

24 - Citra Biner

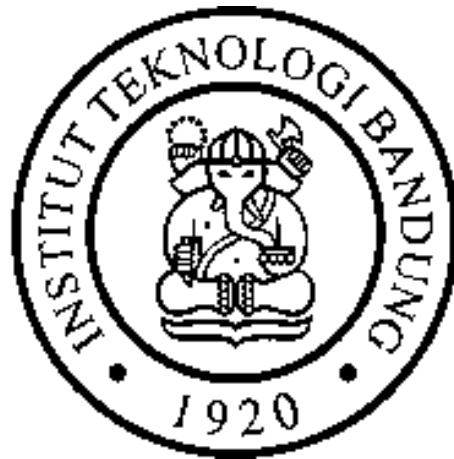
IF4073 Interpretasi dan Pengolahan Citra

Oleh: Rinaldi Munir



Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2023

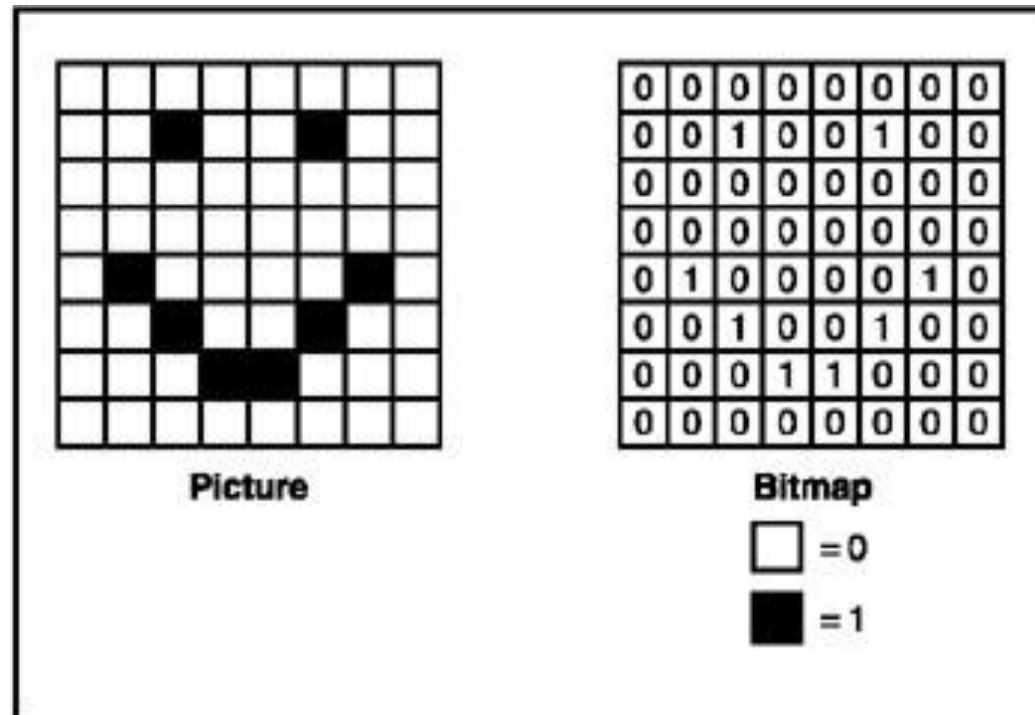
Citra Biner

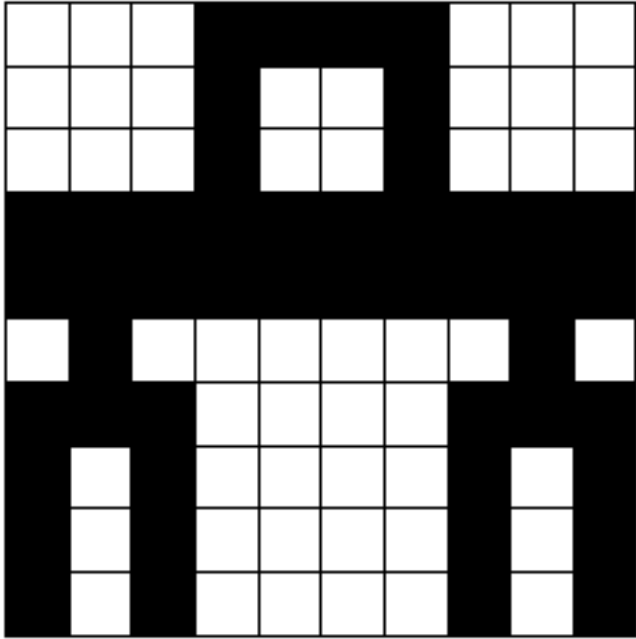




Citra Biner

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





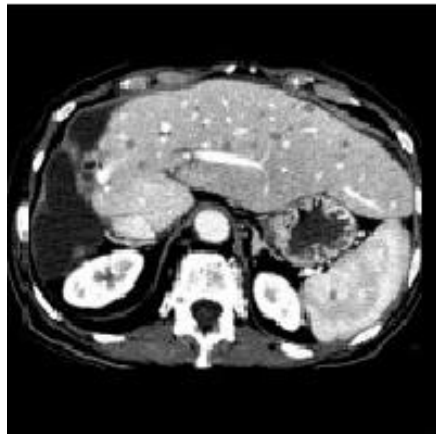
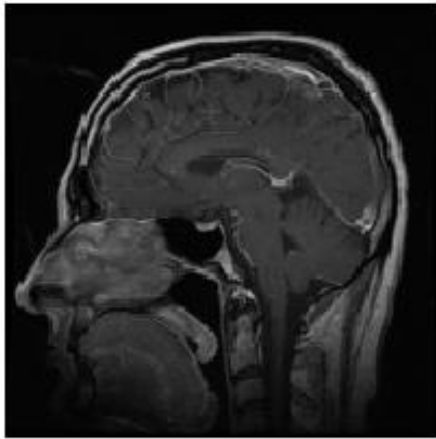
0	0	0	1	1	1	1	0	0	0
0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	1	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	1	1	1
1	0	1	0	0	0	0	1	0	1
1	0	1	0	0	0	0	1	0	1
1	0	1	0	0	0	0	1	0	1

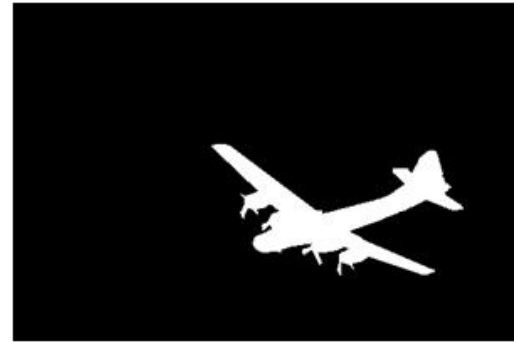
Mengapa perlu citra biner?

