
TUTORIAL 4

Morphological Image Processing

Pengolahan Citra - Semester Gasal 2023/2024

1 Pendahuluan

Pengolahan citra secara morfologis berfokus pada bentuk dan struktur yang ada pada citra. Operasi dilakukan menggunakan *structural element* (*strel*), yakni suatu subcitra (matriks) untuk mencari (*probing*) bentuk/properti pada citra [1]. Bentuk *strel* bebas selama dapat direpresentasikan sebagai suatu citra biner (elemennya 1/0) dengan ukuran tertentu [1]. Berikut adalah contoh *strel* yang berbentuk lingkaran dan belah ketupat.

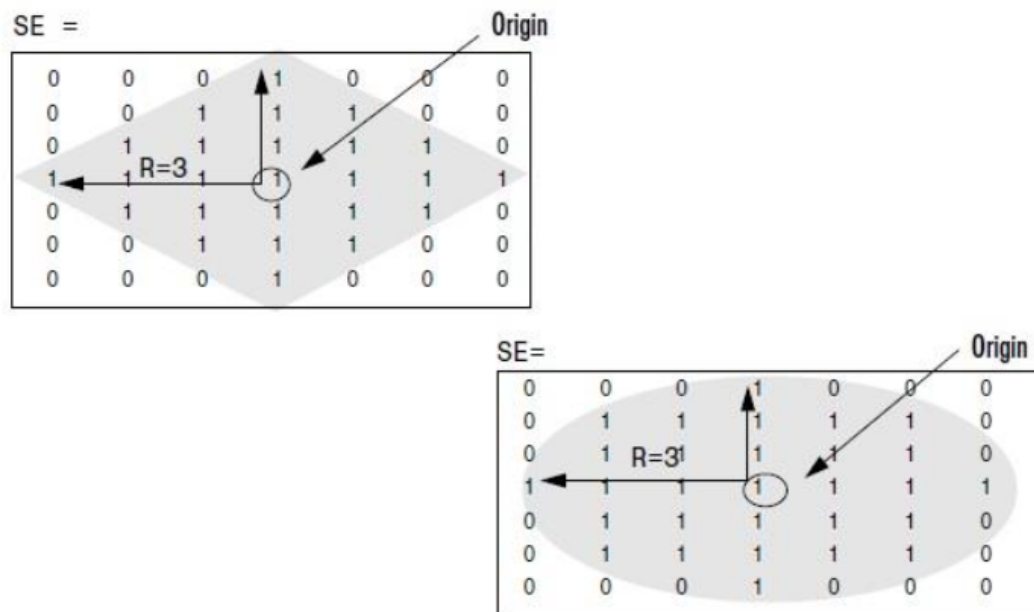


Figure 1: Contoh *Strel* ([sumber](#))

2 Operasi-Operasi Dasar

Kita akan membahas beberapa operasi dasar, yakni erosi, dilasi, *opening*, dan *closing*.

2.1 Erosi

Operasi erosi mengurangi piksel pada *boundaries* dari objek sehingga tampak lebih tipis. Berikut adalah contoh kode untuk melakukan erosi dan hasilnya. Perhatikan bahwa bentuk *strel* berpengaruh terhadap hasilnya.

```
1 # Import packages
2 from skimage import io, color
3 from skimage.morphology import disk, square, diamond, erosion
4 import matplotlib.pyplot as plt
5
6 # Baca citra dan lakukan erosi
7 octagon = color.rgb2gray(io.imread("octagon.jpg"))
8
```

```

9  # Anda bisa ubah bentuk strel, silakan baca dokumentasi
   skimage.morphology
10 strels = {
11     "Disk": disk(10),
12     "Diamond": diamond(10),
13     "Square": square(10)
14 }
15
16 plt.figure(figsize=(15,5))
17 plt.subplot(1,4,1)
18 plt.title('Original Octagon')
19 plt.imshow(octagon, cmap='gray')
20 plt.axis('off')
21
22 plot_counter = 1
23 for shape, strel in strels.items():
24     eroded_octagon = erosion(octagon, strel)
25     plt.subplot(1,4,1+plot_counter)
26     plt.title(f'Eroded Octagon with {shape} Strel')
27     plt.imshow(eroded_octagon, cmap='gray')
28     plt.axis('off')
29     plot_counter += 1
30 plt.show()

