|  |  |  |
| --- | --- | --- |
|  | **Lembar Kerja Mahasiswa**  **Mata Kuliah Pengolahan Citra Digital Praktik (203311-20)**  **Program Studi Informatika**  **Fakultas Sains & Teknologi – Universitas Teknologi Yogyakarta** | |
|  | **Identitas Mahasiswa** | |
| **Nama** | **Arieska Restu Harpian Dwika** |
| **NPM** | **5200411488** |
| **Kelompok Prak** | **Kel. I** |
|  | | |
| **Soal 1.** | | |
| Berdasarkan demo di kelas, lakukan modifikasi pada sintaks segmentasi koin agar mampu menlakukan segmentasi (memberi lingkaran/kotak hijau) pada objek koin dari citra czech\_coin.jpg di bawah ini (citra di-download di elearning):    Pada tugas kali ini Anda tidak perlu membuat GUI seperti biasa, tetapi pastikan munculkan gambar untuk setiap tahapan yang diterapkan pada citra di atas dalam rangka melakukan segmentasi. Anda diperbolehkan menggunakan semua metode yang sudah dipelajari sebelumnya di mata kuliah ini. | | |
| **Hasil Script** | | |
| **//tuliskan script python Anda di sini**  import numpy as np  import cv2  import matplotlib.pyplot as plt  image = cv2.imread("12th Meet (Monday, December, 27th 2021)\Task\Resource\czech\_coin.jpg")  roi = image[0:500, 0:500]  imgGray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  # Image blur : Gaussian blur  k = 15  imgBlur = cv2.GaussianBlur(imgGray, (k, k), 0)  # Threshold image  thresh = cv2.adaptiveThreshold(imgBlur, 255, cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C, cv2.THRESH\_BINARY\_INV, 11, 1)  # kernel = np.ones((3, 3), np.uint8)  kernel = cv2.getStructuringElement(cv2.MORPH\_ELLIPSE, (3,3))  # Morphology  imgErode = cv2.erode(thresh, kernel, *iterations*=2)  imgClosing = cv2.morphologyEx(imgErode, cv2.MORPH\_CLOSE, kernel, *iterations*=6)  imgDilate = cv2.dilate(imgClosing, kernel, *iterations*=2)  contImg = imgDilate.copy()  contours, hierarchy = cv2.findContours(contImg, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)  for cnt in contours:      area = cv2.contourArea(cnt)      if area < 100:          continue      if len(cnt) < 0:          continue      ellipse = cv2.fitEllipse(cnt)      cv2.ellipse(roi, ellipse, (0,255,0), 2)  # Resize image  imgGray = cv2.resize(imgGray, (0,0), None, 0.5, 0.5)  imgBlur = cv2.resize(imgBlur, (0,0), None, 0.5, 0.5)  thresh = cv2.resize(thresh, (0,0), None, 0.5, 0.5)  imgErode = cv2.resize(imgErode, (0,0), None, 0.5, 0.5)  imgClosing = cv2.resize(imgClosing, (0,0), None, 0.5, 0.5)  imgDilate = cv2.resize(imgDilate, (0,0), None, 0.5, 0.5)  # Stacking image  hStack1 = np.hstack([imgGray, imgBlur, thresh])  hStack2 = np.hstack([imgErode, imgClosing, imgDilate])  vStack = np.vstack([hStack1, hStack2])  cv2.imshow('Stages', vStack)  cv2.imshow('Result', roi)  cv2.waitKey(0) | | |
| **Hasil *capture* citra untuk setiap tahapan pengolahan citra yang diterapkan** | | |
| **//paste-kan seluruh tahapan pengolahan citra yang anda lakukan (tidak hanya gambar hasil akhir saja)** | | |