23 - Citra Biner

IF4073 Pemrosesan Citra Digital

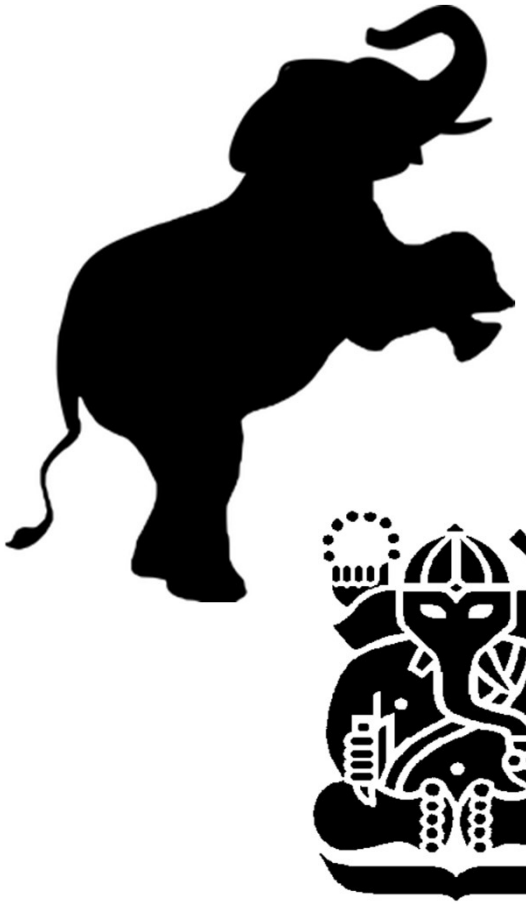
Oleh:

Dr. Ir. Rinaldi, M.T
Dr. Edward Ferdian



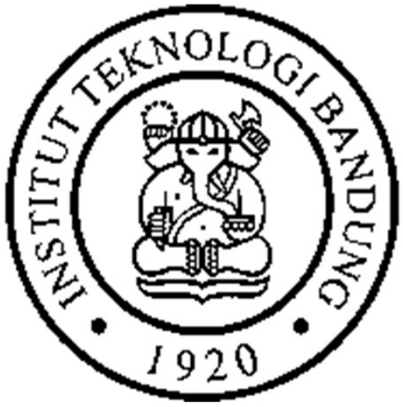
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2026

Citra Biner









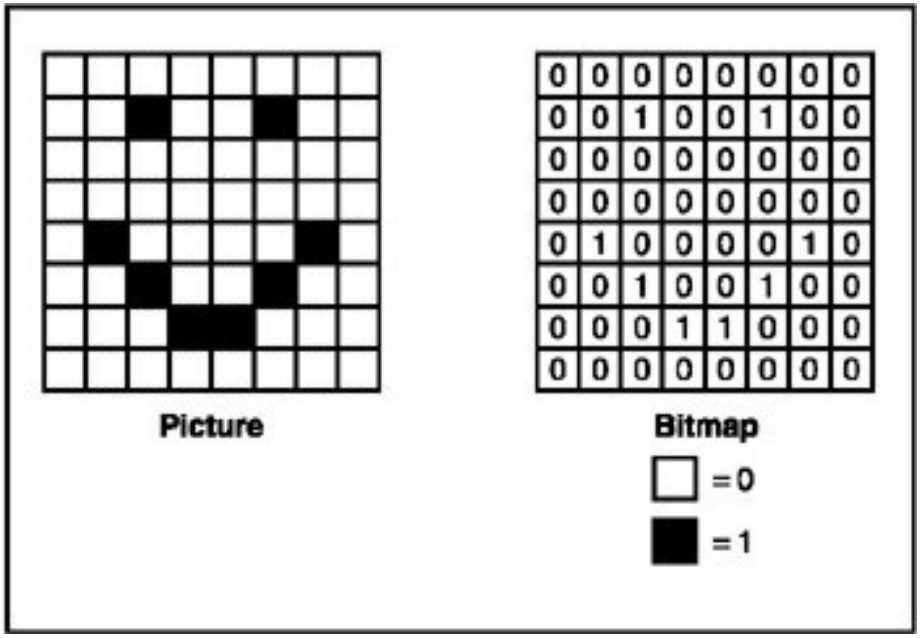


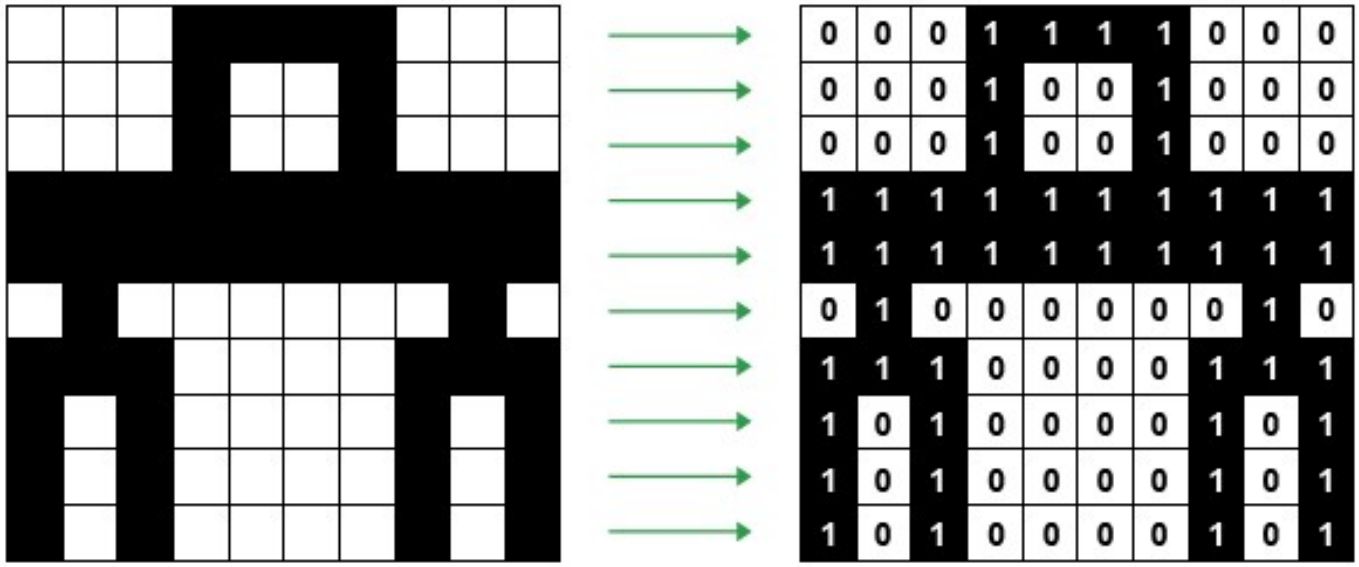

788679 912077



Citra Biner

- Citra biner adalah citra yang memiliki hanya dua nilai *graylevel*, 0 dan 1
- Hitam = 1, putih = 0, atau sebaliknya, bergantung konvensi yang dipakai



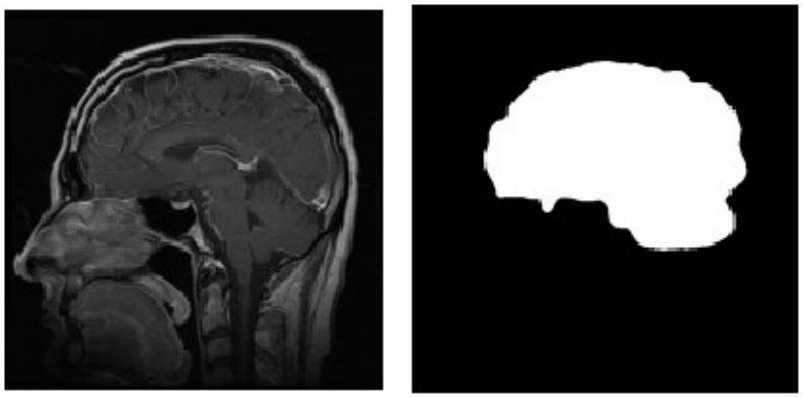

0 0 0 1 1 1 1 0 0 0
0 0 0 1 0 0 1 0 0 0
0 0 0 1 0 0 1 0 0 0
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
0 1 0 0 0 0 0 0 1 0
1 1 1 0 0 0 0 1 1 1
1 0 1 0 0 0 0 1 0 1
1 0 1 0 0 0 0 1 0 1
1 0 1 0 0 0 0 1 0 1

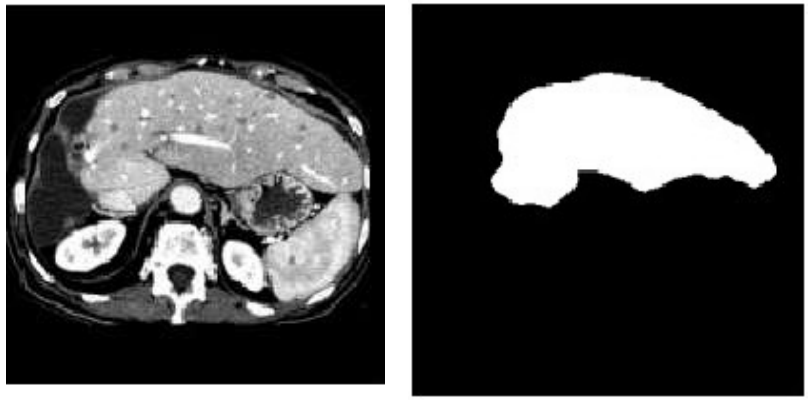
Mengapa perlu citra biner?

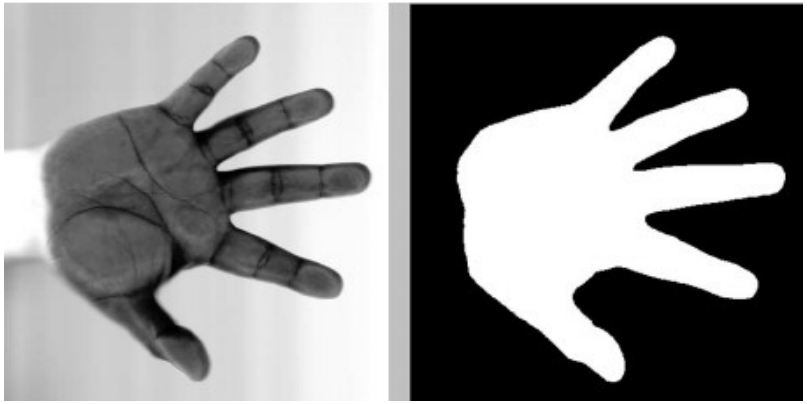
1. Kebutuhan memori untuk setiap pixel sedikit (hanya 1 bit/pixel)
2. Dapat menggunakan operasi logika (AND, OR, NOT) sehingga waktu komputasinya kecil
3. Untuk merepresentasikan citra hasil pendeteksian tepi (*edge images*)



4. Untuk memisahkan (segmentasi) objek dari gambar latar belakangnya → hasil operasi pengambangan



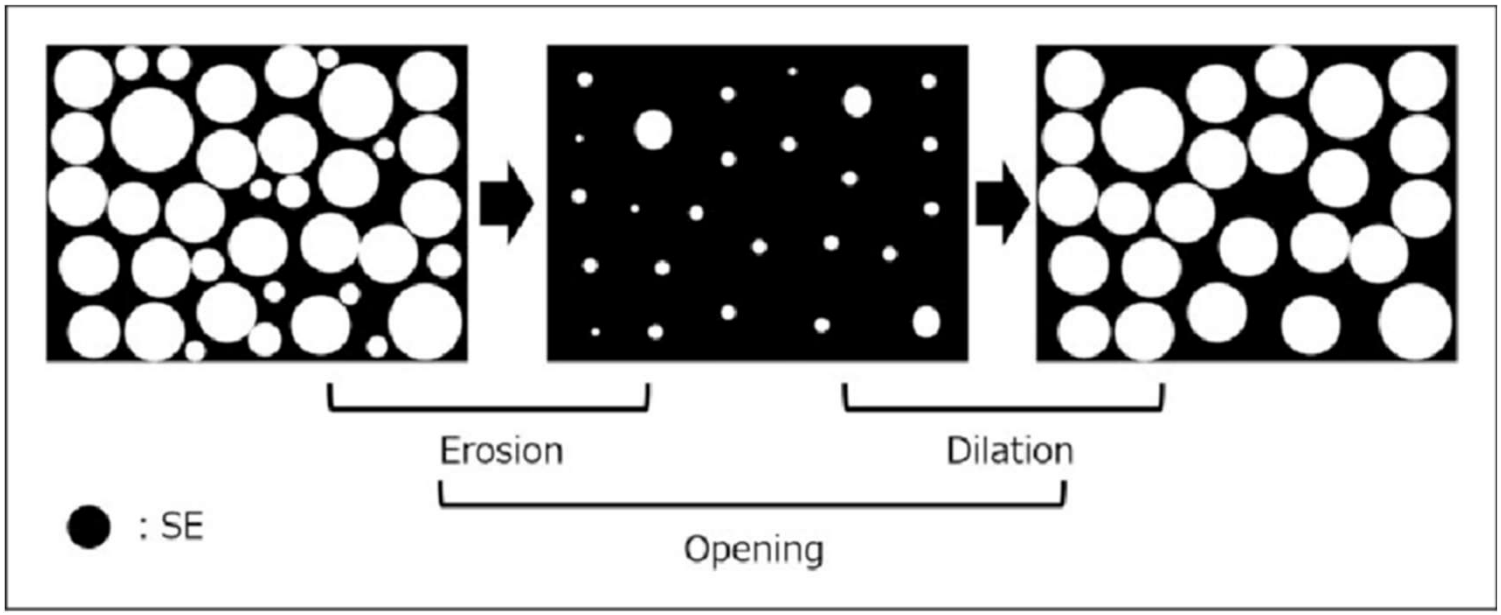








5. Untuk lebih memfokuskan pada analisis bentuk morfologi, yang dalam hal ini nilai *pixel* tidak terlalu penting dibandingkan bentuknya. Setelah objek dipisahkan dari latar belakangnya, properti geometri dan morfologi objek dapat dihitung dari citra biner



6. Untuk menampilkan citra pada piranti luaran yang hanya mempunyai resolusi intensitas satu bit, misalnya pencetak (*printer*).



Run-Length Encoding (RLE)

- RLE (atau RLC) merupakan metode pengkodean citra biner untuk menghasilkan representasi citra yang lebih kompak.

- Dua pendekatan yang digunakan dalam penerapan *RLE* pada citra biner:
  1. Posisi awal kelompok nilai 1 dan panjangnya (*length of runs*)
  2. Panjang *run*, dimulai dengan panjang *run* 1.

- Contoh: Misalkan citra biner adalah sebagai berikut:

1	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0	1	1	0	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Hasil pengkodean dengan metode RLE:

(1) pendekatan pertama:

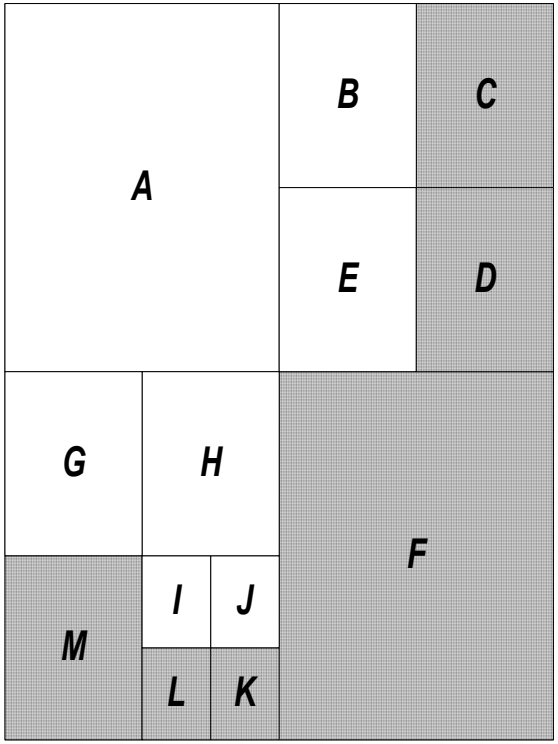
(1, 3) (7, 2) (12, 4) (17, 2) (20, 3)
(5, 13) (19, 4)
(1, 3) (17, 6)

(2) pendekatan kedua

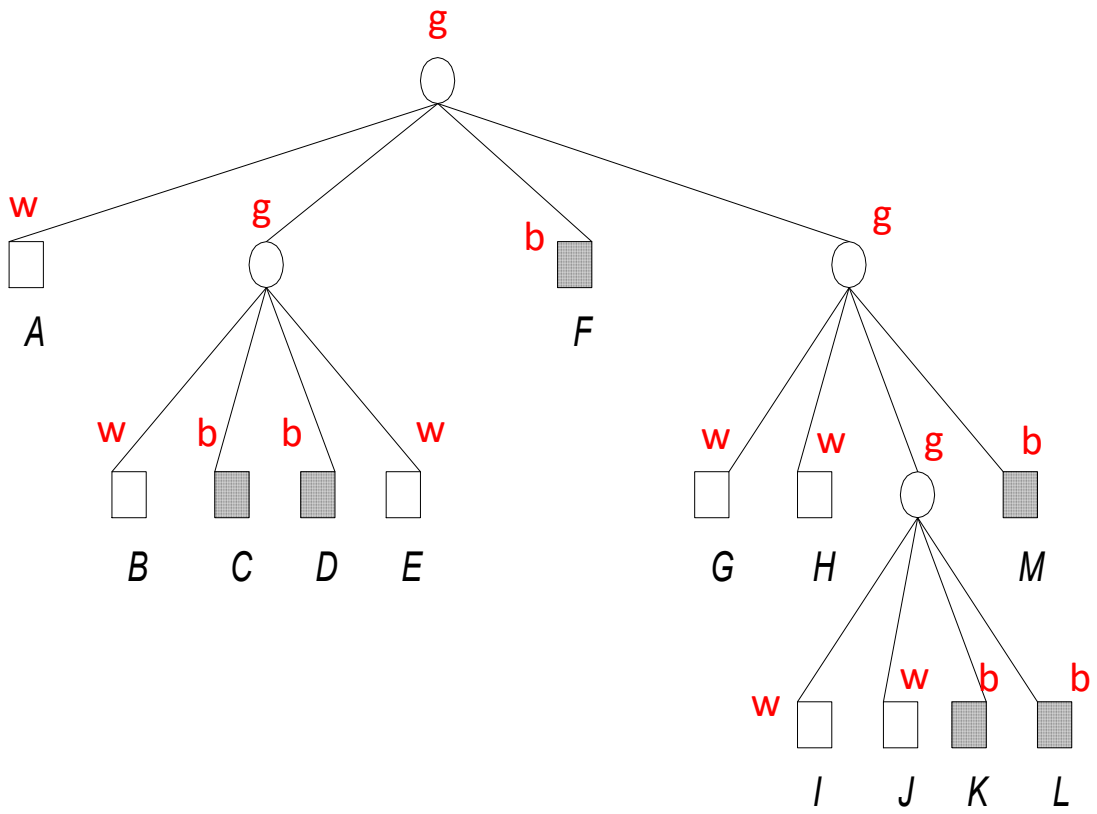
3, 3, 2, 3, 4, 1, 2, 1, 3
0, 4, 13, 1, 4
3, 13, 6