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*)

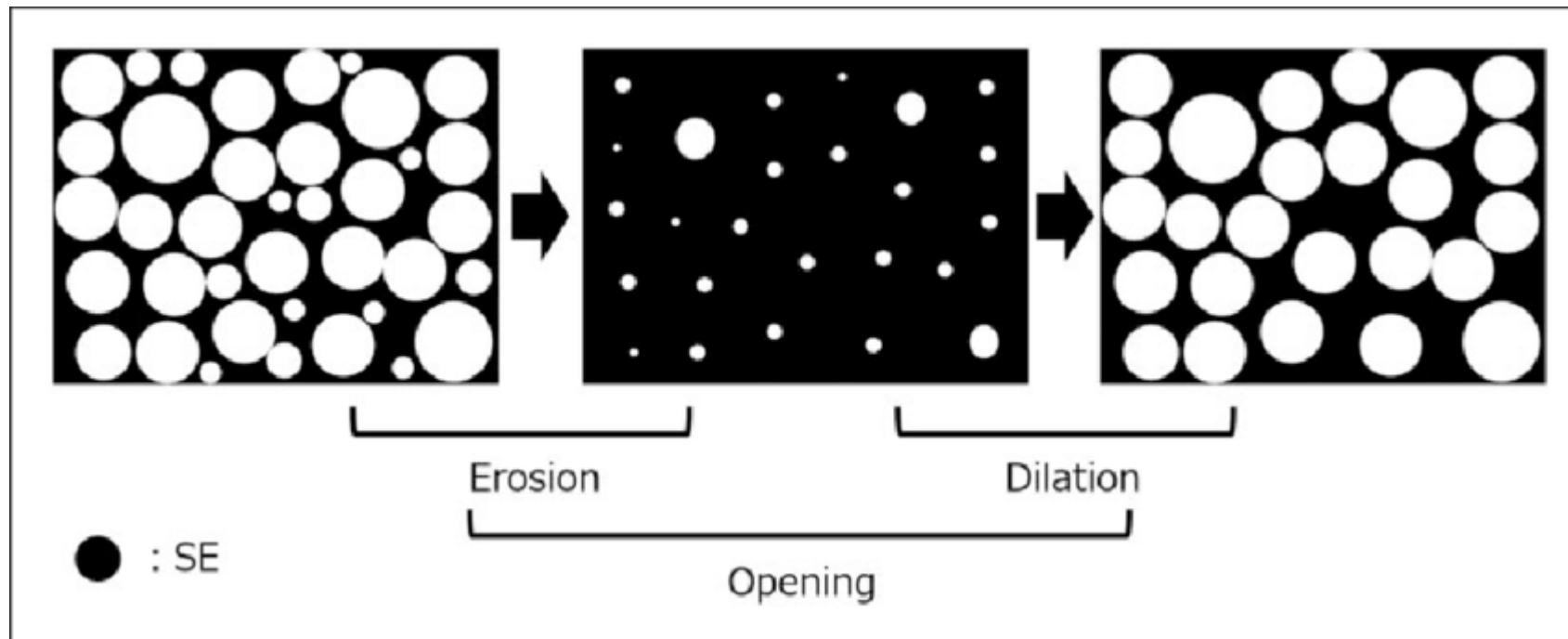


4. Untuk memisahkan (segmentasi) objek dari gambar latar belakangnya → hasil operasi pengambangan





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*).

