



 **itenas**
Institut Teknologi Nasional

VISUAL INNOVATIONS:

**MODUL INOVATIF PEMBELAJARAN
PEMROGRAMAN COMPUTER VISION**

MODUL ALGORITMA WATERSHED

2023

**PROGRAM STUDI INFOMATIKA
INSTITUT TEKNOLOGI NASIONAL
BANDUNG**

By: Irma Amelia Dewi



H. REGION-BASED SEGMENTATION

Judul PRAKTEK H2- ALGORITMA WATERSHED

Deskripsi

Algoritma watershed adalah algoritma yang digunakan untuk segmentasi dan sangat berguna saat mengekstraksi objek yang bersentuhan atau tumpang tindih dalam gambar, seperti gambar koin. Dengan menggunakan metode pemrosesan gambar tradisional seperti threshold dan deteksi kontur, tidak akan dapat mengekstrak setiap koin dari gambar — namun dengan memanfaatkan algoritma watershed, dapat mendeteksi dan mengekstrak setiap koin.

Library

OpenCV2

merupakan *library computer vision* yang dapat digunakan sebagai *library* dalam *digital image processing*

Numpy

merupakan library yang melakukan berbagai operasi matematika pada array

Matplotlib

library Python yang digunakan untuk plotting dan memvisualisasikan data

Estimasi waktu

10 menit

Prerequisite

1. Membuka notebook google colab
2. Menyediakan sebuah citra koin-koin yang berdekatan dan sudah di upload ke dalam penyimpanan sesi google colab



Alur Proses

1. Import library yang dibutuhkan

2. Baca file gambar yang akan di load ke sistem
3. Melakukan operasi Opening, lalu Dilatasi mencari dan kemudian mencari wilayah foreground
4. Memberi penanda dan label pada wilayah atau daerah yang ada di dalamnya.
5. Menerapkan algoritma *Watershed*

Listing program Pada halaman *colab* dapat diketikkan coding untuk melakukan algoritma **Watershed**

```
1. import numpy as np
2. import cv2 as cv
3. from matplotlib import pyplot as plt
4. from google.colab.patches import cv2_imshow
5.
6. img = cv.imread('water_coins.jpg')
7.
8. gray = cv.cvtColor(img,cv.COLOR_BGR2GRAY)
9. ret, thresh = cv.threshold(gray,0,255,cv.THRESH_BINARY_INV+cv.THRESH_OTSU)
10. cv2_imshow(thresh)
11.
12. # noise removal
13. kernel = np.ones((3,3),np.uint8)
14. opening = cv.morphologyEx(thresh,cv.MORPH_OPEN,kernel, iterations = 2)
15.
16. # sure background area
17. sure_bg = cv.dilate(opening,kernel,iterations=3)
18. cv2_imshow(sure_bg)
19.
20. # Finding sure foreground area
21. dist_transform = cv.distanceTransform(opening,cv.DIST_L2,5)
22. ret, sure_fg = cv.threshold(dist_transform,0.7*dist_transform.max(),255,0)
23. cv2_imshow(sure_fg)
24.
25. # Finding unknown region
26. sure_fg = np.uint8(sure_fg)
27. unknown = cv.subtract(sure_bg,sure_fg)
28.
29. # Marker labelling
30. ret, markers = cv.connectedComponents(sure_fg)
31.
32. # Add one to all labels so that sure background is not 0, but 1
33. markers = markers+1
34.
35. # Now, mark the region of unknown with zero
36. markers[unknown==255] = 0
37.
38. markers = cv.watershed(img,markers)
39. img[markers == -1] = [255,0,0]
40.
41. cv2_imshow(markers)
42. cv2_imshow(img)
43.
```