Pohon empatan (*quadtree*)

- Pohon empatan merupakan representasi wilayah (*region*) di dalam citra biner.
- Setiap simpul di dalam pohon-empatan merupakan salah satu dari tiga kategori: putih, hitam, dan abu-abu.
- Algoritma pohon-empatan:
  1. Bagi citra secara rekursif menjadi empat buah upa-wilayah yang berukuran sama.
  2. Untuk setiap upa-wilayah, bila *pixel-pixel* di dalam wilayah tersebut semuanya hitam atau semuanya putih, maka proses pembagian dihentikan.
  3. Sebaliknya, bila *pixel-pixel* di dalam upa-wilayah mengandung baik *pixel* hitam maupun *pixel* putih (kategori abu-abu), maka upa-wilayah tersebut dibagi lagi mejadi empat bagian.
  4. Demikian seterusnya sampai diperoleh upa-wilayah yang semua *pixel*-nya hitam atau semua *pixel*-nya putih



Citra biner

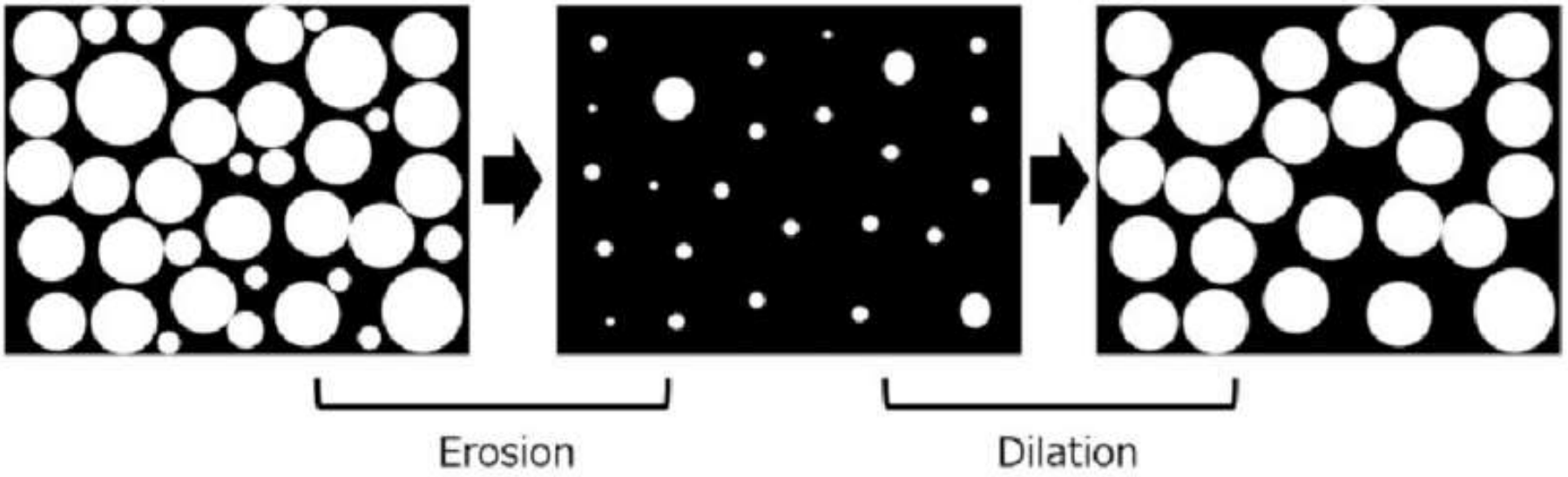


Kode: gwgwbbwbgwwgwwgwwbbb
Di-*decode* sebagai: *g*(*wg*(*wbbw*)*bg*(*wwg*(*wwbb*)*b*))
Keterangan: *b* = *black*, *w* = *white*, *g* = *gray*

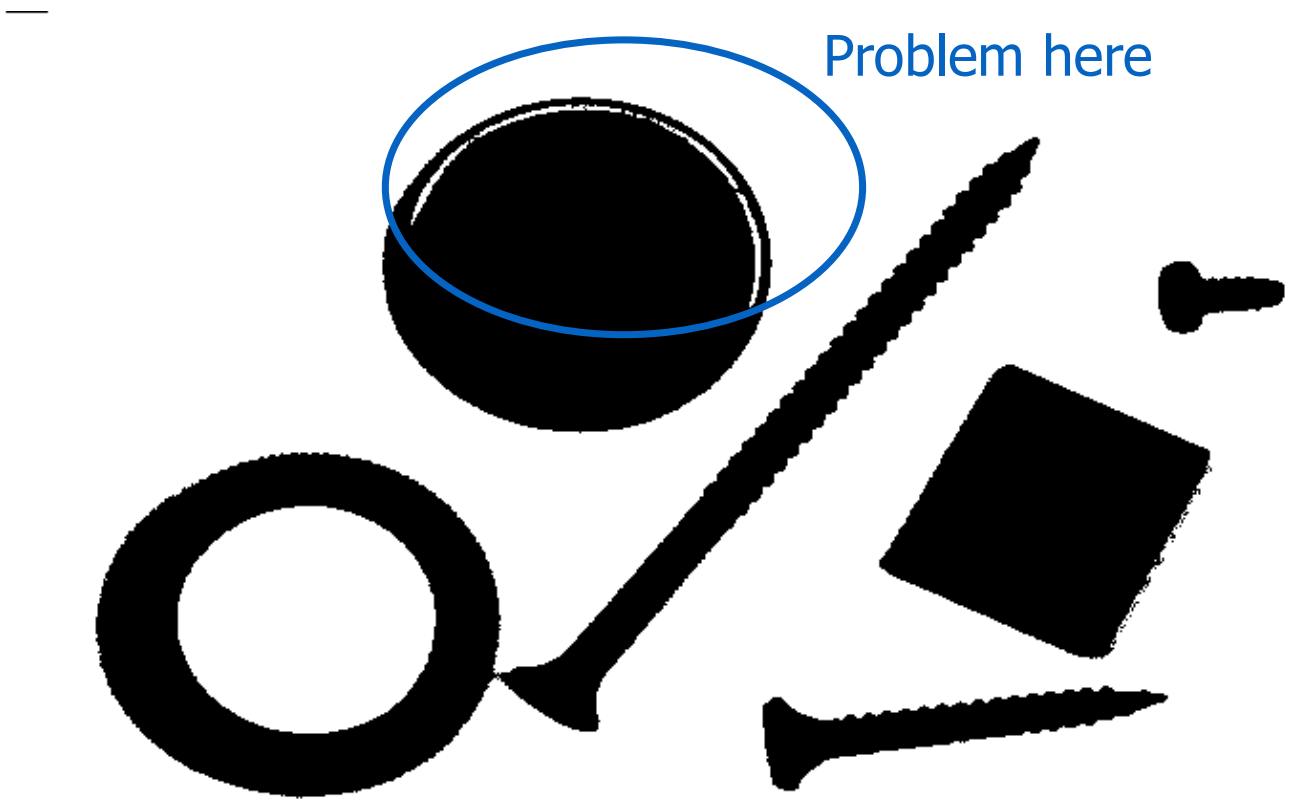
Operasi Morfologi pada Citra Biner

- Morfologi adalah sekumpulan operasi pengolahan citra yang memproses citra berdasarkan bentuknya.

- Sifat operasi morfologi: Operasi morfologi menyederhanakan citra, mengukur dan mempertahankan karakteristik bentuk utama objek.



Selain menyderhanakan bentuk, operasi morfologi juga berguna untuk mengisi *pixel-pixel* yang hilang



Empat operasi morfologi

- Dilasi (*dilation*)
- Pengikisan atau erosi (*erotion*)
- Pembukaan (*opening*)
- Penutupan (*closing*)

Hubungan keempat operasi morfologi tersebut:

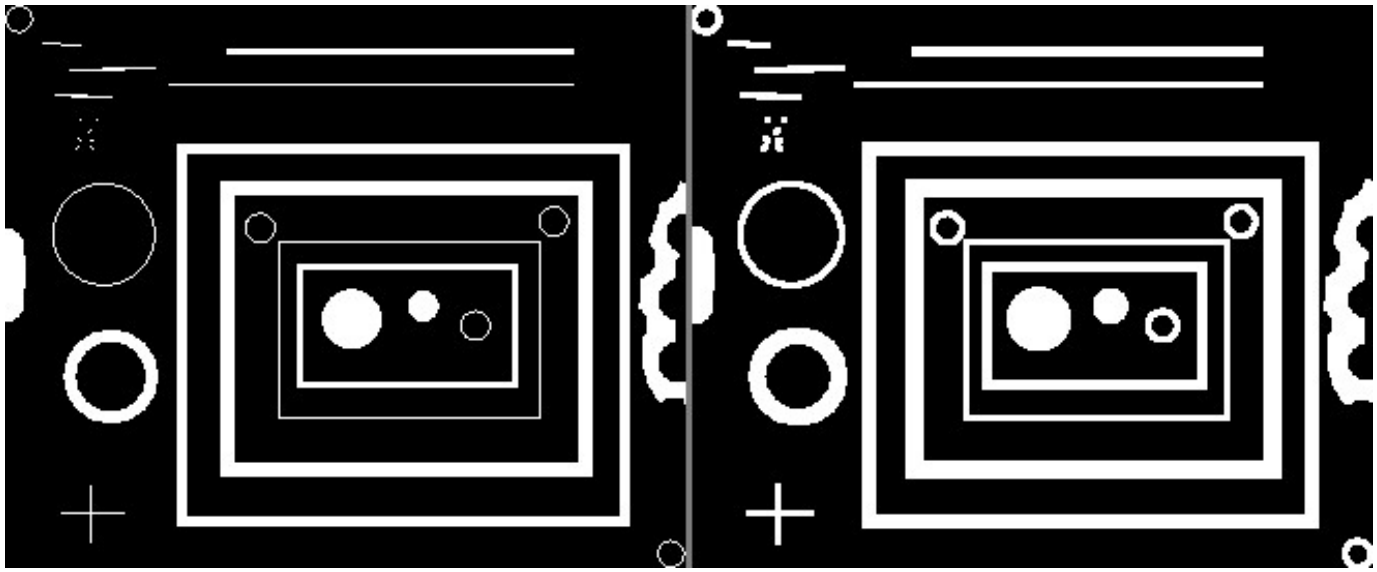
- Operasi **dilasi** bertujuan untuk memperlebar bentuk objek, sedangkan operasi **erosi** mengikis bentuk objek.

- Dilatasi dan erosi adalah operasi dasar, dapat digabungkan menjadi rangkaian yang lebih kompleks.

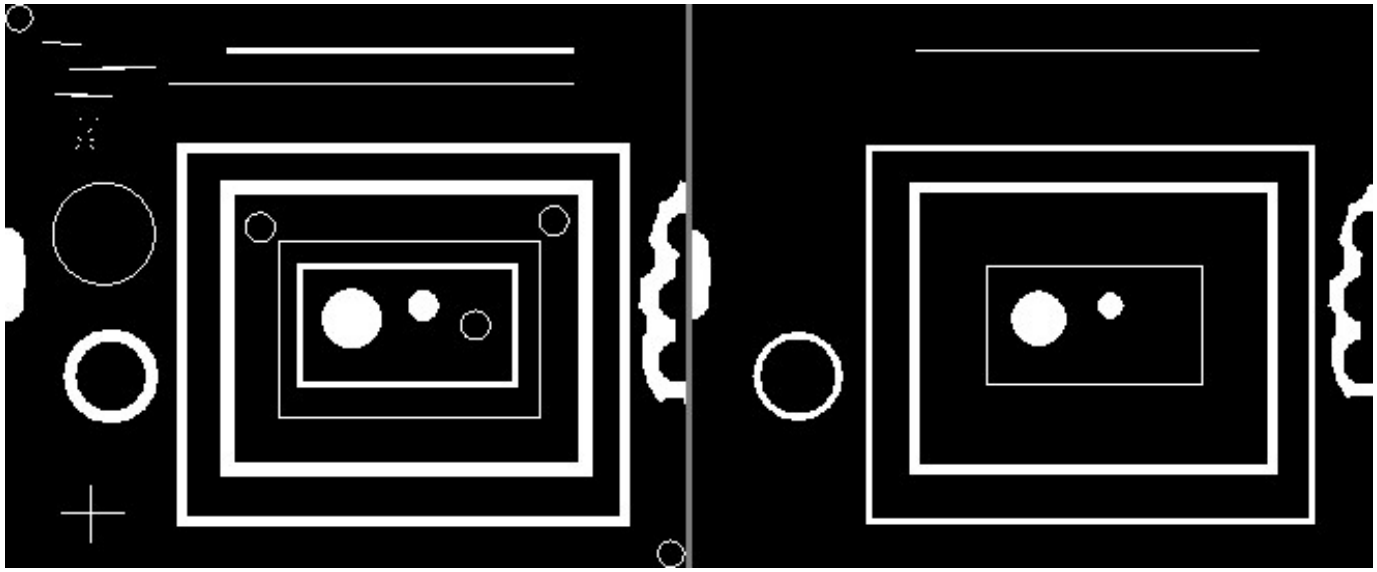
- Erosi dan dilatasi bukan operasi yang dapat dibalik (*not invertible*) --- jika suatu citra dikikis atau melebar, citra aslinya tidak dapat diperoleh kembali.

- Kombinasi erosi dan dilatasi merupakan operasi baru, **pembukaan** (*opening*) dan **penutupan** (*closing*). Keduanya merupakan penapisan secara morfologi yang paling berguna.

Dilasi



Erosi



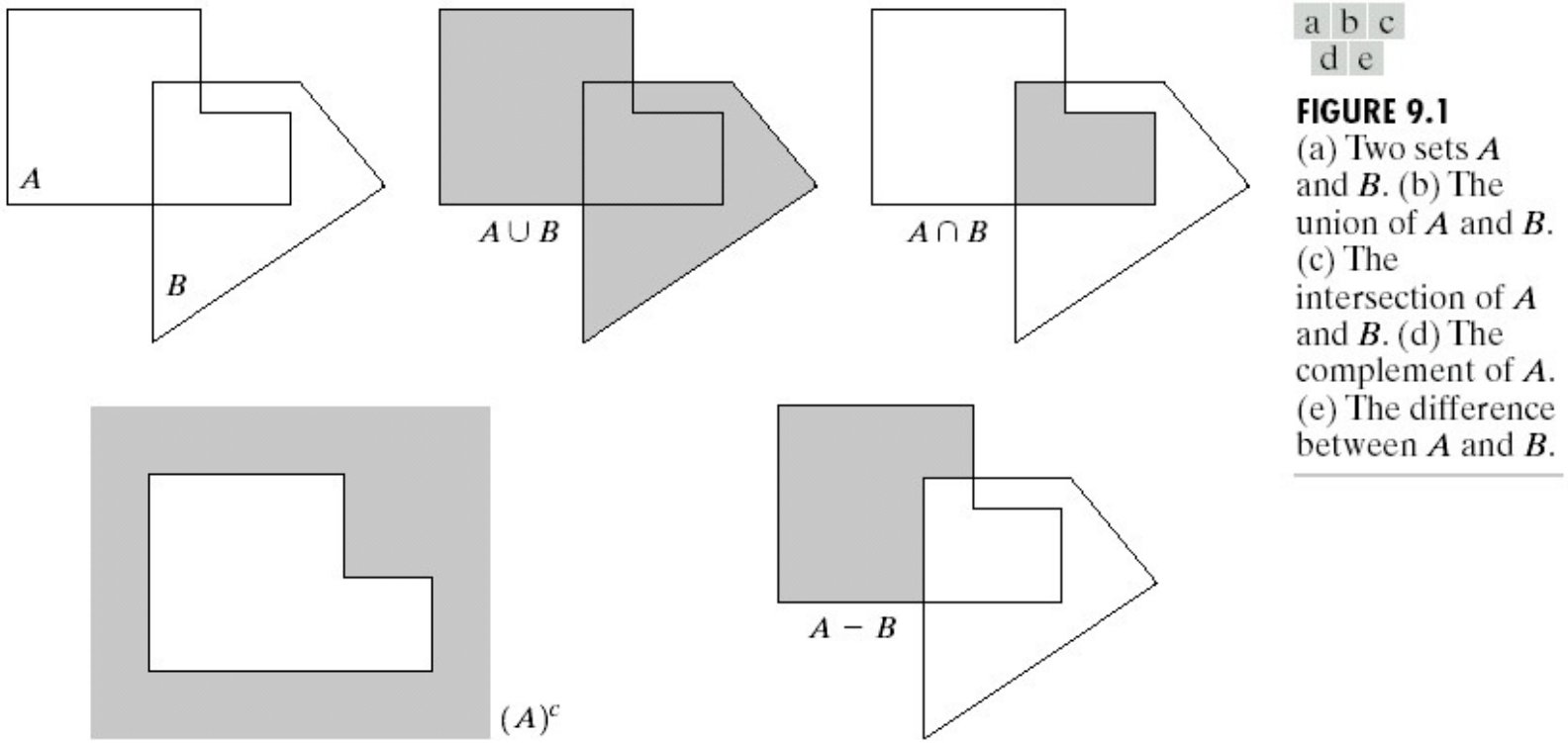
- Erosion 

- Dilation 

- combine to
  - Opening → object
  - Closening → background

keep general shape but smooth with respect to

- Operasi morfologi pada citra biner didasarkan pada operasi di dalam teori himpunan (irisan, gabungan, selisih, komplemen, beda setangkup, dll).

