

# Tugas Analisis Multimedia: Audio, Gambar, Video

**Mata Kuliah:** Sistem & Teknologi Multimedia

**Nama:** Nashwa Putri Laisya

**NIM:** 122140180

---

## 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 # Saya tambahkan matplotlib agar dapat memanggil atribut
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: {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: 3.10.5  
 Librosa: 0.11.0  
 OpenCV: 4.12.0

## Petunjuk Umum Penggeraan

### 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: popsical music
- Sumber: YouTube
- Format file: WAV
- Alasan pemilihan: menarik karena gabungan antara pop dan classical music

**Path file:** `data/audio-popsical-DOTSPF.wav`

---

### A2. TODO: Muat & Metadata

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

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

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

```
In [2]: # Mengatur path audio dengan os agar bisa dijalankan di mana saja
path_audio = os.path.join(os.getcwd(), 'data', 'audio-popsical-DOTSPF.wav')

# Load file audio
y, sr = librosa.load(path_audio)

# Menampilkan sample rate
print(f"Sample Rate: {sr} Hz")

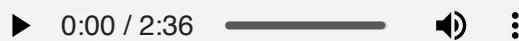
# Menghitung durasi
duration = librosa.get_duration(y=y, sr=sr)
# Menampilkan durasi
print(f"Durasi: {duration:.2f} detik")

# Menampilkan jumlah channel
print(f"Jumlah kanal: {'1 -> Mono' if y.ndim == 1 else '2 -> Stereo'}")

# Menampilkan jumlah total sampel
print(f"Total Sampel: {len(y)}")

# Menampilkan audio player
display(Audio(y, rate=sr))
```

Sample Rate: 22050 Hz  
 Durasi: 156.25 detik  
 Jumlah kanal: 1 -> Mono  
 Total Sampel: 3445282



### A3. TODO: Waveform

**Instruksi:** Plot waveform audio dengan:

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

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

```
In [3]: # Membuat array waktu untuk sumbu x dari detik 0 sampai durasi audio dengan
time = np.linspace(0, duration, len(y))

# Menentukan ukuran figure
plt.figure(figsize=(12, 4))

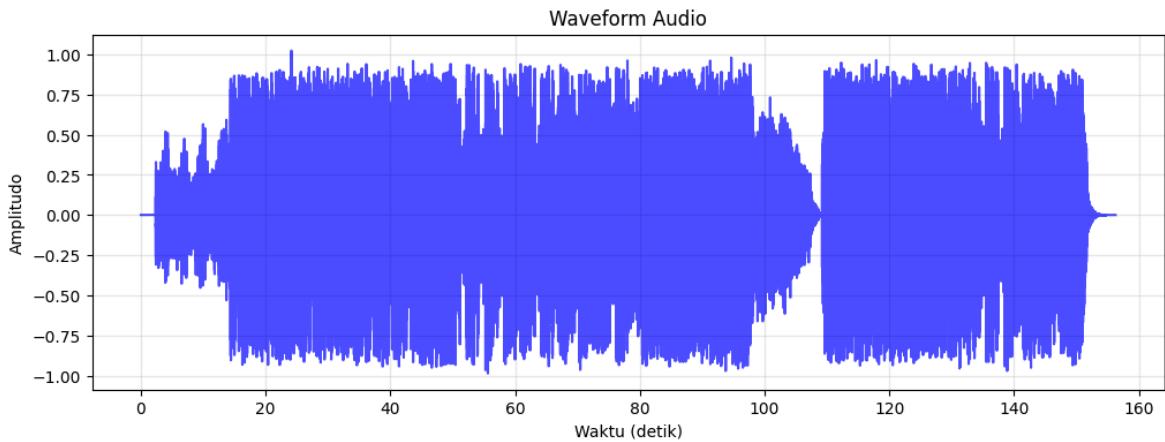
# Membuat plot garis
plt.plot(time, y, color='blue', alpha=0.7)
```

```
# Menambahkan judul
plt.title('Waveform Audio')

# Menambahkan label sumbu x
plt.xlabel('Waktu (detik)')

# Menambahkan label sumbu y
plt.ylabel('Amplitudo')

# Menambahkan grid
plt.grid(True, alpha=0.3)
```



