

Tugas Analisis Multimedia: Audio, Gambar, Video

Mata Kuliah: Sistem & Teknologi Multimedia

Nama: Bagus Andreanto

NIM: 122140017

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__}")
print(f"SoundFile: {sf.__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
SoundFile: 0.13.1
```

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: Musik/lagu Sound Horeg
- Sumber: Data publik (Youtube)
- Format file: WAV
- Alasan pemilihan: Supaya terasa dan terlihat jelas ketika ditampilkan dalam bentuk visual dentuman dari bass sound horeg

Path file: `data/Sound_horeg.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.

Code sederhana untuk menampilkan sample rate, durasi, jumlah kanal, dan jumlah total sample dari file `Sound_horeg.wav`

```
In [2]: # Lokasi file audio
file_path = "data/Sound_horeg.wav"

# Membaca file audio
# Fungsi read mengembalikan nilai data (array amplitudo) dan samplerate (jumlah data, samplerate = sf.read(file_path))

# Jumlah total sampel (panjang array data)
total_samples = len(data)

# Durasi dalam detik = jumlah sampel / samplerate
duration = total_samples / samplerate

# Cek jumlah kanal
# Jika data 1 dimensi -> mono
# Jika data 2 dimensi -> stereo (kolom 0 = kanal kiri, kolom 1 = kanal kanan)
if data.ndim == 1:
    num_channels = 1
else:
    num_channels = data.shape[1]

# Cetak hasil metadata
print("=== Metadata Audio ===")
print("Sample Rate   :", samplerate, "Hz")
print("Durasi         :", round(duration, 2), "detik")
print("Jumlah Kanal    :", num_channels, "(Mono)" if num_channels == 1 else "(Stereo)")
print("Total Sampel    :", total_samples)

=== Metadata Audio ===
Sample Rate   : 48000 Hz
Durasi        : 49.02 detik
Jumlah Kanal   : 2 (Stereo)
Total Sampel   : 2352900
```

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.)

Code sederhana untuk menampilkan visual waveform audio `Sound_horeg.wav` dengan sumbu x untuk waktu, dan sumbu y untuk amplitudo

```
In [3]: # Buat sumbu waktu (dalam detik) sepanjang jumlah sampel
# np.arange(total_samples) = array dari 0 sampai total sampel-1
# dibagi samplerate agar jadi detik
time = np.arange(len(data)) / samplerate

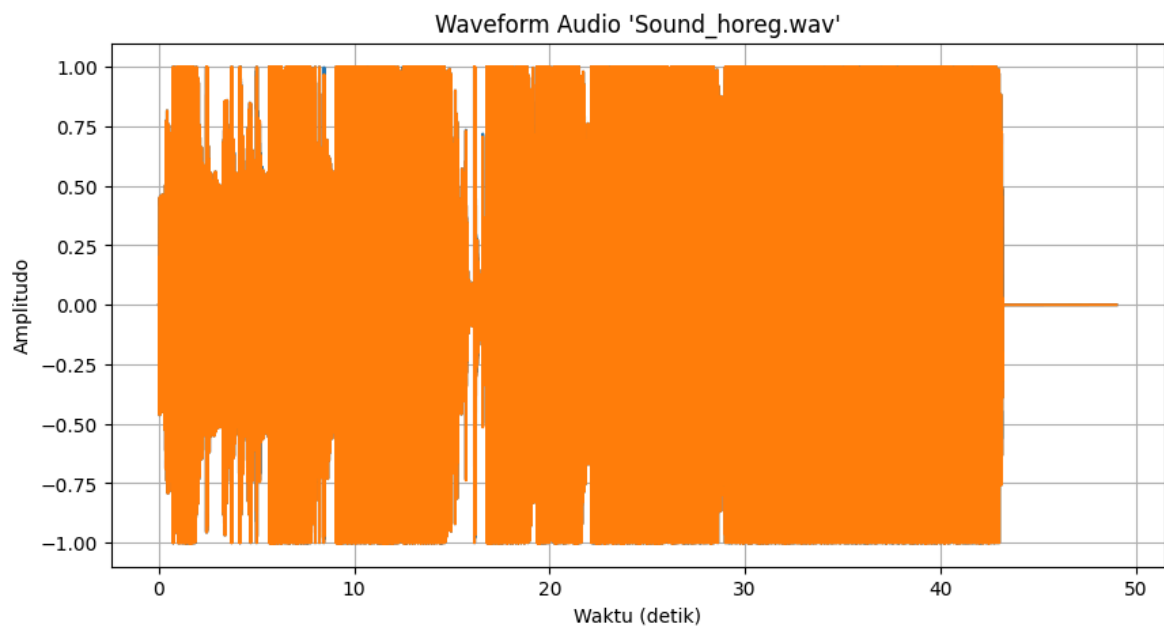
# Buat figure ukuran 10x5
plt.figure(figsize=(10, 5))