```

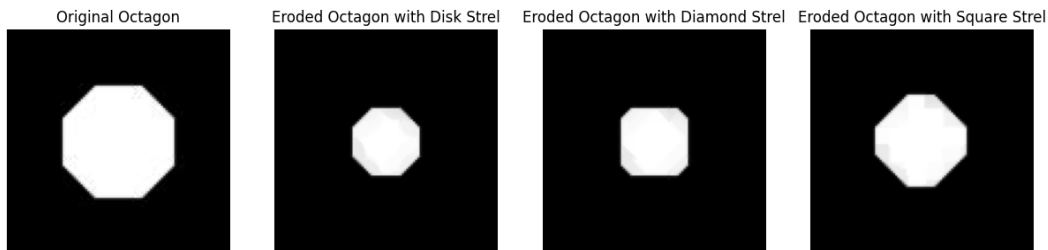


Figure 2: Perbandingan Bentuk *Strel* Pada Erosi

2.2 Dilasi

Berkebalikan dari erosi, operasi dilasi menambahkan piksel pada *boundaries* dari objek sehingga tampak lebih tebal. Berikut adalah contoh kode untuk melakukan dilasi dan hasilnya. Perhatikan bahwa bentuk *strel* berpengaruh terhadap hasilnya.

```

1  # Import packages
2  from skimage import io, color
3  from skimage.morphology import disk, square, diamond, dilation
4  import matplotlib.pyplot as plt
5

```

```

6  # Baca citra dan lakukan dilasi
7  octagon = color.rgb2gray(io.imread("octagon.jpg"))
8
9  # Anda bisa ubah bentuk strel, silakan baca dokumentasi
   skimage.morphology
10 strels = {
11     "Disk": disk(10),
12     "Diamond": diamond(10),
13     "Square": square(10)
14 }
15
16 plt.figure(figsize=(15,5))
17 plt.subplot(1,4,1)
18 plt.title('Original Octagon')
19 plt.imshow(octagon, cmap='gray')
20 plt.axis('off')
21
22 plot_counter = 1
23 for shape, strel in strels.items():
24     dilated_octagon = dilation(octagon, strel)
25     plt.subplot(1,4,1+plot_counter)
26     plt.title(f'Dilated Octagon with {shape} Strel')
27     plt.imshow(dilated_octagon, cmap='gray')
28     plt.axis('off')
29     plot_counter += 1
30 plt.show()

```

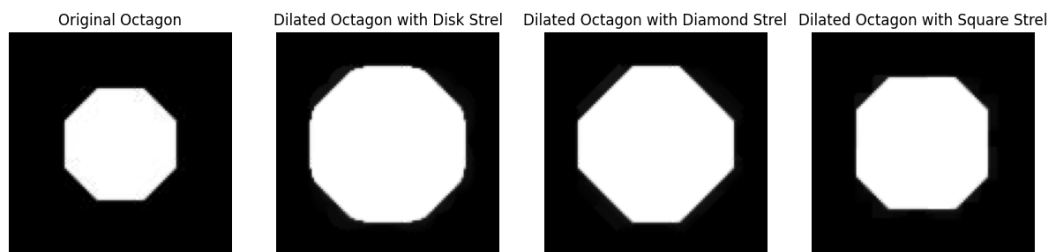


Figure 3: Perbandingan Bentuk *Strel* Pada Dilasi

2.3 Opening

Opening adalah operasi erosi diikuti dilasi menggunakan *strel* yang sama. Operasi ini digunakan untuk menghilangkan objek-objek kecil pada citra dengan tetap mempertahankan objek-objek yang lebih besar. Berikut adalah contoh kode beserta hasilnya.

```

1  # Import packages
2  from skimage import io, color

```

```

3 from skimage.morphology import disk, square, diamond, opening
4 import matplotlib.pyplot as plt
5
6 # Baca citra dan lakukan opening
7 hexagon = color.rgb2gray(io.imread('hexagon.jpg'))
8 strel = disk(3) # Bisa coba strel lain juga
9 opened_hexagon = opening(hexagon, strel)
10
11 plt.figure(figsize=(7,5))
12 plt.subplot(1,2,1)
13 plt.title('Original Hexagon')
14 plt.imshow(hexagon, cmap='gray')
15 plt.axis('off')
16 plt.subplot(1,2,2)
17 plt.title('Opening Operation Result')
18 plt.imshow(opened_hexagon, cmap='gray')
19 plt.axis('off')
20 plt.show()