Only need to print
in **Black & White?**



EPSON

ecotank

UP TO **2** YEARS
INK INCLUDED

Less than half a cent per page

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

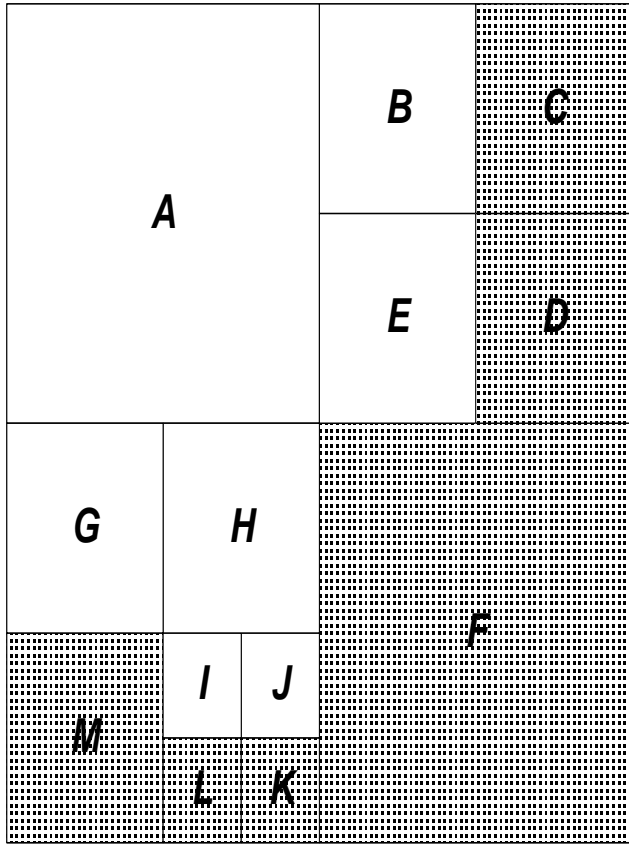
3, 3, 2, 3, 4, 1, 2, 1, 3

0, 4, 13, 1, 4

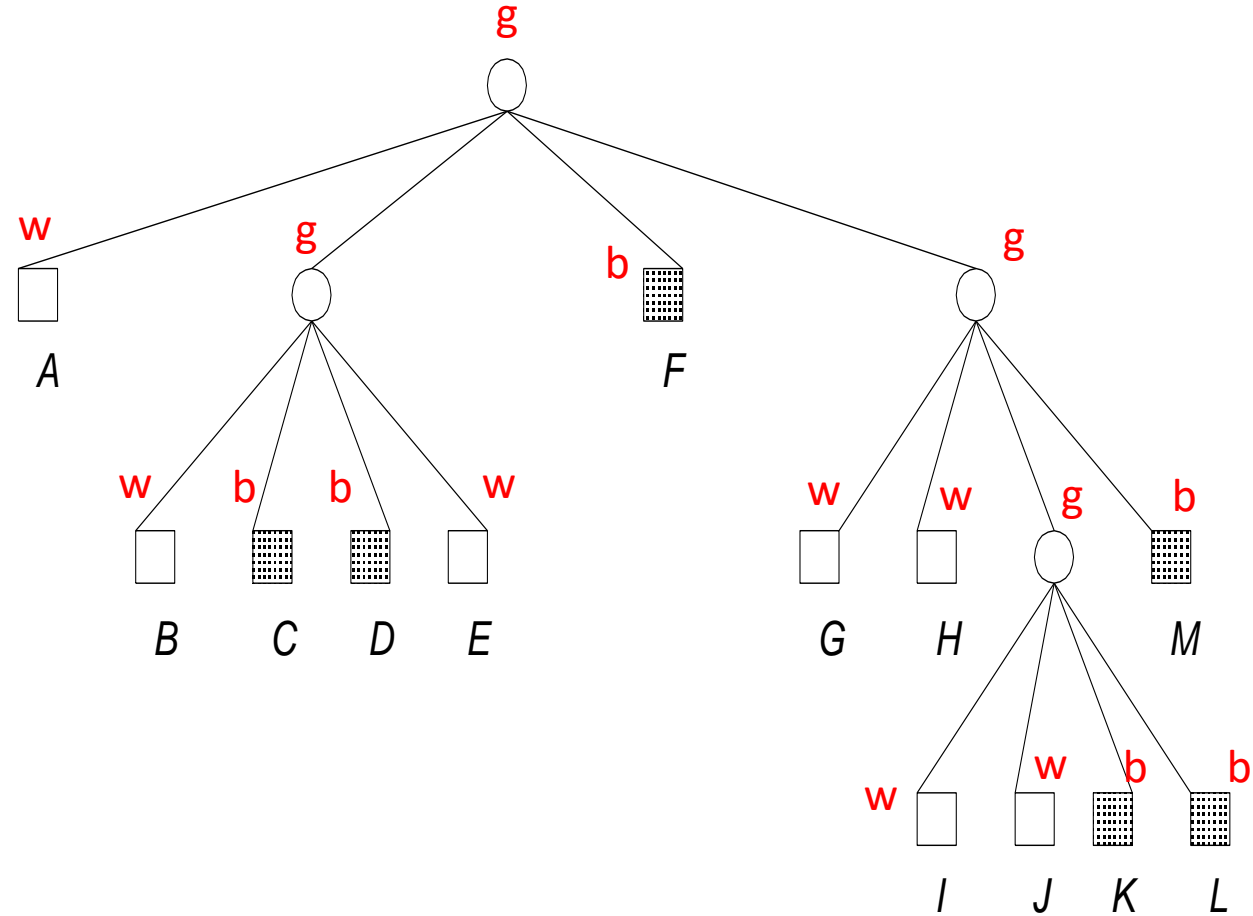
3, 13, 6

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 menjadi empat bagian.
 4. Demikian seterusnya sampai diperoleh upa-wilayah yang semua *pixel*-nya hitam atau semua *pixel*-nya putih



Citra biner



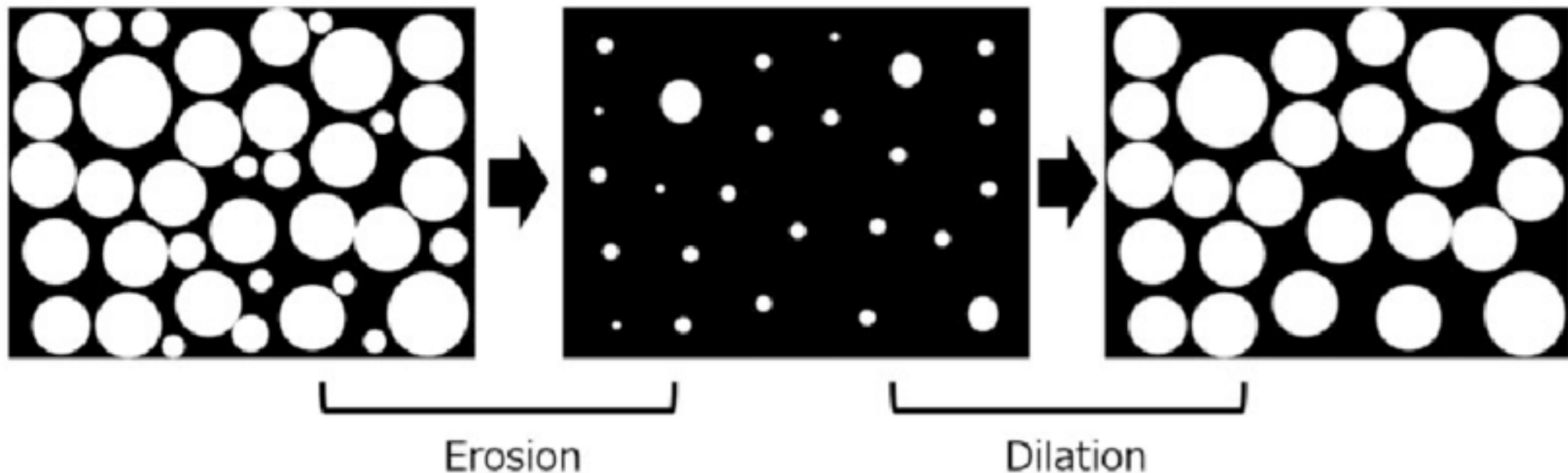
Kode: **gwgwbbwbgwwgwwgwwbbb**

Di-decode sebagai: **$g(wg(wbbw)bg(wwg(wwbb)b))$**

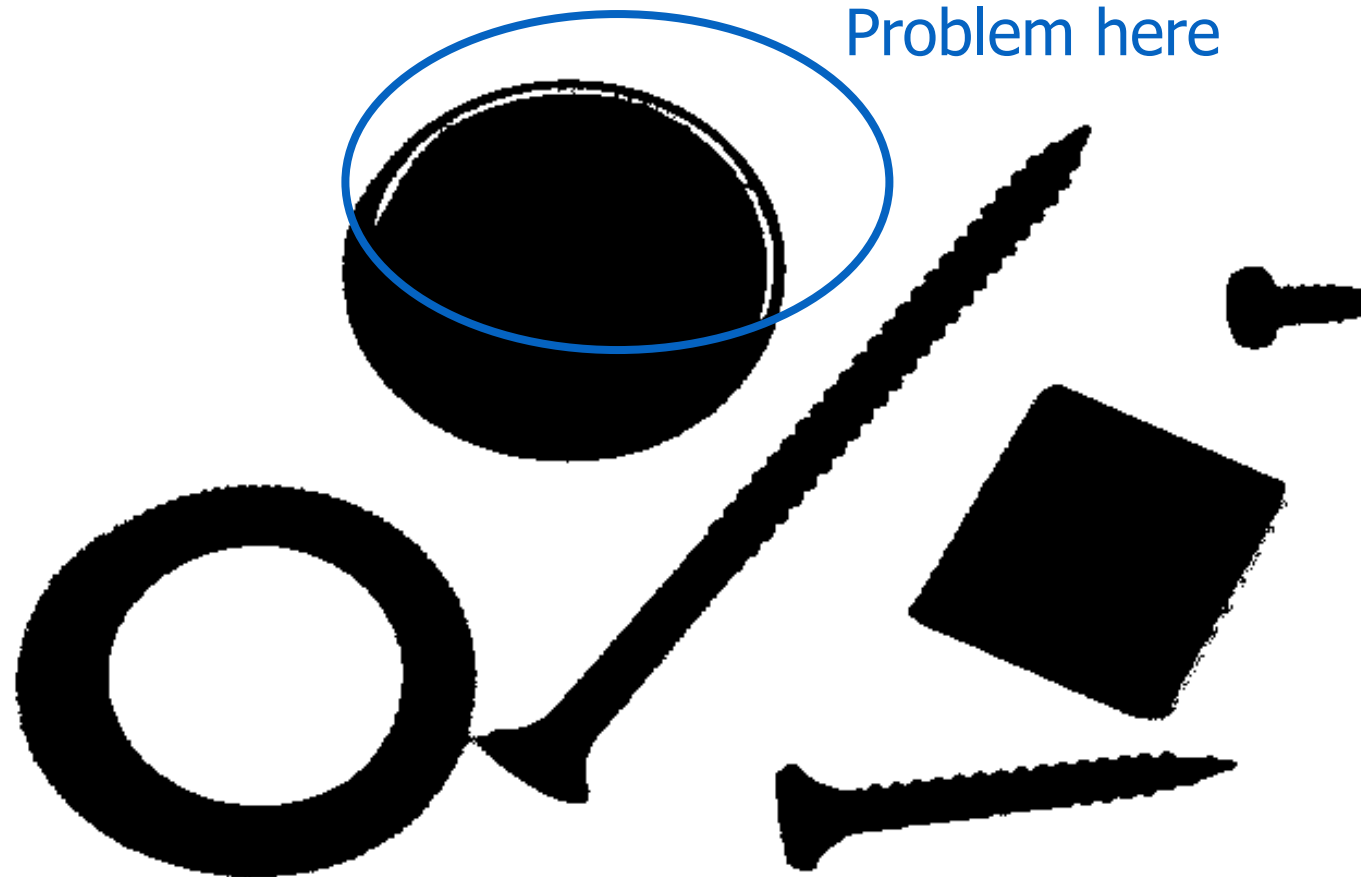
Keterangan: *b = black, w = white, g = gray*