# Plot data audio terhadap waktu
plt.plot(time, data)

# Judul dan label sumbu
plt.title("Waveform Audio 'Sound_horeg.wav'")
plt.xlabel("Waktu (detik)")
plt.ylabel("Amplitudo")

# Grid garis bantu
plt.grid(True)

# Tampilkan grafik
plt.show()
```



Penjelasan dan analisis berdasarkan hasil visualisasi yang didapatkan

- sumbu x = menampilkan waktu atau durasi dari sound horeg ketika di play
- sumbu y = menampilkan nilai amplitudo yaitu getaran gelombang suara yang menentukan keras atau pelan suara yang dihasilkan

Berdasarkan grafik tersebut, di awal lagu nilai amplitudo yang dihasilkan ada disekitar angka 0.50, hal ini karena intro dari sound horeg tersebut masih berupa lirik vokal dan belum ada bass yang di tambahkan.

Kemudian setelah bagian lirik intro hampir selesai dan mulai memasuki pertengahan lagu, bass yang keras mulai di tambahkan dengan frekuensi berulang yang saling

berdekatan, sehingga menghasilkan grafik yang sangat rapat dan nilai amplitudo yang full di angka (1)

Pada detik kurang lebih 43 (s), nilai amplitudo tiba tiba drop di angka 0 dan dengan bentuk grafik datar di bagian akhir yang menunjukkan suara hening di bagian penutup sound horeg tersebut.

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.

Code sederhana untuk menampilkan spectrogram dalam skala log-dB

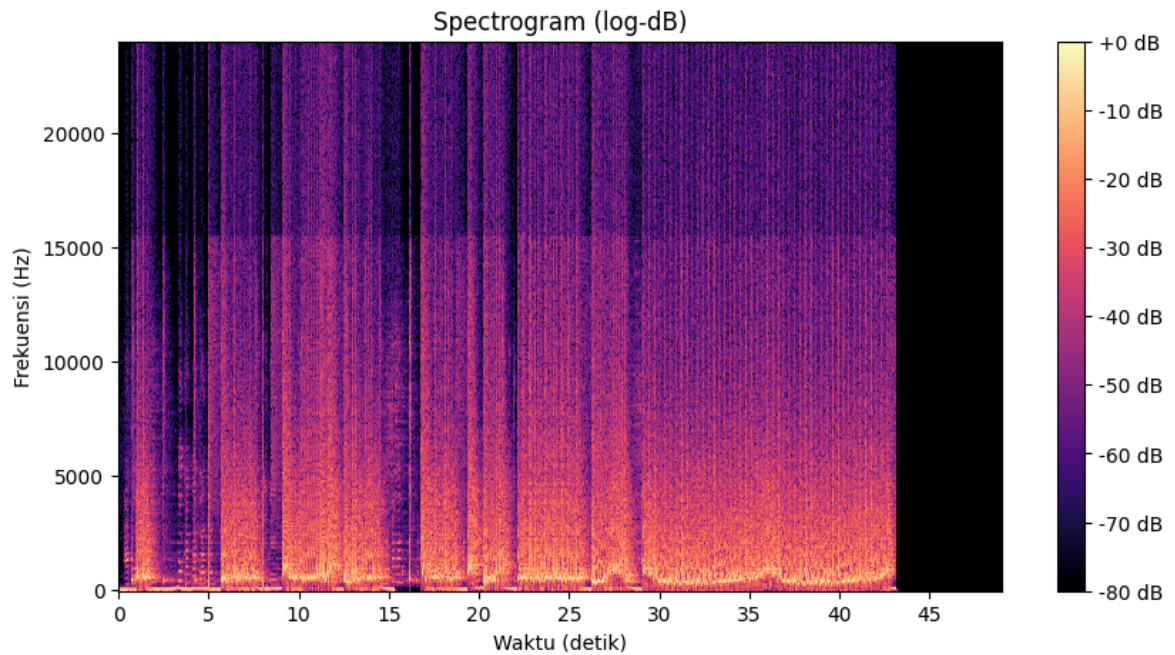
```
In [4]: # Hitung STFT (Short-Time Fourier Transform)
# n_fft = ukuran window (1024 sampel)
# hop_length = seberapa banyak "geser" antar window (256 sampel)
stft = librosa.stft(data.T[0] if data.ndim > 1 else data,
                    n_fft=1024, hop_length=256)