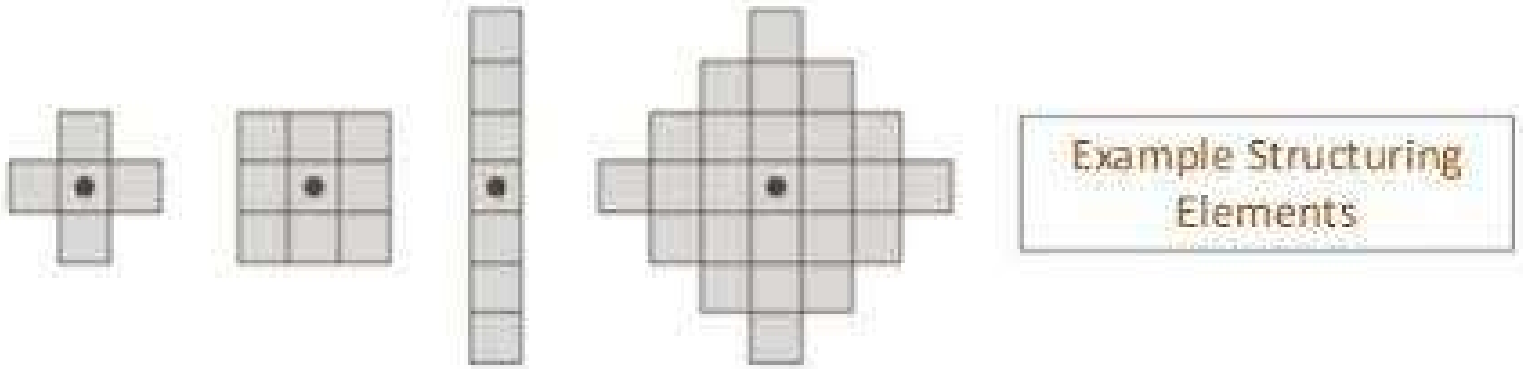


a b c
d e

FIGURE 9.1
(a) Two sets A and B. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B.

Elemen Penstruktur (*Structuring Element* - SE)

- Operasi morfologi menerapkan elemen penstruktur (*structuring element*) pada citra masukan, menghasilkan citra luaran dengan ukuran yang sama.
- SE adalah himpunan kecil elemen bernilai 0 dan 1
- Setiap SE memiliki titik pusat (*origin*)



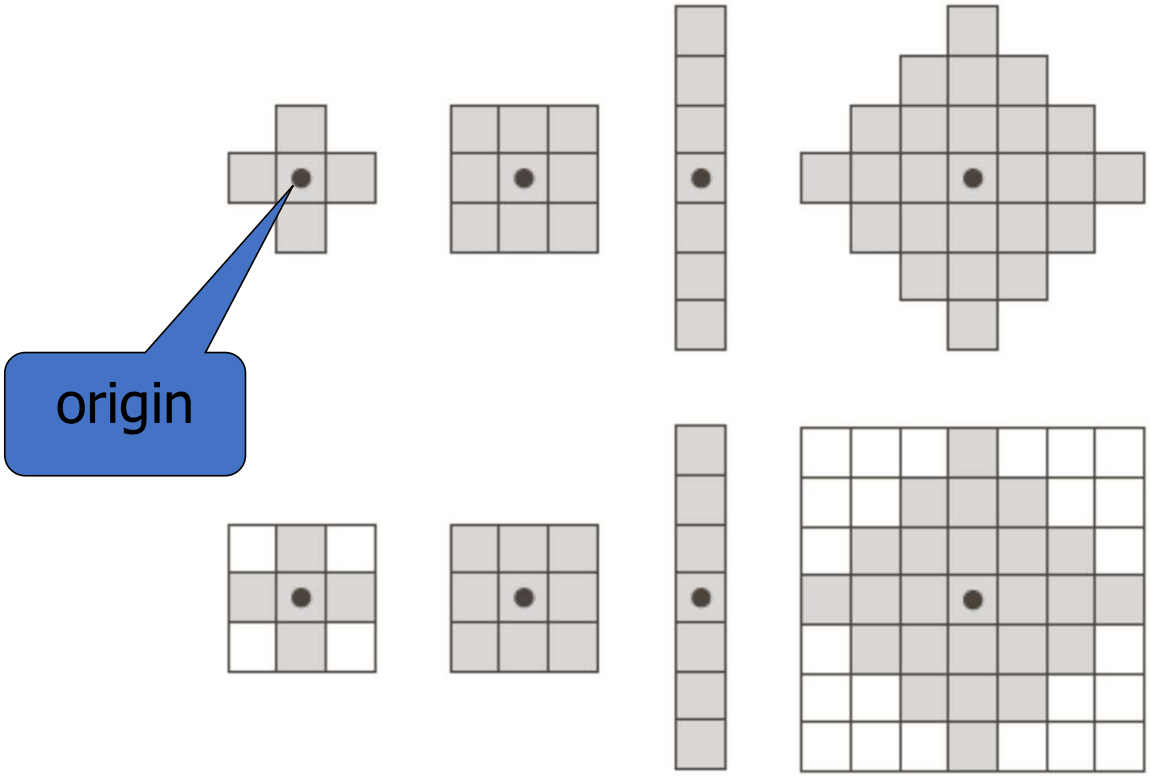
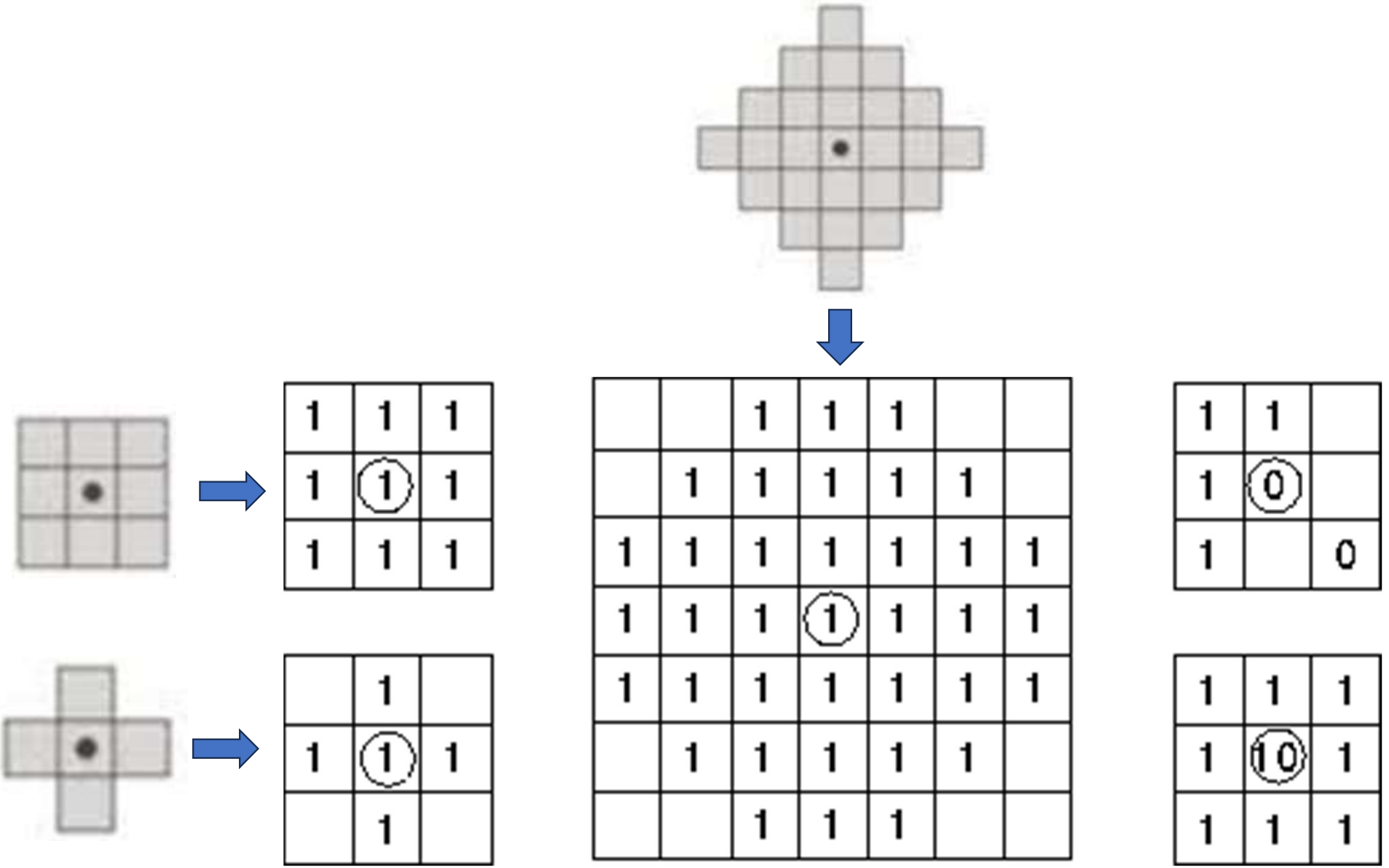
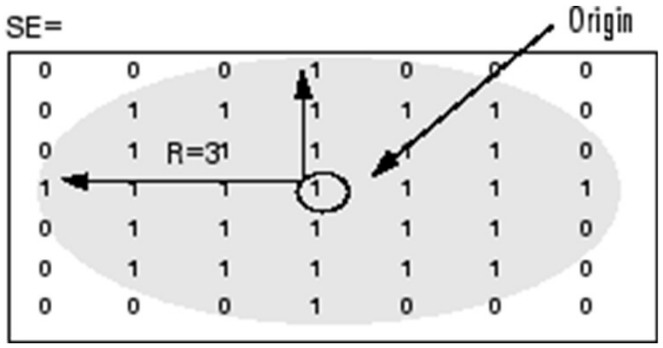


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.


1 1 1
1 1 1
1 1 1
1 1
1 0
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 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 10 1
1 1 1

- Operasi morfologi dapat disesuaikan untuk suatu aplikasi dengan pemilihan elemen penstruktur yang tepat, yang menentukan dengan tepat bagaimana objek akan melebar atau terkikis.

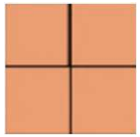
- Fungsi matlab `strel()` dapat membuat berbagai jenis elemen penstruktur:



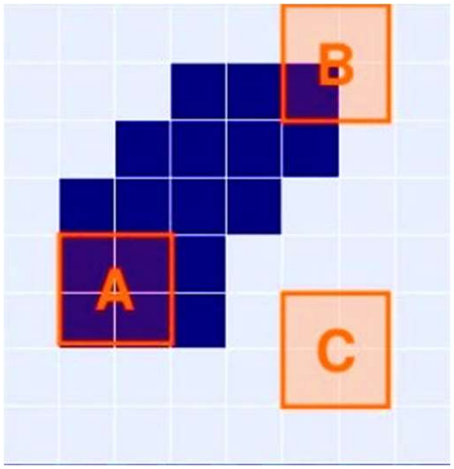
  - dish-shaped,
  - diamond-shaped,
  - ball-shaped,
  - square
  - flat linear dengan panjang LEN
  - sembarang flat atau nonflat dengan spesifikasi tetangganya

Structuring Element

- Hit and Fit in Structuring Element (SE)
  - **Hit :** Any on pixels in the SE covers an on pixel in the image
  - **Fit:** All on pixels in the SW cover on pixels in the image



Structuring Element



Dilasi (*Dilation*)

- Operasi dilatasi atau pelebaran memungkinkan objek untuk mengembang, kemudian berpotensi mengisi lubang-lubang kecil dan menghubungkan objek-objek yang terpisah.

- Cara kerja: letakkan elemen penstruktur pada citra dan menggesernya melintasi citra (seperti pergeseran kernel pada operasi konvolusi):

  1) Jika titik pusat (origin) elemen penstruktur bertepatan dengan '0' pada citra, tidak ada perubahan; berpindah ke piksel berikutnya.

  2) Jika titik pusat (origin) elemen penstruktur bertepatan dengan '1' pada citra, maka semua piksel yang berimpitan dengan elemen penstruktur diubah menjadi ‘1’.

Dilation

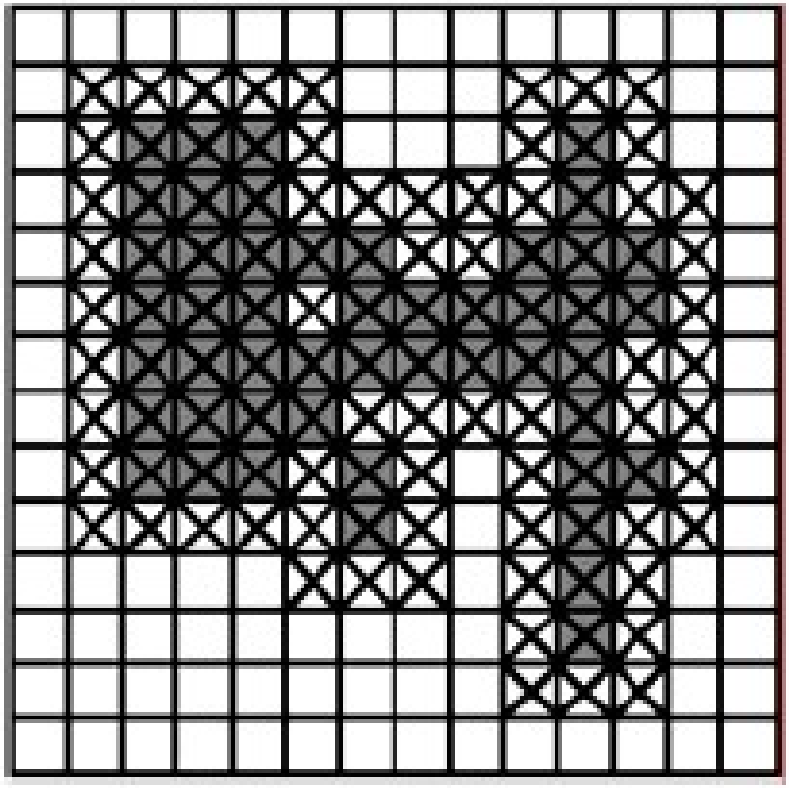
Definition

A **dilation** of an image I by the structure element H is given by the set operation

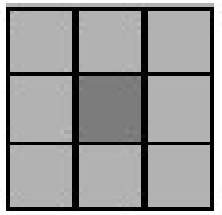
$$I \oplus H = \{(\mathbf{p} + \mathbf{q}) \mid \mathbf{p} \in I, \mathbf{q} \in H\}$$

Alternative definition: Take the union of copies of the structuring element, $H_{\mathbf{p}}$, centered at every pixel location $\mathbf{p}$ in the foreground:

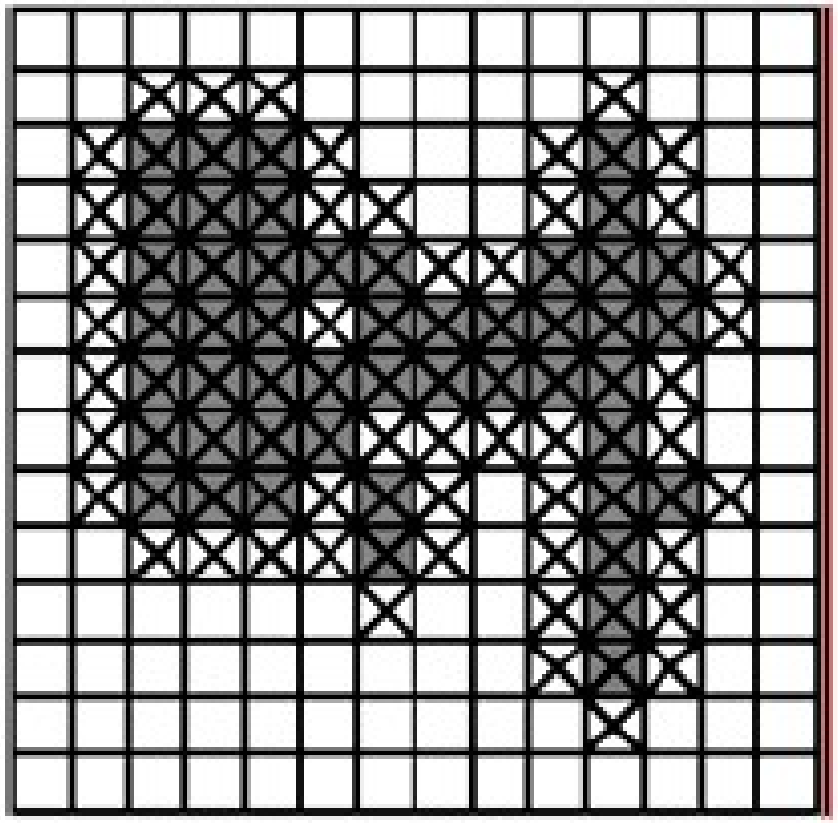
$$I \oplus H = \bigcup_{\mathbf{p} \in I} H_{\mathbf{p}}$$



