

**Laporan Proyek K-Means Clustering
pada Gambar Lemon**



Di susun oleh :

BAYU MAULANA AYASSY – 312210166

CAHYO HIDAYATULLAH – 312210079

YOGA PRATAMA - 312210042

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PELITA BANGSA

BEKASI

2024

Pendahuluan

Dalam bidang pengolahan citra digital, segmentasi gambar adalah salah satu teknik penting yang digunakan untuk menganalisis dan memahami struktur dalam gambar. Segmentasi gambar bertujuan untuk membagi gambar menjadi beberapa bagian atau region yang memiliki karakteristik tertentu, seperti warna, intensitas, atau tekstur yang serupa. Teknik ini sangat berguna dalam berbagai aplikasi, termasuk pengenalan objek, pengindeksan gambar, pemrosesan medis, dan pengawasan video.

K-Means Clustering adalah salah satu algoritma pembelajaran tanpa pengawasan yang paling sering digunakan untuk segmentasi gambar. Algoritma ini bekerja dengan mengelompokkan data ke dalam K cluster yang berbeda berdasarkan jarak antara data dan pusat cluster. Dalam konteks pengolahan citra, data yang dimaksud adalah nilai piksel dari gambar.

Metode Penelitian

1. Membaca Gambar :

Kami memulai dengan membaca gambar lemon menggunakan pustaka OpenCV. Gambar yang digunakan adalah gambar lemon yang berada di tangan seseorang.

2. Konversi Warna :

Gambar yang dibaca oleh OpenCV menggunakan format warna BGR (Blue, Green, Red). Namun, untuk visualisasi yang lebih baik dan kompatibilitas dengan pustaka matplotlib, kami mengonversi gambar tersebut ke format RGB (Red, Green, Blue).

3. Pembentukan Ulang :

Gambar dua dimensi (2D) diubah menjadi susunan piksel satu dimensi (1D) dengan tiga nilai warna (RGB). Langkah ini diperlukan agar data dapat digunakan dalam algoritma K-Means.

4. Konversi ke Tipe Float :

Nilai piksel dikonversi ke tipe float untuk memenuhi persyaratan algoritma K-Means dari OpenCV.

5. Menentukan Kriteria Penghentian :

Kriteria penghentian untuk algoritma K-Means ditentukan berdasarkan dua kondisi: jumlah iterasi maksimum (100 iterasi) dan perubahan minimal dalam posisi pusat cluster (epsilon sebesar 0.85).

6. Clustering :

Algoritma K-Means dijalankan dengan jumlah cluster yang ditetapkan sebagai 8. Proses ini melibatkan inisialisasi acak dari pusat cluster, pengelompokan piksel berdasarkan jarak terdekat ke pusat cluster, dan pembaruan pusat cluster hingga kriteria penghentian terpenuhi.

7. Segmentasi Gambar :

Setelah clustering selesai, setiap piksel dalam gambar asli digantikan oleh nilai pusat cluster terdekat. Hasilnya adalah gambar tersegmentasi yang menunjukkan beberapa bagian berbeda berdasarkan warna.

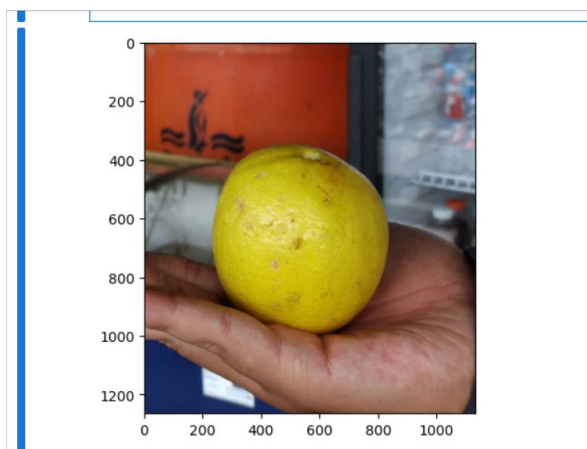
8. Visualisasi :

Gambar asli dan gambar hasil segmentasi ditampilkan untuk membandingkan dan mengilustrasikan hasil segmentasi.

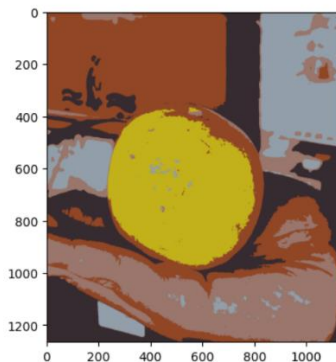
Hasil

Gambar asli lemon dan gambar hasil segmentasi ditampilkan di bawah ini :

Gambar Asli :



Gambar Hasil Segmentasi :



Pada gambar hasil segmentasi, kita dapat melihat bahwa gambar lemon terbagi menjadi beberapa bagian dengan warna yang berbeda. Setiap bagian menunjukkan cluster yang berbeda berdasarkan kesamaan warna. Hasil ini menggambarkan bagaimana algoritma K-Means dapat digunakan untuk mengidentifikasi dan memisahkan fitur-fitur dalam gambar.

Kesimpulan

Algoritma K-Means telah berhasil mempartisi gambar lemon menjadi beberapa cluster berdasarkan warna. Hasil segmentasi menunjukkan bahwa gambar lemon dapat dibagi menjadi beberapa bagian dengan warna yang berbeda. Proyek ini memberikan wawasan tentang bagaimana K-Means dapat diterapkan dalam pengolahan citra untuk segmentasi gambar. Meskipun hasilnya bergantung pada jumlah cluster yang dipilih, metode ini tetap efektif untuk mengenali pola dan struktur dalam data gambar.

Code Python

```
```python

import numpy as np
import matplotlib.pyplot as plt
import cv2

Membaca gambar, gunakan gambar sesuai dengan yang dimiliki
image = cv2.imread('/mnt/data/Screenshot 2024-07-09 132624.png')

Mengubah warna ke RGB
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image)
plt.title('Gambar Asli')
plt.show()

Membentuk ulang gambar menjadi susunan piksel 2D dengan 3 nilai warna (RGB)
pixel_vals = image.reshape((-1, 3))

Mengonversi ke tipe float
pixel_vals = np.float32(pixel_vals)

Menentukan kriteria agar algoritme berhenti
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.85)
```

**# Melakukan k-means clustering dengan jumlah cluster yang ditetapkan sebagai 3**

k = 8

```
retval, labels, centers = cv2.kmeans(pixel_vals, k, None, criteria, 10,
cv2.KMEANS_RANDOM_CENTERS)
```

**# Mengonversi data menjadi nilai 8-bit**

```
centers = np.uint8(centers)
```

```
segmented_data = centers[labels.flatten()]
```

**# Membentuk ulang data menjadi dimensi gambar asli**

```
segmented_image = segmented_data.reshape((image.shape))
```

```
plt.imshow(segmented_image)
```

```
plt.title('Gambar Hasil Segmentasi')
```

```
plt.show()
```

```
'''
```