**LAPORAN UAS PENGOLAHAN CITRA**

Disusun Oleh:

Rahmat Hidayat - 312210565

Satria Dwi Aprianto - 312210490

Syahril Haryanto - 312210668

Farhan Zulfahriansyah - 312210494

Robby Firmansyah - 312210643

Noufal Ariq Fadhurrahman - 312210526

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PELITA BANGSA**

**2022**

**BAB I**

**PENDAHULUAN**

Pada era digital saat ini, pengolahan gambar merupakan bagian yang penting dalam berbagai aplikasi teknologi seperti pengenalan objek, visi komputer, dan analisis data visual. Dalam laporan ini, kami akan menjelaskan proses pengolahan gambar menggunakan dua library utama, yaitu OpenCV (Open Source Computer Vision Library) dan Matplotlib. OpenCV digunakan untuk membaca, memanipulasi, dan mengolah gambar dengan berbagai operasi seperti konversi warna dan penyesuaian ukuran. Sementara itu, Matplotlib digunakan untuk menampilkan gambar-gambar yang telah diolah dalam bentuk visual yang dapat dipahami.

Langkah-langkah yang akan dijelaskan dalam laporan ini mencakup pembacaan beberapa gambar dari file, pengecekan keberhasilan pembacaan gambar, konversi warna dari BGR (format bawaan OpenCV) ke RGB (format umum untuk visualisasi), penyesuaian ukuran gambar agar memiliki dimensi yang seragam, penggabungan gambar secara horizontal, dan akhirnya menampilkan gambar hasil gabungan.

Melalui laporan ini, diharapkan pembaca dapat memahami teknik dasar dalam pengolahan gambar menggunakan Python serta aplikasi dari dua library yang kuat, yaitu OpenCV dan Matplotlib, dalam konteks pengolahan dan visualisasi gambar. Langkah-langkah yang dijelaskan akan memperlihatkan bagaimana integrasi antara kedua library ini dapat digunakan untuk mencapai tujuan pengolahan gambar secara efisien dan efektif.

**BAB II**

**LAPORAN**

1. Import Library

Pada langkah pertama, kita mengimpor library yang diperlukan:

import numpy as np

import matplotlib.pyplot as plt

import cv2

2. Membaca Gambar

image = cv2.imread(image\_path)

if image is None:

print(f"Error: Gambar di {image\_path} tidak ditemukan atau tidak bisa dibaca.")

return None, None

3. Ubah Warna ke RGB

image = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)

4. Membentuk Ulang Gambar Menjadi Susunan Piksel 2D dengan 3 Nilai Warna (RGB)

pixel\_vals = image.reshape((-1, 3))

5. Mengkonversikan ke Tipe Float

pixel\_vals = np.float32(pixel\_vals)

6. Melakukan K-means Clustering

retval, labels, centers = cv2.kmeans(pixel\_vals, k, None, criteria, 10, cv2.KMEANS\_RANDOM\_CENTERS)

7. Mengonversi Data Menjadi Nilai 8-bit

centers = np.uint8(centers)

segmented\_data = centers[labels.flatten()]

8. Membentuk Ulang Data Menjadi Dimensi Gambar Asli

segmented\_image = segmented\_data.reshape((image.shape))

return image, segmented\_image

9. Segmentasi Gambar

image\_path1 = 'images/Motor.jpg'

image\_path2 = 'images/Motor 2.jpg'

image\_path3 = 'images/Motor 3.jpg'

original\_image1, segmented\_image1 = segment\_image(image\_path1)

original\_image2, segmented\_image2 = segment\_image(image\_path2)

original\_image3, segmented\_image3 = segment\_image(image\_path3)

10. Menampilkan Gambar Asli dan Gambar Tersegmentasi

if all(img is not None for img in [original\_image1, segmented\_image1, original\_image2, segmented\_image2, original\_image3, segmented\_image3]):

plt.figure(figsize=(15, 15))

plt.subplot(3, 2, 1)

plt.imshow(original\_image1)

plt.title("Gambar Asli 1")

plt.subplot(3, 2, 2)

plt.imshow(segmented\_image1)

plt.title("Gambar Tersegmentasi 1")

plt.subplot(3, 2, 3)

plt.imshow(original\_image2)

plt.title("Gambar Asli 2")

plt.subplot(3, 2, 4)

plt.imshow(segmented\_image2)

plt.title("Gambar Tersegmentasi 2")

plt.subplot(3, 2, 5)

plt.imshow(original\_image3)

plt.title("Gambar Asli 3")

plt.subplot(3, 2, 6)

plt.imshow(segmented\_image3)