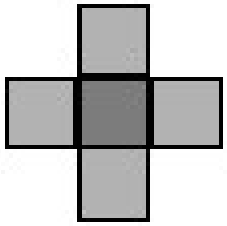
SE=



X menyatakan pixel yang diubah menjadi 1



SE=



BW image

	(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)	(0,9)	(0,10)	(0,11)	(0,12)	(0,13)
(0,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(1,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(2,0)	0	0	1	1	0	0	0	0	0	0	0	0	0	0
(3,0)	0	0	1	1	1	1	0	0	0	0	0	0	0	0
(4,0)	0	0	1	1	1	1	0	0	0	0	0	0	0	0
(5,0)	0	0	1	1	0	0	0	0	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(7,0)	0	0	0	0	0	0	1	1	1	1	1	1	0	0
(8,0)	0	0	0	0	0	0	1	1	1	1	1	1	0	0
(9,0)	0	0	0	0	1	1	1	1	1	1	1	1	0	0
(10,0)	0	0	0	0	1	1	1	1	1	1	1	1	0	0
(11,0)	0	0	0	0	0	0	1	1	1	1	1	1	0	0
(12,0)	0	0	0	0	0	0	1	1	1	1	1	1	0	0
(13,0)	0	0	0	0	0	0	1	1	1	1	1	1	0	0
(14,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(15,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(16,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0

x

kernel

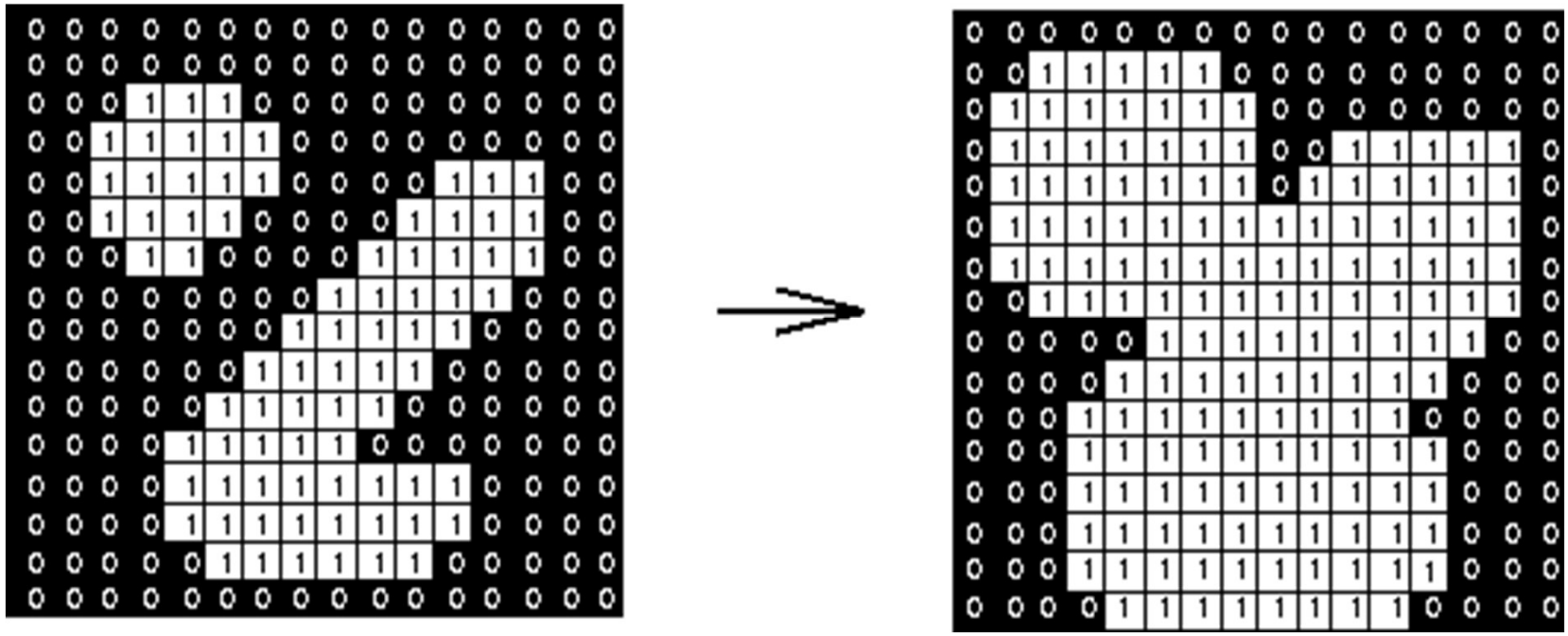
	(0,0)	(0,1)	(0,2)	
(0,0)	1	1	1	(0,2)
(1,0)	1	1	1	(1,2)
(2,0)	1	1	1	(2,2)
	(2,0)	(2,1)	(2,2)	

→

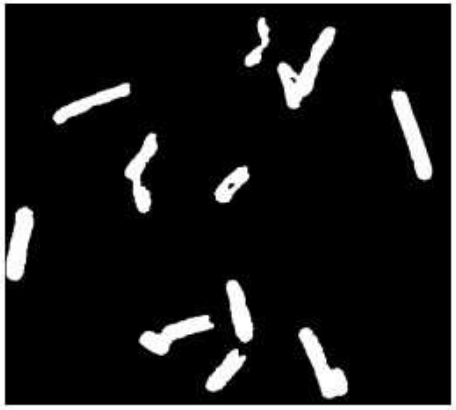
dilated BW image

	(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)	(0,9)	(0,10)	(0,11)	(0,12)	(0,13)
(0,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(1,0)	0	1	1	1	1	0	0	0	0	0	0	0	0	0
(2,0)	0	1	1	1	1	1	1	0	0	0	0	0	0	0
(3,0)	0	1	1	1	1	1	1	0	0	0	0	0	0	0
(4,0)	0	1	1	1	1	1	1	0	0	0	0	0	0	0
(5,0)	0	1	1	1	1	1	1	0	0	0	0	0	0	0
(6,0)	0	1	1	1	1	1	1	1	1	1	1	1	1	0
(7,0)	0	0	0	0	0	1	1	1	1	1	1	1	1	0
(8,0)	0	0	0	1	1	1	1	1	1	1	1	1	1	0
(9,0)	0	0	0	1	1	1	1	1	1	1	1	1	1	0
(10,0)	0	0	0	1	1	1	1	1	1	1	1	1	1	0
(11,0)	0	0	0	1	1	1	1	1	1	1	1	1	1	0
(12,0)	0	0	0	0	0	1	1	1	1	1	1	1	1	0
(13,0)	0	0	0	0	0	1	1	1	1	1	1	1	1	0
(14,0)	0	0	0	0	0	1	1	1	1	1	1	1	1	0
(15,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(16,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0

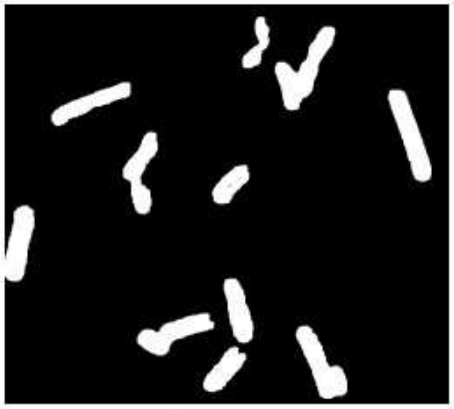
https://medium.com/@sasasulakshi/opencv-morphological-dilation-and-erosion-fab65c29efb3



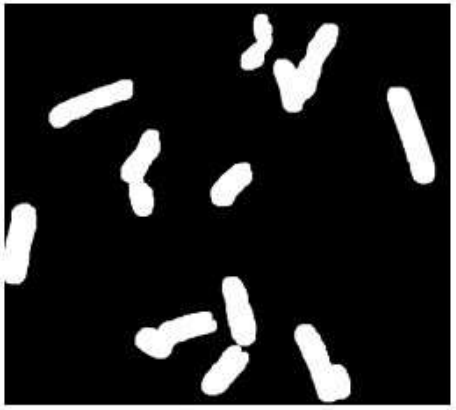
Dilation: a 3×3 square structuring element
(www.cs.princeton.edu/~pshilane/class/mosaic/).



Original (701x781)

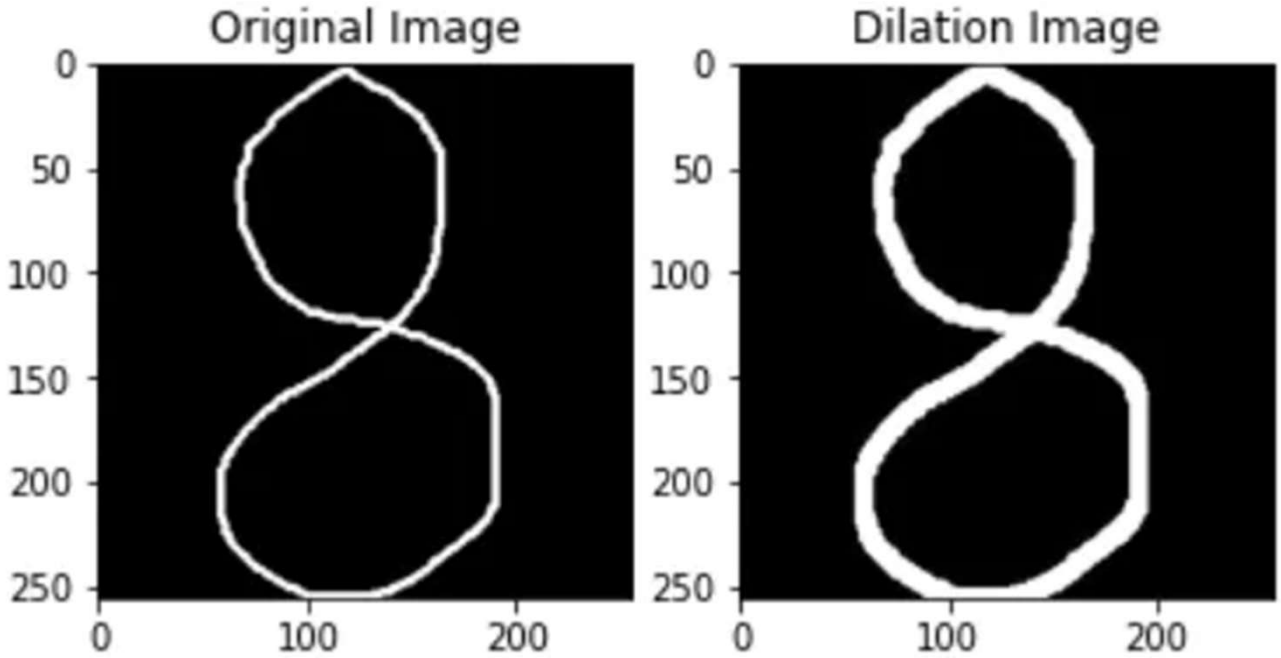


dilation with
3x3 structuring element



dilation with
7x7 structuring element

Dilatasi dengan SE berbentuk bujursangkar 3 x 3



https://medium.com/analytics-vidhya/morphological-transformations-of-images-using-opencv-image-processing-part-2-f64b14af2a38

```
%Read image
I = imread('ford.jpg');

figure('Name', 'original');
imshow(I);

%create structuring elements
% 11-by-11 square
se1 = strel('square',11);

%Apply dilation operation
figure('Name', 'Dilate');
Idilate1 = imdilate(I,se1);

%Show the result image
subplot(1,1,1), imshow(Idilate1),
title('11x11 square');
```



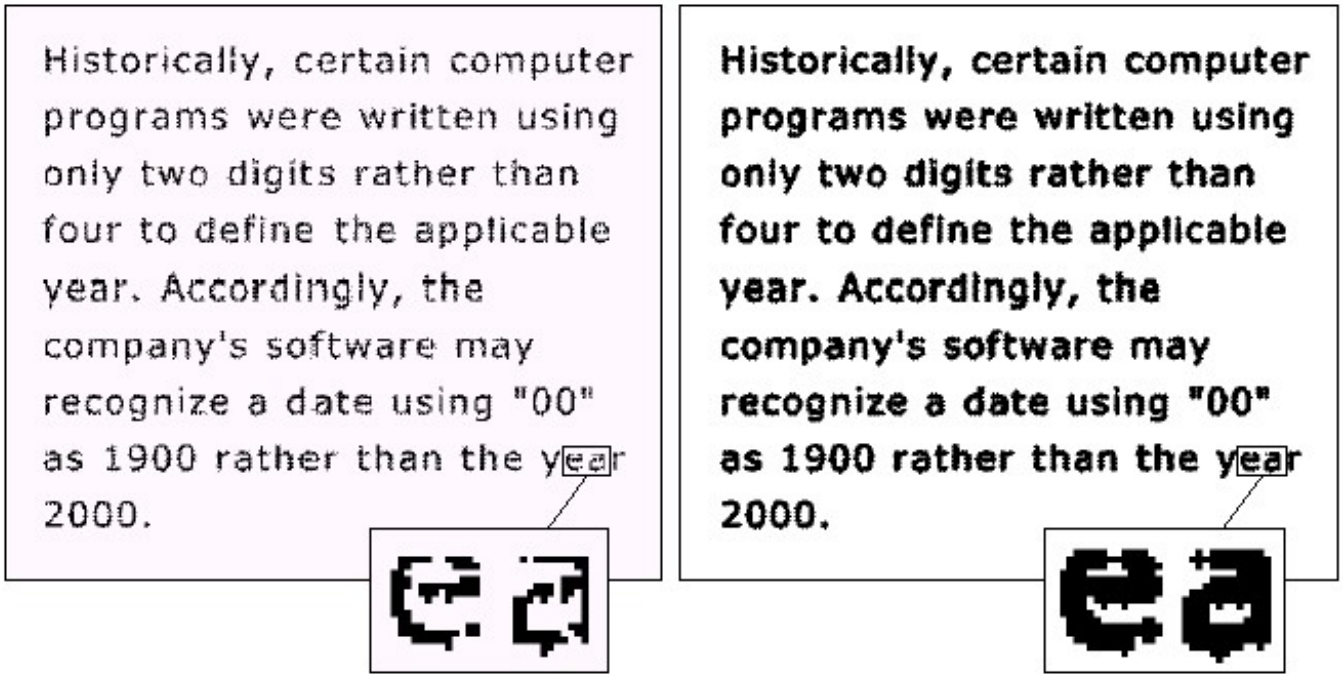
original image

11x11 square



Image after dilation

Dilation : Bridging gaps



a c
b

FIGURE 9.5
(a) Sample text of poor resolution with broken characters (magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

0	1	0
1	1	1
0	1	0

Erosi (*erosion*)

- Erosi mengecilkan objek dengan mengikis (mengikis) batas-batasnya.

- Cara kerja: letakkan elemen penstruktur pada citra dan menggesernya melintasi citra (seperti pergeseran kernel pada operasi konvolusi):

  Untuk setiap pixel pada citra:

  1. Pusatkan structuring element (SE) di atas pixel tersebut.

  2. Periksa area sekeliling pixel sesuai bentuk SE.

  3. Jika semua posisi SE yang bernilai 1 *cocok* dengan pixel objek (1) di citra:
     - maka pixel pusat tetap 1.

  4. Jika salah satu posisi SE = 1 tidak cocok (di citra bernilai 0):
     - maka pixel pusat diubah menjadi 0.

Erosion

Definition

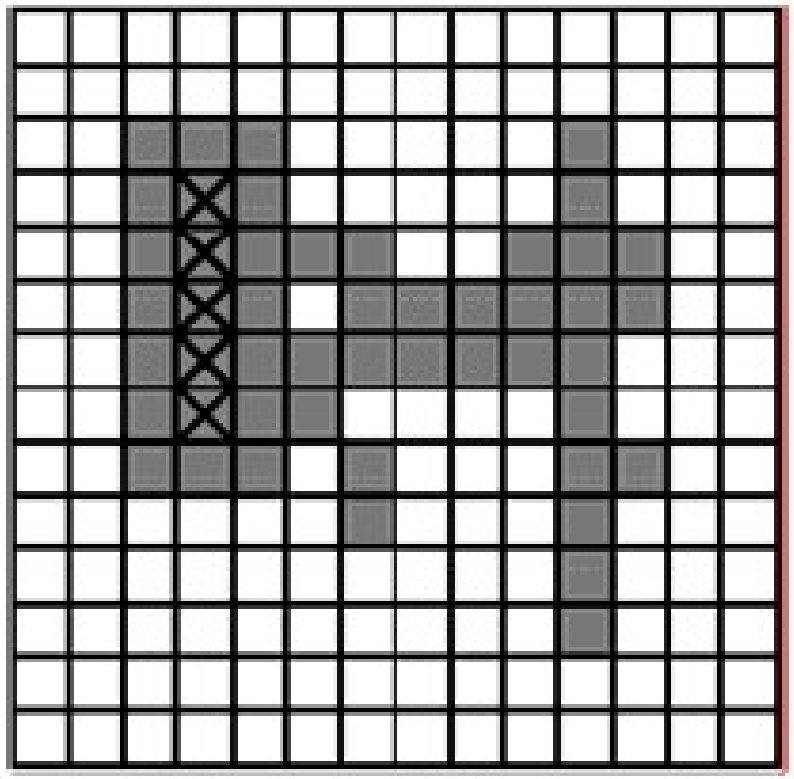
A **erosion** of an image I by the structure element H is given by the set operation

$$I \ominus H = \{\mathbf{p} \in \mathbb{Z}^2 \mid (\mathbf{p} + \mathbf{q}) \in I, \text{for every } \mathbf{q} \in H\}$$

Alternative definition: Keep only pixels $\mathbf{p} \in I$ such that $H_{\mathbf{p}}$ fits inside I:

$$I \ominus H = \{\mathbf{p} \mid H_{\mathbf{p}} \subseteq I\}$$

Erosion



Untuk setiap piksel pada citra:

1.Pusatkan structuring element (SE) di atas piksel tersebut.

2.Periksa area sekeliling piksel sesuai bentuk SE.

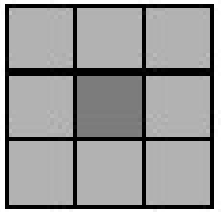
3.Jika **semua posisi SE yang bernilai 1** *cocok* dengan piksel objek (1) di citra:

maka piksel pusat **tetap 1**.

4.Jika salah satu posisi SE = 1 tidak cocok (di citra bernilai 0):

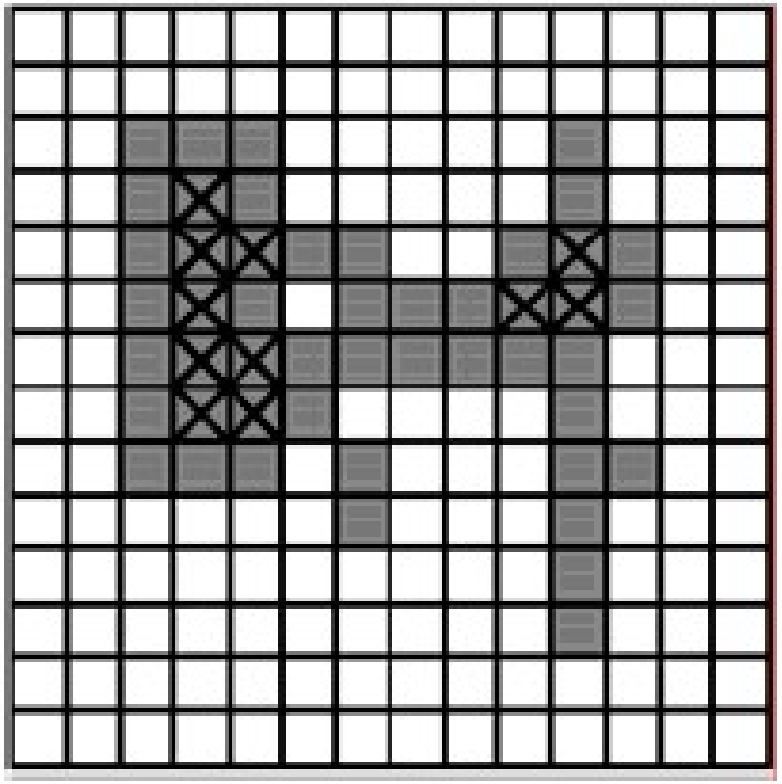
maka piksel pusat **diubah menjadi 0**.

