

Anggota Kelompok :

1. Abiyanfaras Danuyasa (312210103)
2. Birrham Efendi Lubis (312210272)
3. Muhammad Zidan Fadillah (312210277)

Kelas : TI.22.A.2

Mata Kuliah : Pengolahan Citra

Laporan Project : Aplikasi Segmentasi Gambar Menggunakan K-Means Clustering dengan Streamlit

1. Pendahuluan

Proyek ini bertujuan untuk membuat aplikasi web interaktif menggunakan Streamlit yang dapat melakukan segmentasi gambar berdasarkan warna menggunakan algoritma K-Means Clustering. Aplikasi ini memungkinkan pengguna untuk mengunggah gambar, mengatur jumlah cluster (k), dan melihat hasil segmentasi serta persentase warna dari setiap segmen yang dihasilkan.

2. Pustaka yang Digunakan

Proyek ini menggunakan beberapa pustaka Python, yaitu:

- `streamlit`: Untuk membuat antarmuka web interaktif.
- `numpy`: Untuk operasi numerik dan manipulasi array.
- `cv2` (OpenCV): Untuk pemrosesan gambar.
- `PIL` (Pillow): Untuk memuat dan memanipulasi gambar.
- `matplotlib.colors`: Untuk konversi warna dan manipulasi warna.

3. Implementasi Kode

Berikut adalah penjelasan rinci mengenai implementasi kode:

A. Mengimpor Pustaka yang Diperlukan

```
python
Salin kode
import streamlit as st
import numpy as np
import cv2
from PIL import Image
import matplotlib.colors as mcolors
```

B. Fungsi Segmentasi Gambar

Fungsi `segment_image` menggunakan algoritma K-Means Clustering untuk membagi gambar menjadi beberapa segmen berdasarkan warna.

```
python
Salin kode
def segment_image(image, k, max_iter=100, epsilon=0.85):
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
    pixel_vals = image.reshape((-1, 3))
    pixel_vals = np.float32(pixel_vals)
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER,
max_iter, epsilon)
    _, labels, centers = cv2.kmeans(pixel_vals, k, None, criteria, 10,
cv2.KMEANS_RANDOM_CENTERS)
    centers = np.uint8(centers)
    segmented_data = centers[labels.flatten()]
    segmented_image = segmented_data.reshape((image.shape))
    segmented_image = cv2.cvtColor(segmented_image, cv2.COLOR_BGR2RGB)
    segment_percentages = [(np.sum(labels == i) / len(labels)) * 100 for i
in range(k)]
    return segmented_image, centers, segment_percentages
```

C. Fungsi Mengenali Warna

Fungsi `get_color_name` digunakan untuk mengenali nama warna berdasarkan nilai RGB dari pusat cluster yang dihasilkan oleh K-Means.

```
python
Salin kode
def get_color_name(center):
    colors = {
        "Merah": [255, 0, 0],
        "Hijau": [0, 255, 0],
        "Biru": [0, 0, 255],
        "Kuning": [255, 255, 0],
        "Cyan": [0, 255, 255],
        "Magenta": [255, 0, 255],
        "Putih": [255, 255, 255],
        "Hitam": [0, 0, 0],
        "Oranye": [255, 165, 0],
        "Abu-abu": [128, 128, 128]
    }
    color_name = "Tidak diketahui"
    min_dist = float('inf')
    for name, color in colors.items():
        dist = np.linalg.norm(center - np.array(color))
        if dist < min_dist:
            min_dist = dist
            color_name = name
    return color_name
```

D. Konfigurasi Halaman Streamlit

Konfigurasi halaman Streamlit untuk judul, ikon, dan layout halaman.

```
python
Salin kode
st.set_page_config(
    page_title="K-Means",
    page_icon=":art:",
    layout="wide"
)
```

