

Tugas Analisis Multimedia: Audio, Gambar, Video

Mata Kuliah: Sistem & Teknologi Multimedia

Nama: Arkan Hariz Chandrawinata Liem

NIM: 122140038

Deskripsi Tugas

Tugas ini bertujuan untuk memahami representasi dasar data multimedia (audio, gambar, dan video) melalui praktik langsung memuat data, visualisasi, dan ekstraksi informasi fundamental. Anda akan bekerja dengan tiga jenis media berbeda untuk menganalisis karakteristik temporal (audio), spasial (gambar), dan spatio-temporal (video).

Fokus tugas adalah pada pemahaman konsep dasar representasi multimedia dan kemampuan interpretasi hasil visualisasi, **bukan** pada manipulasi atau transformasi lanjutan data multimedia.

CATATAN PENTING: PRESENTASI ACAK & KEJUJURAN AKADEMIK

Sebagian mahasiswa akan dipilih secara ACAK untuk presentasi singkat (5-10 menit) menjelaskan kode dan interpretasi hasil mereka. Jika Anda:

- Tidak mampu menjelaskan kode yang Anda kumpulkan
- Hanya menyalin-tempel tanpa pemahaman
- Bergantung sepenuhnya pada AI tanpa memahami konsep

Maka nilai tugas Anda akan diberikan 0 (nol).

Gunakan referensi dan AI sebagai alat bantu pembelajaran, tetapi pastikan Anda memahami setiap baris kode dan dapat menjelaskan logika di baliknya.

```
In [1]: # Import Library (Satu-satunya sel kode dalam template ini)
import numpy as np
import matplotlib.pyplot as plt
import librosa
import soundfile as sf
from PIL import Image
import cv2
from IPython.display import Audio, HTML, display
import os

# 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.matplotlib.__version__}")
print(f"Librosa: {librosa.__version__}")
print(f"OpenCV: {cv2.__version__}")

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

Library versions:
NumPy: 2.2.6
Matplotlib: 3.10.6
Librosa: 0.11.0
OpenCV: 4.12.0

Petunjuk Umum Pengerjaan



Cara Menggunakan Template

- Gunakan notebook ini sebagai kerangka kerja utama
- Tulis penjelasan (markdown) **SEBELUM** menaruh kode agar maksud dan tujuan jelas
- Tambahkan sel kode di tempat yang sudah disediakan (tandai dengan TODO)
- Semua plot/gambar harus diberi judul, label sumbu, dan keterangan singkat



Standar Visualisasi

- Setiap plot harus memiliki judul yang deskriptif
- Label sumbu X dan Y harus jelas
- Gunakan colorbar untuk plot yang memerlukan skala warna
- Berikan interpretasi singkat setelah setiap visualisasi



Struktur Data yang Direkomendasikan

- Buat folder `data/` di direktori yang sama dengan notebook
- Gunakan nama file yang deskriptif (contoh: `audio_musik_piano.wav`, `gambar_pemandangan_gunung.jpg`)
- Dokumentasikan sumber data jika menggunakan dataset publik



Larangan

- **Jangan** menaruh seluruh pekerjaan dalam satu sel kode yang sangat panjang
- **Jangan** menempel hasil output tanpa interpretasi atau analisis
- **Jangan** bergantung sepenuhnya pada AI - pahami dan kuasai kode Anda








Persiapan Presentasi Acak





- Pastikan Anda memahami setiap baris kode yang ditulis
- Latih menjelaskan logika dan alur pemikiran Anda
- Siapkan penjelasan untuk setiap visualisasi dan interpretasinya

Checklist Kelengkapan (Centang saat selesai)





Bagian Audio

-  Muat audio dan tampilkan metadata (durasi, sample rate, jumlah kanal)
-  Tampilkan waveform dengan label sumbu yang jelas
-  Tampilkan spectrogram dalam skala log-dB dengan colorbar
-  Tampilkan MFCC (minimal 13 koefisien) sebagai heatmap
-  Berikan interpretasi dan analisis untuk setiap visualisasi audio





Bagian Gambar

-  Tampilkan gambar dengan benar dalam format RGB
-  Tampilkan informasi dasar (dimensi, jumlah kanal, dtype)
-  Tampilkan histogram warna untuk channel R, G, B
-  Berikan analisis hubungan histogram dengan kesan visual gambar

Bagian Video

-  Tampilkan metadata video (resolusi, fps, frame count, durasi)
-  Tampilkan 3 frame representatif (awal, tengah, akhir)
-  Konversi BGR ke RGB dengan benar untuk visualisasi
-  Analisis kesesuaian parameter video dengan use case

Analisis & Dokumentasi

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

Pendahuluan

Apa itu Data Multimedia?

Data multimedia adalah informasi yang dikodekan dalam berbagai format untuk merepresentasikan dunia nyata:

- **Audio (1D):** Sinyal satu dimensi yang berubah terhadap waktu
 - Contoh: musik, suara, speech
 - Representasi: amplitudo vs waktu
- **Gambar (2D):** Matriks nilai intensitas dalam ruang dua dimensi
 - Contoh: foto, ilustrasi, grafik

- Representasi: intensitas pixel pada koordinat (x,y)
- **Video (2D + Waktu):** Rangkaian frame (gambar) yang ditampilkan berurutan
 - Contoh: film, rekaman, animasi
 - Representasi: frame berubah terhadap waktu dengan frame rate tertentu

Tujuan Tugas

Memahami representasi dasar dan teknik visualisasi fundamental untuk setiap jenis media multimedia, termasuk:

- Cara memuat dan membaca file multimedia
- Ekstraksi informasi metadata yang penting
- Visualisasi yang informatif dan mudah dipahami
- Interpretasi hasil analisis secara kontekstual

Cara Kerja

1. Isi setiap bagian sesuai instruksi yang diberikan
2. Tambahkan sel kode di tempat yang ditandai dengan "TODO"
3. Berikan interpretasi dan analisis setelah setiap visualisasi
4. Pastikan semua plot memiliki judul, label, dan keterangan yang jelas

Bagian A — Audio

A1. Deskripsi Data

TODO: Jelaskan audio yang akan Anda analisis:

- Jenis audio: **pidato proklamasi**
- Sumber: **youtube**
- Format file: **WAV**
- Alasan pemilihan: **karena pidato presiden pertama di momen yang berharga yaitu kemerdekaan indonesia dan audio yang unik seperti suara radio lama.**

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

A2. TODO: Muat & Metadata

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

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

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

