

Anggota Kelompok :

1. Mohamad Mahdi Alethea (312210195)
2. Veronika Natalia Kala(312210690)

Kelas : TI.22.A.2

Mata Kuliah : Pengolahan Citra

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

Pendahuluan

Segmentasi gambar adalah teknik pemrosesan gambar yang memisahkan gambar menjadi beberapa bagian atau segmen yang lebih bermakna dan mudah dianalisis. Salah satu metode yang digunakan untuk segmentasi gambar adalah K-Means Clustering. Dalam laporan ini, kita akan membahas implementasi segmentasi gambar menggunakan K-Means Clustering dengan bantuan pustaka Streamlit untuk membuat antarmuka pengguna.

Pustaka yang Digunakan

1. **Streamlit** : Untuk membuat antarmuka pengguna web yang interaktif.
2. **NumPy** : Untuk manipulasi array dan komputasi numerik.
3. **OpenCV** : Untuk pemrosesan gambar.
4. **Pillow** : Untuk membuka dan memproses gambar.
5. **Matplotlib** : Untuk membantu dalam konversi warna.

Fungsi Utama

1. **segment_image**: Fungsi ini digunakan untuk melakukan segmentasi gambar menggunakan algoritma K-Means Clustering.
 - **Parameter:**
 - **Image** : Gambar input dalam format array.
 - **K** : Jumlah cluster yang diinginkan.
 - **max_iter** : Jumlah iterasi maksimum untuk algoritma K-Means.
 - **Epsilon** : Kriteria konvergensi untuk algoritma.
 - **Output:**
 - **segmented_image** : Gambar yang telah tersegmentasi.
 - **Centers** : Pusat-pusat cluster.
 - **segment_percentages** : Persentase masing-masing segmen dalam gambar.

2. **get_color_name:** Fungsi ini digunakan untuk mengenali warna berdasarkan nilai RGB dari pusat cluster.
 - **Parameter:**
 - center: Nilai RGB dari pusat cluster.
 - **Output:**
 - Nama warna yang paling mendekati nilai RGB yang diberikan.

Konfigurasi Halaman Streamlit

Konfigurasi halaman mencakup pengaturan tampilan antarmuka pengguna, termasuk gaya CSS untuk mempercantik tampilan aplikasi.

Implementasi Antarmuka Pengguna

- **Upload Gambar:** Pengguna dapat mengunggah gambar yang ingin mereka segmentasikan.
- **Pengaturan Parameter:** Pengguna dapat mengatur jumlah cluster (k) yang diinginkan melalui slider di sidebar.
- **Proses Segmentasi:** Setelah mengatur parameter, pengguna dapat menekan tombol "Segmentasikan Gambar" untuk memulai proses segmentasi.
- **Tampilan Hasil:** Gambar tersegmentasi ditampilkan bersama dengan persentase dan warna dari masing-masing segmen.

Kode Lengkap

Berikut adalah kode lengkap implementasi aplikasi segmentasi gambar menggunakan K-Means Clustering dengan Streamlit:

```
import streamlit as st
import numpy as np
import cv2
from PIL import Image
import matplotlib.pyplot as plt
import matplotlib.colors as mcolersp

# Fungsi untuk melakukan segmentasi gambar menggunakan K-Means Clustering
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

# Fungsi untuk mengenali warna berdasarkan nilai RGB
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

# Konfigurasi halaman Streamlit
st.set_page_config(
    page_title="K-Means",
    page_icon=":art:",
    layout="wide"
)

# Menambahkan gaya CSS untuk mempercantik tampilan
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
)

# Tampilan aplikasi
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
)

```

```

# Layout kolom
col1, col2 = st.columns((2, 1))

# Unggah gambar di kolom pertama
with col1:
    uploaded_file = st.file_uploader("Choose Images...", 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)

# Parameter segmentasi di kolom kedua
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)

# Menjalankan aplikasi Streamlit
if __name__ == '__main__':
    st.set_option('deprecation.showPyplotGlobalUse', False)

```

- Spesifikasi Handphone dan Gambar

Type Hp	Redmi Note 11
Dimension	2160x3840
Widht	2160
Height	3840
Horizontal Resolution	96
Vertical Resolution	96
Bith depth	24



Gambar Asli



Gambar Tersegmentasi

Persentase Warna Setiap Segmen:

Segmen 1 (Putih): 38.60%



Segmen 2 (Abu-abu): 4.91%



Segmen 3 (Abu-abu): 21.36%



Segmen 4 (Abu-abu): 5.57%



Segmen 5 (Abu-abu): 29.57%