E. Menambahkan Gaya CSS

Menambahkan gaya CSS untuk mempercantik tampilan aplikasi.

```
python
Salin kode
st.markdown(
    """
    <style>
    .stApp {
        background-color: #f0f0f0;
    }
    .st-bw {
        background-color: white;
        padding: 20px;
        border-radius: 10px;
        box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.1);
    }
    .segmented-image {
        border-radius: 10px;
        box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.1);
        margin-top: 20px;
    }
    .sidebar .sidebar-content {
        background-color: #ffffff;
        box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.1);
        border-radius: 10px;
        padding: 20px;
        margin-top: 20px;
    }
    .sidebar .sidebar-content h2 {
        font-size: 24px;
        font-weight: bold;
        margin-bottom: 10px;
    }
    .sidebar .sidebar-content p {
        font-size: 16px;
        line-height: 1.5;
    }
    .sidebar .sidebar-content .widget.stSlider {
        margin-top: 20px;
    }
    </style>
    """,
    unsafe_allow_html=True
)
```

```
)
```

F. Tampilan Aplikasi

Menampilkan judul aplikasi.

```
python
Salin kode
st.markdown(
    """
    <div class="title-wrapper">
        <h6 style="font-size: 30px; text-align: center; color: #FFD700;
            text-shadow: 2px 2px 4px #000000;">Segmentasi Gambar
Menggunakan K-Means Clustering</h6>
    </div>
    """,
    unsafe_allow_html=True
)
```

G. Layout Kolom dan Unggah Gambar

Membagi layout menjadi dua kolom dan memungkinkan pengguna untuk mengunggah gambar.

```
python
Salin kode
col1, col2 = st.columns((2, 1))

# Unggah gambar di kolom pertama
with col1:
    uploaded_file = st.file_uploader("Unggah gambar...", type=["jpg",
"jpeg", "png"])
    if uploaded_file is not None:
        image = np.array(Image.open(uploaded_file))
        st.image(image, caption='Gambar Asli', use_column_width=True)
```

H. Parameter Segmentasi

Menampilkan slider di sidebar untuk memilih jumlah cluster dan tombol untuk memicu proses segmentasi.

```
python
Salin kode
with col2:
    st.sidebar.header("Pengaturan")
    k = st.sidebar.slider("Jumlah Cluster (k)", min_value=2, max_value=10,
value=3)

    if st.sidebar.button("Segmentasikan Gambar"):
        with st.spinner('Sedang memproses...'):
            segmented_image, centers, segment_percentages =
segment_image(image, k)
            st.success('Selesai!')

    st.image(segmented_image, caption='Gambar Tersegmentasi',
use_column_width=True)
```

```

        st.subheader("Persentase Warna Setiap Segmen:")
        for i, (center, percentage) in enumerate(zip(centers,
segment_percentages)):
            color_name = get_color_name(center)
            st.write(f"Segmen {i + 1} ({color_name}): {percentage:.2f}%")

            # Tambahkan plot warna untuk setiap segmen
            color_rgb = np.uint8([[center]])
            color_rgb = cv2.cvtColor(color_rgb, cv2.COLOR_BGR2RGB)
            color_hex = mcolors.rgb2hex(color_rgb.squeeze() / 255.)
            st.write(f'<div style="width: 50px; height: 50px; background-
color: {color_hex}; border-radius: 50%; display: inline-block;"></div>',
unsafe_allow_html=True)

```

I. Menjalankan Aplikasi Streamlit

- Menggunakan `if __name__ == '__main__':` untuk memastikan aplikasi dijalankan dengan benar saat skrip dieksekusi.