```
In [15]: # Path audio
PATH_AUDIO = os.path.join(os.getcwd(), 'data', 'pidato_proklamasi.wav')

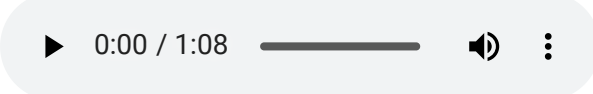
y = None
sr = None
source_info = ""

# Memuat file audio lokal
try:
    if os.path.exists(PATH_AUDIO):
        y, sr = librosa.load(PATH_AUDIO, sr=None)
        source_info = f"File audio berhasil dimuat dari {PATH_AUDIO}"
    else:
        source_info = f"File audio tidak ditemukan di {PATH_AUDIO}."
except:
    print(f"Tidak dapat memuat {PATH_AUDIO}")

print(source_info)

# Display audio
Audio(data=y, rate=sr)
```

File audio berhasil dimuat dari d:\ALIEM1\Semester_7\MULTIMEDIA\PERTEMUAN_2\Tugas\data\pidato_proklamasi.wav

Out[15]: 

```
In [16]: # Metadata
print("📦 Metadata Audio")
print(f"🎵 Sample rate (Hz): {sr:,} Hz")
print(f"🕒 Durasi (detik): {len(y)/sr:.2f} detik")
print(f"🔊 Jumlah kanal (mono/stereo):")
print(f"{'Mono' if y.ndim == 1 else f'Multi-kanal ({y.ndim} kanal)'}")
print(f"📊 Jumlah total sample: {len(y):,}")
```

📦 Metadata Audio
 🎵 Sample rate (Hz): 48,000 Hz
 🕒 Durasi (detik): 68.58 detik
 🔊 Jumlah kanal (mono/stereo):
 Mono
 📊 Jumlah total sample: 3,291,847

A3. TODO: Waveform

Instruksi: Plot waveform audio dengan:

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

Analisis yang diperlukan: Berdasarkan waveform, terdapat beberapa pola berulang. Pola berulang ini ditunjukkan dengan adanya lonjakan amplitudo yang besar/tinggi kemudian setelah amplitudo menurun/kecil. Adanya terjadi lonjakan amplitudo yang

besar/tinggi karena suara yang keras ketika Ir. Soekarno sedang membacakan proklamasi nya. Sedangkan amplitudo yang rendah/kecil ketika Ir. Soekarno berhenti sejenak setelah membaca potong-potong proklamasi layak nya sedang mengambil nafas. Kemudian menuju akhir suara, amplitudo mulai berkurang dan bernilai 0 yang dimana suara sudah mulai menghilang perlahan sehingga tidak ada lagi suara.

