dari tiap detiknya dan seberapa halus gerakannya.

---

## Refleksi Pembelajaran

### 3 Poin yang Saya Pelajari:

1. Setiap jenis Data (audio, gambar, video) mempunyai spesifikasinya yang berbeda beda dan menganalisisnya berbeda beda juga.
2. Informasi Tiap data yang dimiliki itu penting karna dapat mengetahui kulitas serta ukuran pada data yang kita miliki
3. Bisa mempelajari dari library yang beru digunakan seperti CV2

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

1. Bagaimana cara komputer menangkap banyaknya frame dalam video ?
  2. Mengapa CV2 bisa sangat canggih dalam mendeteksi warna ?
- 

## Sumber Data & Referensi

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

- **Audio:** [Link Sumber](#)
- **Gambar:** [Link Sumber](#)
- **Video:** [Link Sumber](#)
- **Referensi teknis:**
  - <https://support.google.com/youtube/answer/6375112?hl=id&co=GENIE.Platform%3DDesktop>
  -

## Rubrik Penilaian

### Distribusi Bobot Penilaian

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

---

### Detail Kriteria Penilaian

#### Kelengkapan (35%)

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

#### Kualitas Visualisasi (20%)

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

#### Analisis & Interpretasi (30%)

- Interpretasi menunjukkan pemahaman konsep

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

## Kerapihan & Struktur (10%)

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

## Originalitas & Penguasaan (5%)

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

## Proporsi Penilaian Total

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

## Aturan Kejujuran Akademik

### Penggunaan Referensi & AI yang Diperbolehkan

Anda **BOLEH** menggunakan:

- Dokumentasi resmi library (NumPy, Matplotlib, Librosa, OpenCV)
- Tutorial dan contoh kode dari sumber terpercaya
- AI tools (ChatGPT, GitHub Copilot, dll.) sebagai **alat bantu pembelajaran**
- Diskusi dengan teman untuk pemahaman konsep

### Syarat & Batasan WAJIB

Namun Anda **HARUS**:

-  **Memahami setiap baris kode** yang Anda masukkan ke notebook
-  **Menulis interpretasi dengan kata-kata sendiri**, bukan hasil copy-paste
-  **Mencantumkan sumber data dan referensi** yang digunakan, termasuk transkrip percakapan dengan AI dalam link atau teks
-  **Mampu menjelaskan logika dan alur pemikiran** saat presentasi acak

### Pelanggaran yang Berakibat Nilai 0

- **Plagiarisme atau penyalinan buta** dari sumber manapun
- **Copy-paste kode tanpa pemahaman** dan tidak dapat menjelaskan

- Menggunakan AI untuk mengerjakan seluruh **tugas** tanpa pembelajaran personal
- Tidak dapat menjawab pertanyaan dasar tentang kode yang dikumpulkan
- Menyalin pekerjaan teman atau bekerjasama dalam pengerjaan individual

## Persiapan Presentasi Acak

### Kemungkinan pertanyaan yang akan ditanyakan:

- "Jelaskan mengapa Anda menggunakan parameter ini di STFT?"
- "Apa arti dari pola yang terlihat di MFCC?"
- "Mengapa perlu konversi BGR ke RGB?"
- "Interpretasikan hasil histogram yang Anda buat"
- "Bagaimana cara kerja spectrogram?"

### Tips sukses:

- Pahami konsep dasar setiap teknik yang digunakan
- Latih menjelaskan dengan bahasa sederhana
- Siapkan justifikasi untuk setiap pilihan parameter
- Kuasai interpretasi setiap visualisasi yang dibuat

## Panduan Pengumpulan

### Berkas yang Harus Dikumpulkan

#### Wajib:

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

### Informasi Pengumpulan

### Checklist Sebelum Submit

- Semua cell sudah dijalankan dan menampilkan output
  - Nama file sesuai format: `NIM_Worksheet2.ipynb` dan `NIM_Worksheet2.pdf`
  - Semua TODO sudah diisi dengan lengkap
  - Analisis dan interpretasi sudah ditulis untuk setiap bagian
  - Sumber data dan referensi sudah dicantumkan
- 

### Export ke PDF:

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