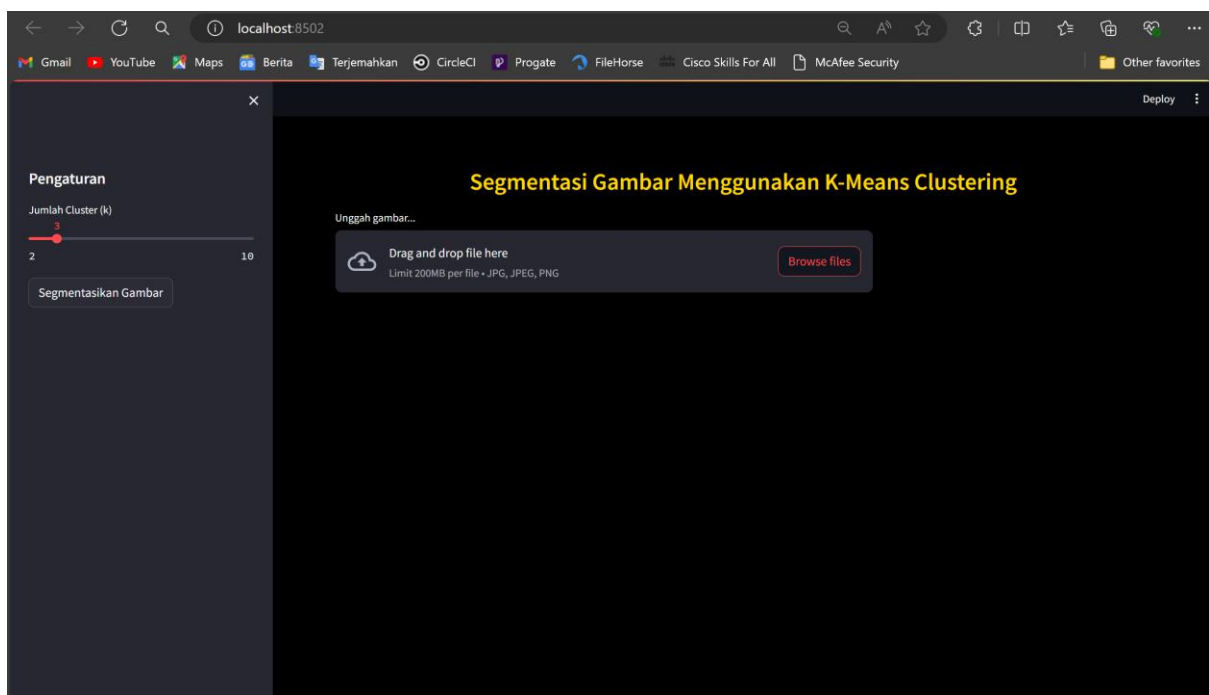
```

python
Salin kode
if __name__ == '__main__':
    st.set_option('deprecation.showPyplotGlobalUse', False)

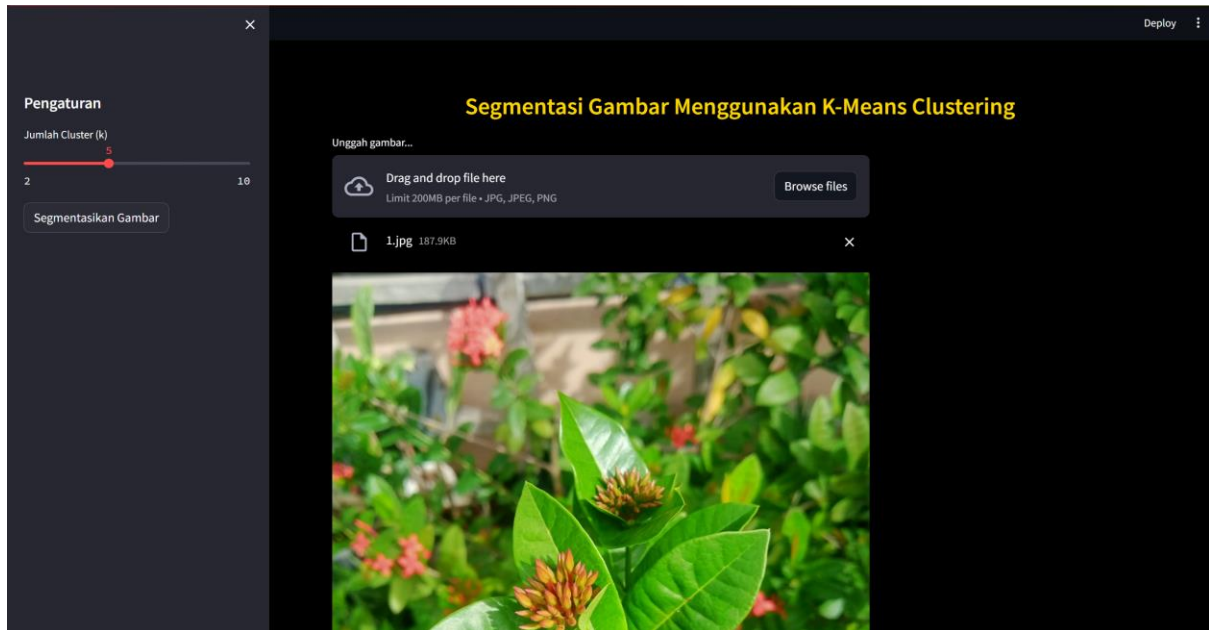
```