### Analisis:

Berdasarkan waveform yang terbentuk, saya dapat melihat pola gelombang yang dihasilkan dari audio, misalnya dari detik 0 s.d detik 15 amplitudo yang terbentuk masih kecil kemudian setelah detik 15 menuju detik 20, amplitudo mulai naik. Biasanya pola seperti ini menunjukkan pola intro. Detik 20 s.d detik 95 menunjukkan amplitudo yang tinggi dan konsisten (tidak terlalu ekstrim perubahannya), amplitudo tinggi menunjukkan kalau audio yang dihasilkan keras. Dari detik 100 s.d detik 110 amplitudonya menurun seperti ada jeda pada audio. Kemudian setelah jeda di detik 110, amplitudo kembali menjadi tinggi lagi dari detik 110 s.d detik 150 yang artinya suara yang dihasilkan keras dan konsisten. Terakhir pada detik 150 s.d 155 amplitudo mengecil mendekati nol yang biasanya menunjukkan kalau bagian ini adalah outro.

## A4. TODO: Spectrogram log-dB

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

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

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

```
In [4]: # Menghitung STFT
D = librosa.stft(y, n_fft=1024, hop_length=256)

# Mengubah ke skala log-dB (amplitudo ke dB)
```

```

DB = librosa.amplitude_to_db(abs(D), ref=np.max)

# Menentukan ukuran figure
plt.figure(figsize=(12, 6))

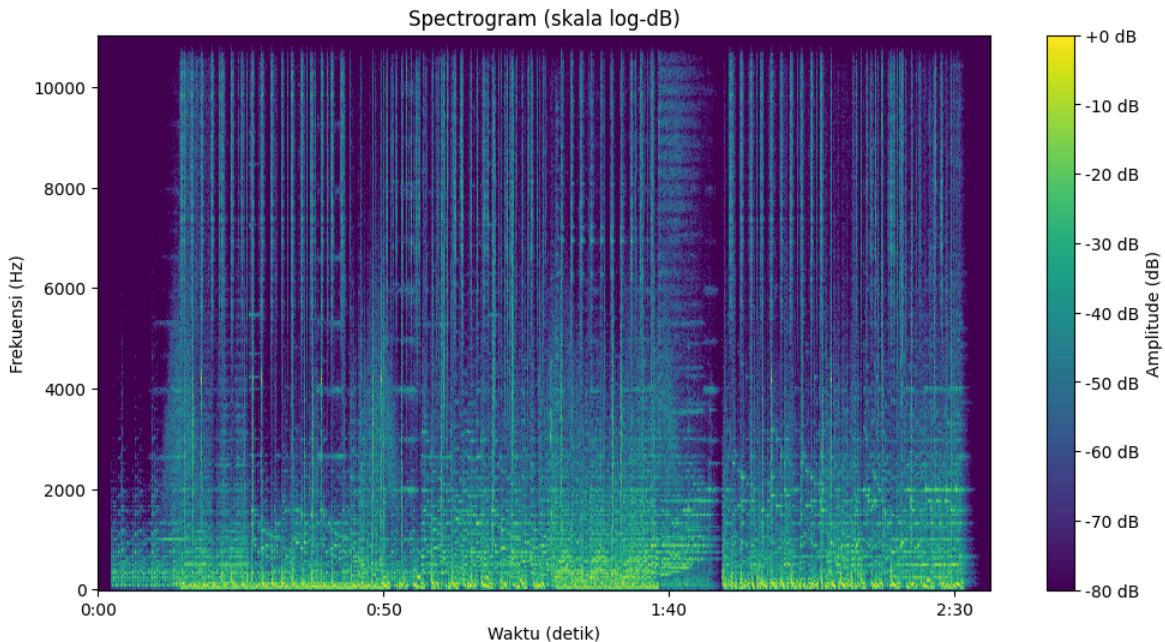
# Menampilkan spectrogram dalam skala log-dB
img = librosa.display.specshow(DB, sr=sr, hop_length=256, x_axis='time',

# Menambahkan colorbar
plt.colorbar(img, format='%.2f dB', label='Amplitude (dB)')

# Menambahkan label dan judul
plt.title('Spectrogram (skala log-dB)')
plt.xlabel('Waktu (detik)')
plt.ylabel('Frekuensi (Hz)')

```

Out[4]: Text(0, 0.5, 'Frekuensi (Hz)')