SE=

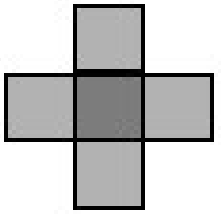


X menyatakan pixel yang tersisa (bernilai 1), pixel-pixel lainnya menjadi 0

Erosion



SE=



BW image

	(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)	(0,9)	(0,10)	(0,11)	(0,12)	(0,13)
(0,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(1,0)	0	0	0	0	0	0	0	0	0	0	0	1	1	0
(2,0)	0	0	1	1	0	0	0	0	0	0	0	1	1	0
(3,0)	0	0	1	1	1	1	0	0	0	0	0	1	1	0
(4,0)	0	0	1	1	1	1	0	0	0	0	0	1	1	0
(5,0)	1	1	1	1	1	1	1	1	1	1	1	1	1	1
(6,0)	0	0	1	1	0	0	0	0	0	0	0	0	0	0
(7,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(8,0)	0	0	0	0	0	0	1	1	1	1	1	1	0	0
(9,0)	0	0	0	0	0	0	1	1	1	1	1	1	0	0
(10,0)	0	0	0	0	1	1	1	1	1	1	1	1	0	0
(11,0)	0	1	1	1	1	1	1	1	1	1	1	1	0	0
(12,0)	0	1	1	1	0	0	1	1	1	1	1	1	0	0
(13,0)	0	1	1	1	0	0	1	1	1	1	1	1	0	0
(14,0)	0	0	0	0	0	0	1	1	1	1	1	1	0	0
(15,0)	0	0	0	0	0	0	1	1	0	1	1	0	0	0
(16,0)	0	0	0	0	0	0	1	1	0	1	1	0	0	0

X

kernel

	(0,0)	(0,1)	(0,2)	
(0,0)	1	1	1	(0,2)
(1,0)	1	1	1	(1,2)
(2,0)	1	1	1	(2,2)
	(2,0)	(2,1)	(2,2)	

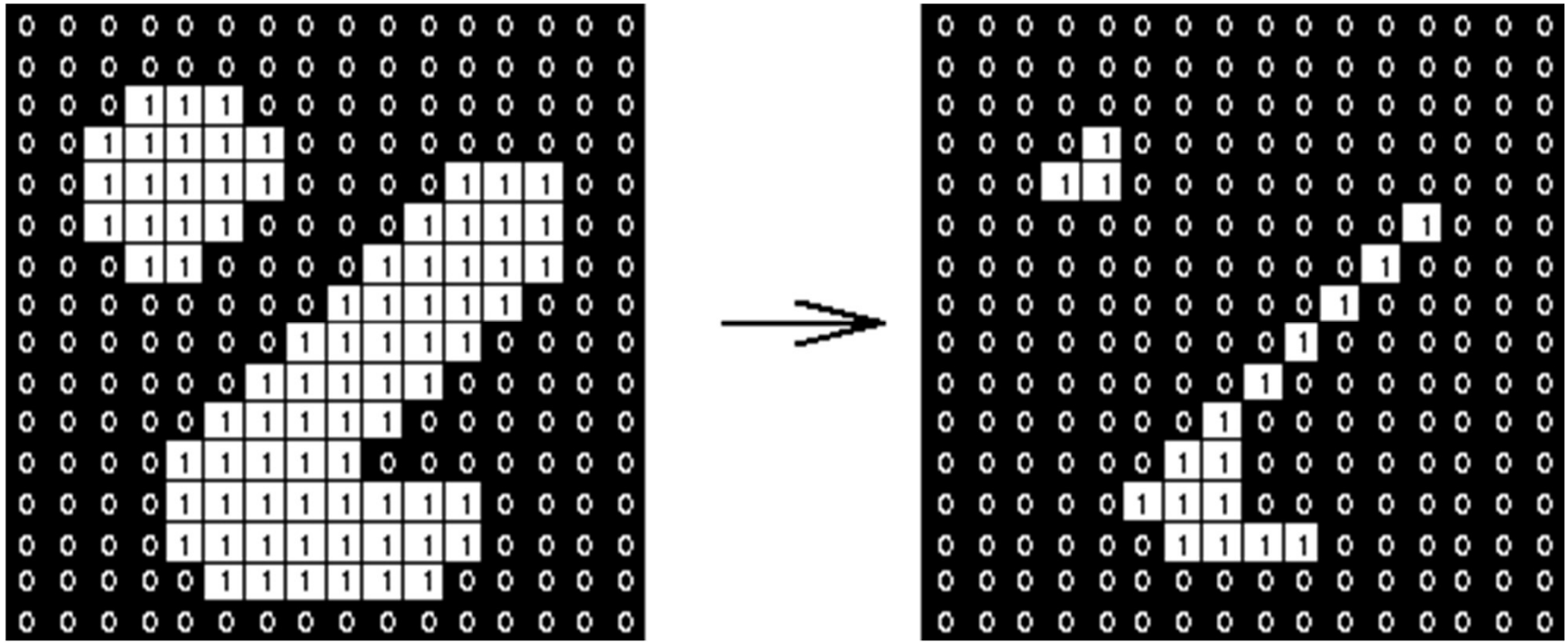
→

eroded BW image

	(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)	(0,9)	(0,10)	(0,11)	(0,12)	(0,13)
(0,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(1,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(2,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(3,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(4,0)	0	0	0	1	1	0	0	0	0	0	0	0	0	0
(5,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(6,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(7,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(8,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(9,0)	0	0	0	0	0	0	0	1	1	1	1	0	0	0
(10,0)	0	0	0	0	0	0	0	1	1	1	1	0	0	0
(11,0)	0	0	0	0	0	0	0	1	1	1	1	0	0	0
(12,0)	0	0	1	0	0	0	0	1	1	1	1	0	0	0
(13,0)	0	0	0	0	0	0	0	1	1	1	1	0	0	0
(14,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(15,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(16,0)	0	0	0	0	0	0	0	0	0	0	0	0	0	0

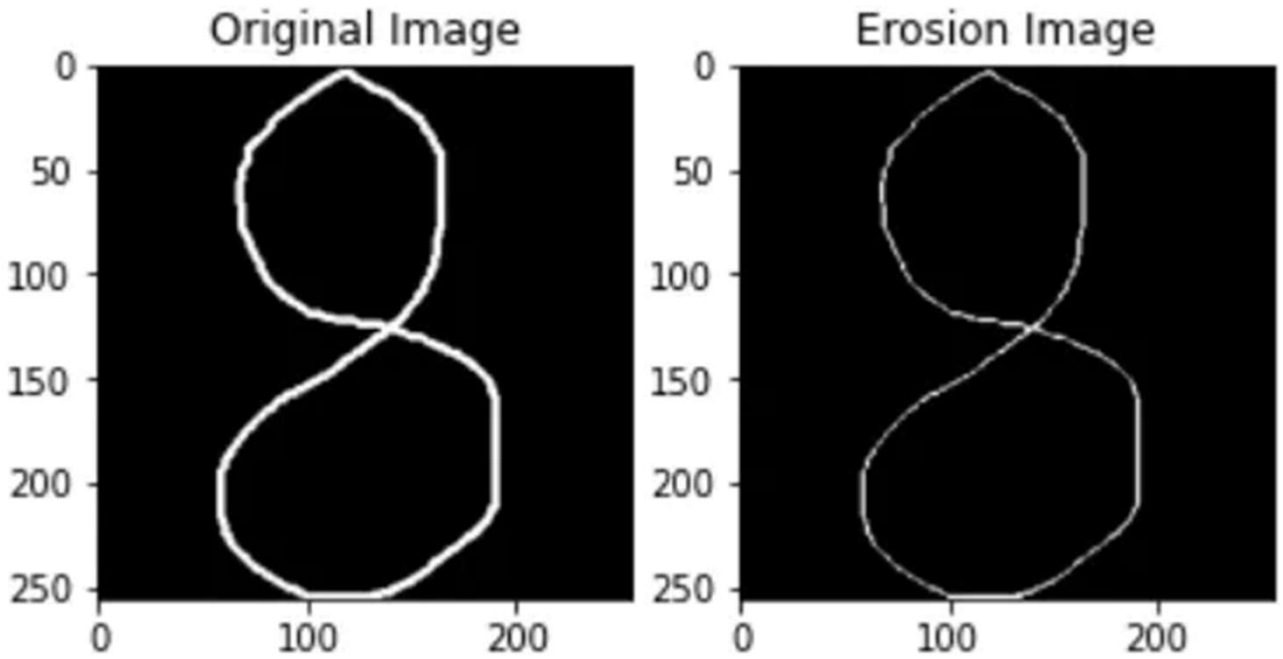
https://medium.com/@sasasulakshi/opencv-morphological-dilation-and-erosion-fab65c29efb3

Erosi dengan SE berbentuk bujursangkar 3 x 3



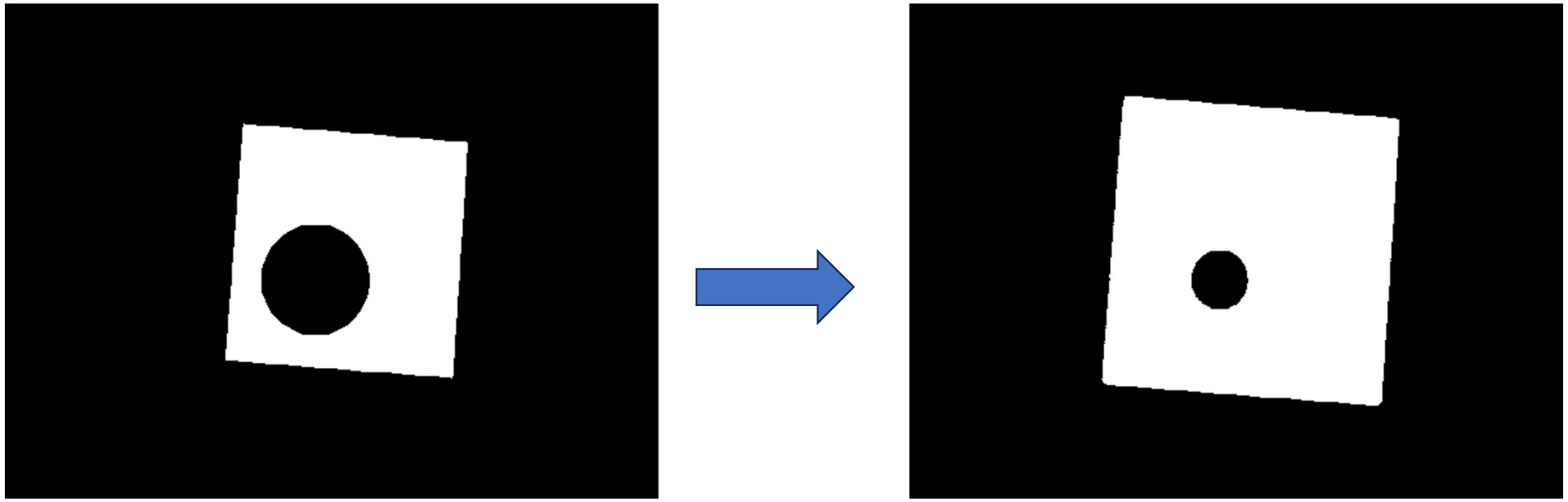
https://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm

Erosi dengan SE berbentuk bujursangkar 3 x 3

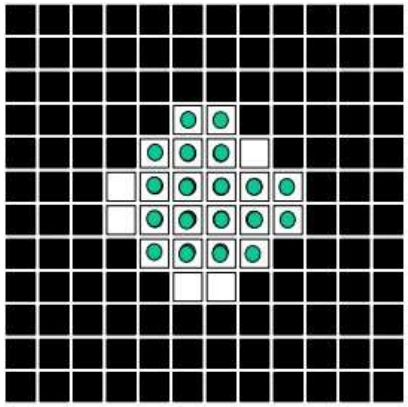


https://medium.com/analytics-vidhya/morphological-transformations-of-images-using-opencv-image-processing-part-2-f64b14af2a38

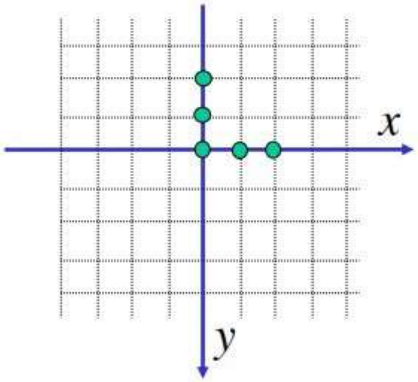
Erosi dua kali dengan SE berbentuk disk dengan diameter 11 pixel



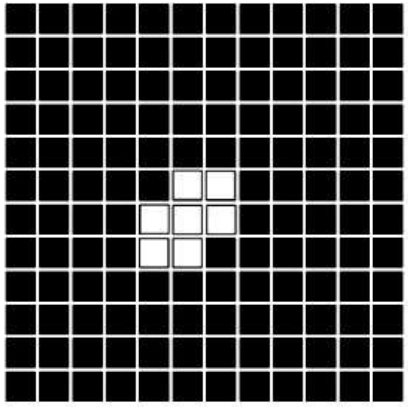
https://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm



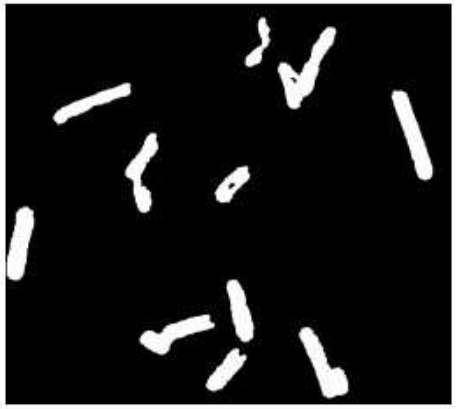
$f[x,y]$



Π_{xy}



$g[x,y]$



Original (701x781)



erosion with
3x3 structuring element



erosion with
7x7 structuring element

```
%Read image
I = imread('ford.jpg');

figure('Name', 'original');
imshow(I);

%create structuring elements
% 11-by-11 square
se1 = strel('square',11);

%Apply erosion operation
figure('Name', 'Erode');
Ierode1 = imerode(I,se1);

%Show the result image
subplot(1,1,1), imshow(Ierode1),
title('11x11 square');
```



original image

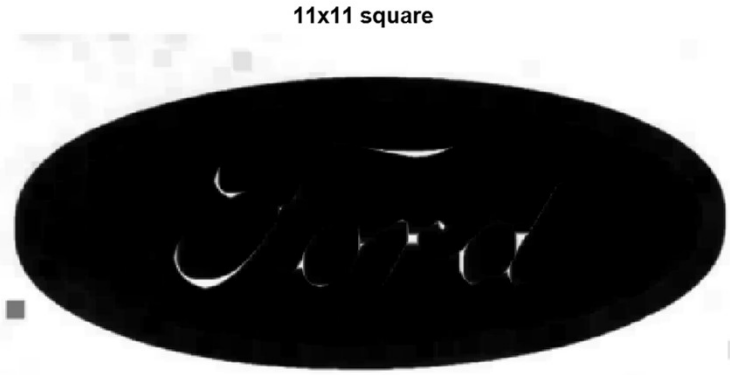
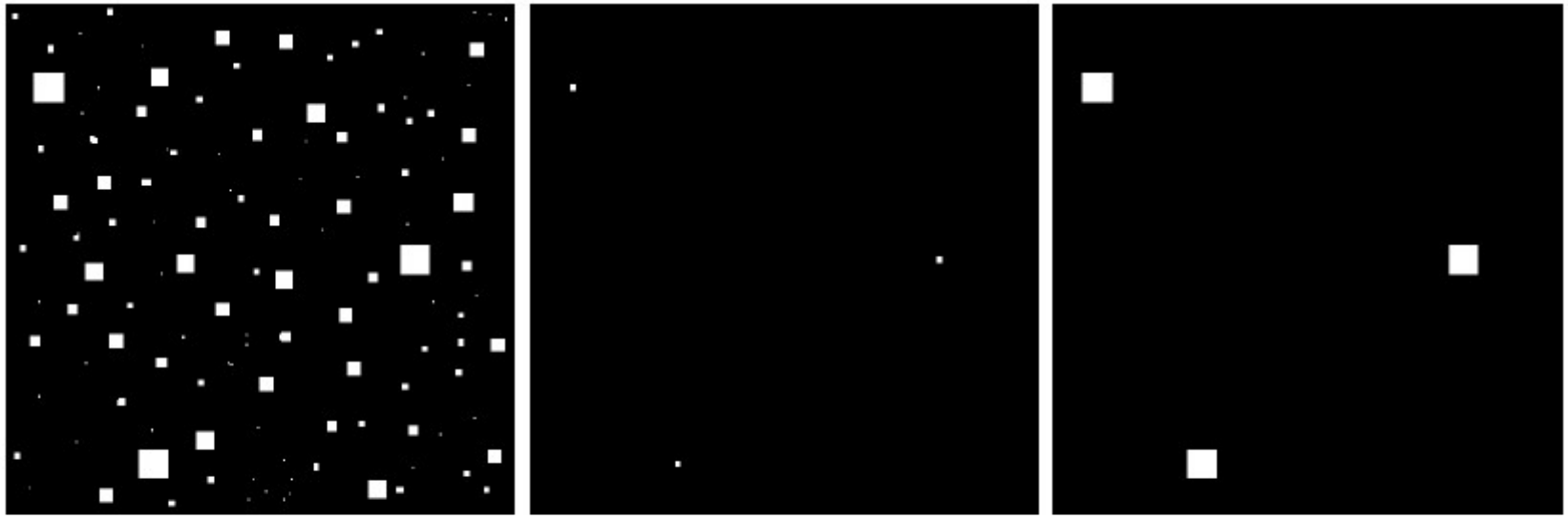