5. Penggunaan Aplikasi Segmentasi Menggunakan K-Means Clustering

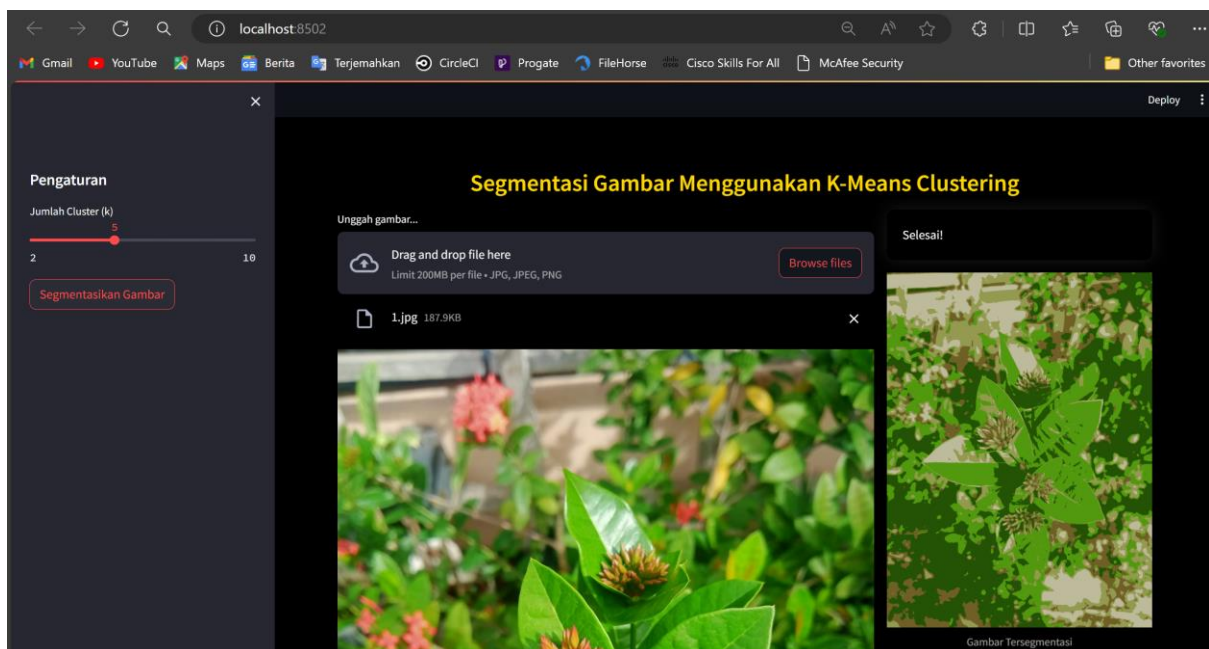
- Pertama kita harus memasukkan gambar dari file ke aplikasi



