

## BAB IX

### MORFOLOGI

#### Materi:

- Definisi dan Fungsi Morfologi
- Dilasi
- Erosi
- *Opening*
- *Closing*

#### Tujuan Praktikum:

- Mahasiswa dapat mengimplementasikan operasi-operasi morfologi di OpenCV-Python.

#### A. Penyajian

##### 1. Definisi dan Fungsi Morfologi

Kata morfologi biasa Kita dengar di cabang biologi yaitu yang berhubungan dengan bentuk dan struktur hewan dan tumbuhan. Sama halnya dalam konteks morfologi matematis yaitu sebagai alat untuk mengekstraksi komponen gambar yang berguna dalam representasi dan deskripsi bentuk wilayah, seperti batas dan kerangka. Teknik morfologi juga berguna untuk *pre-processing* dan *post-processing* citra, seperti penipisan dan pemangkasan, misalnya untuk memperbaiki hasil segmentasi [1]. Morfologi adalah proses mengidentifikasi bentuk dengan basis region (citra bertipe biner & *grayscale*). Adapun operasi-operasi morfologi yang sering digunakan ialah dilasi, erosi, *opening*, dan *closing*.

##### 2. Dilasi

Merupakan proses penggabungan titik-latar 0 menjadi bagian dari objek 1 berdasarkan *structuring element* (SE) yang digunakan. Notasi dilasi dapat ditulis berdasarkan persamaan di bawah ini.

$$D(A, S) = A \oplus S$$

Keterangan:

$D(A, S)$  : Fungsi dilasi

$A$  : Citra

$S$  : *Structure element*

$\oplus$  : Simbol dilasi

Fungsi Dilasi pada OpenCV [2]:

```
dst = cv.dilate(src, kernel, anchor, iterations, borderType, borderValue)
```

Keterangan:

src : input image; jumlah channel bisa berapa saja, tetapi kedalamannya harus salah satu dari CV\_8U, CV\_16U, CV\_16S, CV\_32F or CV\_64F.

dst : output image dengan ukuran dan tipe yang sama dengan src.

kernel : *structuring element* yang digunakan untuk dilasi; jika element = Mat (), *structuring element* persegi 3 x 3 digunakan. Kernel dapat dibuat menggunakan `getStructureElement` ([https://docs.opencv.org/4.1.2/d4/d86/group\\_imgproc\\_filter.html](https://docs.opencv.org/4.1.2/d4/d86/group_imgproc_filter.html))

anchor : posisi jangkar di dalam elemen; nilai default (-1, -1) berarti bahwa jangkar berada di pusat elemen.

iterations : number of times dilation is applied.

borderType : metode ekstrapolasi piksel, lihat `BorderTypes` ([https://docs.opencv.org/4.1.2/d2/de8/group\\_core\\_array.html](https://docs.opencv.org/4.1.2/d2/de8/group_core_array.html))

borderValue: *border value* untuk nilai border yang konstan

### 3. Erosi

Merupakan proses penghapusan titik-titik objek (1) menjadi bagian dari latar (0), berdasarkan *structuring element* (SE) yang digunakan. Notasi erosi dapat ditulis berdasarkan persamaan di bawah ini.

$$E(A, S) = A \otimes S$$

Keterangan:

$E(A, S)$  : Fungsi erosi

A : Citra

S : *Structure element*

$\otimes$  : Simbol erosi

Fungsi Erosi pada OpenCV [2]:

```
Dst = cv.erode(src, kernel, anchor, iterations, borderType, borderValue)
```

Keterangan:

src : input image; jumlah channel bisa berapa saja, tetapi kedalamannya harus salah satu dari CV\_8U, CV\_16U, CV\_16S, CV\_32F or CV\_64F.

dst : output image dengan ukuran dan tipe yang sama dengan src.