```
In [4]: t = np.linspace(0, len(y)/sr, len(y))

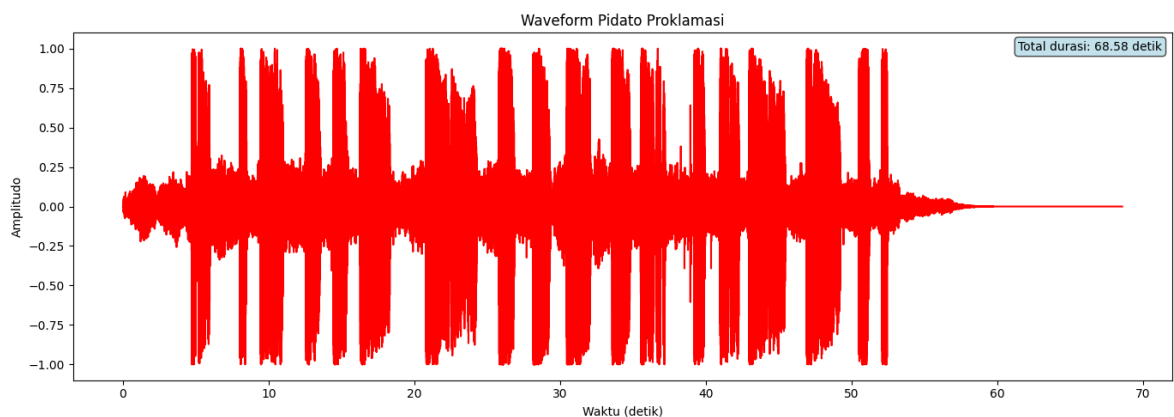
# Setup visualisasi waveform
fig, ax = plt.subplots(figsize=(14, 5))

# Plot waveform
ax.plot(t, y, color='red')
ax.set_title('Waveform Pidato Proklamasi')
ax.set_xlabel('Waktu (detik)')
ax.set_ylabel('Amplitudo')

# Informasi durasi
duration = f"Total durasi: {len(y)/sr:.2f} detik"
ax.text(0.86, 0.95, duration, transform=ax.transAxes,
        bbox=dict(boxstyle="round,pad=0.3", facecolor="lightblue", alpha=0.7))

plt.tight_layout()
plt.show()

Audio(data=y, rate=sr)
```



Out[4]:

A4. TODO: Spectrogram log-dB

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

- Gunakan parameter standar ($n_{\text{fft}}=1024$, $\text{hop_length}=256$)
- Tampilkan dengan colorbar
- Label sumbu: waktu (detik) dan frekuensi (Hz)

Analisis yang diperlukan: Pada spectrogram, dapat mengetahui dan melihat frekuensi dari suatu audio dibandingkan bentuk waveform yang menampilkan seberapa keras dan

pelan suatu audio pada waktu tertentu. Pada spectrogram juga terdapat indikator warna untuk menentukan intensitas dari suatu energi suara dalam dB.

```
In [17]: n_fft = 1024
hop_length = 256
window = 'hann'

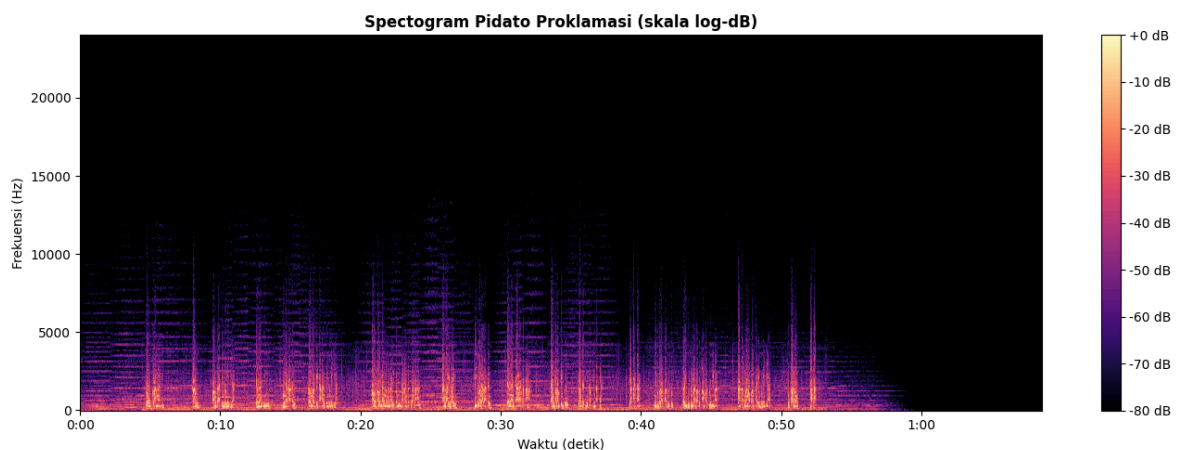
D = librosa.stft(y, n_fft=n_fft, hop_length=hop_length, window=window)
magnitude = np.abs(D)
magnitude_db = librosa.amplitude_to_db(magnitude, ref=np.max)