# Ubah ke skala magnitude (energi)
spectrogram = np.abs(stft)

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

# Plot spectrogram
plt.figure(figsize=(10, 5))
librosa.display.specshow(spectrogram_db,
                        sr=samplerate,
                        hop_length=256,
                        x_axis='time',
                        y_axis='hz',
                        cmap='magma')

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



Analisis perbedaan yang didapatkan dari spectrogram dibandingkan dengan waveform

Waveform:

Visualisasi waveform hanya menampilkan nilai amplitudo-nya saja, jadi informasi yang didapatkan hanya terbatas pada keras atau pelan suara yang dihasilkan pada setiap detik lagu

Spectrogram:

Sedangkan untuk spectrogram menampilkan data yang lebih lengkap yaitu ada intensitas warna (kuning - hitam), dimana warna terang menunjukkan bahwa suara keras di titik tersebut, sedangkan warna yang semakin gelap menunjukkan suara semakin kecil atau pelan di titik tersebut.

Kemudian juga terdapat sumbu y (frekuensi) yang menunjukkan frekuensi yang dihasilkan di waktu (s) tersebut. Berdasarkan hasil visualisasi, untuk frekuensi dari rentang 15000 - 20000 di dominasi dengan warna gelap. Hal ini menunjukkan sebaran untuk suara frekuensi tinggi yang tidak terdengar jelas oleh manusia berada di rentan tersebut. **(biasanya suara treble atau high)**

Dan untuk di rentang 0 - 15000 di dominasi dengan warna terang, dimana suara yang dihasilkan bisa terdengar jelas oleh manusia, dalam konteks sound horeg di sini adalah bagian **Bass-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?

Code sederhana untuk menampilkan heatmap 13 Koefisien MFCC

```
In [5]: # Hitung MFCC (Mel-Frequency Cepstral Coefficients)
# n_mfcc = jumlah koefisien yang dihitung (13 sesuai instruksi)
# n_fft dan hop_length disamakan dengan STFT agar konsisten
mfcc = librosa.feature.mfcc(y=data.T[0] if data.ndim > 1 else data,
                             sr=samplerate,
                             n_mfcc=13,
                             n_fft=1024,
                             hop_length=256)

# mfcc.shape = (13, jumlah_frame)
# sumbu Y = koefisien (1..13), sumbu X = frame waktu

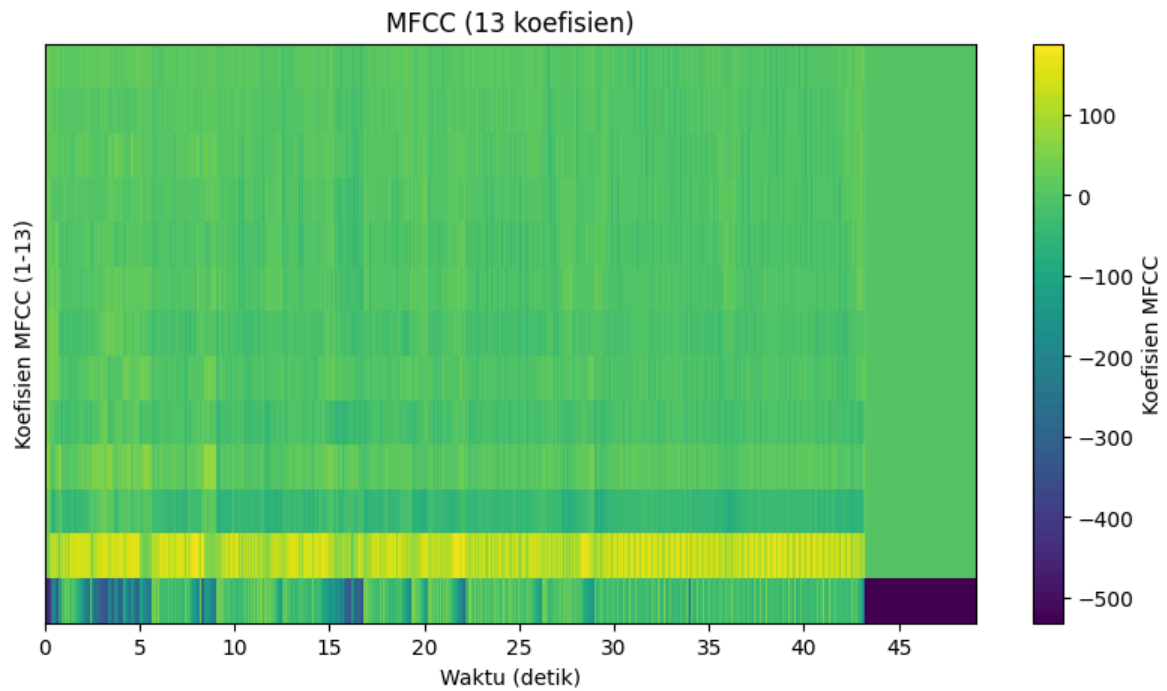
# Buat figure
plt.figure(figsize=(10, 5))

# Plot heatmap MFCC
librosa.display.specshow(mfcc,
                          x_axis='time',
                          sr=samplerate,
                          hop_length=256,
                          cmap='viridis')

# Tambahkan colorbar
plt.colorbar(label="Koefisien MFCC")

# Judul dan Label
plt.title("MFCC (13 koefisien)")
plt.xlabel("Waktu (detik)")
plt.ylabel("Koefisien MFCC (1-13)")

# Tampilkan plot
plt.show()
```