```

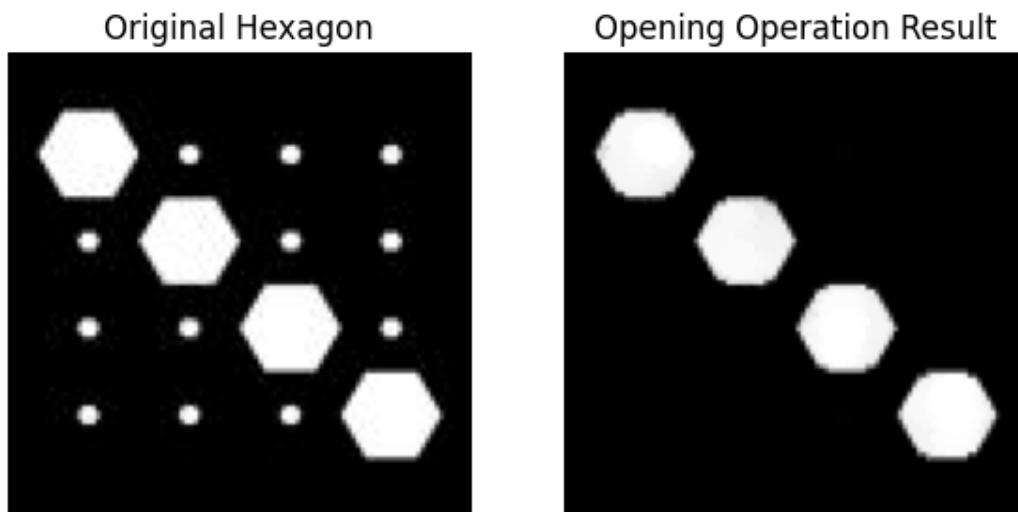


Figure 4: Hasil Operasi *Opening*

2.4 Closing

Closing adalah kebalikan dari *opening*, yakni operasi dilasi diikuti oleh erosi menggunakan *strel* yang sama. Operasi ini digunakan untuk mengisi *gap* antara piksel dari objek-objek yang ada pada citra. Berikut adalah contoh kode beserta hasilnya.

```

1 # Import packages
2 from skimage import io, color
3 from skimage.morphology import disk, square, diamond, closing

```

```

4 import matplotlib.pyplot as plt
5
6 # Baca citra dan lakukan closing
7 star = color.rgb2gray((io.imread('star.jpg')))
8 strel = disk(7)
9 closed_star = closing(star, strel)
10
11 plt.figure(figsize=(10,5))
12 plt.subplot(1,2,1)
13 plt.title('Original Star')
14 plt.imshow(star, cmap='gray')
15 plt.axis('off')
16 plt.subplot(1,2,2)
17 plt.title('Closing Operation Result')
18 plt.imshow(closed_star, cmap='gray')
19 plt.axis('off')
20 plt.show()

```

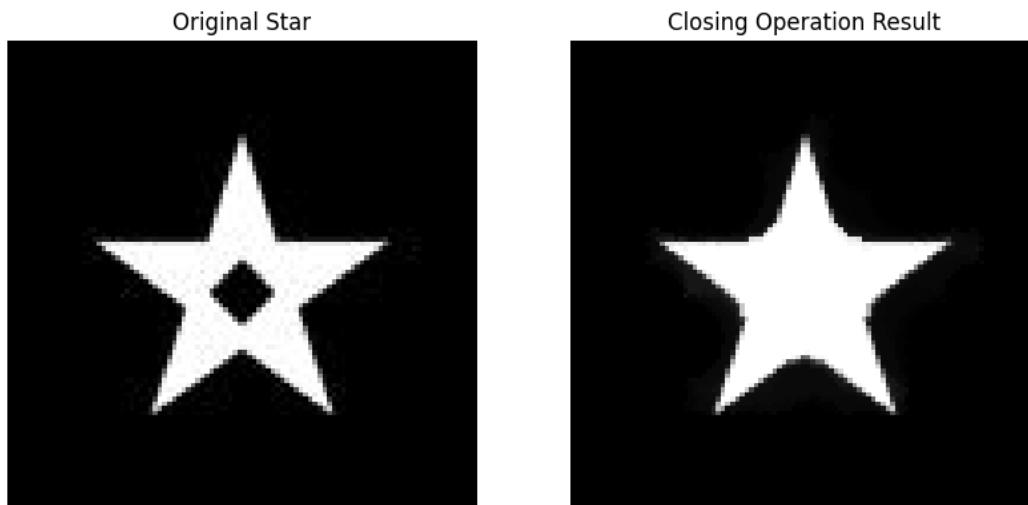


Figure 5: Hasil Operasi *Closing*

3 Operasi-Operasi Lain

3.1 *Top-Hat Filter*

Top-Hat Filter didefinisikan sebagai selisih antara citra asal dengan hasil *opening* dari citra tersebut. Operasi ini berguna untuk mengekstrak detail-detail terang pada latar yang gelap. Berikut adalah contoh kode dan hasilnya.

```

1 # Import packages
2 from skimage import io, color