# Setup plot
fig, ax = plt.subplots(figsize=(14, 5))

# Plot spectrogram
specto = librosa.display.specshow(magnitude_db, sr=sr,
                                   hop_length=256, x_axis='time',
                                   y_axis='hz', ax=ax)
ax.set_title('Spectrogram Pidato Proklamasi (skala log-dB)', fontweight='bold')
ax.set_xlabel('Waktu (detik)')
ax.set_ylabel('Frekuensi (Hz)')
plt.colorbar(specto, ax=ax, format="%+2.0f dB")

plt.tight_layout()
plt.show()

print(f"Ukuran FFT: {n_fft}")
print(f"Hop length: {hop_length}")
print(f"Window: {window}")
print(f"Magnitude shape: {magnitude.shape}")
```



Ukuran FFT: 1024
 Hop length: 256
 Window: hann
 Magnitude shape: (513, 12859)

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: Berdasarkan visualisasi, pola dari MFCC relatif stabil dengan variasi yang rendah sepanjang audio, walaupun masih terdapat perubahan pada bagian tertentu. Pola yang stabil ini dapat dimaknai dengan karakter suara yang cenderung konsisten, sedangkan perubahan yang terjadi menggambarkan perbedaan saat berpidato seperti artikulasi, intonasi, dan/atau penekanan kata.

```
In [6]: n_mels = 64
f_max = sr // 2

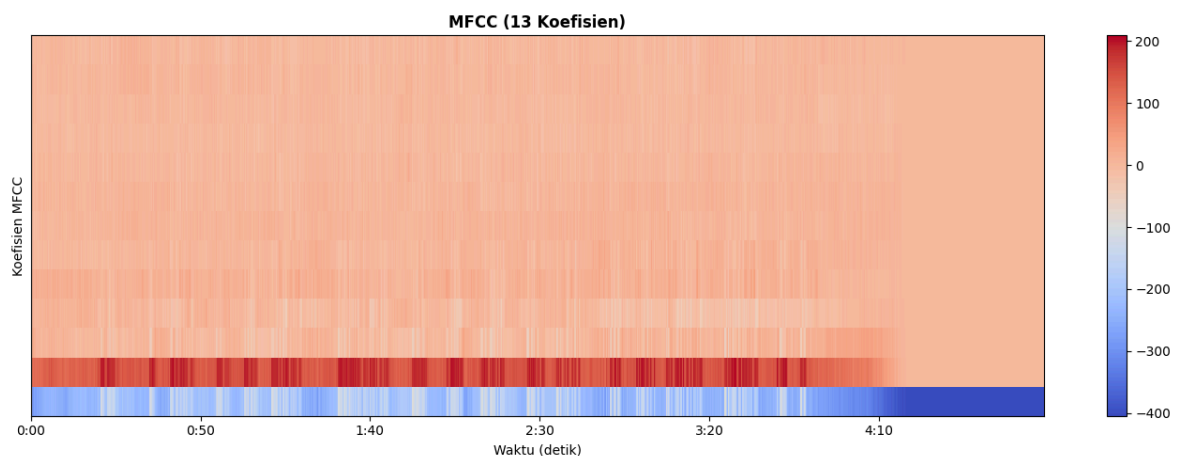
# Hitung MFCC
n_mfcc = 13
mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=n_mfcc, n_fft=n_fft,
                             hop_length=hop_length, n_mels=n_mels)