Image after erosion

Erosion : eliminating irrelevant detail



a b c

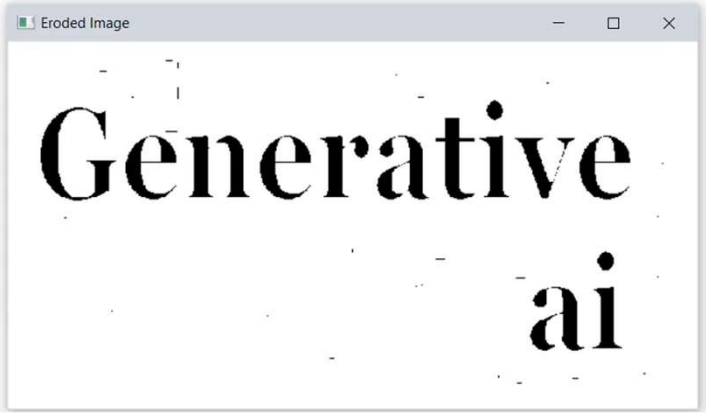
FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

structuring element B = 13x13 pixels of gray level 1

Erosi: menghilangkan bagian yang tidak perlu di dalam citra

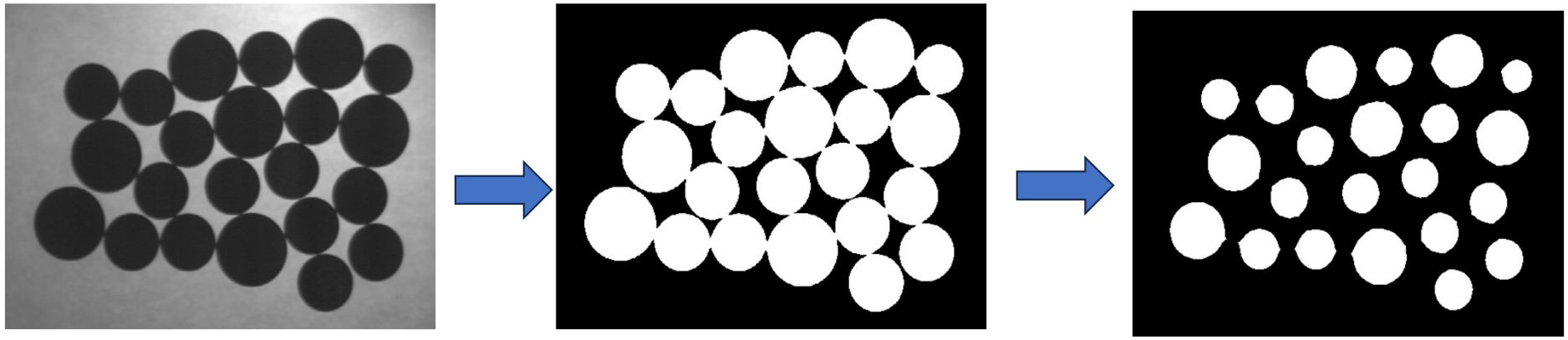


Hasil erosi →



https://medium.com/@sasasulakshi/opencv-morphological-dilation-and-erosion-fab65c29efb3

- Kegunaan lain operasi erosi: mencacah (*counting*) jumlah objek



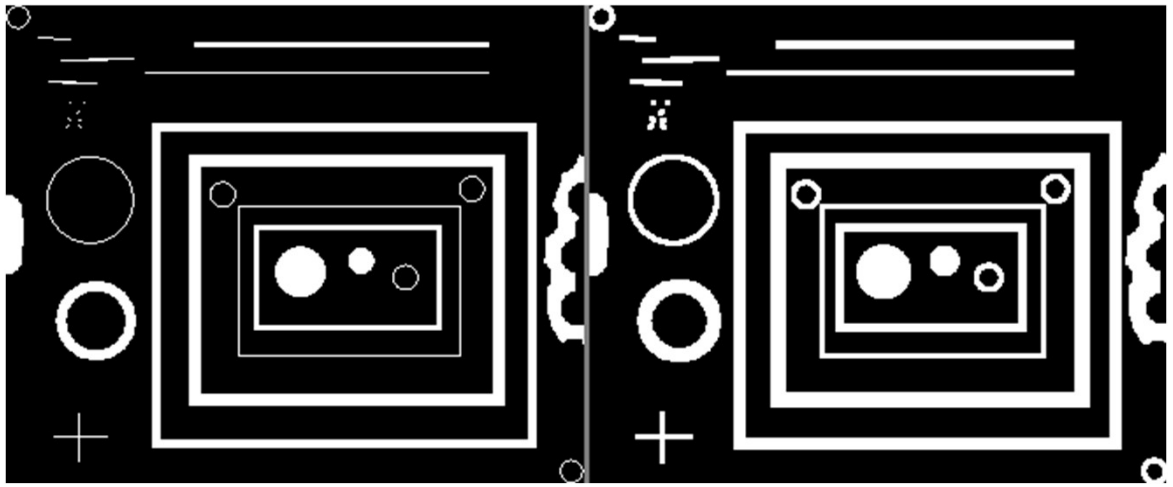
Citra grayscale

Citra biner

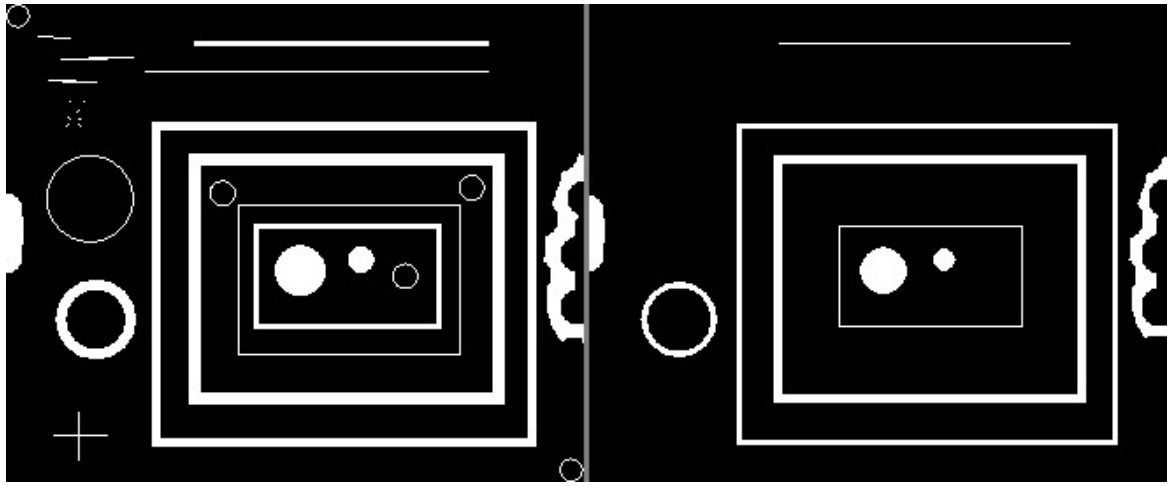
Hasil erosi dua kali dengan SE berbentuk disk dengan dimeter 11

Perbedaan hasil dilatasi dengan erosi

Dilasi



Erosi



Sumber: Mathwork

Pembukaan (*Opening*)

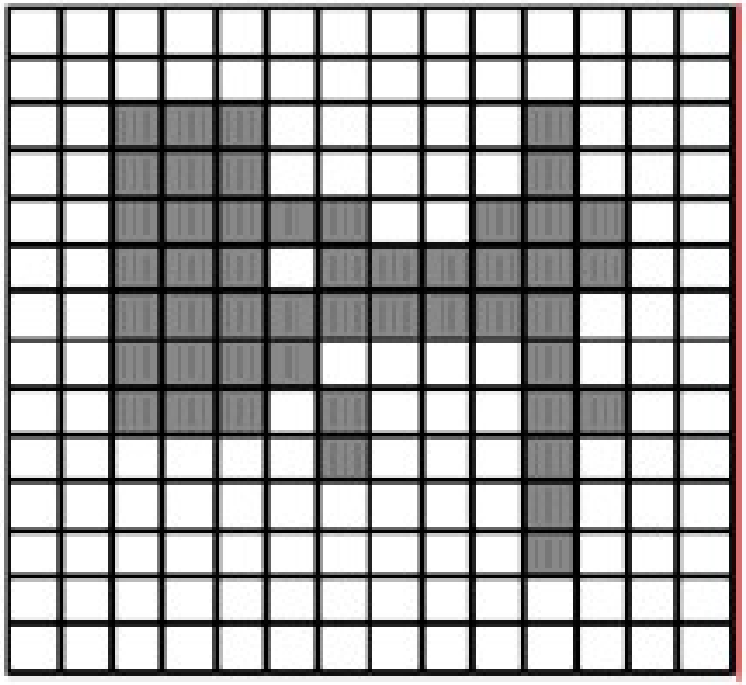
- Operasi pembukaan terdiri dari erosi yang diikuti dengan dilatasi dan dapat digunakan untuk menghilangkan semua pixel di wilayah yang terlalu kecil untuk memuat elemen penstruktur.

- Dalam hal ini elemen penstruktur sering disebut *probe*, karena ia menyelidiki citra untuk mencari objek kecil untuk disaring dari citra.

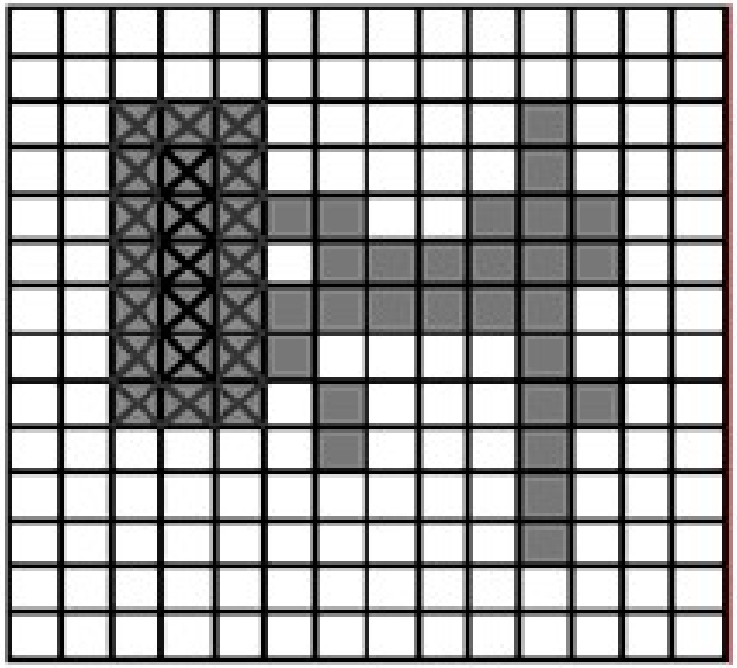
- Notasi: $A \circ B = (A \ominus B) \oplus B$

Opening

B=



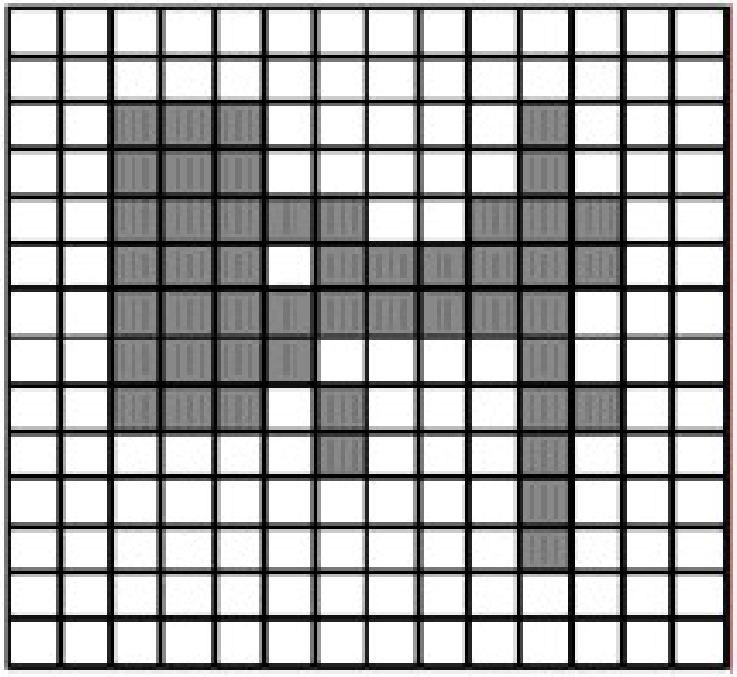
A



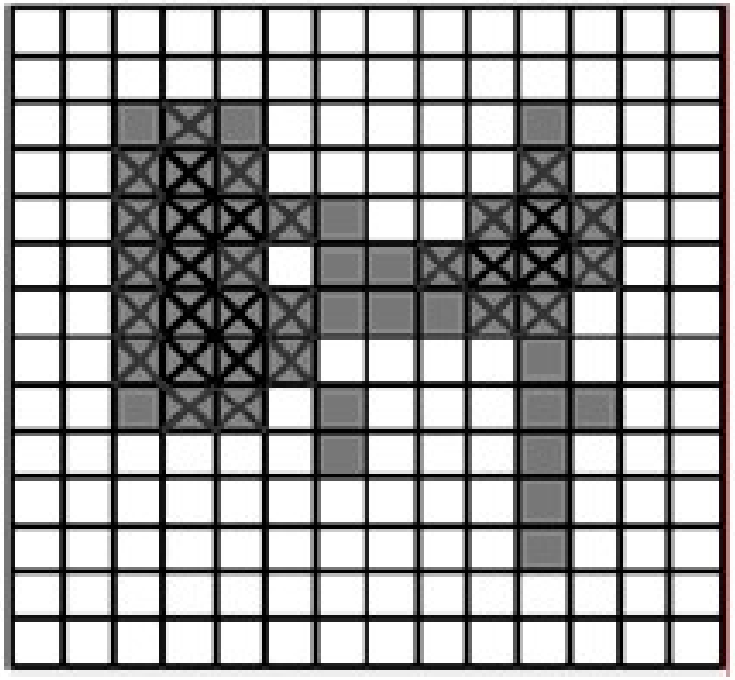
$A \ominus B$ $A \circ B$

Opening

B=



A



A⊖B A∘B

```
%Read image
I = imread('ford.jpg');

figure('Name', 'original');
imshow(I);

%create structuring elements
% 11-by-11 square
se1 = strel('square',11);

%Apply the open operation
figure('Name', 'Open');
Iopen1 = imopen(I,se1);

%Show the result image
subplot(1,1,1), imshow(Iopen1), title('11 x
11 square');
```



original image

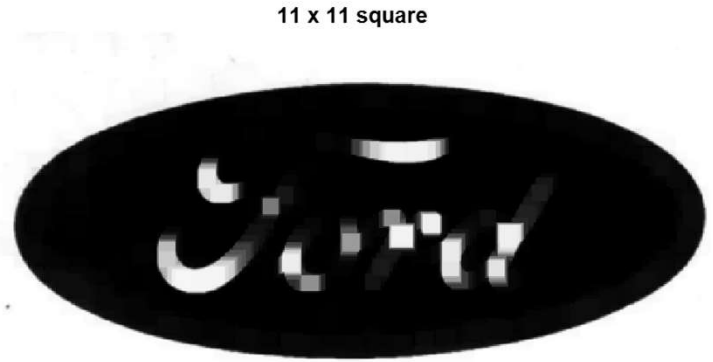
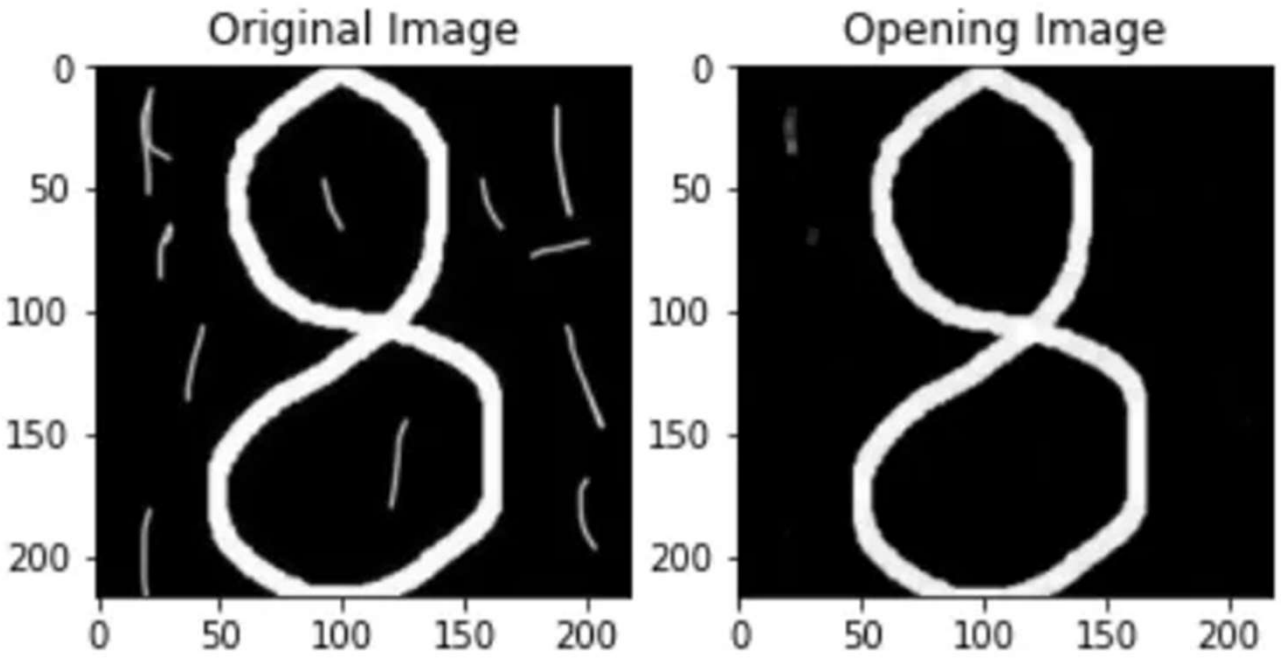
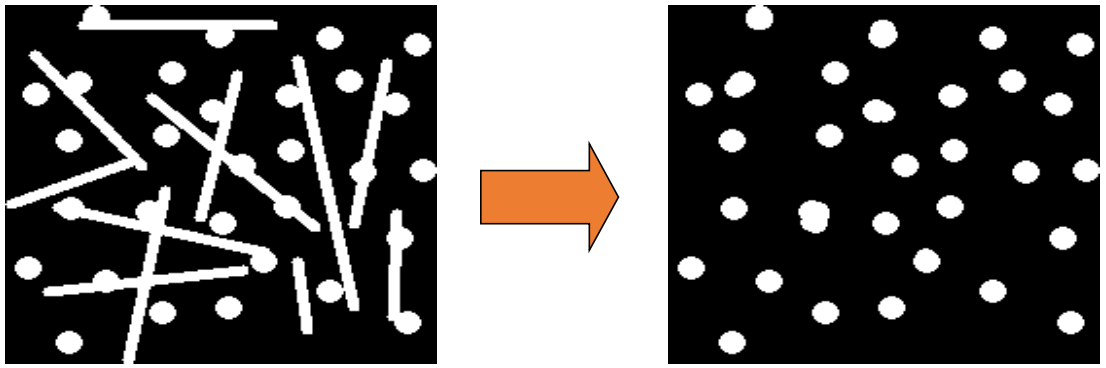


Image after opening

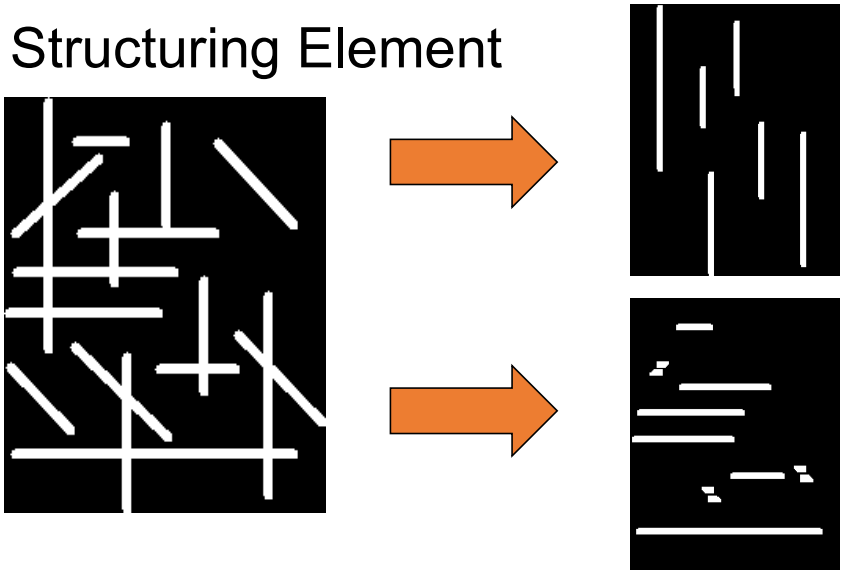
Opening dengan SE berbentuk bujursangkar 4 x 4



Opening with a 11 pixel diameter disc:



3x9 and 9x3 Structuring Element



Penutupan (*closing*)

- Penutupan terdiri dari dilatasi yang diikuti erosi dan menghubungkan objek-objek yang berdekatan.