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



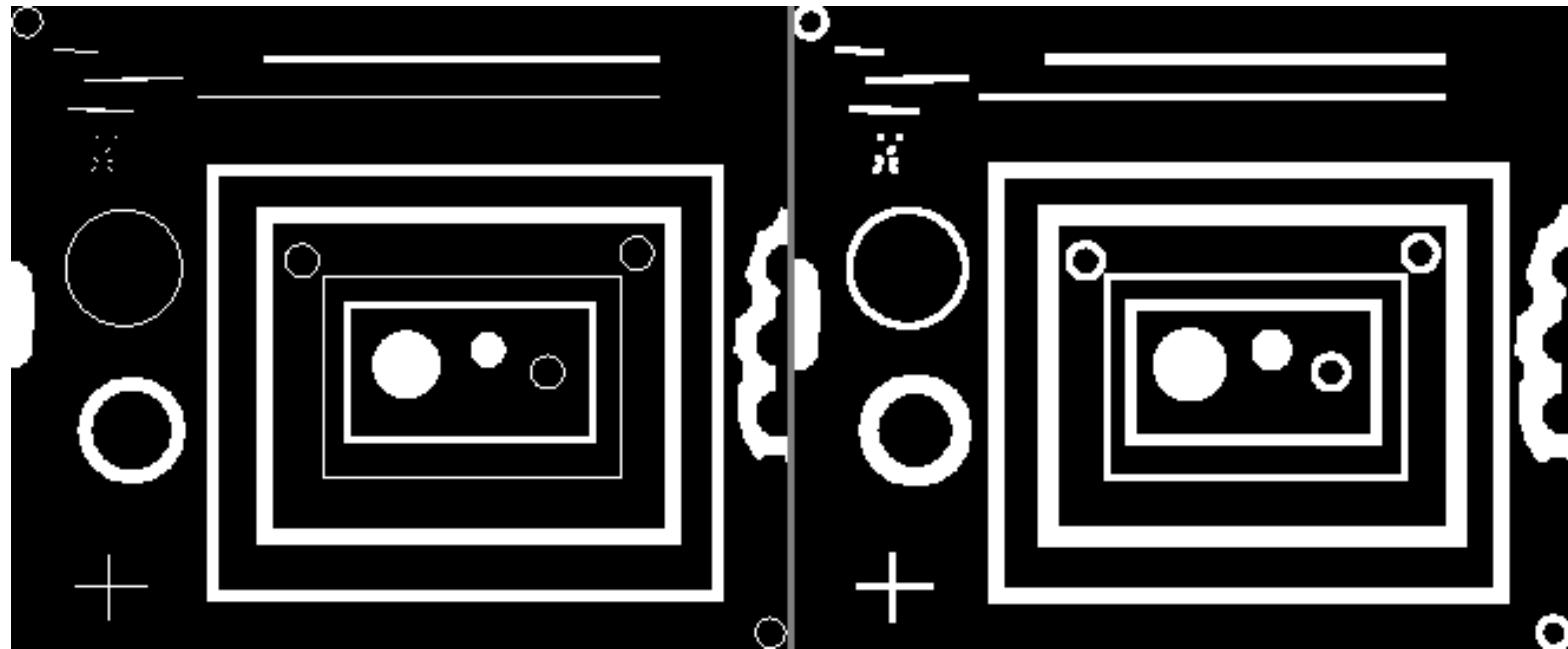
Empat operasi morfologi

- Dilatasi (*dilation*)
- Pengikisan atau erosi (*erosion*)
- Pembukaan (*opening*)
- Penutupan (*closing*)

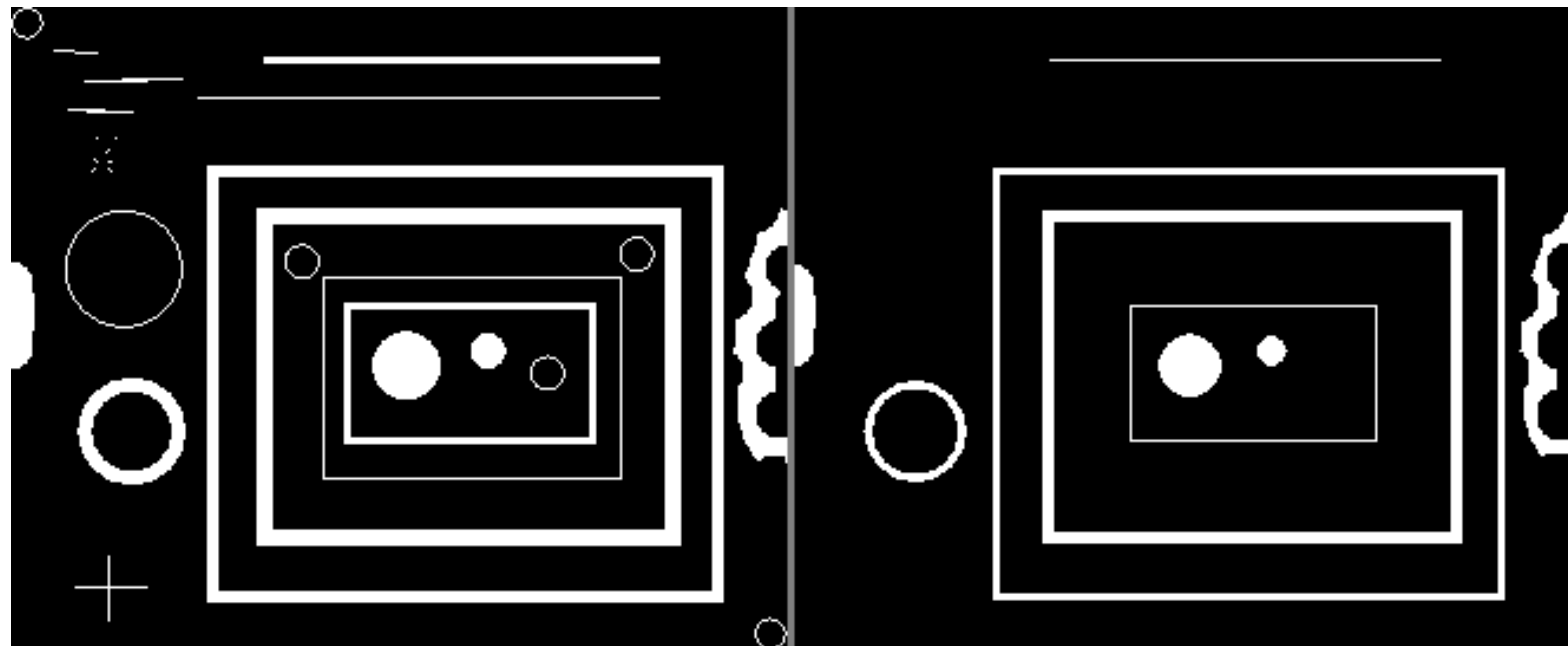
Hubungan keempat operasi morfologi tersebut:

- Operasi **dilatasi** 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.