# Setup plot
fig, ax = plt.subplots(figsize=(14, 5))

# Plot MFCC
mfcc_heatmap = librosa.display.specshow(mfcc, x_axis='time', ax=ax)
ax.set_title('MFCC (13 Koefisien)', fontweight='bold')
ax.set_xlabel('Waktu (detik)')
ax.set_ylabel('Koefisien MFCC')
plt.colorbar(mfcc_heatmap, ax=ax)

plt.tight_layout()
plt.show()

print(f"Jumlah Mel Bins: {n_mels}")
print(f"Jumlah Koefisien MFCC: {n_mfcc}")
print(f"MFCC shape: {mfcc.shape}")
```



Jumlah Mel Bins: 64
 Jumlah Koefisien MFCC: 13
 MFCC shape: (13, 12859)

A6. Analisis Ringkas (Wajib)

Jawab pertanyaan berikut:

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

Jawaban Anda: Waveform memberikan informasi tentang amplitudo sinyal audio yang berubah terhadap waktu, sehingga informasi yang diberikan waveform lebih fokus terhadap seberapa keras atau pelan-nya suara. Dibandingkan dengan spectogram yang memberikan informasi tentang frekuensi terhadap waktu dengan intensitasnya, sehingga dapat melihat seperti struktur nada, detail artikulasi yang tidak dapat ditangkap oleh waveform.

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

Jawaban Anda: Berdasarkan visualisasi MFCC pada audio tersebut, saya dapat mempelajari karakteristik spektral suara yang ditangkap dalam bentuk koefisien yang mewakili pola timbre/warna suara. Pola MFCC yang stabil/konsisten menunjukkan ciri suara yang konsisten sedangkan terdapat variasi yang muncul menunjukkan terjadi nya perubahan artikulasi, tekanan dalam berbicara.

Bagian B — Gambar

B1. Deskripsi Data

TODO: Jelaskan gambar yang akan Anda analisis:

- Jenis gambar: **pemandangan**
- Sumber: **internet (freepik)**
- Format file: **JPG**
- Alasan pemilihan: **suka dengan pemandangan gunung terutama gunung fuji yang berada di Jepang**

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

B2. TODO: Baca & Tampilkan (RGB)

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

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

Analisis yang diperlukan: Gambar yang dimuat adalah gambar pemandangan dari gunung fuji yang berada di Jepang. Pada gambar tersebut, objek yang dominan adalah dedaunan dan juga gunung fuji karena foto tersebut diambil dari kejauhan dan mungkin difoto sebagai berikut untuk estetika. Kondisi pencahayaan bagus dan cerah karena foto diambil di siang hari dan cuaca mendukung (tidak gelap/mendung). Menurut saya dari segi komposisi warna bagus karena foto diambil pada musim kemarau dilihat dari warna dedaunan yang berwarna coklat kemerahan.

```
In [18]: # Path image
PATH_IMAGE = os.path.join(os.getcwd(), 'data', 'pemandangan_gunung.jpg')
```

```

img_pil = None
source_info_image = ""

try:
    if os.path.exists(PATH_IMAGE):
        img_pil = Image.open(PATH_IMAGE)
        if img_pil.mode != 'RGB':
            img_pil = img_pil.convert('RGB')
        else:
            print(f"{img_pil} sudah RGB")
            source_info_image = f"File gambar berhasil dimuat dari {PATH_IMAGE}"
        else:
            source_info_image = f"File gambar tidak ditemukan di {PATH_IMAGE}."
except:
    print(f"File {PATH_IMAGE} tidak dapat memuat")

print(source_info_image)

plt.figure(figsize=(12, 8))
plt.title('Pemandangan Gunung Fuji Jepang', fontweight='bold')
plt.imshow(img_pil)
plt.axis('off')

```

<PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=4928x3280 at 0x2E1030A4310
> sudah RGB

File gambar berhasil dimuat dari d:\ALIEM1\Semester_7\MULTIMEDIA\PERTEMUAN_2\Tugas\data\pemandangan_gunung.jpg

Out[18]: (np.float64(-0.5), np.float64(4927.5), np.float64(3279.5), np.float64(-0.5))

Pemandangan Gunung Fuji Jepang



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: karena tahap yang penting sebelum gambar di input ke dalam machine learning/deep learning. Selain itu, informasi gambar dapat menentukan strategi pemrosesan yang tepat serta mengoptimalkan memory.