### Analisis:

Informasi yang didapatkan dari spectrogram jauh lebih jelas dibandingkan dari waveform. Informasi yang bisa didapatkan dari waveform hanya informasi secara general, misalnya bagian yang keras atau pelan, intro, outro, dll. Sedangkan dari spectrogram, informasi yang didapat tidak hanya terbatas pada interpretasi amplitudo, tetapi juga interpretasi dari frekuensi dan intensitas yang dihasilkan audio, seperti warna kuning pada spectrogram, dari detik awal sampai akhir audio, warna kuning secara konsisten ditampilkan pada spectrogram. Artinya, dari awal hingga akhir, audio menghasilkan suara frekuensi rendah dengan intensitas tinggi secara konsisten. Pada menit ke 1:40 s.d 2:00 warna kuning mulai memudar, artinya suara berfrekuensi rendah perlakan menghilang.

## A5. TODO: MFCC

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

- Sumbu X: waktu (frame)

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

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

```
In [5]: # Menghitung MFCC dengan 13 koefisien
mfccs = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)

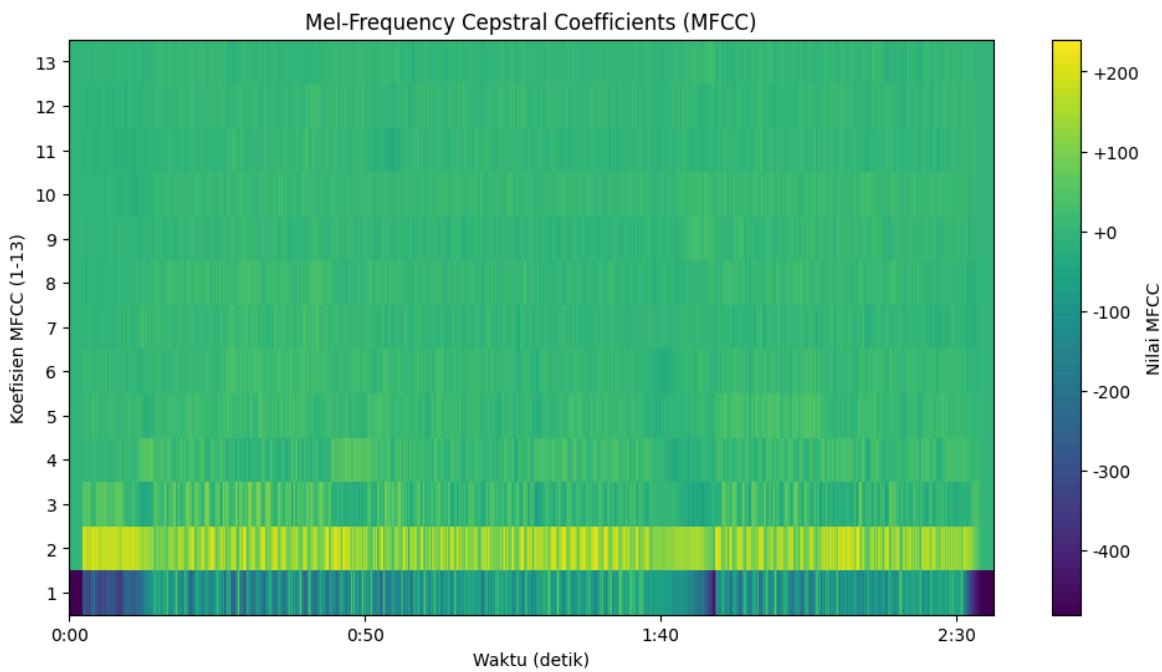
# Menentukan ukuran figure
plt.figure(figsize=(12, 6))

# Menampilkan MFCC sebagai heatmap
img = librosa.display.specshow(mfccs, x_axis='time', cmap='viridis')

# Menambahkan colorbar
plt.colorbar(img, format='%.2f', label='Nilai MFCC')

# Menambahkan judul dan label
plt.title('Mel-Frequency Cepstral Coefficients (MFCC)')
plt.xlabel('Waktu (detik)')
plt.ylabel('Koefisien MFCC (1-13)')
# Mengatur label sumbu y untuk menunjukkan 13 koefisien
plt.yticks(np.arange(0, 13), np.arange(1, 14)) # Label 1-13 for MFCC
```

```
Out[5]: ([<matplotlib.axis.YTick at 0x145f20410>,
<matplotlib.axis.YTick at 0x145f22350>,
<matplotlib.axis.YTick at 0x145f565d0>,
<matplotlib.axis.YTick at 0x145f56d50>,
<matplotlib.axis.YTick at 0x145f574d0>,
<matplotlib.axis.YTick at 0x145f57c50>,
<matplotlib.axis.YTick at 0x145f8c410>,
<matplotlib.axis.YTick at 0x145f8cb90>,
<matplotlib.axis.YTick at 0x145f8d310>,
<matplotlib.axis.YTick at 0x145f8da90>,
<matplotlib.axis.YTick at 0x145f8e210>,
<matplotlib.axis.YTick at 0x145f8e990>,
<matplotlib.axis.YTick at 0x145f8f110>],
[Text(0, 0, '1'),
 Text(0, 1, '2'),
 Text(0, 2, '3'),
 Text(0, 3, '4'),
 Text(0, 4, '5'),
 Text(0, 5, '6'),
 Text(0, 6, '7'),
 Text(0, 7, '8'),
 Text(0, 8, '9'),
 Text(0, 9, '10'),
 Text(0, 10, '11'),
 Text(0, 11, '12'),
 Text(0, 12, '13')])
```



