

1. **Dimas Firmansyah (312210267)**
2. **Aditya Putra Wijaya (312210207)**
3. **Muhammad Radja (312210623)**

Kelas : TI.22.A.2

Mata Kuliah : Pengolahan Citra

Laporan Aplikasi Segmentasi Gambar Menggunakan K-Means Clustering dengan Streamlit

I. Pendahuluan

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

1.1 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.

II. Implementasi Kode

Berikut adalah penjelasan rinci mengenai implementasi kode:

2.1 Mengimpor Pustaka yang Diperlukan

```
import cv2
import numpy as np
from sklearn.cluster import KMeans
import streamlit as st
```

2.2 Fungsi load_image

Fungsi load_image Memuat gambar dari path yang diberikan dan mengonversinya dari format BGR (default OpenCV) ke RGB.

```
def load_image(image_path):
    image = cv2.imread(image_path)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    return image
```

2.3 Fungsi reshape_image

Fungsi reshape_image Mengubah gambar menjadi array 2D di mana setiap baris adalah piksel dengan 3 nilai warna (RGB) dan mengonversi nilai piksel ke tipe data float32.

```
def reshape_image(image):
    pixel_values = image.reshape((-1, 3))
    pixel_values = np.float32(pixel_values)
    return pixel_values
```

2.4 Fungsi apply_kmeans

Menerapkan algoritma KMeans untuk clustering piksel dan mengembalikan label untuk setiap piksel dan pusat dari setiap cluster.

```
def apply_kmeans(pixel_values, k):
    kmeans = KMeans(n_clusters=k, random_state=42)
    labels = kmeans.fit_predict(pixel_values)
    centers = kmeans.cluster_centers_
    return labels, centers
```

2.5 Fungsi create_segmented_image

- Mengonversi pusat cluster ke tipe data uint8.
- Menggunakan label untuk membentuk kembali gambar yang disegmentasi.
- Mengembalikan gambar yang disegmentasi.

```
def create_segmented_image(labels, centers, image_shape):
    centers = np.uint8(centers)
    segmented_image = centers[labels.flatten()]
    segmented_image = segmented_image.reshape(image_shape)
    return segmented_image
```

2.6 Fungsi `calculate_color_percentage`

Menghitung persentase setiap warna berdasarkan jumlah piksel yang termasuk dalam setiap cluster.

Mengembalikan dictionary dengan warna (RGB) dan persentasenya.

```
def calculate_color_percentage(labels, centers):
    unique, counts = np.unique(labels, return_counts=True)
    total_pixels = len(labels)
    percentages = {tuple(np.uint8(centers[i])): count / total_pixels
    * 100 for i, count in zip(unique, counts)}
    return percentages
```

2.7 Fungsi `display_color_percentages`

- Menampilkan persentase setiap warna dalam kolom yang diberikan.
- Membuat div berwarna sesuai dengan warna dari setiap cluster.

```
def display_color_percentages(percentages, col):
    for color, percentage in percentages.items():
        col.write(f"Color {color}: {percentage:.2f}%")
        col.write(
            f"<div style='background-color: rgb({color[0]},
{color[1]}, {color[2]}); width: 50px; height: 50px;'></div>",
            unsafe_allow_html=True
        )
```

2.8 Fungsi main

- st.title: Menampilkan judul aplikasi Streamlit.
- st.file_uploader: Mengunggah file gambar.
- Jika file diunggah, gambar diubah menjadi array numpy dan dikonversi ke format RGB.
- st.slider: Menyediakan slider untuk memilih jumlah cluster (k).
- Memanggil fungsi untuk reshaping gambar, menerapkan KMeans, dan membuat gambar yang disegmentasi.
- Menampilkan gambar asli dan gambar yang disegmentasi dalam kolom terpisah.
- Menampilkan persentase warna di kolom ketiga.

```
def main():
    st.title("Segmentasi Gambar dengan KNN")

    uploaded_file = st.file_uploader("Choose an image...",
    type=["jpg", "jpeg", "png"])

    if uploaded_file is not None:
        # Convert the uploaded file to an image
        image = np.array(bytearray(uploaded_file.read()),
dtype=np.uint8)
        image = cv2.imdecode(image, cv2.IMREAD_COLOR)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

        k = st.slider("Number of clusters (k)", min_value=2,
max_value=10, value=4)

        # Reshape and apply KMeans
        pixel_values = reshape_image(image)
        labels, centers = apply_kmeans(pixel_values, k)

        # Create segmented image
        segmented_image = create_segmented_image(labels, centers,
image.shape)

        # Display original and segmented images side by side
        col1, col2, col3 = st.columns(3)

        with col1:
            st.image(image, caption='Gambar Asli',
use_column_width=True)

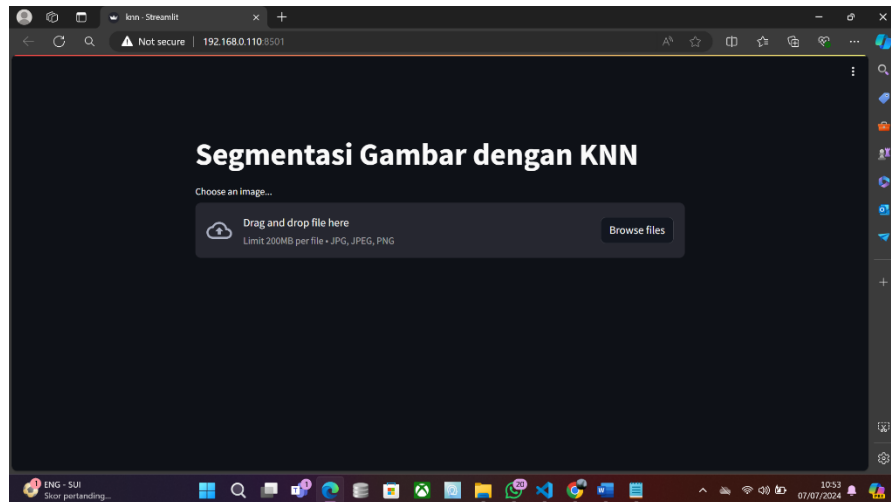
        with col2:
            st.image(segmented_image, caption='Segmentasi Gambar',
use_column_width=True)

        with col3:
            st.write("Persentase segmentasi gambar:")
            percentages = calculate_color_percentage(labels,
centers)
            display_color_percentages(percentages, col3)

if __name__ == "__main__":
    main
```