kernel : *structuring element* yang digunakan untuk dilasi; jika element = Mat (), *structuring element* persegi 3 x 3 digunakan. Kernel dapat dibuat menggunakan `getStructureElement` ([https://docs.opencv.org/4.1.2/d4/d86/group\\_imgproc\\_filter.html](https://docs.opencv.org/4.1.2/d4/d86/group_imgproc_filter.html))

anchor : posisi jangkar di dalam elemen; nilai default (-1, -1) berarti bahwa jangkar berada di pusat elemen.

iterations : number of times dilation is applied.

borderType: metode ekstrapolasi piksel, lihat `BorderTypes` ([https://docs.opencv.org/4.1.2/d2/de8/group\\_core\\_array.html](https://docs.opencv.org/4.1.2/d2/de8/group_core_array.html))

borderValue: *border value* untuk nilai border yang konstan

#### 4. Opening dan Closing

*Opening* adalah proses erosi yang diikuti dengan dilasi. Efek yang dihasilkan adalah menghilangnya objek-objek kecil dan kurus, memecah objek pada titik-titik yang kurus, dan secara umum menghaluskan batas objek besar tanpa mengubah area objek secara signifikan. *Opening* berguna untuk **menghaluskan citra, menghilangkan tonjolan yang tipis, dan memisahkan objek**. Notasi *opening* dapat ditulis berdasarkan persamaan di bawah ini.

$$A \circ S = (A \otimes S) \oplus S$$

*Closing* adalah proses dilasi yang diikuti dengan erosi. Efek yang dihasilkan adalah mengisi lubang kecil pada objek, menggabungkan objek-objek yang berdekatan, dan secara umum menghaluskan batas dari objek besar tanpa mengubah area objek secara signifikan. Teknik ini berguna untuk menghaluskan citra, menghilangkan lubang yang kecil. Notasi *closing* dapat ditulis berdasarkan persamaan di bawah ini.

$$A \circ S = (A \oplus S) \otimes S$$

Fungsi Opening pada OpenCV [2]:

`dst = cv.morphologyEx(src, op, kernel, anchor, iterations, borderType, borderValue)`

Keterangan:

src : input image; jumlah channel bisa berapa saja, tetapi kedalamannya harus salah

satu  
dari CV\_8U, CV\_16U, CV\_16S, CV\_32F or CV\_64F.

dst : output image dengan ukuran dan tipe yang sama dengan src.

op : tipe dari operasi morphological, untuk closing: cv.MORPH\_CLOSE dan untuk opening: cv.MORPH\_OPEN, lihat **MorphTypes**  
([https://docs.opencv.org/4.1.2/d4/d86/group\\_imgproc\\_filter.html](https://docs.opencv.org/4.1.2/d4/d86/group_imgproc_filter.html))

kernel : *structuring element* yang digunakan untuk dilasi; jika element = Mat (), *structuring element* persegi 3 x 3 digunakan. Kernel dapat dibuat menggunakan `getStructureElement`  
([https://docs.opencv.org/4.1.2/d4/d86/group\\_imgproc\\_filter.html](https://docs.opencv.org/4.1.2/d4/d86/group_imgproc_filter.html))

anchor : posisi jangkar di dalam elemen; nilai default (-1, -1) berarti bahwa jangkar berada di pusat elemen.

iterations : number of times dilation is applied.

borderType: metode ekstrapolasi piksel, lihat **BorderTypes**  
([https://docs.opencv.org/4.1.2/d2/de8/group\\_core\\_array.html](https://docs.opencv.org/4.1.2/d2/de8/group_core_array.html))

borderValue: *border value* untuk nilai border yang konstan

### Contoh Structuring Element

```
# Rectangular Kernel
>>> cv.getStructuringElement(cv.MORPH_RECT,(5,5))
array([[1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1]], dtype=uint8)
# Elliptical Kernel
>>> cv.getStructuringElement(cv.MORPH_ELLIPSE,(5,5))
array([[0, 0, 1, 0, 0],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [1, 1, 1, 1, 1],
       [0, 0, 1, 0, 0]], dtype=uint8)
# Cross-shaped Kernel
>>> cv.getStructuringElement(cv.MORPH_CROSS,(5,5))
array([[0, 0, 1, 0, 0],
       [0, 0, 1, 0, 0],
       [1, 1, 1, 1, 1],
       [0, 0, 1, 0, 0],
       [0, 0, 1, 0, 0]], dtype=uint8)
```