### Analisis:

Pola MFCC relatif stabil, seperti yang ditunjukkan oleh koefisien-2 yang stabil berwarna kuning. Ada sedikit perubahan untuk koefisien-1 (selain intro dan outro), yaitu di menit 1:40 s.d 2:00, warna di koefisien-1 menjadi gelap karena ada jeda atau transisi pada audio. Potensi maknanya, MFCC yang stabil biasanya menandakan kalau suara musik yang dihasilkan konsisten atau mungkin berulang.

## A6. Analisis Ringkas (Wajib)

### Jawab pertanyaan berikut:

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

*Jawaban Anda:* Waveform hanya menunjukkan perubahan amplitudo (keras-pelan) terhadap waktu, sehingga kita hanya bisa mengidentifikasi bagian intro, jeda, dan outro, atau kapan audio keras/pelan secara global. Spectrogram memberikan informasi yang jauh lebih lengkap: selain amplitudo, juga memperlihatkan frekuensi dan intensitas suara terhadap waktu. Dengan spectrogram kita dapat melihat pola frekuensi rendah/tinggi, intensitasnya (melalui warna), dan perubahan spektrum suara sepanjang durasi audio.

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

*Jawaban Anda:* Dari visualisasi MFCC audio ini, terlihat bahwa pola koefisien MFCC relatif stabil sepanjang durasi audio, terutama pada koefisien ke-2 yang konsisten berwarna kuning. Hal ini menunjukkan bahwa karakter suara audio cenderung konstan atau berulang. Selain itu, terdapat perubahan warna signifikan pada koefisien-1 sekitar menit 1:40–2:00 yang kelihatan lebih gelap. Ini mengindikasikan adanya jeda atau transisi pada audio di bagian tersebut. Secara

umum, stabilitas pola MFCC dapat diartikan sebagai musik atau suara dengan instrumen dan pola frekuensi yang konsisten, sedangkan perubahan pola MFCC menunjukkan adanya variasi, transisi, atau jeda dalam audio.

## Bagian B — Gambar

### B1. Deskripsi Data

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

- Jenis gambar: lukisan
- Sumber: Wikipedia
- Format file: JPG
- Alasan pemilihan: salah satu lukisan yang bagus menurut saya, Stanczyk ini seorang pelawak kerajaan, tapi ada banyak hal lain di balik pekerjaannya sebagai pelawak

**Path file:** `data/gambar-lukisan-stanczyk.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 [6]: # Mengatur path gambar dengan os agar bisa dijalankan di mana saja
path_gambar = os.path.join(os.getcwd(), 'data', 'gambar-lukisan-stanczyk')

# Baca gambar dengan OpenCV
img_bgr = cv2.imread(path_gambar)

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

# Menentukan ukuran figure
plt.figure(figsize=(12, 8))

# Menampilkan gambar RGB
plt.imshow(img_rgb)

# Menambahkan judul
plt.title('Lukisan Stańczyk (Jan Matejko, 1862)', pad=20)

# Menghilangkan axis
plt.axis('off')
```

```
Out[6]: (np.float64(-0.5), np.float64(5765.5), np.float64(4288.5), np.float64(-0.5))
```

Lukisan Stańczyk (Jan Matejko, 1862)



#### Analisis:

Dari gambar, objek dominan yang terlihat adalah pria dewasa berpakaian badut yang duduk di kursi besar. Pencahayaan difokuskan ke objek dominan sehingga area belakangnya lebih gelap. Warna didominasi warna merah dan warna-warna gelap seperti hitam dan cokelat

---

### B3. TODO: Informasi Dasar

**Instruksi:** Tampilkan informasi metadata gambar:

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

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