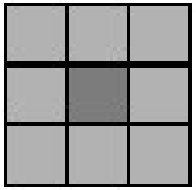
- Dapat digunakan untuk mengisi lubang dan celah kecil.

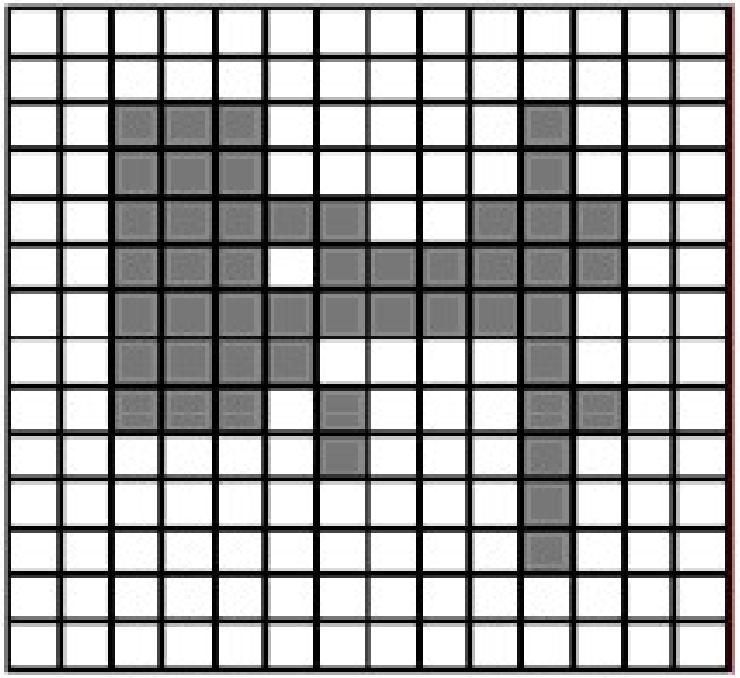
- Notasi: $A \bullet B = (A \oplus B) \ominus B$

Closing

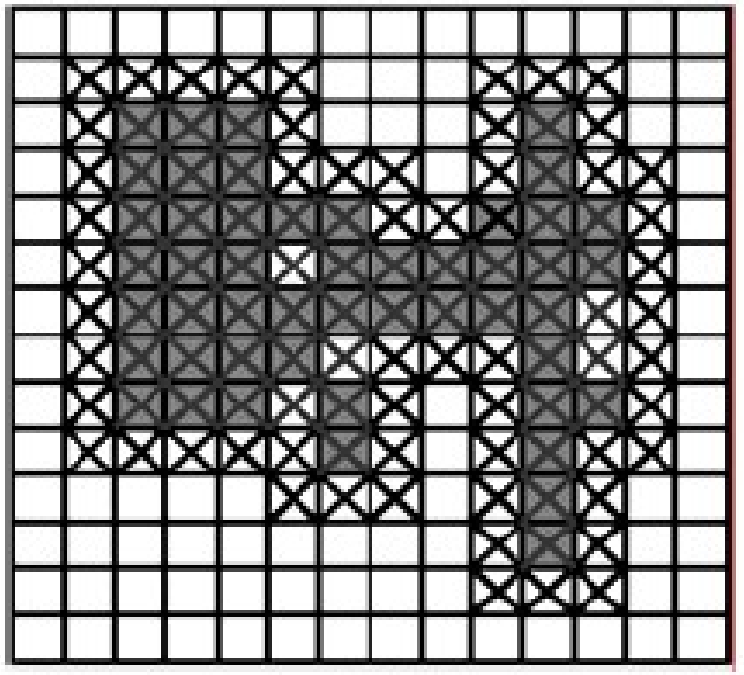


B=





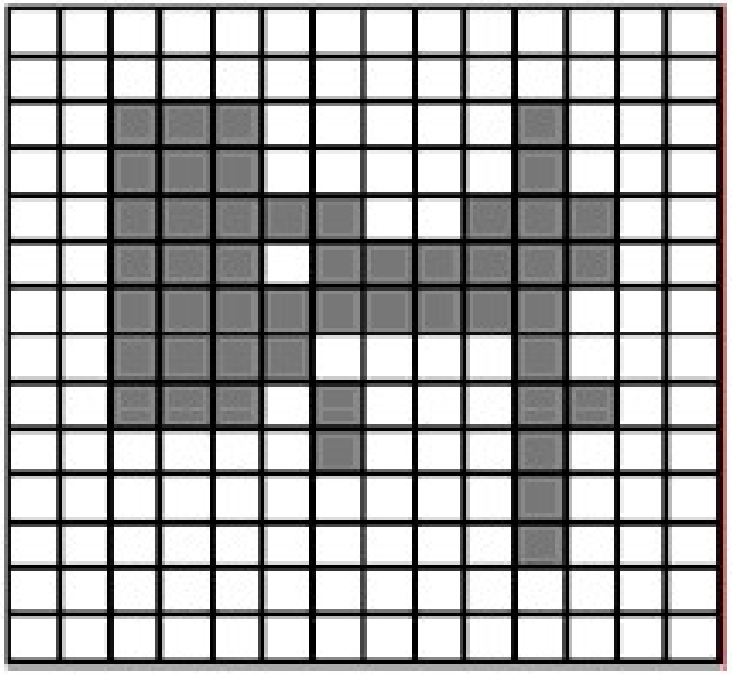
A



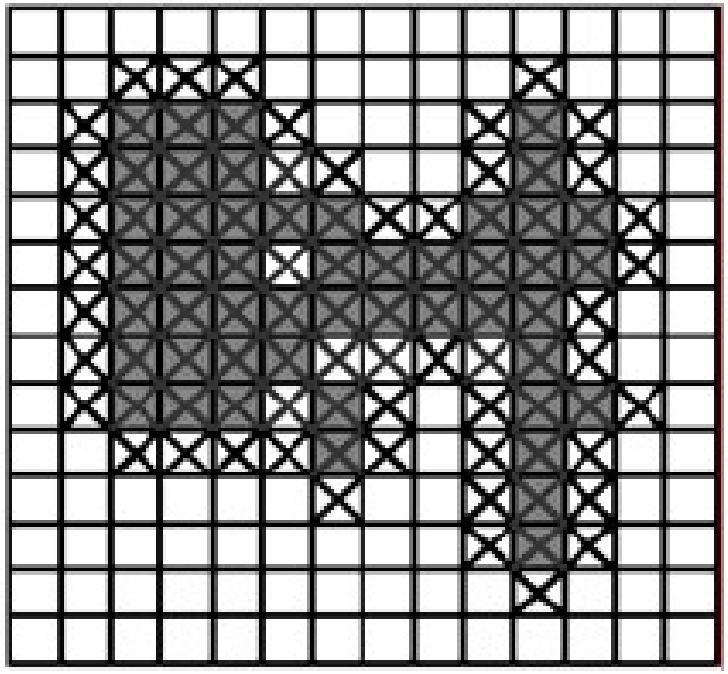
A⊕B A • B

Closing

B=



A



A⊕B A • B

```
%Read image
I = imread('ford.jpg');

figure('Name', 'original');
imshow(I);

%create structuring elements
% 11-by-11 square
se1 = strel('square',11);

%Apply close operation
figure('Name', 'Close');
Iclose1 = imclose(I,se1);

%Show the result image
subplot(1,1,1), imshow(Iclose1),
title('11x11 square');
```



original image

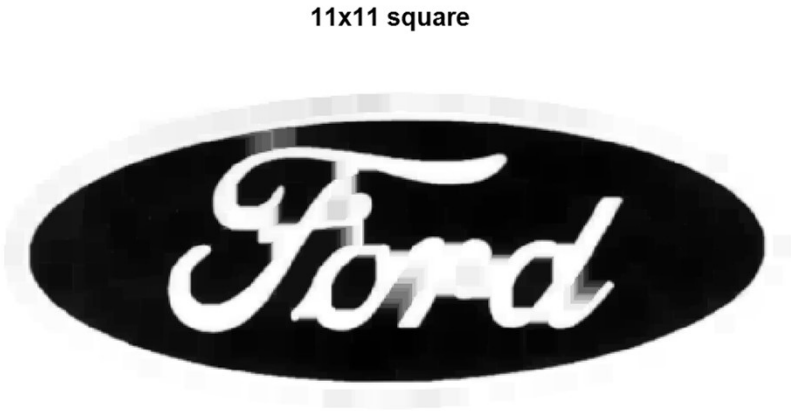
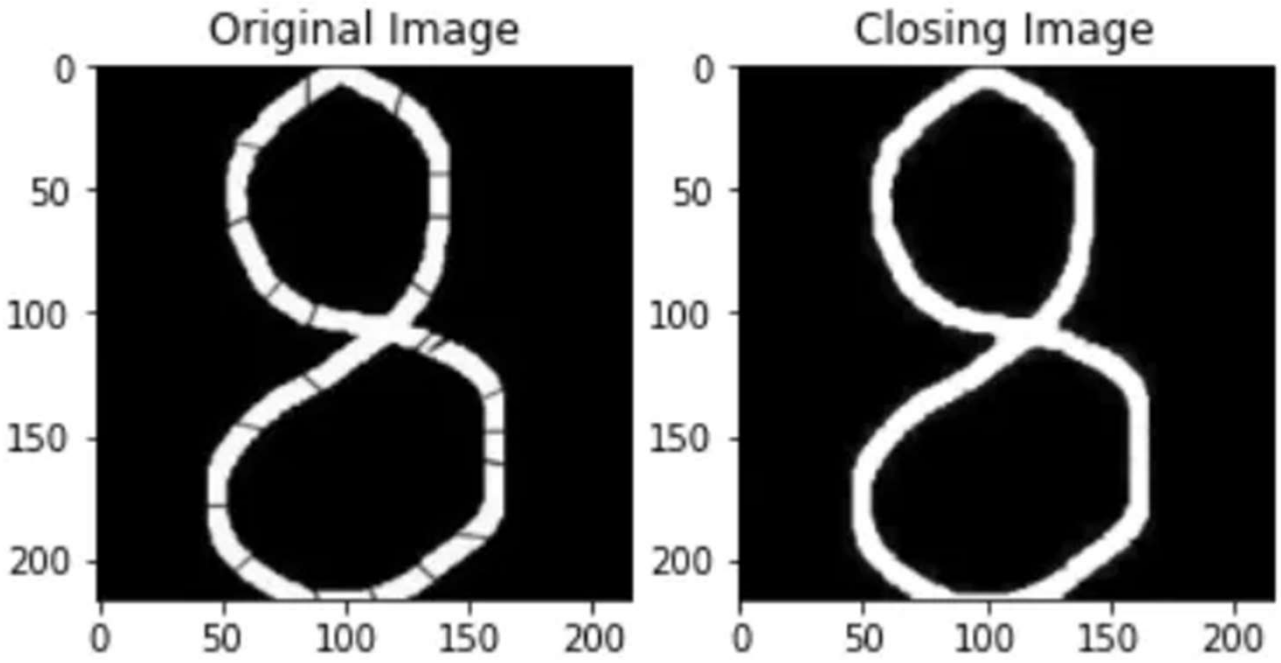
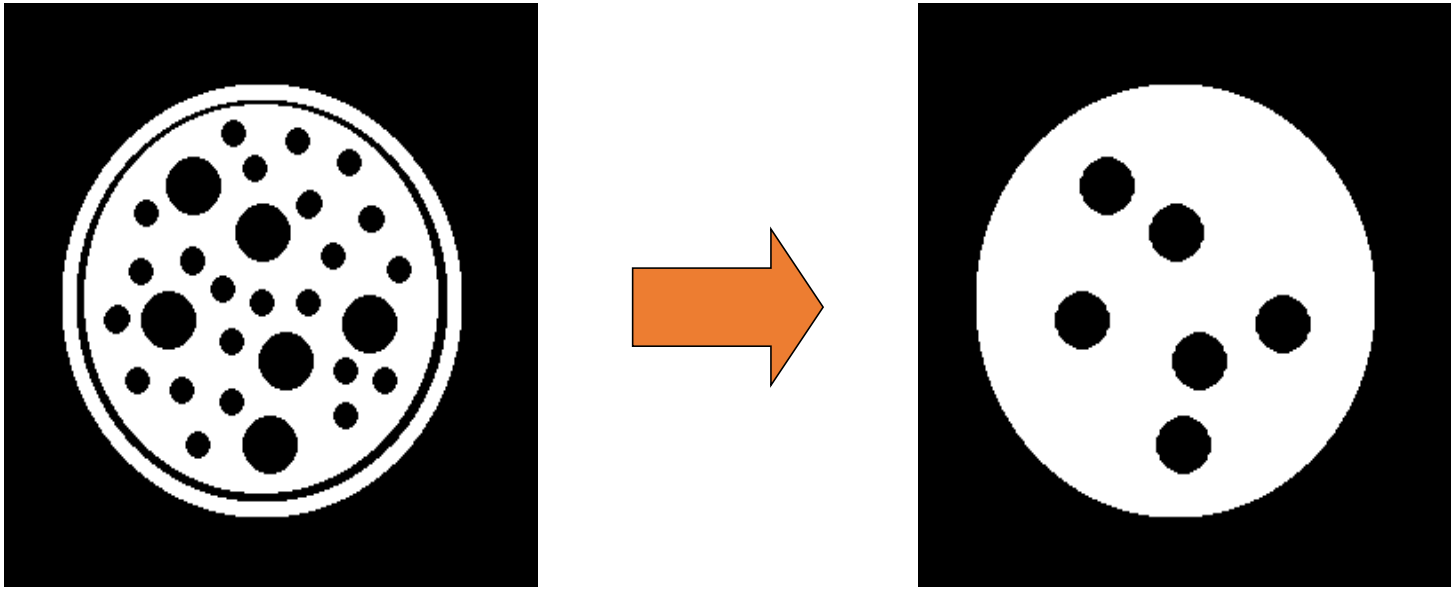


Image after closing

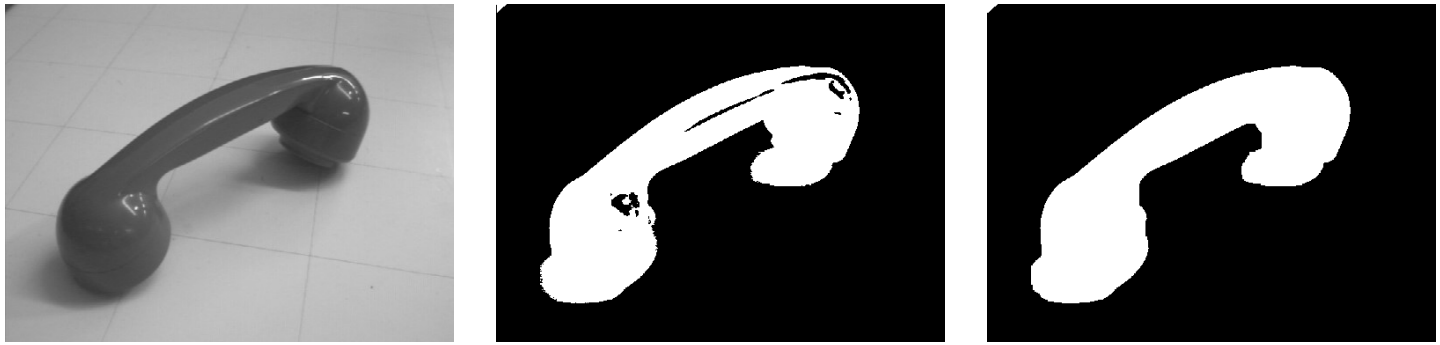
Closing dengan SE berbentuk bujursangkar 4 x 4



Closing operation with a 22 pixel disc, closes small holes in the foreground.



Threshold, closing with disc of size 20.