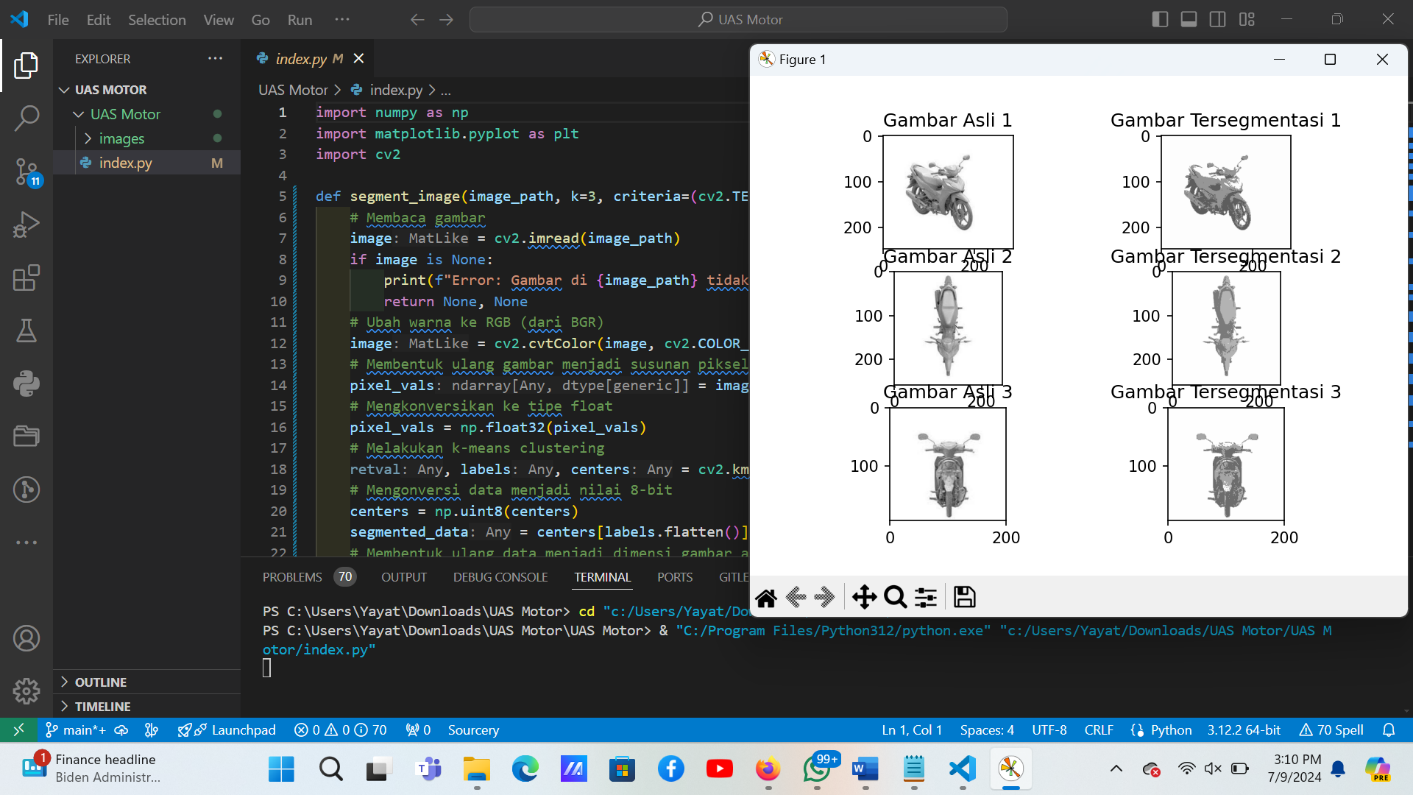
plt.title("Gambar Tersegmentasi 3")