Dilatasi



Erosi



- Erosion

shrink

- Dilation

grow

- combine to

- Opening



object

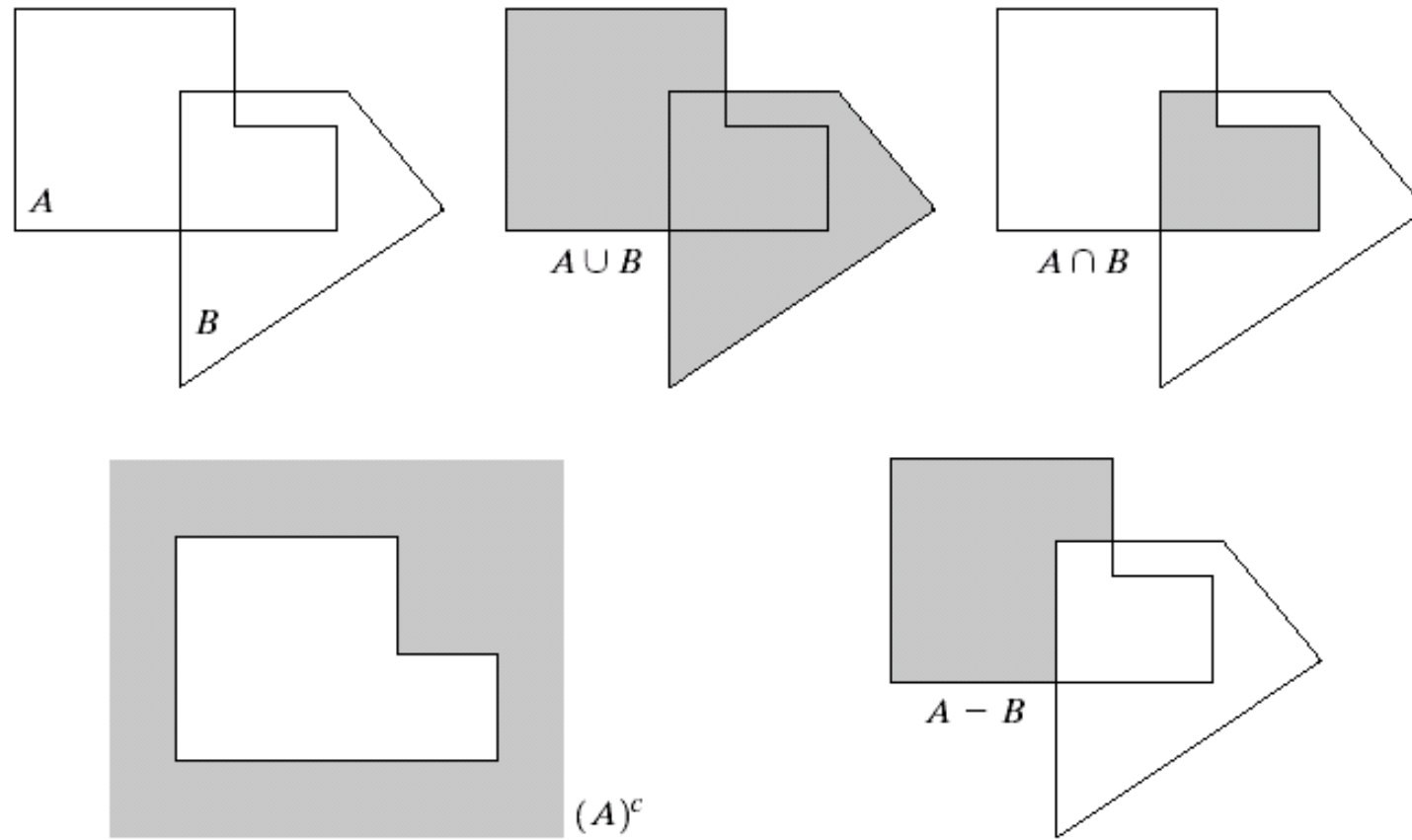
- Closing



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).

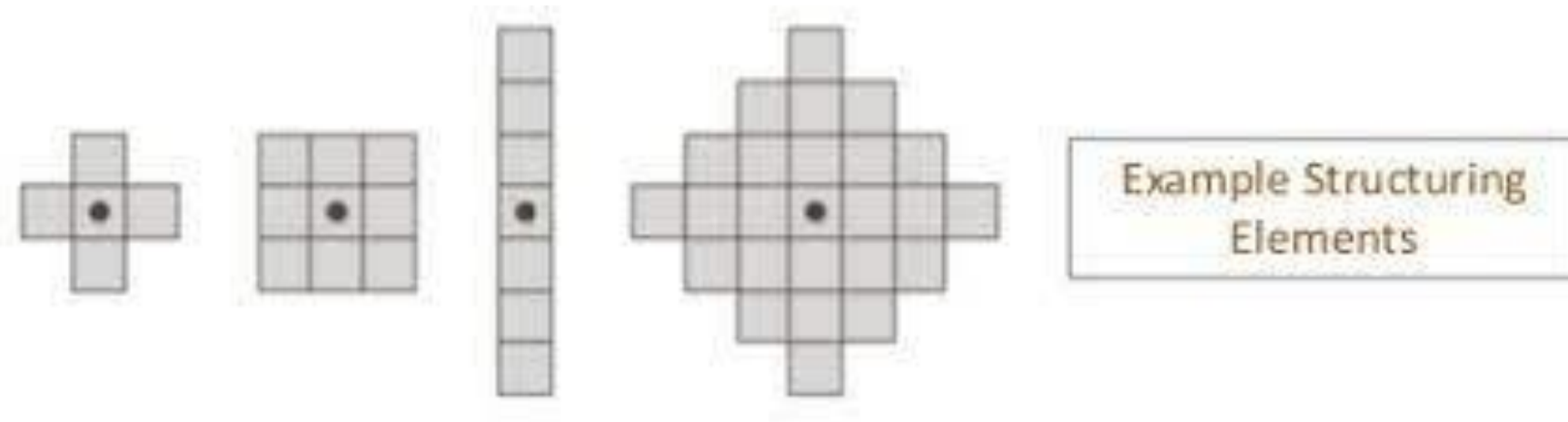


a	b	c
d	e	

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*)