Analisis Interpretasi Pola MFCC

- **MFCC Koefisien 1 (Baris 1):** Didominasi oleh warna hijau, yang menandakan nilai koefisien sedang, dengan di beberapa momen terdapat warna biru (gelap) yang menandakan pola ritme yang tidak stabil
- **MFCC Koefisien 2 (Baris 2):** Sangat di dominasi warna kuning (terang) yang menandakan nilai koefisien tinggi dan sangat rapat hampir di seluruh waktu
- **MFCC Koefisien 3-13 (Baris 3-13):** memiliki nilai sedang dengan ritme yang berulang sehingga menghasilkan pola yang stabil

Makna dari pola yang tidak stabil di awal dan tiba tiba stabil ketika memasuki pertengahan lagu adalah, karena di awal terdapat lirik intro bersamaan dengan beberapa nada yang bervariasi serta jeda setiap nada yang masih terbilang random dan belum membentuk ritme yang stabil. Sedangkan dipertengahan lagu pola nada yang muncul mulai membentuk ritme yang stabil karena hanya ada bass dengan nada berulang.

A6. Analisis Ringkas (Wajib)

Jawab pertanyaan berikut:

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

Jawaban Anda:

Informasi yang didapatkan dari waveform hanya terbatas pada nilai amplitudo, yaitu nilai yang merepresentasikan keras atau pelannya suara yang dihasilkan.

Sedangkan spectrogram menampilkan data yang lebih detail, yaitu frekuensi (tinggi / rendah) dalam satuan Hz dan intensitas warna (gelap / terang) untuk suara keras atau pelan.

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

Jawaban Anda:

MFCC menampilkan pola koefisien yang berbeda-beda untuk setiap karakteristik frekuensi, dengan beberapa koefisien menunjukkan pola yang stabil dan berulang, sementara yang lain menunjukkan variasi yang lebih dinamis.

Jadi dibagian awal audio, pola koefisien yang dihasilkan masih tidak stabil, karena karakteristik nada masih bervariasi dan belum membentuk ritme yang jelas. Namun ketika memasuki pertengahan lagu, pola koefisien mulai stabil sehingga membentuk ritme yang berulang.

Bagian B — Gambar

B1. Deskripsi Data

TODO: Jelaskan gambar yang akan Anda analisis:

- Jenis gambar: Foto Kucing
- Sumber: Dataset Publik
- Format file: JPG
- Alasan pemilihan: Saya catlover

Path file: `data/Gambar_kucing.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).