```
In [20]: print("📁 Metadata Image")
img_array = np.array(img_pil)
print("Dimensi (Height x Width):")
print(f"{img_pil.size[0]} x {img_pil.size[1]} pixels")
print(f"Jumlah Kanal: {img_array.shape[2]} kanal")
print(f"Tipe data (dtype): {img_array.dtype}")
print(f"Mode warna: {img_pil.mode}")
print("Ukuran file dalam memori:")
print(f"{img_array.nbytes:,} bytes ({img_array.nbytes/1024:.1f} KB)")
```

```
📁 Metadata Image
Dimensi (Height x Width):
4928 x 3280 pixels
Jumlah Kanal: 3 kanal
Tipe data (dtype): uint8
Mode warna: RGB
Ukuran file dalam memori:
48,491,520 bytes (47355.0 KB)
```

B4. TODO: Histogram Warna

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

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

Analisis yang diperlukan: Warna yang dominan adalah warna biru karena histogram menunjukkan frekuensi jauh lebih tinggi pada intensitas rendah dibanding channel lain. Kontras dari gambar juga memiliki kontras yang cukup baik karena persebaran nilai dari gelap hingga terang. Untuk sebaran intensitasnya channel biru memiliki puncak frekuensi yang tinggi pada intensitas rendah, channel merah memiliki distribusi yang lebih merata di tengah, memperlihatkan bahwa warna merah sedang hingga terang. Untuk channel hijau cenderung rendah sehingga warna hijau tidak dominan.

```
In [9]: red_channel = img_array[:, :, 0].flatten()
green_channel = img_array[:, :, 1].flatten()
blue_channel = img_array[:, :, 2].flatten()

fig, axes = plt.subplots(1, 3, figsize=(18, 5))

# Red Channel
axes[0].hist(red_channel, bins=256, range=(0, 255), color='red', alpha=0.7)
axes[0].set_title('Red Channel Distribution', fontweight='bold')
```

```

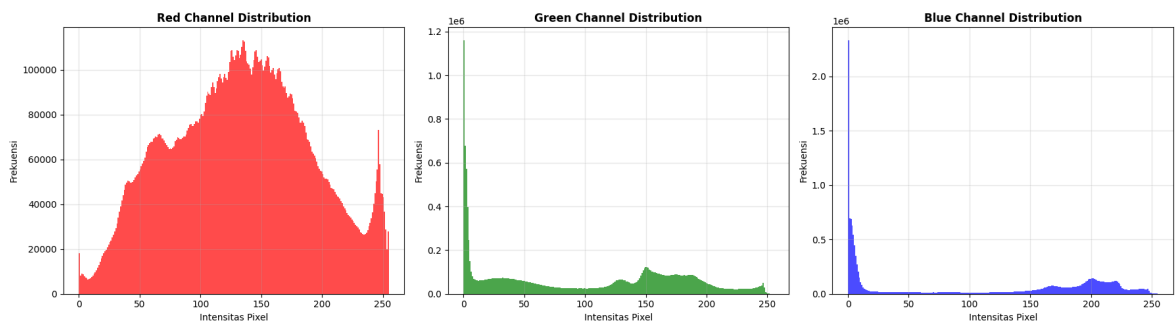
axes[0].set_xlabel('Intensitas Pixel')
axes[0].set_ylabel('Frekuensi')
axes[0].grid(True, alpha=0.3)

# Green Channel
axes[1].hist(green_channel, bins=256, range=(0, 255), color='green', alpha=0.7)
axes[1].set_title('Green Channel Distribution', fontweight='bold')
axes[1].set_xlabel('Intensitas Pixel')
axes[1].set_ylabel('Frekuensi')
axes[1].grid(True, alpha=0.3)

# Blue Channel
axes[2].hist(blue_channel, bins=256, range=(0, 255), color='blue', alpha=0.7)
axes[2].set_title('Blue Channel Distribution', fontweight='bold')
axes[2].set_xlabel('Intensitas Pixel')
axes[2].set_ylabel('Frekuensi')
axes[2].grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

```



B5. Analisis Ringkas (Wajib)

Jawab pertanyaan berikut:

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

Jawaban Anda: Pola histogram dengan kesan visual dari gambar sangat berhubungan. Dominasi warna biru berdasarkan histogramnya, memberi petunjuk bahwa gambar tersebut memiliki banyak area berwarna biru. Dikarenakan hal tersebut, gambar/foto terkesan dingin dan gelap di beberapa bagian. Channel merah dan hijau memiliki distribusi yang menyebar di rentang menengah hingga tinggi yang membuat warna bervariasi, terutama pada bagian foto yang cerah. Overall, sebaran histogram yang ada menjelaskan kenapa gambar terlihat lebih dominan pada warna biru dibandingkan kedua warna lainnya yaitu merah dan hijau.

Bagian C — Video

C1. Deskripsi Data

TODO: Jelaskan video yang akan Anda analisis:

- Jenis video: **animasi**
- Sumber: **youtube**
- Durasi target: **7 detik**
- Alasan pemilihan: **karena saya suka menonton anime dan video yang saya ambil adalah ciri khas dari anime tersebut**

Path file: data/____.mp4 (isi nama file Anda nanti di kode)

C2. TODO: Baca & Metadata

Instruksi: Baca video dengan OpenCV dan tampilkan metadata:

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

Analisis yang diperlukan: Parameter-parameter tersebut sangat penting untuk sinkronisasi antara video dan audio, penting untuk menentukan sampling rate, penting untuk preprocessing seperti resize dan frame extraction, dan kepentingan" lainnya untuk memproses suatu vidio.