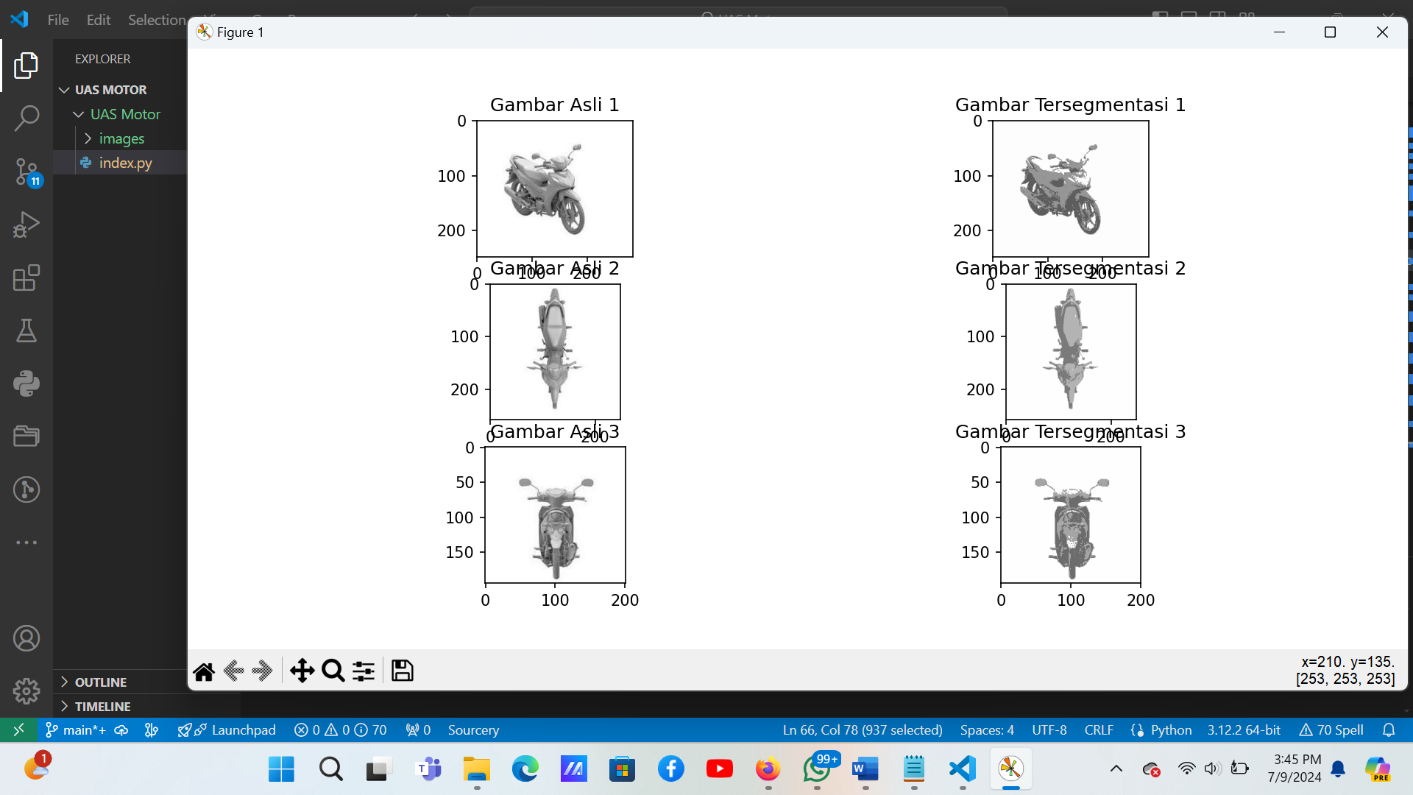
plt.show()

else:

print("Satu atau lebih gambar tidak ditemukan atau tidak bisa diproses.")

11. Hasil Output





**BAB III**

**KESIMPULAN**

Dalam laporan ini, telah dibahas proses pengolahan gambar menggunakan OpenCV dan Matplotlib dalam lingkungan Python. Langkah-langkah yang dilakukan meliputi pembacaan gambar dari file, konversi warna dari BGR ke RGB, penyesuaian ukuran gambar untuk memastikan dimensi yang seragam, penggabungan gambar secara horizontal, dan akhirnya menampilkan gambar gabungan.

Penggunaan OpenCV sebagai library utama untuk operasi pengolahan gambar seperti pembacaan, konversi warna, dan penyesuaian ukuran memberikan fleksibilitas dan kekuatan dalam manipulasi data gambar. OpenCV menyediakan fungsi-fungsi yang efisien untuk menangani berbagai aspek pengolahan gambar, termasuk manipulasi intensif seperti resizing dan konversi warna.

Matplotlib, di sisi lain, digunakan untuk visualisasi hasil pengolahan gambar. Dengan Matplotlib, gambar-gambar yang telah diolah dengan OpenCV dapat ditampilkan secara intuitif dalam bentuk visual yang dapat dipahami oleh pengguna. Penyajian visual yang baik dari gambar-gambar ini penting dalam banyak aplikasi, seperti dalam riset ilmiah, presentasi data, dan aplikasi industri.

Penerapan kombinasi OpenCV dan Matplotlib dalam laporan ini menunjukkan integrasi yang kuat antara pengolahan data gambar dan visualisasi. Proses ini tidak hanya mengilustrasikan kemampuan teknis dari kedua library tersebut, tetapi juga memberikan gambaran yang jelas tentang bagaimana teknologi ini dapat digunakan dalam berbagai konteks aplikatif, dari riset akademis hingga pengembangan produk komersial.

Diharapkan laporan ini dapat memberikan pemahaman yang lebih baik tentang penggunaan OpenCV dan Matplotlib dalam pengolahan dan visualisasi gambar, serta mendorong eksplorasi lebih lanjut terhadap kemungkinan-kemungkinan baru dalam analisis visual dan pengolahan data gambar secara umum. Kedua library ini memberikan fondasi yang kokoh untuk pengembangan solusi-solusi inovatif dalam bidang pengolahan citra dan visi komputer.