(Tambahkan sel kode di sini)

Code sederhana untuk menampilkan gambar kucing dalam format RGB

```
In [6]: # Lokasi file gambar
image_path = "data/Gambar_kucing.jpg"

# Baca gambar dengan OpenCV (hasil dalam format BGR)
```

```

img_bgr = cv2.imread(image_path)

# Cek apakah file terbaca
if img_bgr is None:
    raise FileNotFoundError(f"Gambar tidak ditemukan: {image_path}")

# Konversi BGR ke RGB untuk tampilan yang benar
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

# Informasi dasar gambar
print("Ukuran (tinggi, lebar, kanal):", img_rgb.shape)
print("Tipe data:", img_rgb.dtype)
mean_color = img_rgb.mean(axis=(0,1))
print("Rata-rata warna (R, G, B):", np.round(mean_color, 1))
brightness = img_rgb.mean()
print("Rata-rata kecerahan (0..255):", round(brightness, 1))

# Tampilan gambar dalam format RGB
plt.figure(figsize=(8, 6))
plt.imshow(img_rgb)
plt.title("Gambar kucing (RGB)")
plt.axis('off')
plt.show()

```

Ukuran (tinggi, lebar, kanal): (6000, 4000, 3)
 Tipe data: uint8
 Rata-rata warna (R, G, B): [189.5 164.7 145.3]
 Rata-rata kecerahan (0..255): 166.5

Gambar kucing (RGB)



Analisis

- Objek dominan: Wajah kucing
- Kondisi pencahayaan: Pencahayaan soft dari belakang objek
- Komposisi warna: Didominasi warna oranye dari bulu kucing, hijau dari mata, dengan vibe cahaya hangat

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.

Code sederhana untuk menampilkan informasi metadata dari `Gambar_kucing.jpg`

```
In [7]: # Menggunakan kembali img_rgb dari B2
height, width, channels = img_rgb.shape
dtype = img_rgb.dtype

# Mode warna sederhana berdasarkan jumlah channel
if channels == 1:
    mode = "Grayscale"
elif channels == 3:
    mode = "RGB"
elif channels == 4:
    mode = "RGBA (RGB + Alpha)"
else:
    mode = f"Tidak standar ({channels} channel)"

# Estimasi ukuran file dalam memori (bytes)
memory_size = img_rgb.nbytes
memory_size_kb = memory_size / 1024
memory_size_mb = memory_size / (1024 * 1024)

print("=== Informasi Dasar Gambar ===")
print(f"Dimensi (Height x Width): {height} x {width}")
print(f"Jumlah Kanal          : {channels}")
print(f"Tipe Data (dtype)       : {dtype}")
print(f"Mode Warna              : {mode}")
print(f"Ukuran dalam Memori     : {memory_size_kb:.2f} KB ({memory_size_mb:.2f} M

=== Informasi Dasar Gambar ===
Dimensi (Height x Width): 6000 x 4000
Jumlah Kanal          : 3
Tipe Data (dtype)     : uint8
Mode Warna            : RGB
Ukuran dalam Memori   : 70312.50 KB (68.66 MB)
```

Analisis Informasi Metadata dari gambar

1. **Dimensi (Height x Width):** Untuk mengetahui resolusi gambar, jadi jika ukuran gambar terlalu besar atau kecil, kita bisa menyesuaikan resolusi sesuai kebutuhan analisis atau model yang akan digunakan.
2. **Jumlah Kanal:** Menunjukan bahwa gambar tersebut memiliki 3 kanal (RGB) yang artinya gambar tersebut berwarna dan bukan grayscale (1 kanal).
3. **Type Data (dtype):** Untuk mengetahui rentang nilai pixel (0-255 untuk uint8) dan memastikan kompatibilitas dengan algoritma pemrosesan gambar yang akan digunakan.
4. **Mode Warna:** Mengetahui mode warna yaitu RGB, yang akan dipakai ketika melakukan analisis berbasis warna.
5. **Ukuran dalam Memori:** Untuk mengetahui ukuran dari gambar tersebut, sehingga dapat disesuaikan untuk tahapan preprocessing atau analisis lanjutan.

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?