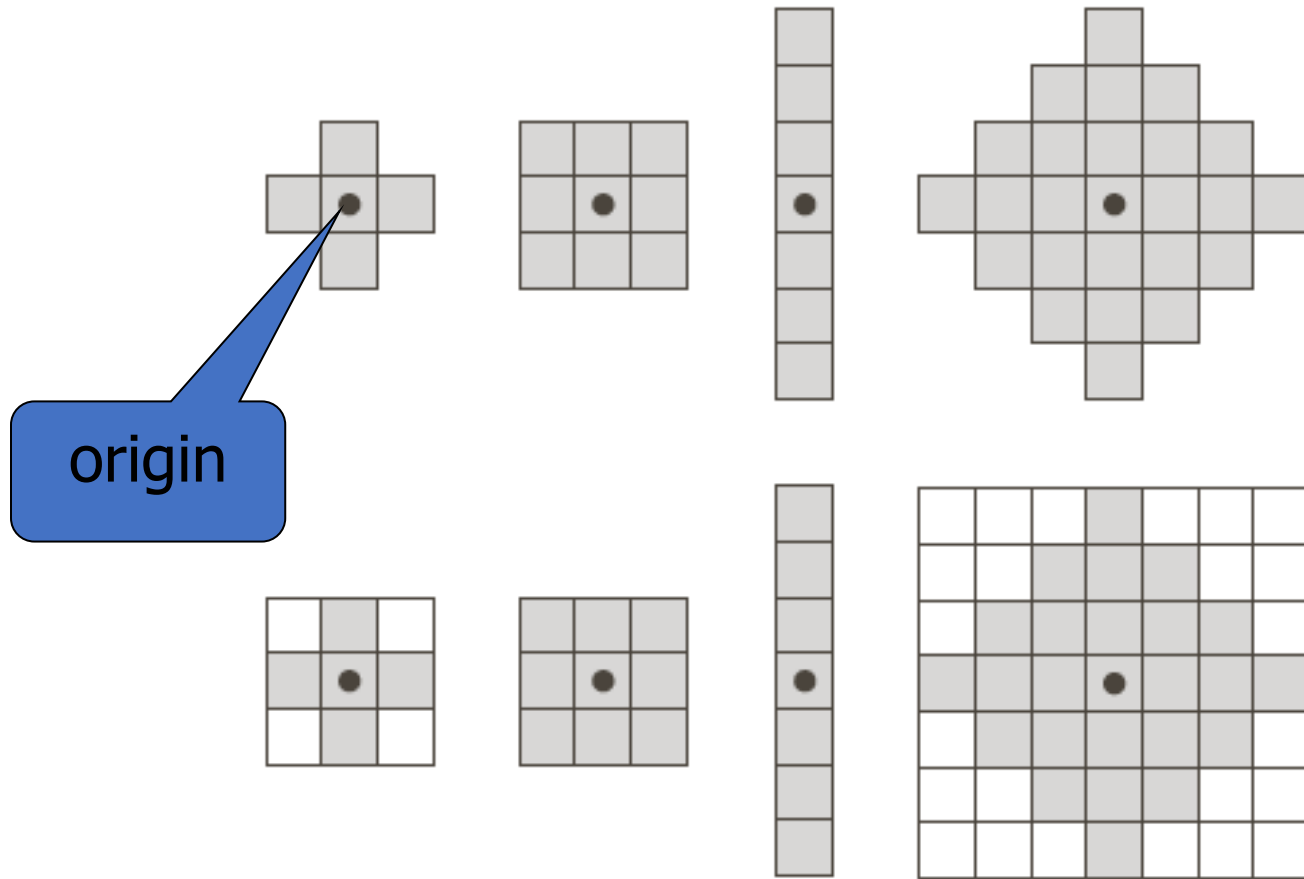
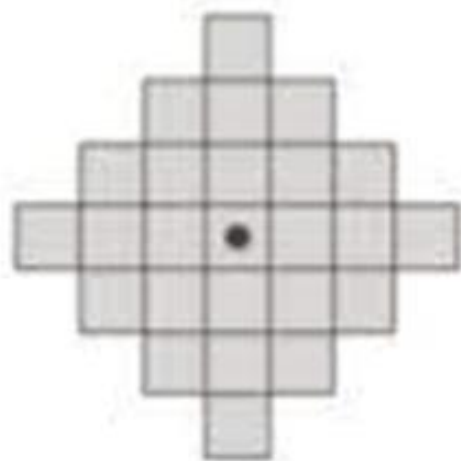
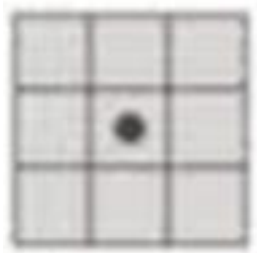


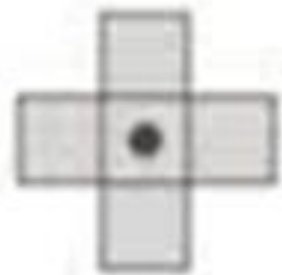
FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.



		1	1	1		
	1	1	1	1	1	
1	1	1	1	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	

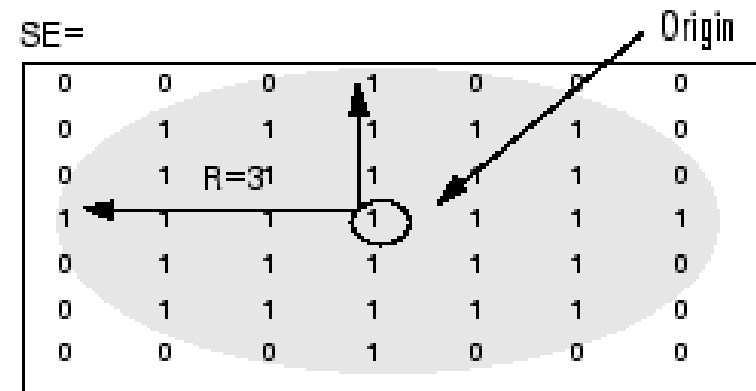
1	1	
1	0	
1		0

1	1	1
1	1	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



Dilatasi (*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, lakukan operasi logika OR pada semua piksel dalam elemen penstruktur.

Dilation

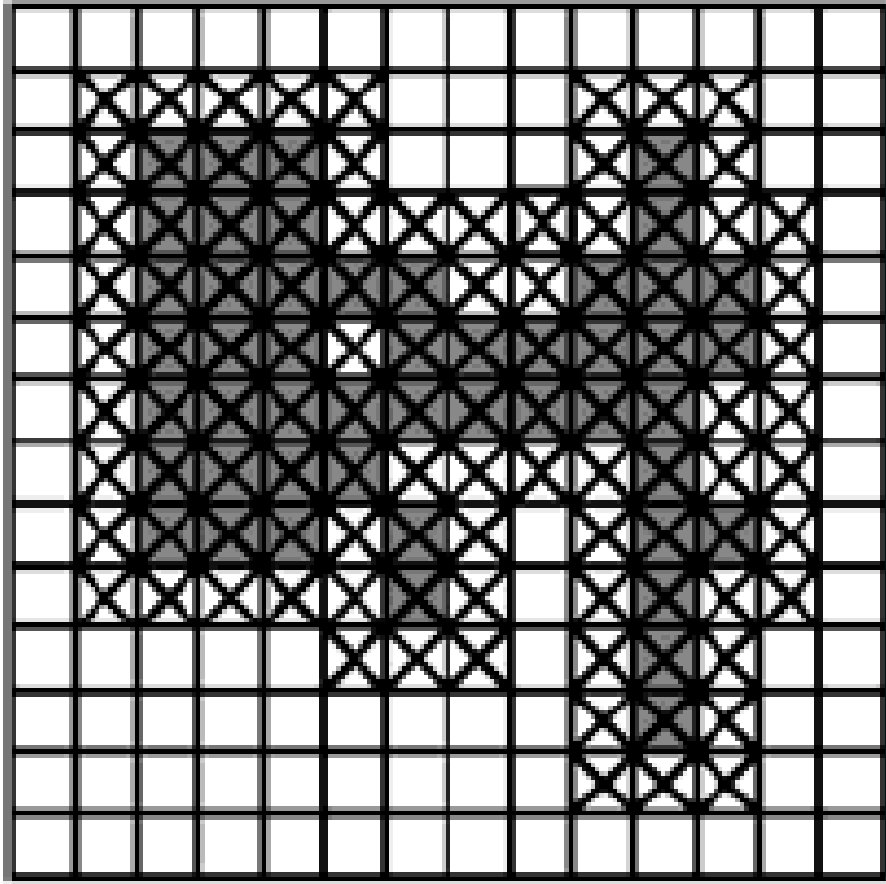
Definition

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

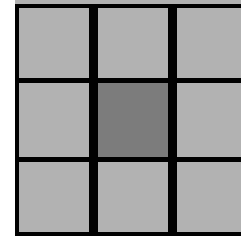
$$I \oplus H = \{(\mathbf{p} + \mathbf{q}) \mid \mathbf{p} \in I, \mathbf{q} \in H\}$$