```
In [10]: # Path video
PATH_VIDEO = os.path.join(os.getcwd(), 'data', 'animasi_bertarung_3.mp4')

print("Membaca video...")
print(f"File video: {PATH_VIDEO}")

cap = cv2.VideoCapture(PATH_VIDEO)

if cap.isOpened():
    print("Video berhasil dibuka.")
    source_info_video = f"File video berhasil dimuat dari {PATH_VIDEO}"
    video_loaded = True
else:
    print(f"Gagal memuat video {PATH_VIDEO}")

print(f"Status: {source_info_video}")
print(f"Video loaded: {video_loaded}")
```

```
Membaca video...
File video: d:\ALIEM1\Semester_7\MULTIMEDIA\PERTEMUAN_2\Tugas\data\animasi_bertarung_3.mp4
Video berhasil dibuka.
Status: File video berhasil dimuat dari d:\ALIEM1\Semester_7\MULTIMEDIA\PERTEMUAN_2\Tugas\data\animasi_bertarung_3.mp4
Video loaded: True
```

```
In [11]: if video_loaded == True:
    print("Metadata Video")

    # Resolusi
    width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
```

```

height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
print(f"Resolusi (Width x Height): {width} x {height} pixels")

# Frame rate
fps = cap.get(cv2.CAP_PROP_FPS)
print(f"Frame rate (FPS): {fps:.2f} fps")

# Total frame
frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
print(f"Jumlah total frame: {frame_count:,} frames")

# Durasi
duration = frame_count / fps if fps > 0 else None
print(f"Durasi (detik): {duration:.2f} detik")

# Klasifikasi resolusi
if height < 480:
    res_class = "SD (Sub-480p)"
elif height < 720:
    res_class = "SD (480p)"
elif height < 1080:
    res_class = "HD (720p)"
elif height < 1440:
    res_class = "Full HD (1080p)"
elif height < 2160:
    res_class = "2K (1440p)"
elif height < 4320:
    res_class = "4K (2160p)"

print(f"Klasifikasi resolusi: {res_class}")
else:
    print("Tidak dapat mengekstrak metadata karena video tidak dimuat.")

```

Metadata Video

Resolusi (Width x Height): 640 x 354 pixels

Frame rate (FPS): 30.00 fps

Jumlah total frame: 225 frames

Durasi (detik): 7.50 detik

Klasifikasi resolusi: SD (Sub-480p)

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

Instruksi: Ambil dan tampilkan 3 frame representatif:

- Frame pertama (index 0)
- Frame tengah (index $\sim \text{total_frame}/2$)
- Frame terakhir (index $\text{total_frame}-1$)
- **Konversi BGR→RGB** sebelum ditampilkan
- Subplot dengan judul frame dan timestamp

Analisis yang diperlukan: Berdasarkan antar frame yang sudah ada, masing-masing frame memiliki resolusi yang berbeda. Frame awal dan akhir terlihat resolusi yang jernih, sedangkan frame tengah terlihat sedikit buram atau resolusi nya tidak setinggi frame awal dan akhir.

Deskripsikan perbedaan visual antar frame dan apa yang dapat dipelajari dari sampel frame ini.


```

In [21]: # Menentukan frame yang akan diambil
frame_pertama = 0
frame_kedua = frame_count // 2
frame_ketiga = max(0, frame_count - 1)

key_frames = [
    (frame_pertama, "Frame Awal"),
    (frame_kedua, "Frame Tengah"),
    (frame_ketiga, "Frame Akhir")
]

fig, axes = plt.subplots(1, 3, figsize=(18, 6))

extract_frames = []

for i, (frame_idx, title) in enumerate(key_frames):
    # Set posisi frame
    cap.set(cv2.CAP_PROP_POS_FRAMES, frame_idx)

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

    if ret:
        # Konversi dari BGR ke RGB untuk display yang benar
        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        extract_frames.append(frame_rgb)