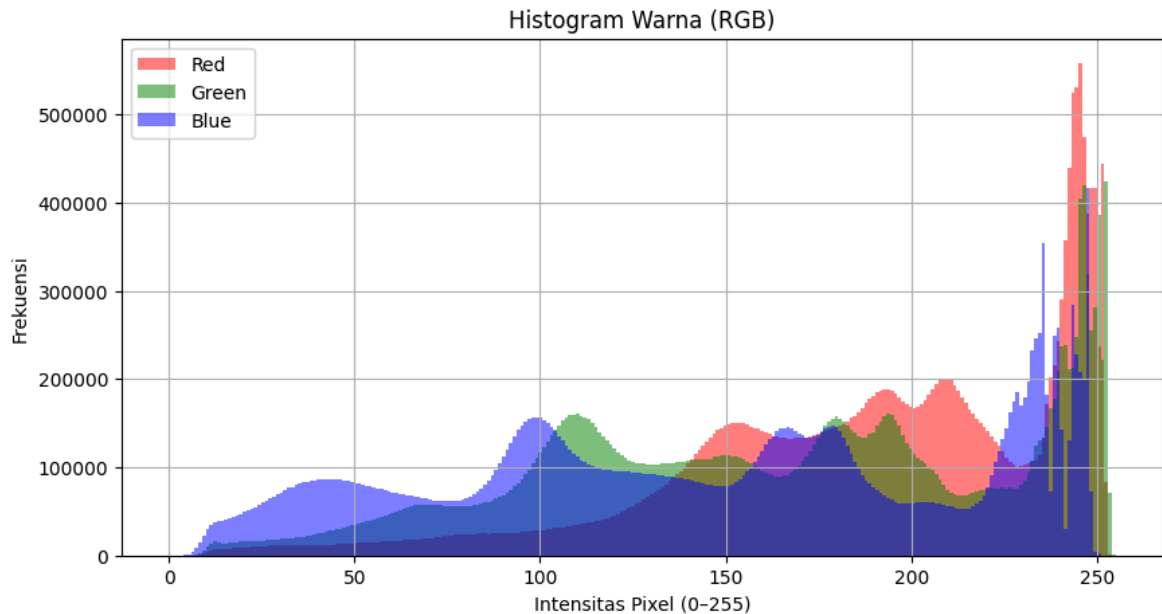
Code sederhana untuk menampilkan histogram untuk channel R, G, B

```
In [8]: # Memisahkan channel
R = img_rgb[:, :, 0].ravel() # Kanal merah
G = img_rgb[:, :, 1].ravel() # Kanal hijau
B = img_rgb[:, :, 2].ravel() # Kanal biru

# Plot histogram overlay
plt.figure(figsize=(10, 5))

plt.hist(R, bins=256, range=(0, 255), color='red', alpha=0.5, label='Red')
plt.hist(G, bins=256, range=(0, 255), color='green', alpha=0.5, label='Green')
plt.hist(B, bins=256, range=(0, 255), color='blue', alpha=0.5, label='Blue')

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



Analisis Histogram Warna

1. **Channel Dominan:** Channel merah dan hijau lebih dominan dengan nilai frekuensi yang lebih tinggi dari pada channel biru. Hal ini sesuai dengan komposisi warna gambar yang didominasi dengan vibes warna hangat seperti orange dan hijau untuk mata kucing.
2. **Kontras Gambar:** Kontras dari gambar terlihat cukup rendah, hal ini dibuktikan dengan sebaran intensitas pixel yang tidak mencakup nilai 0 hingga 255 secara penuh, melainkan sebagian besar pixel berada di rentang menengah (50-200).
3. **Sebaran Intensitas:** Sebaran intensitas pada channel R dan G menunjukkan variasi yang baik, sedangkan channel B lebih terkonsentrasi pada nilai rendah, mencerminkan sedikitnya warna biru dalam gambar. Hal ini sesuai dengan vibes hangat dari gambar kucing tersebut.

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:

Hubungan antara pola histogram yang didapatkan sesuai dengan kesan visual dari gambar tersebut. Histogram menunjukkan bahwa gambar memiliki nuansa hangat dengan dominasi warna merah dan hijau, serta kontras yang cukup rendah. Sebaran intensitas yang terbatas pada rentang menengah juga mencerminkan kesan visual yang lembut dan tidak terlalu mencolok.

Hal ini sesuai dengan gambar kucing yang memiliki pencahayaan soft dan warna bulu oranye yang hangat, serta mata hijau yang menonjol. Kontras yang rendah pada

histogram juga mencerminkan pencahayaan yang tidak terlalu keras, sehingga memberikan kesan soft dan warm.

Bagian C — Video

C1. Deskripsi Data

TODO: Jelaskan video yang akan Anda analisis:

- Jenis video: Video Kucing Happy
- Sumber: Data Publik (Youtube)
- Durasi target: 12 s
- Alasan pemilihan: Saya catlover

Path file: `data/video_kucing_happy.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.

Code sederhana untuk menampilkan metadata dari file video

`video_kucing_happy.mp4`