Note that opening is the **dual** of closing i.e. opening the foreground pixels with a particular structuring element is equivalent to closing the background pixels with the same element.

Operasi morfologi secara berulang

- Kita dapat menerapkan satu atau beberapa operasi pada suatu citra secara berulang (iterative).

- Citra masukan ---(menerapkan sebuah operasi) → citra luaran 1 --- (terapkan operasi lagi) → citra luaran 2 …dan seterusnya…..hingga mendapatkan citra yang kita inginkan

Hasil program Matlab operasi dilatasi berulang kali



Original image



after 1 dilation



after 5th dilations

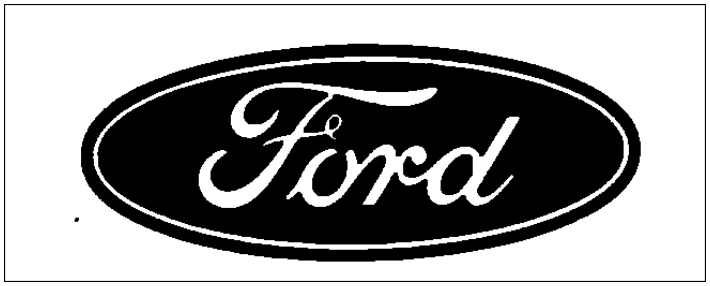


after inf dilations

Hasil program Matlab operasi erosi berulang kali



Original image



after 1 erosion



after 5th erosions

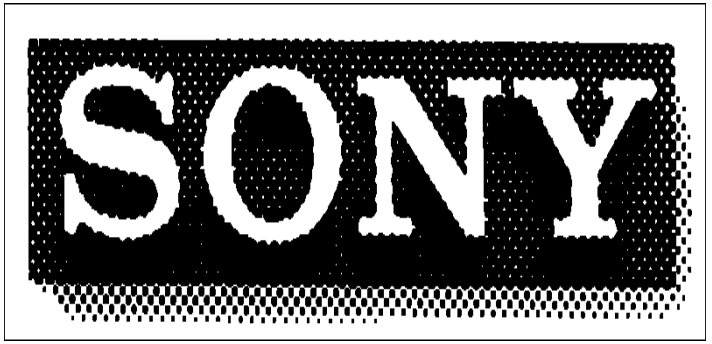


after inf erosions

Hasil program Matlab operasi erosi berulang kali



Original image



after 1 erosion



after 5th erosions



after inf erosions

Hasil program Matlab operasi opening berulang kali



Original image



after 1 opening



after 5th openings

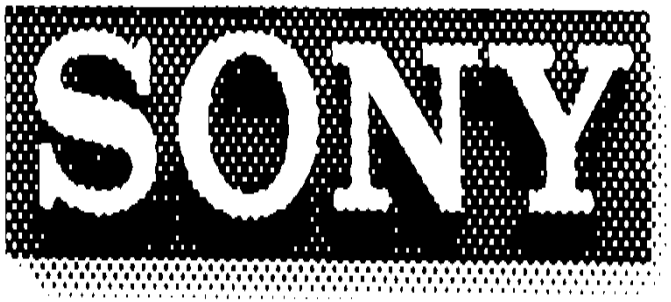


after inf openings

Hasil program Matlab operasi opening berulang kali



Original image



after 1 opening



after 5th openings

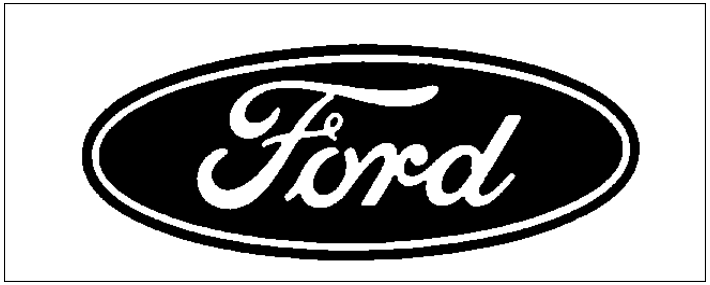


after inf openings

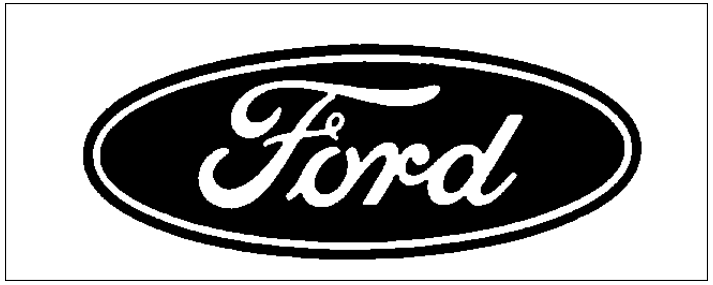
Hasil program Matlab operasi closing berulang kali



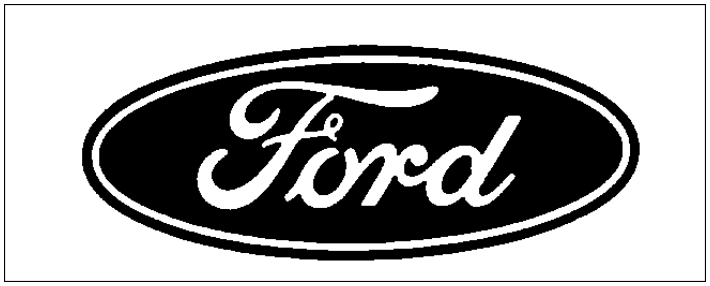
Original image



after 1 closing



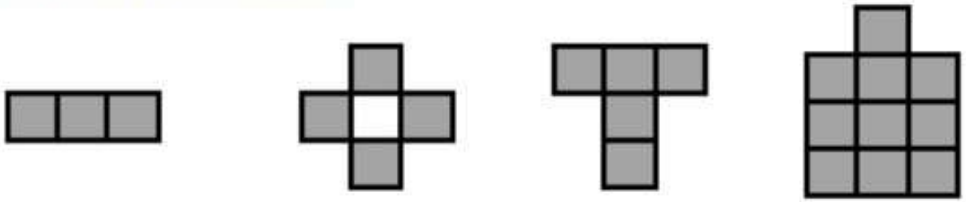
after 5th closings



after inf closnings

Hit-or-Miss Transformation ⊛ (HMT)

- find location of one shape among a set of shapes "template matching"



- composite SE: object part (B1) and background part (B2)
- does B1 ***fits the object while, simultaneously,*** B2 misses the object, i.e., ***fits the background?***

$$A \circledast B = (A \ominus X) \cap [A^c \ominus (W - X)]$$

Hit-or-Miss Transformation

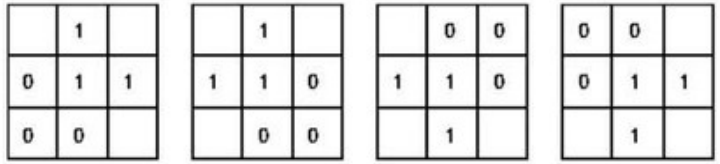
In general, if we are looking for a particular shape in an image, we design two structural elements: B1 which is the same shape, and B2 which fits around the shape. We then write B=(B1,B2) and the Hit-and-miss transform as

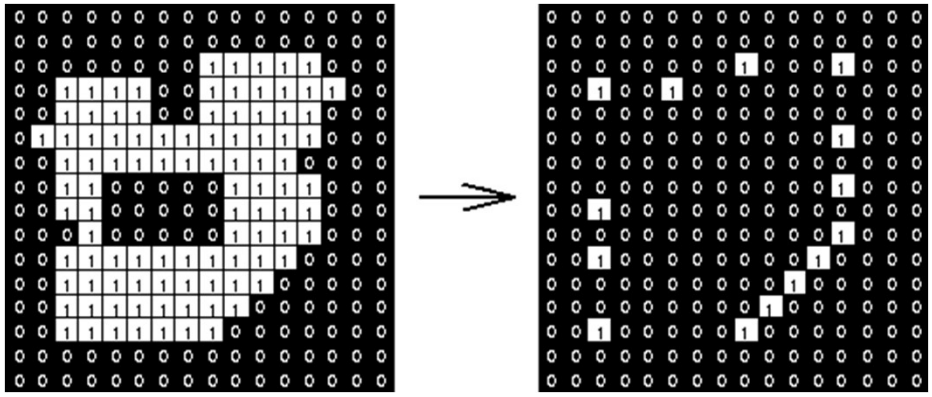
$$A \circledast B = (A \ominus B_1) \cap (\overline{A} \ominus B_2)$$

The transform searches a binary image A using a pair of disjoint structuring elements $B = (B_1, B_2)$

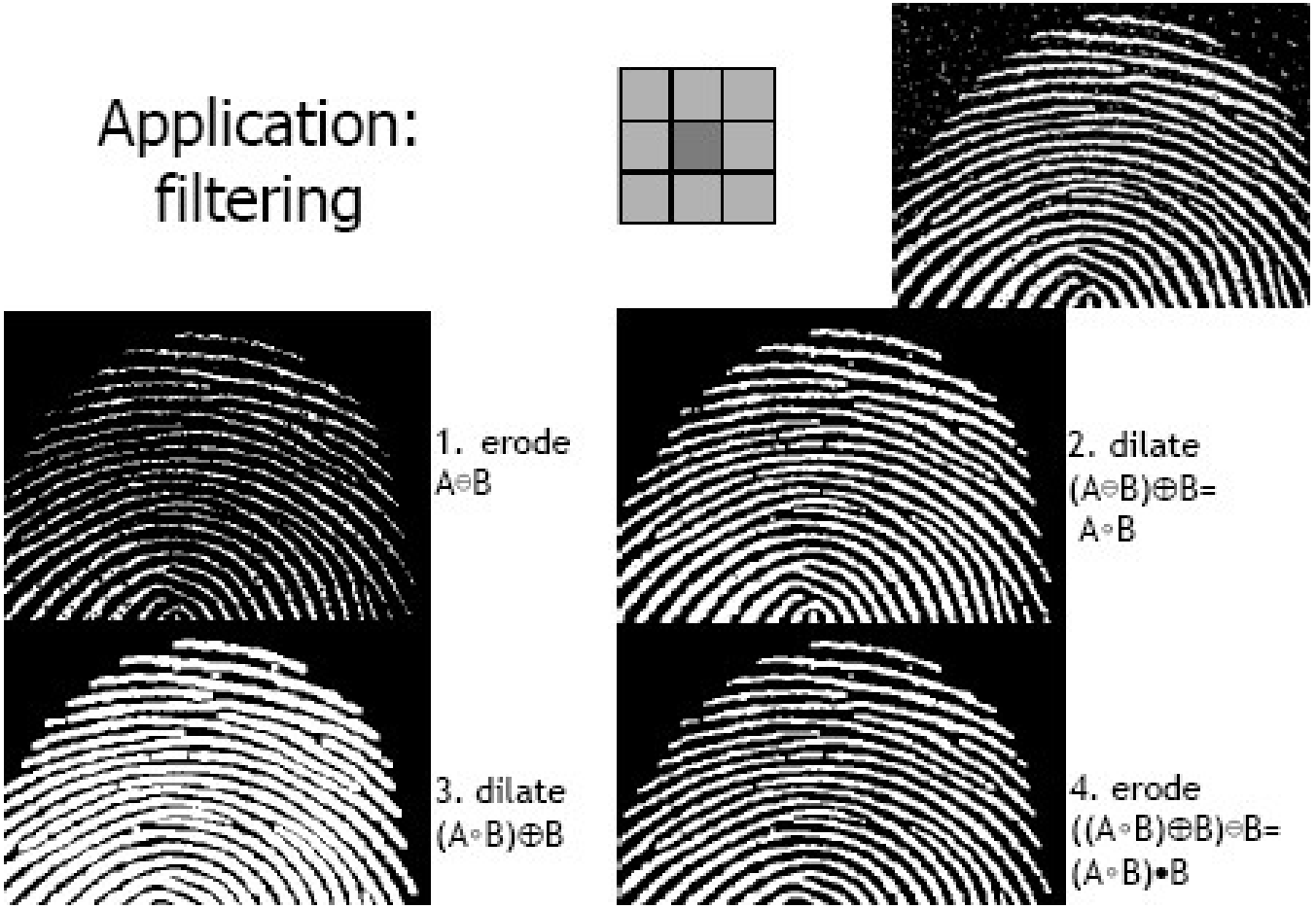
- **B_1 (The Hit):** A structuring element that matches the foreground object pixels (1s). [1]
- **B_2 (The Miss):** A structuring element that matches the background pixels (0s). [1]

Hit-or-Miss Transformation





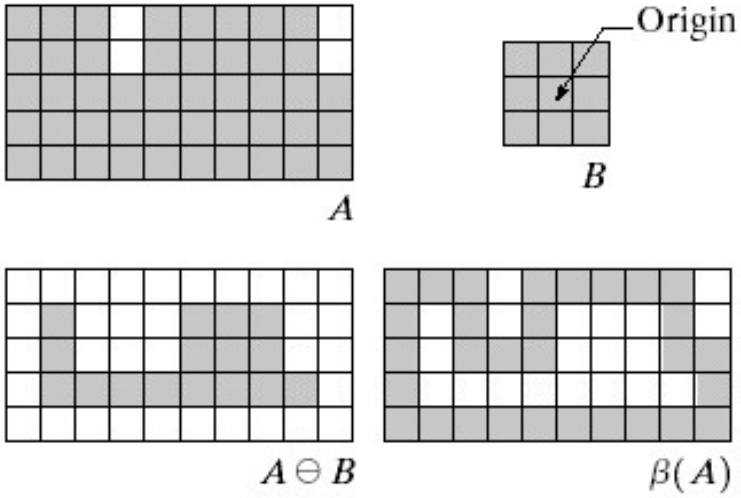
Application: 1. filtering



Application: 2.Boundary Extraction

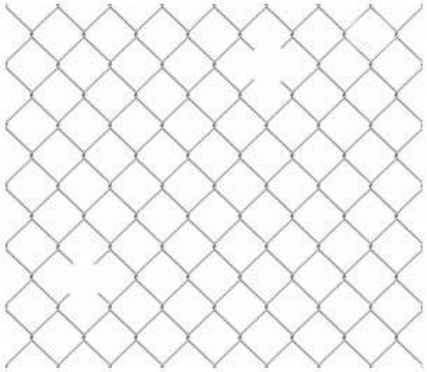
a b
c d

FIGURE 9.13 (a) Set A. (b) Structuring element B. (c) A eroded by B. (d) Boundary, given by the set difference between A and its erosion.

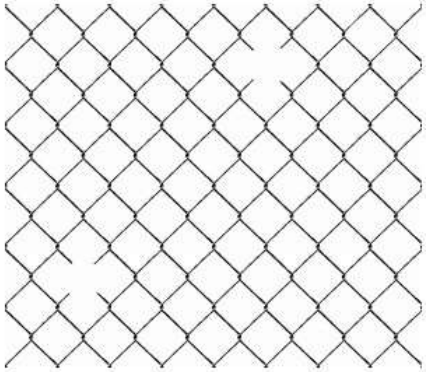


$$\beta(A) = A - (A \ominus B)$$

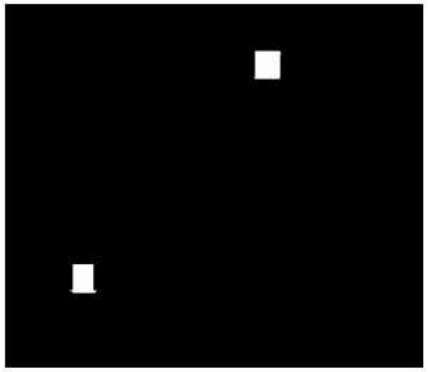
Example: chain link fence hole detection



Original grayscale image
Fence (1023 x 1173*)*

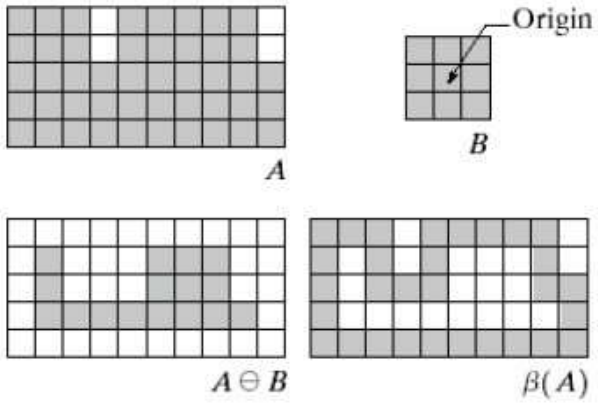


Fence thresholded
using Otsu's method

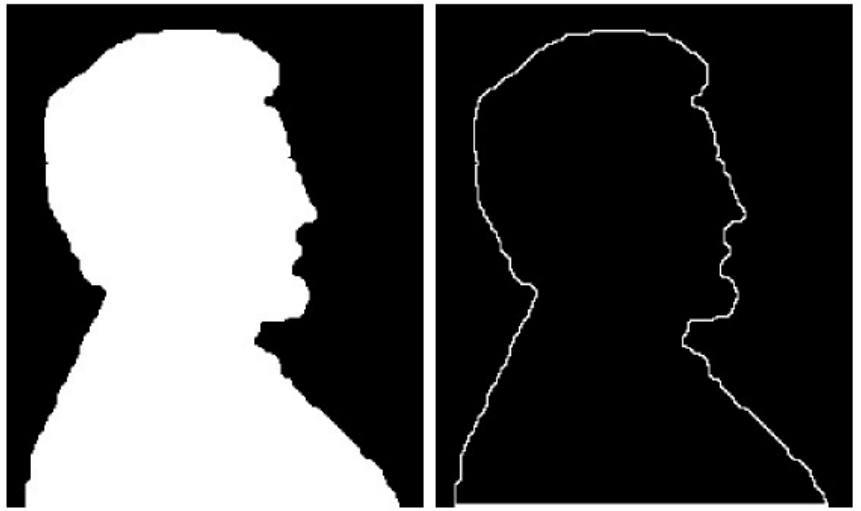


Erosion with 151x151
"cross" structuring element

Example: Boundary Extraction



$$\beta(A) = A - (A \ominus B)$$



a b

FIGURE 9.14
(a) A simple binary image, with 1's represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

Referensi

Materi operasi morfologi di dalam PPT ini diambil dari sini:

1. Xueyan Li, *CIS581 Presentation, Morphological Operations*
2. Digital Image Processing, *Lecture 5 Morphological Image Processing*