        # Tampilkan frame
        axes[i].imshow(frame_rgb)
        axes[i].set_title(f'{title}\n(Frame #{frame_idx})', fontsize=12,
                          fontweight='bold')
        axes[i].axis('off')

        # Tambahkan informasi timestamp
        timestamp = frame_idx / fps
        axes[i].text(0.02, 0.98, f'Waktu: {timestamp:.2f}s',
                     transform=axes[i].transAxes, fontsize=10,
                     bbox=dict(boxstyle="round,pad=0.3", facecolor="yellow",
                               alpha=0.8), verticalalignment='top')

    else:
        # Jika gagal membaca frame
        axes[i].text(0.5, 0.5, f'❌ Gagal\nmembaca\n{title}',
                     transform=axes[i].transAxes, fontsize=12,
                     horizontalalignment='center', verticalalignment='center')
        axes[i].set_title(f'{title} (Error)', fontsize=12)
        axes[i].axis('off')

plt.tight_layout()
plt.show()

print(f"✅ Frame berhasil diekstrak!")
print(f"📺 Resolusi frame: {width} x {height}")
print(f"🕒 Frame yang diekstrak:")
for frame_idx, title in key_frames:
    timestamp = frame_idx / fps
    print(f"    {title}: Frame #{frame_idx} (waktu {timestamp:.2f}s)")

```



- ✓ Frame berhasil diekstrak!
- 📊 Resolusi frame: 640 × 354
- 🔄 Frame yang diekstrak:
 - Frame Awal: Frame #0 (waktu 0.00s)
 - Frame Tengah: Frame #112 (waktu 3.73s)
 - Frame Akhir: Frame #224 (waktu 7.47s)

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: Berdasarkan metadata, fps dari video ini adalah 30 fps dan resolusi video nya standard definition (SD). Menurut saya untuk fps nya tidak masalah tetapi untuk resolusi tidak sesuai. Resolusi tidak sesuai karena masih tidak terlalu jelas/masih buram karena resolusi yang standar definition atau 480p. Mungkin minimal nya menurut saya sekitar 720p supaya tidak terlalu buram.

Perbandingan & Kesimpulan

Perbandingan Representasi Media

TODO: Bandingkan secara ringkas representasi dan visualisasi ketiga media:

Audio (1D - Temporal)

- Representasi: sinyal satu dimensi yang mengandung informasi temporal.
- Visualisasi utama: Waveform untuk visualisasi struktur temporal, spectrogram untuk mengubah sinyal audio dari domain waktu ke domain frekuensi-waktu, MFCC untuk ringkasan spektral.
- Informasi yang diperoleh: Pola amplitudo, frekuensi, karakteristik suara, dan durasi.

Gambar (2D - Spasial)

- Representasi: sinyal dua dimensi (2D) yang terdiri dari kumpulan pixel yang disusun dalam bentuk grid.
- Visualisasi utama: Tampilan gambar dan histogram.
- Informasi yang diperoleh: grafik penyebaran intensitas dari masing-masing channel.

Video (2D + Waktu - Spatio-temporal)

- Representasi: rangkaian gambar (frame) yang ditampilkan berurutan pada kecepatan tertentu yang disebut frame rate.
 - Visualisasi utama: video dan/atau frame video
 - Informasi yang diperoleh: frame rate, total frame, durasi, resolusi.
-

Refleksi Pembelajaran

3 Poin yang Saya Pelajari:

1. Saya mempelajari untuk mendapatkan informasi dari sebuah audio berdasarkan waveform atau spectrogram.
2. Saya mempelajari untuk menampilkan histogram dari sebuah image untuk melihat distribusi sebuah warna/channel yang bisa digunakan untuk analisis seperti mengetahui warna yang dominan.
3. Saya mempelajari supaya komputer dapat melihat dan memuat sebuah video, kemudian mengambil frame frame tertentu dari video tersebut.

2 Hal yang Masih Membingungkan/Ingin Diperdalam:

1. Saya masih bingung dalam penggunaan MFCC pada audio.
 2. Saya sempat mengalami kesulitan saat membuat histogram untuk image.
-

Sumber Data & Referensi

TODO: Cantumkan semua sumber data dan referensi yang digunakan:

- **Audio:** [Audio Proklamasi](#)
- **Gambar:** [Freepik](#)
- **Video:** [Video Animasi Bertarung](#)
- **Referensi teknis:** [History 1 ChatGPT](#), [History 2 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

Aspek Penilaian	Bobot	Deskripsi
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