IV. Penggunaan Aplikasi Segmentasi Menggunakan K-Means Clustering

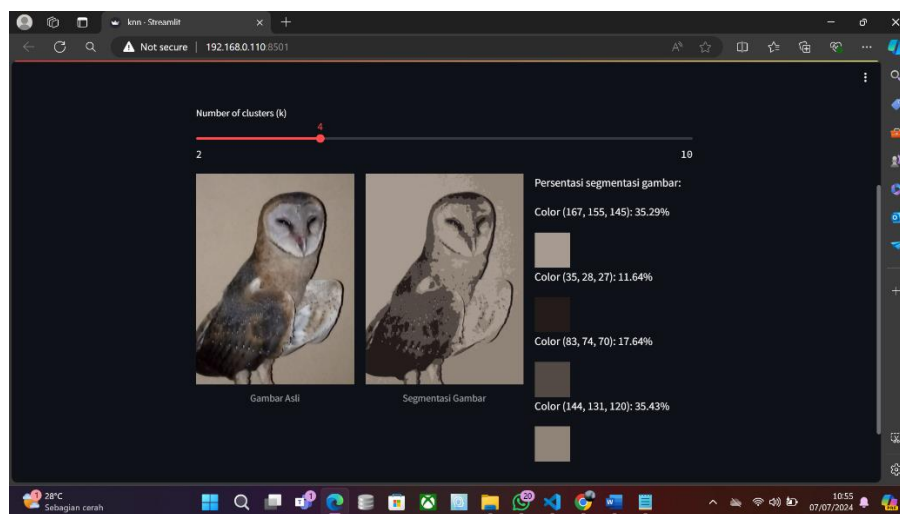
5.1 Pertama masukan gambar ke aplikasi



5.2 Kemudian kita mengatur Jumlah Cluster yang di inginkan atau yang ingin disegmentasikan gambar nya

5.3 Gambar sudah di segmentasi seperti gambar di bawah ini

5.4 Deskripsi presentase warna dari setiap segmen



VI. Contoh Perbandingan Gambar

Disini kita akan mencoba membandingkan beberapa foto yang ditangkap dari kamera yang berbeda. Kami juga menyiapkan deskripsi gambar dan jenis kamera yang di pakai untuk mengetahui segmentasi gambar.

- Spesifikasi handphone, segmentasi gambar, dan kadar warna dalam gambar

| | |
|-----------------------|-----------------|
| Tipe Hp | Infinix Hot 12i |
| Dimension | 4160x3120 |
| Widht | 4160 |
| Height | 3120 |
| Horizontal Resolution | 96 |
| Vertical Resolution | 96 |
| Bith depth | 24 |



Gambar Asli



Gambar tersegmentasi

Persentasi segmentasi gambar:

Color (170, 182, 208): 22.45%



Color (183, 125, 66): 46.27%



Color (56, 33, 48): 17.61%

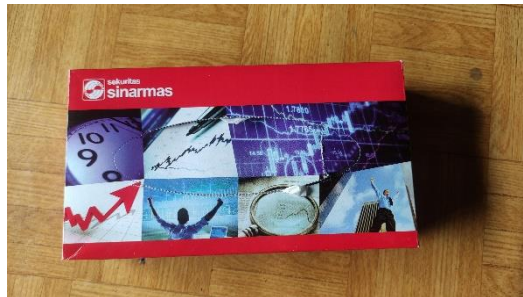


Color (217, 43, 56): 13.67%



- **Spesifikasi handphone, segmentasi gambar, dan kadar warna dalam gambar**

| | |
|-----------------------|-------------|
| Tipe Hp | Poco X3 NFC |
| Dimension | 4160x2342 |
| Widht | 4160 |
| Height | 2342 |
| Horizontal Resolution | 96 |
| Vertical Resolution | 96 |
| Bith depth | 24 |



Gambar Asli



Gambar tersegmentasi

Persentasi segmentasi gambar:

Color (40, 30, 48): 18.12%



Color (175, 138, 82): 33.02%



Color (160, 66, 45): 31.21%



Color (153, 185, 213): 17.66%



VII. 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.