```
In [9]: # Path file video
video_path = "data/video_kucing_happy.mp4"

# Buka video dengan OpenCV
cap = cv2.VideoCapture(video_path)

if not cap.isOpened():
    print("Gagal membuka file video.")
else:
    # Ambil metadata dasar
    width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fps = cap.get(cv2.CAP_PROP_FPS)
    total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    duration = total_frames / fps if fps > 0 else 0

    # Klasifikasi resolusi
    if width >= 3840 and height >= 2160:
```



```

        resolution_class = "4K"
    elif width >= 1920 and height >= 1080:
        resolution_class = "Full HD"
    elif width >= 1280 and height >= 720:
        resolution_class = "HD"
    else:
        resolution_class = "SD"

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

# Tutup video
cap.release()

```

```

=== Metadata Video ===
Resolusi      : 720 x 1280 (SD)
Frame rate (fps)  : 30.00
Total Frame     : 362
Durasi         : 12.07 detik

```

Analisis Parameter Video

1. **Resolusi (Width x Height):** Menampilkan detail visual hingga ukuran dari video, sehingga proses analisis dapat dilakukan dengan lebih baik.
2. **Frame Rate (fps):** Untuk menampilkan seberapa halus gerakan dalam video tersebut.
3. **Jumlah Total Frame:** Digunakan untuk mengetahui total durasi video berdasarkan frame rate.
4. **Durasi (detik):** Durasi video memberikan konteks waktu untuk analisis, membantu dalam memahami panjang dan konten video secara keseluruhan.

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

Instruksi: Ambil dan tampilkan 3 frame representatif:

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

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

Code untuk menampilkan 3 frame representatif dari video `video_kucing_happy.mp4`

```

In [10]: # Buka video
cap = cv2.VideoCapture(video_path)

```

```

# Ambil metadata
total_frame = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
fps = cap.get(cv2.CAP_PROP_FPS) or 30.0 # fallback jika fps = 0
durasi = total_frame / fps if fps > 0 else 0

print(f"Total frame: {total_frame}, FPS: {fps:.2f}, Durasi: {durasi:.2f} detik")

# Fungsi ambil frame dengan validasi
def get_valid_frame(cap, index):
    cap.set(cv2.CAP_PROP_POS_FRAMES, index)
    ret, frame = cap.read()
    if ret:
        return cv2.cvtColor(frame, cv2.COLOR_BGR2RGB), index
    # kalau gagal, mundur 1 frame sampai ketemu valid
    for i in range(index-1, -1, -1):
        cap.set(cv2.CAP_PROP_POS_FRAMES, i)
        ret, frame = cap.read()
        if ret:
            return cv2.cvtColor(frame, cv2.COLOR_BGR2RGB), i
    return None, None

# Target frame: awal, tengah, akhir
target_frames = [0, total_frame // 2, total_frame - 1]
frames = []

for idx in target_frames:
    frame, valid_idx = get_valid_frame(cap, idx)
    if frame is not None:
        timestamp = valid_idx / fps
        frames.append((frame, valid_idx, timestamp))
    else:
        print(f"✗ Tidak bisa ambil frame {idx}")

cap.release()