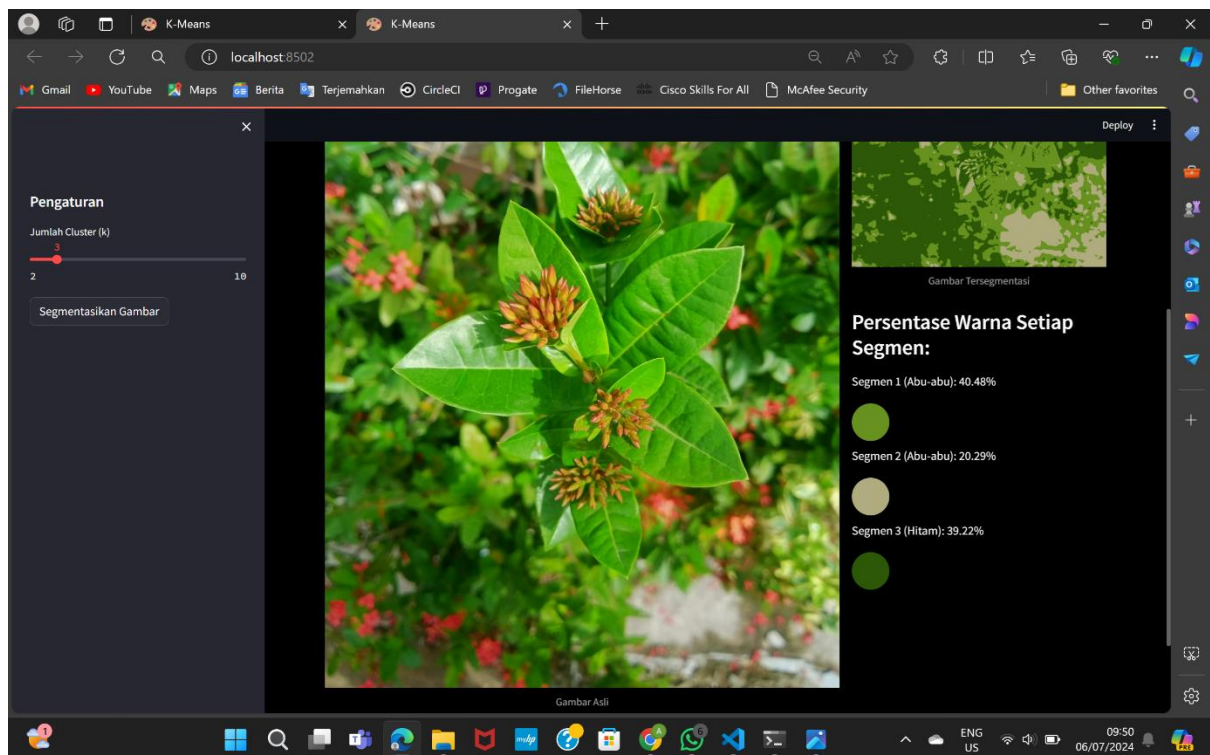
- Kemudian kita mengatur Jumlah Cluster yang di inginkan atau yang ingin di segemntasikan gambar nya



- Gambar sudah di segmentasi seperti gambar di bawah ini



- Deskripsi presentase warna dari setiap segmen



- Contoh perbandingan gambar dari gambar satu dan lain
- Spesifikasi Handphone Dan Gambar

Tipe Hp	Samsung Galaxy A51
Dimension	1200x1600
Widht	1200
Height	1600
Horizontal Resolution	96
Vertical Resolution	96
Bith depth	24



Gambar Asli



Gambar Tersegmentasi

Persentase Warna Setiap Segmen:

Segmen 1 (Abu-abu): 11.89%



Segmen 2 (Abu-abu): 27.62%



Segmen 3 (Hitam): 31.81%



Segmen 4 (Abu-abu): 10.79%



Segmen 5 (Abu-abu): 17.89%



Gambar Presentase Segmen



Gambar Asli



Gambar Tersegmentasi

Persentase Warna Setiap Segmen:

Segmen 1 (Abu-abu): 22.42%



Segmen 2 (Abu-abu): 20.95%



Segmen 3 (Hitam): 18.18%



Segmen 4 (Abu-abu): 18.88%



Segmen 5 (Abu-abu): 19.57%



Gambar Presentase Segment

- **Spesifikasi Handphone Dan Gambar**

Type Hp	Oppo A31
Dimension	1182x1564
Widht	1182
Height	1564
Horizontal Resolution	96
Vertical Resolution	96
Bith depth	24



Gambar Asli



Gambar Tersegmentasi

Persentase Warna Setiap Segmen:

Segmen 1 (Abu-abu): 16.27%



Segmen 2 (Hitam): 7.28%



Segmen 3 (Abu-abu): 22.22%



Segmen 4 (Abu-abu): 29.15%



Segmen 5 (Abu-abu): 25.07%



Gambar Presentase Segment



Gambar Asli



Gambar Tersegmentasi

Persentase Warna Setiap Segmen:

Segmen 1 (Abu-abu): 18.23%



Segmen 2 (Abu-abu): 19.94%



Segmen 3 (Hitam): 19.90%



Segmen 4 (Abu-abu): 22.81%



Segmen 5 (Abu-abu): 19.12%



Gambar Presentase Segment

6. Kesimpulan

Proyek ini berhasil membuat aplikasi web interaktif untuk segmentasi gambar menggunakan K-Means Clustering. Pengguna dapat mengunggah gambar, memilih jumlah cluster, dan melihat hasil segmentasi serta persentase warna dari setiap segmen yang dihasilkan. Aplikasi ini memberikan wawasan tentang distribusi warna dalam gambar dan dapat digunakan dalam berbagai aplikasi analisis citra.