## B. Latihan

1. Berikut ini merupakan latihan membuat program yang dapat melakukan dilasi, erosi, *opening*, *closing*.

### Morfologi.py

```
import numpy as np
import cv2 as cv

def morfologi(img, se):
    dst_erosi = cv.erode(img, se, iterations = 1)
    dst_dilate = cv.dilate(img, se, iterations = 1)
    dst_opening = cv.morphologyEx(img, cv.MORPH_OPEN, se, iterations = 1)
    dst_closing = cv.morphologyEx(img, cv.MORPH_CLOSE, se, iterations = 1)
    return dst_erosi, dst_dilate, dst_opening, dst_closing

def main():
    img1 = cv.imread('circbw.tif',0)
    s_element = cv.getStructuringElement(cv.MORPH_RECT,(3,3))
    dst_erosi, dst_dilate, dst_opening, dst_closing = morfologi(img1, s_element)

    cv.imshow('dst_erosi',dst_erosi)
    cv.imshow('dst_dilate',dst_dilate)
    cv.imshow('dst_opening',dst_opening)
    cv.imshow('dst_closing',dst_closing)
    cv.waitKey(0)
    cv.destroyAllWindows()
```

2. Ingat kembali pada latihan di Modul 3: Ruang Warna Pengolahan Citra, dalam latihan tersebut kita mencoba mensegmentasi objek apel warna merah. Berikut ini adalah kode programnya:

```
import numpy as np
import cv2 as cv

#membaca citra RGB
rgb = cv.imread('red_apple.jpg')
ycrCb = cv.cvtColor(rgb, cv.COLOR_BGR2YCrCb)

lower_blue = np.array([40,150,50])
upper_blue = np.array([200,255,150])
# Threshold the HSV image to get only blue colors
mask = cv.inRange(ycrCb, lower_blue, upper_blue)

res = cv.bitwise_and(rgb,rgb, mask= mask)

cv.imshow('src', rgb)
cv.imshow('rgb', mask)
cv.imshow('objek', res)
cv.waitKey(0)
cv.destroyAllWindows()
```

Output:



**Gambar 1. Citra Output Hasil Segmentasi tanpa Dilakukan Morfologi**

Dari hasil menggunakan segmentasi masih terlihat belum sempurna, yaitu masih ada noise pada latarbelakang dan bentuk apel yang masih ada lubang kosong di dalamnya. Hasil segmentasi tersebut dapat diperbaiki dengan morfologi sebagai berikut.

```
import numpy as np
import cv2 as cv

#membaca citra RGB
rgb = cv.imread('red_apple.jpg')
yrcrb = cv.cvtColor(rgb, cv.COLOR_BGR2YCrCb)

lower_blue = np.array([40,150,50])
upper_blue = np.array([200,255,150])
# Threshold the HSV image to get only blue colors
img_biner = cv.inRange(yrcrb, lower_blue, upper_blue)

#mendefinisikan structuring element 3x3 dan 5x5
se_3 = cv.getStructuringElement(cv.MORPH_RECT,(3,3))
se_5 = cv.getStructuringElement(cv.MORPH_RECT,(5,5))

#operasi morfologi
dst_dilate = cv.dilate(img_biner, se_3, iterations = 1)
dst_erosi = cv.erode(dst_dilate, se_3, iterations = 2)
dst_dilate2 = cv.dilate(dst_erosi, se_5, iterations = 2)
dst_erosi2 = cv.erode(dst_dilate2, se_5, iterations = 3)
dst_dilate3 = cv.dilate(dst_erosi2, se_3, iterations = 1)

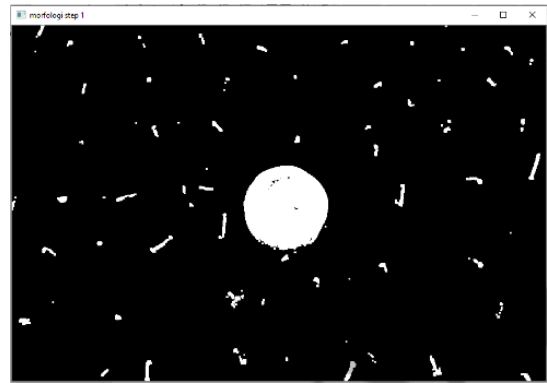
res = cv.bitwise_and(rgb,rgb, mask= dst_dilate3)

cv.imshow('src', rgb)
cv.imshow('rgb', dst_dilate3)
cv.imshow('objek', res)
cv.waitKey(0)
cv.destroyAllWindows()
```