```

```

3 from skimage.morphology import disk, square, diamond,
   white_tophat
4 import matplotlib.pyplot as plt
5
6 # Baca citra dan lakukan top-hat filter
7 cosmos = color.rgb2gray(io.imread('cosmos.jpg'))
8
9 strel = disk(20)
10 res = white_tophat(cosmos, strel)
11
12 # Show image
13 plt.figure(figsize=(20,10))
14 plt.subplot(1,2,1)
15 plt.title('Original Cosmos')
16 plt.imshow(cosmos, cmap='gray')
17 plt.axis('off')
18 plt.subplot(1,2,2)
19 plt.title('Top-Hat Filter Results')
20 plt.imshow(res, cmap='gray')
21 plt.axis('off')
22 plt.show()

```

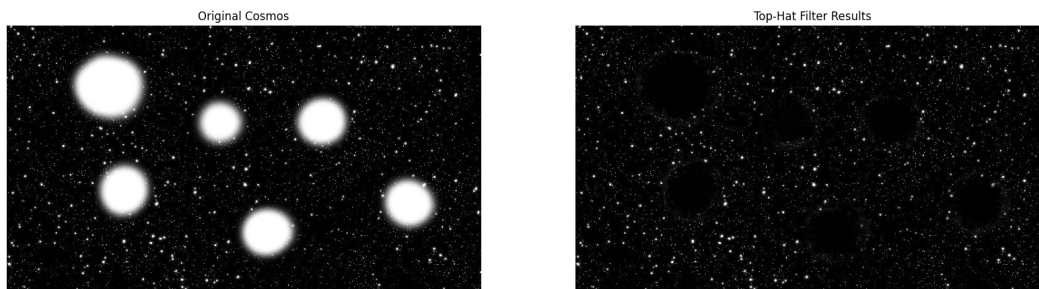


Figure 6: Hasil Operasi *Top-Hat Filter* ([sumber citra](#))

3.2 *Bottom-Hat Filter*

Bottom-Hat Filter didefinisikan sebagai selisih antara hasil *closing* dari suatu citra dengan citra asalnya. Tujuannya adalah untuk mengekstrak detail-detail gelap pada latar yang terang. Berikut adalah contoh kode dan hasilnya.

```

1 # Import packages
2 from skimage import io, color
3 from skimage.morphology import disk, square, diamond,
   black_tophat
4 import matplotlib.pyplot as plt
5
6 # Baca citra dan lakukan bottom-hat filter

```

```

7 cells = color.rgb2gray(io.imread('cells.jpg'))
8
9 strel = disk(20)
10 res = black_tophat(cells, strel)
11
12 # Show image
13 plt.figure(figsize=(20,10))
14 plt.subplot(1,2,1)
15 plt.title('Original Cells')
16 plt.imshow(cells, cmap='gray')
17 plt.axis('off')
18 plt.subplot(1,2,2)
19 plt.title('Bottom-Hat Filter Results')
20 plt.imshow(res, cmap='gray')
21 plt.axis('off')
22 plt.show()

```

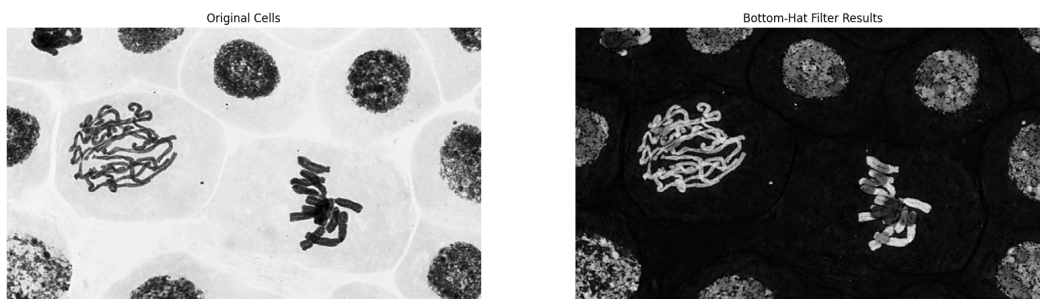


Figure 7: Hasil Operasi *Bottom-Hat Filter* ([sumber citra](#))

References

- [1] A. "Arymurthy and L. Rahadiani, "Image transform and mathematical morphology," pPT Pengcit.