```
In [7]: # Mendapatkan dimensi gambar (Height × Width)
height, width, channels = img_rgb.shape
# Menampilkan dimensi gambar
print(f"Dimensi gambar: {height} x {width} pixels")
```

```
# Menampilkan jumlah channel
print(f"Jumlah kanal: {channels}")

# Menampilkan tipe data
print(f"Tipe data: {img_rgb.dtype}")

# Menghitung ukuran file dalam memori (bytes)
memory_size = img_rgb.nbytes

# Mode warna berdasarkan jumlah channel
mode_warna = {1: 'Grayscale', 3: 'RGB', 4: 'RGBA'}.get(channels, 'Unknown')
print(f"Mode warna: {mode_warna}")

# Menampilkan ukuran file dalam memori dalam MB
print(f"Ukuran dalam memori: {memory_size/1024/1024:.2f} MB")
```

Dimensi gambar: 4289 × 5766 pixels

Jumlah kanal: 3

Tipe data: uint8

Mode warna: RGB

Ukuran dalam memori: 70.75 MB

### Analisis:

Informasi ini penting untuk tahap pre-processing atau analisis lanjutan karena dimensi, jumlah kanal, tipe data, mode warna, dan ukuran file menentukan cara kita membaca, mengubah, dan memproses gambar. Misalnya, dimensi dan jumlah kanal dibutuhkan untuk menyesuaikan ukuran input model atau mengonversi ke format tertentu (grayscale/RGB), tipe data memengaruhi rentang nilai piksel dan normalisasi, sedangkan ukuran file berpengaruh pada efisiensi memori dan waktu komputasi. Jadi dengan informasi ini, kita bisa merancang pipeline yang tepat, mencegah error, dan memastikan hasil analisis atau model yang lebih akurat.

## 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 [8]:

```
# Menentukan ukuran figure
plt.figure(figsize=(12, 6))

# Menghitung histogram untuk setiap channel
colors = ['red', 'green', 'blue']
channels = cv2.split(img_rgb)

# Plot histogram untuk setiap channel
for i, (color, channel) in enumerate(zip(colors, channels)):
    hist = cv2.calcHist([channel], [0], None, [256], [0, 256])
```

```

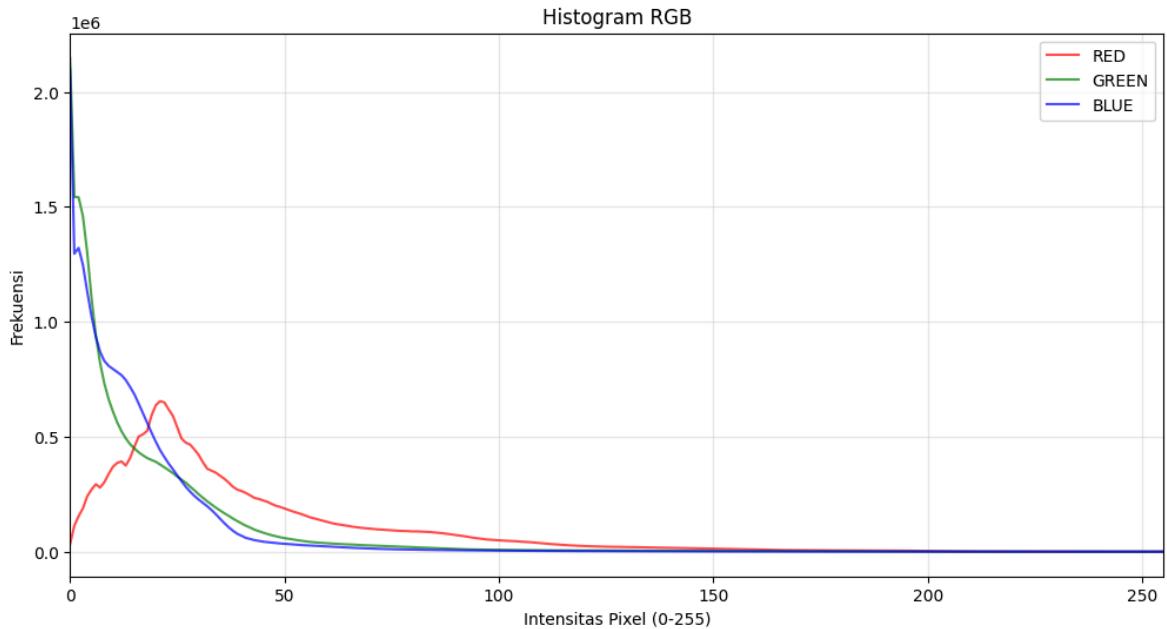
plt.plot(hist, color=color, alpha=0.7, label=color.upper())