# Plot hasil
plt.figure(figsize=(15,5))
for i, (frame, idx, timestamp) in enumerate(frames):
    plt.subplot(1,3,i+1)
    plt.imshow(frame)
    plt.axis("off")
    plt.title(f"Frame {idx}\n{timestamp:.2f} detik")
plt.tight_layout()
plt.show()

```

Total frame: 362, FPS: 30.00, Durasi: 12.07 detik



Analisis Perbedaan Visual Antar Frame

1. **Frame Pertama (Awal):** Pada frame awal, visual yang ditampilkan masih terlihat stabil dan jelas. Gambar kucing juga masih terlihat fokus dan tidak blur
2. **Frame Kedua (Tengah):** Pada frame tengah, terlihat adanya gerakan pada kucing, sehingga gambar mulai terlihat blur. Hal ini menandakan bahwa kucing sedang bergerak aktif di frame tersebut.
3. **Frame Ketiga (Akhir):** Pada frame akhir, gerakan kucing semakin terlihat jelas, dan gambar menjadi sedikit kurang blur dari frame sebelumnya, hal ini menandakan bahwa kucing mulai melambatkan gerakannya.

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 hasil metadata yang didapatkan, video kucing happy ini memiliki resolusi 720 x 1280 (SD) dengan frame rate 30 fps.

Maka untuk media sosial video tersebut dirasa kurang baik, karena resolusi dan frame rate yang kurang untuk standar media sosial yang biasanya menggunakan resolusi minimal Full HD (1080p) dengan frame rate 60 fps agar tampilan video lebih halus dan jernih.

sedangkan untuk keperluan kuliah daring atau presentasi, resolusi dan fps video ini sudah cukup memadai, karena mengingat beban kerja dalam memproses video serta ditambah dengan banyaknya user ketika melakukan meeting online menyebabkan bandwidth yang digunakan menjadi besar, sehingga resolusi dan fps video yang terlalu tinggi akan memberatkan kinerja sistem.

Perbandingan & Kesimpulan

Perbandingan Representasi Media

TODO: Bandingkan secara ringkas representasi dan visualisasi ketiga media:

Audio (1D - Temporal)

- Representasi: Gelombang suara serta frekuensi
- Visualisasi utama: Waveform, Spectrogram, MFCC
- Informasi yang diperoleh: Durasi, frekuensi dominan, pola perubahan suara

Gambar (2D - Spasial)

- Representasi: Piksel gambar dalam format RGB
- Visualisasi utama: Histogram warna, Heatmap
- Informasi yang diperoleh: Dimensi, jumlah kanal, channel dominan, kontras

Video (2D + Waktu - Spatio-temporal)

- Representasi: Piksel video dalam format BGR
 - Visualisasi utama: Frame representatif, Histogram warna
 - Informasi yang diperoleh: Resolusi, fps, frame count, durasi
-

Refleksi Pembelajaran

3 Poin yang Saya Pelajari:

1. Menampilkan waveform, spectrogram, dan MFCC untuk analisis audio
2. Menampilkan histogram warna dan analisis hubungan histogram dengan kesan visual gambar
3. Menampilkan metadata video dan analisis kesesuaian parameter video dengan use case

2 Hal yang Masih Membingungkan/Ingin Diperdalam:

1. Bagian MFCC masih kurang memahami cara membaca visualisasinya
 2. Bagian representatif dari video karena mengalami beberapa error ketika mengerjakan bagian tersebut
-

Sumber Data & Referensi

TODO: Cantumkan semua sumber data dan referensi yang digunakan:

- **Audio:** [Sound Horeg Tukang Tambal Ban Full Bass](#)
- **Gambar:** [Gambar Kucing](#)
- **Video:** [Video Kucing Happy](#)
- **Referensi teknis:** [Referensi GPT](#)

Rubrik Penilaian

Distribusi Bobot Penilaian

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

Detail Kriteria Penilaian

Kelengkapan (35%)

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

Kualitas Visualisasi (20%)

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

Analisis & Interpretasi (30%)

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

Kerapihan & Struktur (10%)

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

Orisinalitas & Penguasaan (5%)

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





Proporsi Penilaian Total

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

Aturan Kejujuran Akademik





Penggunaan Referensi & AI yang Diperbolehkan

Anda **BOLEH** menggunakan:

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

Syarat & Batasan WAJIB

Namun Anda **HARUS**:

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

Pelanggaran yang Berakibat Nilai 0

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

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

Persiapan Presentasi Acak

Kemungkinan pertanyaan yang akan ditanyakan:

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

Tips sukses:

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

Panduan Pengumpulan

Berkas yang Harus Dikumpulkan

Wajib:

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



Informasi Pengumpulan



Checklist Sebelum Submit

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

Export ke PDF:

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