Output:



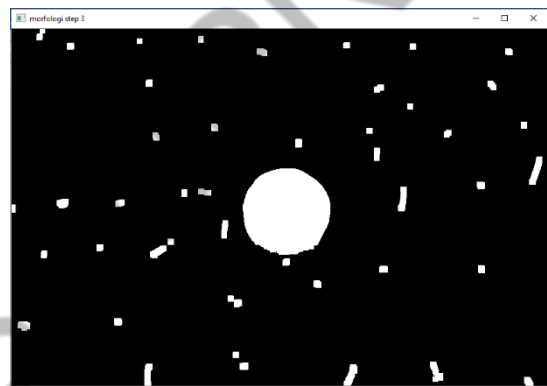
Citra Awal



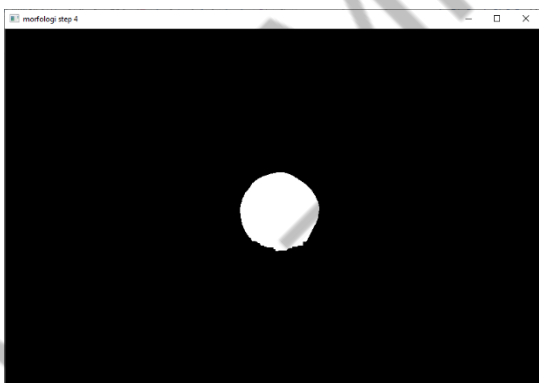
Hasil Dilasi dengan SE 3x3 satu kali iterasi



Hasil Erosi dengan SE 3x3 dua kali iterasi



Hasil Dilasi dengan SE 5x5 dua kali iterasi



Hasil Erosi dengan SE 5x5 tiga kali iterasi



Citra Hasil

**Gambar 2. Citra Output Latihan No.2**



### C. Lembar Kerja Praktikum

1. Burung jalak bali merupakan satu-satunya satwa endemik Pulau Bali yang masih tersisa. Sebagai mahasiswa informatika yang memiliki jiwa bela Negara, ingin berkontribusi terhadap kelestarian burung tersebut di Indonesia. Maka akan dibuat aplikasi berbasis pengolahan citra digital untuk mendeteksi jumlah burung jalak bali yang ada di suatu wilayah. Program membaca gambar, dan keluarannya adalah jumlah burung jalak bali yang ada di foto tersebut. Gunakan Teknik image enhancement, image restoration, segmentasi citra, post-processing dengan morfologi, dan hitung jumlah burung yang ada di gambar tersebut.



**Gambar 3. Foto Burung Jalak Bali**

### Referensi

- [1] R. C. Gonzalez and R. E. Woods, "Digital Image Processing, 4th Edition," *J. Electron. Imaging*, p. 1019, Jan. 2018.
- [2] A. Mordvintsev and A. K., "OpenCV-Python Tutorials," 2019. [https://docs.opencv.org/4.1.2/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/4.1.2/d6/d00/tutorial_py_root.html).