# Menambahkan judul dan label
plt.title('Histogram RGB')
plt.xlabel('Intensitas Pixel (0-255)')
plt.ylabel('Frekuensi')
plt.legend()
plt.grid(True, alpha=0.3)

# Mengatur rentang sumbu x
plt.xlim([0, 255])

```

Out[8]: (0.0, 255.0)



### Analisis:

Channel yang dominan adalah channel biru karena warna biru memiliki frekuensi tertinggi yaitu 2,0. Kontras gambar tergolong rendah karena intensitas pixel yang memiliki frekuensi tinggi berada di rentang 0 - 50 yang artinya warna yang dihasilkan cenderung gelap, jadi tidak kontras. Sebaran intensitas pixel kebanyakan berada di rentang 0 - 50, dan intensitas tinggi hanya sedikit, yang artinya hanya sedikit warna terang pada gambar dan dominan warna gelap.

## 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 mayoritas pixel berada di intensitas rendah (0 – 50), sehingga secara visual gambar akan kelihatan gelap dengan kontras rendah. Dominasi channel biru di area intensitas rendah, artinya warna gambar yang cenderung kebiruan, gelap atau dingin. Karena hanya sedikit pixel di area intensitas tinggi, gambar terlihat kurang terang dan tidak memiliki area putih yang signifikan. Dengan kata lain, distribusi intensitas pada histogram sangat memengaruhi kesan

visual: puncak di kiri = gelap, puncak di kanan = terang, dan perbedaan antara channel RGB menentukan warna dominan yang kita lihat.

## Bagian C — Video

### C1. Deskripsi Data

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

- Jenis video: interview
- Sumber: YouTube
- Durasi target: 18 detik
- Alasan pemilihan: video interview bocil yang menurut saya lucu dan video ini sempat viral juga di tahun 2014

**Path file:** `data/video-apparently-kid.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 [9]: # Mengatur path video dengan os agar bisa dijalankan di mana saja
path_video = os.path.join(os.getcwd(), 'data', 'video-apparently-kid.mp4')

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

# Dapatkan metadata video
width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
fps = cap.get(cv2.CAP_PROP_FPS)
total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
duration = total_frames / fps

# Tentukan klasifikasi resolusi
def get_resolution_class(width, height):
    if width >= 3840 and height >= 2160:
        return "4K"
    elif width >= 2560 and height >= 1440:
        return "2K"
    elif width >= 1920 and height >= 1080:
        return "Full HD"
```

```

    elif width >= 1280 and height >= 720:
        return "HD"
    else:
        return "SD"

resolution_class = get_resolution_class(width, height)

# Tampilkan metadata
print(f"Resolusi: {width} x {height} pixels")
print(f"Frame rate: {fps} fps")
print(f"Total frame: {total_frames}")
print(f"Durasi: {duration:.2f} detik")
print(f"Klasifikasi resolusi: {resolution_class}")

# Tutup video capture
cap.release()

```

Resolusi: 640 × 360 pixels  
 Frame rate: 29.961016631768892 fps  
 Total frame: 528  
 Durasi: 17.62 detik  
 Klasifikasi resolusi: SD

#### Analisis:

Parameter-parameter tersebut penting karena menentukan bagaimana video akan diproses, ditampilkan, dan dianalisis. Resolusi (Width × Height) berpengaruh pada detail visual dan ukuran data yang harus diolah. Frame rate (fps) memengaruhi kelancaran gerakan serta kebutuhan komputasi saat ekstraksi frame atau analisis per frame. Jumlah total frame dan durasi (detik) membantu menghitung seberapa banyak data yang akan diproses. Klasifikasi resolusi (HD, Full HD, 4K) penting untuk menyesuaikan kebutuhan aplikasi—misalnya streaming, deteksi objek, atau pelatihan model—karena kualitas dan ukuran file memengaruhi performa sistem, penggunaan memori, dan waktu komputasi.

## 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 [10]: # Buka video capture
cap = cv2.VideoCapture(path_video)

# Hitung frame tengah
middle_frame = total_frames // 2

```

```

# Buat figure dengan 3 subplot untuk frame awal, tengah, dan akhir
fig, axes = plt.subplots(1, 3, figsize=(15, 5))

# Daftar frame yang akan diambil
frame_indices = [0, middle_frame, total_frames-1]
frame_titles = ['Frame Pertama', 'Frame Tengah', 'Frame Terakhir']

# Loop untuk mengambil dan menampilkan frame
for i, (frame_idx, title) in enumerate(zip(frame_indices, frame_titles)):
    # Set posisi frame yang diinginkan
    cap.set(cv2.CAP_PROP_POS_FRAMES, frame_idx)

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

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

        # Tampilkan frame
        axes[i].imshow(frame_rgb)
        axes[i].axis('off')

        # Hitung timestamp dalam detik
        timestamp = frame_idx / fps

        # Tambahkan judul dengan timestamp
        axes[i].set_title(f'{title}\n(t = {timestamp:.2f}s)')

    # Tambahkan judul utama
plt.suptitle('Frame Representatif Video', fontsize=14, y=1.05)

# Rapikan layout
plt.tight_layout()

# Tutup video capture
cap.release()

```

Frame Representatif Video



### Analisis:

Perbedaan visual:

- Frame pertama ( $t = 0.00$  s) -> Noah belum memegang mikrofon, orang-orang beraktivitas di belakangnya
- Frame tengah ( $t = 8.81$  s) -> Noah sedang memegang mikrofon yang diarahkan ke mulutnya yang sedang berbicara, matanya tertuju ke kamera, aktivitas di belakangnya berubah

- Frame terakhir ( $t = 17.59$  s) -> Mikrofon yang tadinya ada di dekat mulut Noah sekarang berada di area bahunya, Noah melihat ke arah kirinya, aktivitas di belakangnya berubah

Hal yang dapat dipelajari dari sampel frame ini adalah bagaimana alur dan isi video berubah dari awal hingga akhir. Kita bisa melihat pergeseran posisi dan ekspresi objek utama (Noah), adanya interaksi dengan mikrofon, dan perubahan latar belakang yang menunjukkan aktivitas orang-orang. Dengan mengambil frame representatif di awal, tengah, dan akhir, kita dapat memahami konten, konteks, dan perubahan adegan video tanpa harus memutar keseluruhan durasi, sehingga bermanfaat untuk analisis isi, segmentasi adegan, atau pembuatan ringkasan visual.

---

## 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:* Fps dan resolusi video ini sudah sangat sesuai untuk use case berita di tahun 2014, karena banyak portal berita online maupun televisi yang saat itu menggunakan resolusi SD atau HD rendah untuk keperluan tayang cepat dan hemat bandwidth. Video SD juga lebih cepat dimuat dan lebih kecil ukurannya, cocok dengan koneksi internet yang pada 2014 rata-rata lebih lambat dibanding sekarang. Dan frame rate ~30 fps sudah cukup mulus untuk liputan berita dan tidak terlihat patah-patah.

# Perbandingan & Kesimpulan

## Perbandingan Representasi Media

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

### Audio (1D - Temporal)

- Representasi: Audio direpresentasikan dengan gelombang amplitudo dari suara yang dihasilkan terhadap waktu
- Visualisasi utama: Waveform (secara global) dan Spectrogram (struktur lebih detail)
- Informasi yang diperoleh: Dengan waveform kita bisa melihat pola audio secara global, seperti keras/pelannya suara, intro, outro, dan jeda. Sedangkan dengan spectrogram kita bisa melihat struktur audio dengan lebih jelas, seperti intensitas dari frekuensi suara, juga karakter nada, apakah ritme berulang atau berubah-ubah.

## Gambar (2D - Spasial)

- Representasi: Gambar direpresentasikan dengan matriks pixel (baris \* kolom) yang menyimpan nilai warna (channel RGB)
- Visualisasi utama: Tampilan citra (image) dan histogram untuk mengetahui komposisi/persebaran warna
- Informasi yang diperoleh: Dari suatu citra/gambar, kita dapat melihat komposisi warna pada citra tersebut dengan menggunakan histogram warna. Dari histogram warna, kita juga bisa mengetahui apakah gambar tersebut memiliki kontras yang tinggi atau tidak, dan kita bisa mengetahui warna yang dominan.

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

- Representasi: Video direpresentasikan dengan kumpulan frame gambar yang dinamis terhadap waktu
  - Visualisasi utama: Pemutaran video secara langsung atau dengan potongan-potongan frame per waktu untuk melihat perubahan/pergerakan frame
  - Informasi yang diperoleh: Video berisi gambar/frame yang berubah-ubah terhadap waktu, termasuk pergerakan objek dan perubahan latar
- 

## Refleksi Pembelajaran

### 3 Poin yang Saya Pelajari:

1. Saya berhasil mempelajari cara memuat dan membaca file multimedia, seperti audio, image/citra, dan video
2. Dari tiap multimedia, ada informasi yang bisa diambil atau diekstrak yang nantinya bisa membantu manusia dan juga mesin untuk mengolah data multimedia tersebut secara lanjut
3. Informasi yang dapat diesktrak dari data multimedia tersebut dapat divisualisasikan dengan plot-plot yang sesuai, dari plot tersebut kita bisa memahami makna dari data multimedia tersebut

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

1. Saya masih bingung kenapa video tidak ditampilkan, padahal audio dan citra bisa didisplay. Kenapa video hanya bisa ditampilkan potongan framenya saja
- 

## Sumber Data & Referensi

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

- **Audio:** [https://youtu.be/c1TCURWuOg8?si=QFiEZpkyqc\\_hozMI](https://youtu.be/c1TCURWuOg8?si=QFiEZpkyqc_hozMI)
- **Gambar:** <https://en.wikipedia.org/wiki/Sta%C5%84czyk>
- **Video:** <https://youtu.be/rz5TGN7eUcM>

- **Referensi teknis:** <https://chatgpt.com/share/68c52580-1d24-8004-a32e-5e425e946f50> & GitHub Copilot

# 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; Penggunaan</b>	<b>5%</b>	Pemahaman saat presentasi acak

## Detail Kriteria Penilaian

### Kelengkapan (35%)

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

### Kualitas Visualisasi (20%)

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

### Analisis & Interpretasi (30%)

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

### Kerapihan & Struktur (10%)

- Markdown terstruktur dengan heading yang konsisten

- Kode bersih, terkompartemen, dan mudah dibaca
- Dokumentasi yang memadai
- Flow logical dari satu bagian ke bagian lain

### Orisinalitas & Penguasaan (5%)

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

## Proporsi Penilaian Total

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

## Aturan Kejujuran Akademik

### Penggunaan Referensi & AI yang Diperbolehkan

Anda **BOLEH** menggunakan:

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

### Syarat & Batasan WAJIB

Namun Anda **HARUS**:

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

### Pelanggaran yang Berakibat Nilai 0

- **Plagiarisme atau penyalinan buta** dari sumber manapun
- **Copy-paste kode tanpa pemahaman** dan tidak dapat menjelaskan
- **Menggunakan AI untuk mengerjakan seluruh tugas** tanpa pembelajaran personal

- **Tidak dapat menjawab pertanyaan dasar** tentang kode yang dikumpulkan
- **Menyalin pekerjaan teman** atau bekerjasama dalam pengerjaan individual

## Persiapan Presentasi Acak

**Kemungkinan pertanyaan yang akan ditanyakan:**

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

**Tips sukses:**

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

## Panduan Pengumpulan

### Berkas yang Harus Dikumpulkan

**Wajib:**

1. **Notebook Jupyter** (.ipynb) dengan nama:  
**NIM\_Nama\_TugasMultimedia.ipynb**
    - Contoh: **123456789\_JohnDoe\_TugasMultimedia.ipynb**
  2. **PDF hasil render dari notebook**
- 



### Informasi Pengumpulan

---



### Checklist Sebelum Submit

- Semua cell sudah dijalankan dan menampilkan output
  - Nama file sesuai format: **NIM\_Worksheet2.ipynb** dan **NIM\_Worksheet2.pdf**
  - Semua TODO sudah diisi dengan lengkap
  - Analisis dan interpretasi sudah ditulis untuk setiap bagian
  - Sumber data dan referensi sudah dicantumkan
-

## Export ke PDF:

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