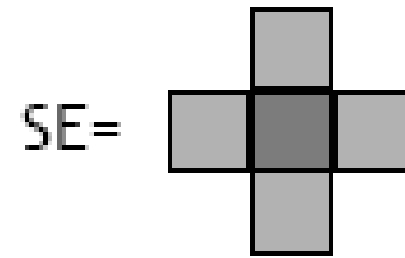
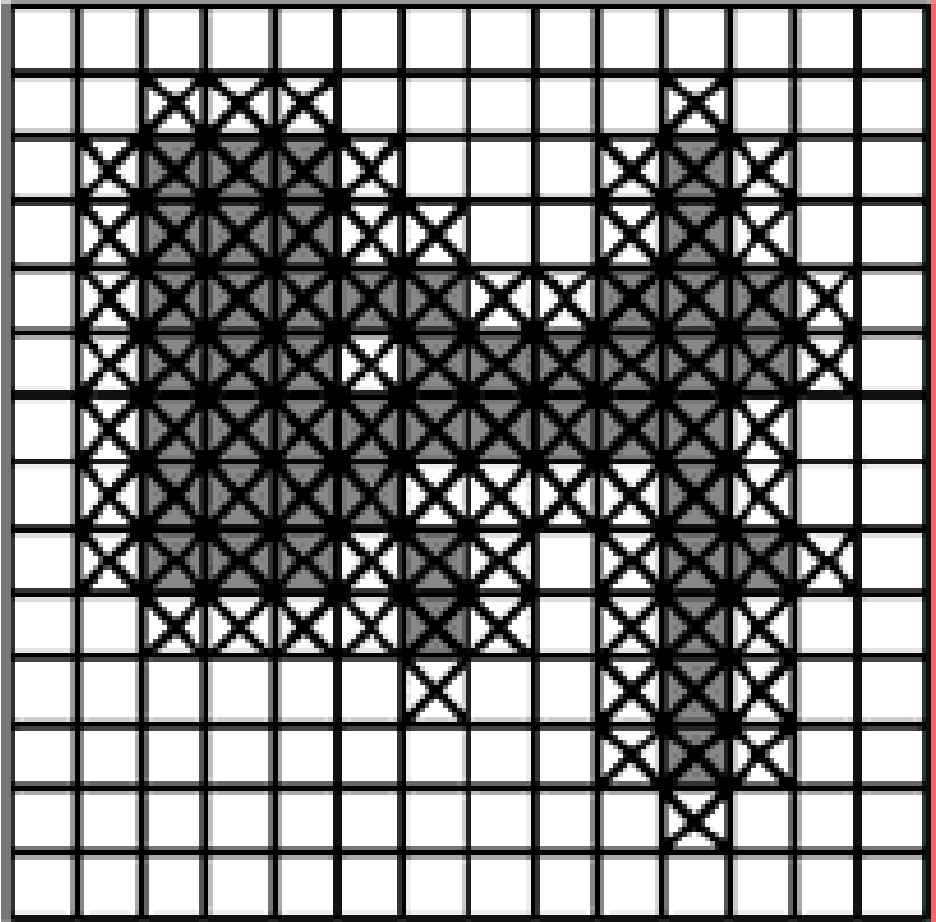
Alternative definition: Take the union of copies of the structuring element, $H_{\mathbf{p}}$, centered at every pixel location \mathbf{p} in the foreground:

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



SE =





```
%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



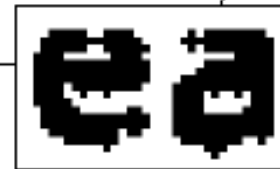
Image after dilation

Dilation : Bridging gaps

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



0	1	0
1	1	1
0	1	0

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.

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):
 - 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, dan salah satu pixel '1' pada elemen penstruktur melampaui objek (pixel '1') pada citra, maka ubah pixel '1' pada citra tersebut menjadi '0';

Erosion

Definition

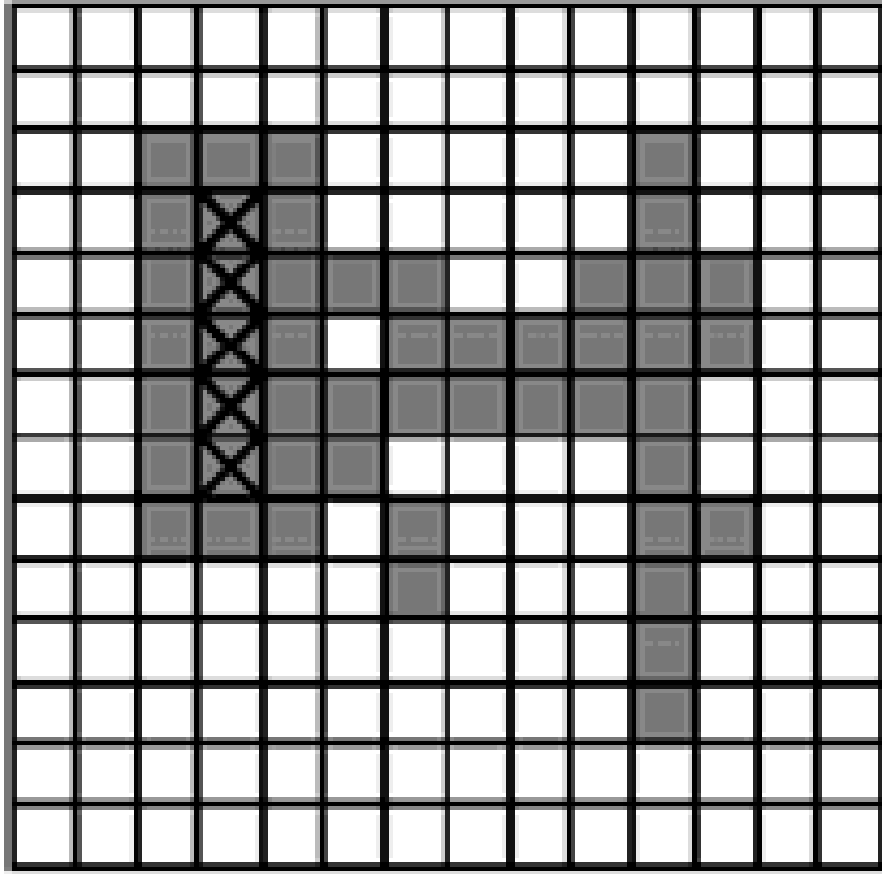
A **erosion** of an image I by the structure element H is given by the set operation

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

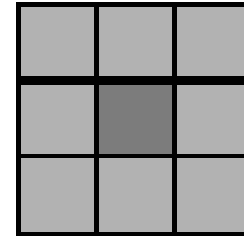
Alternative definition: Keep only pixels $\mathbf{p} \in I$ such that $H_{\mathbf{p}}$ fits inside I :

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

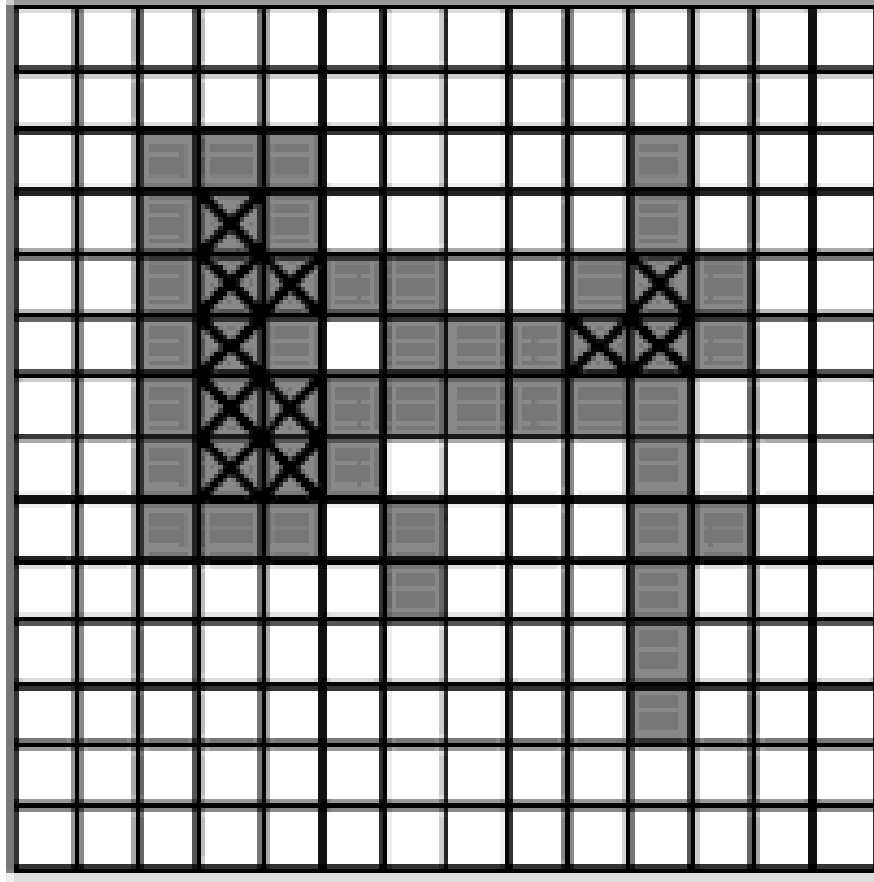
Erosion



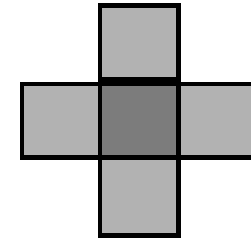
SE=



Erosion



SE=





original image

```
%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');
```

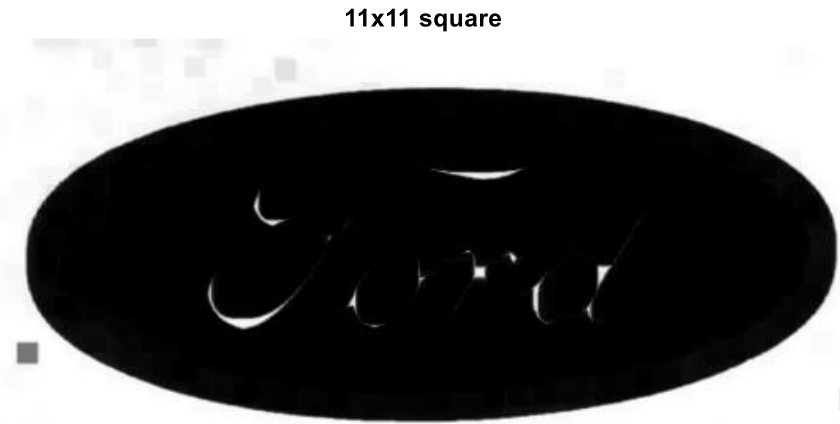
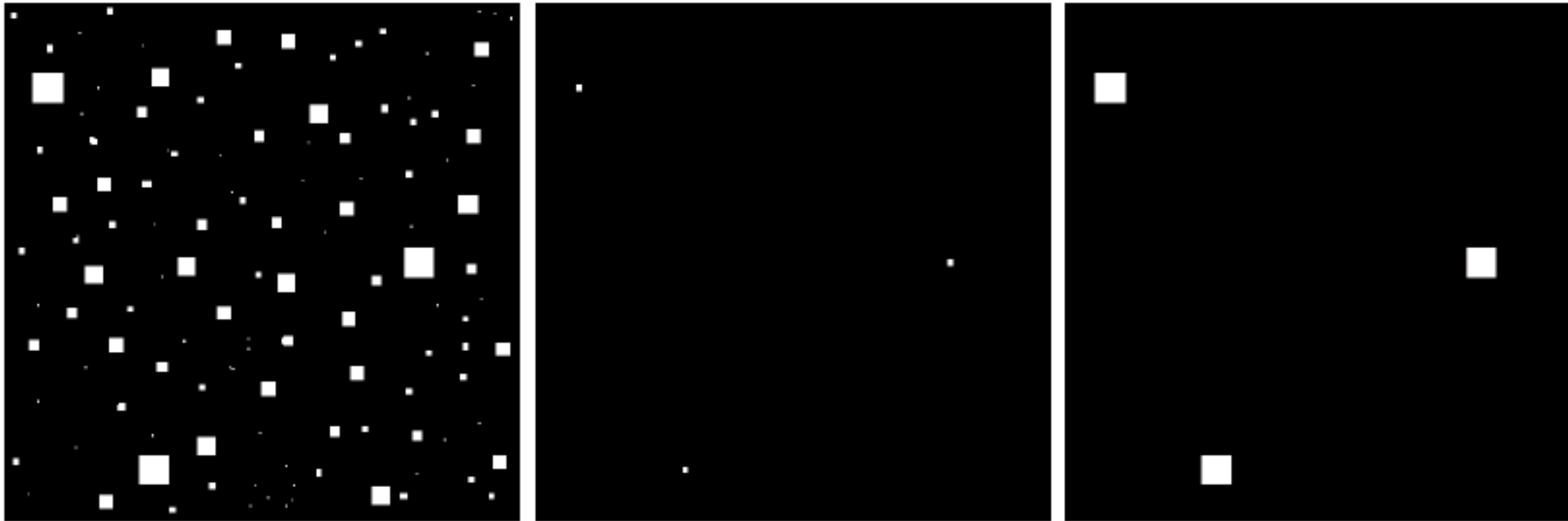


Image after erosion

Erosion : eliminating irrelevant detail



a b c

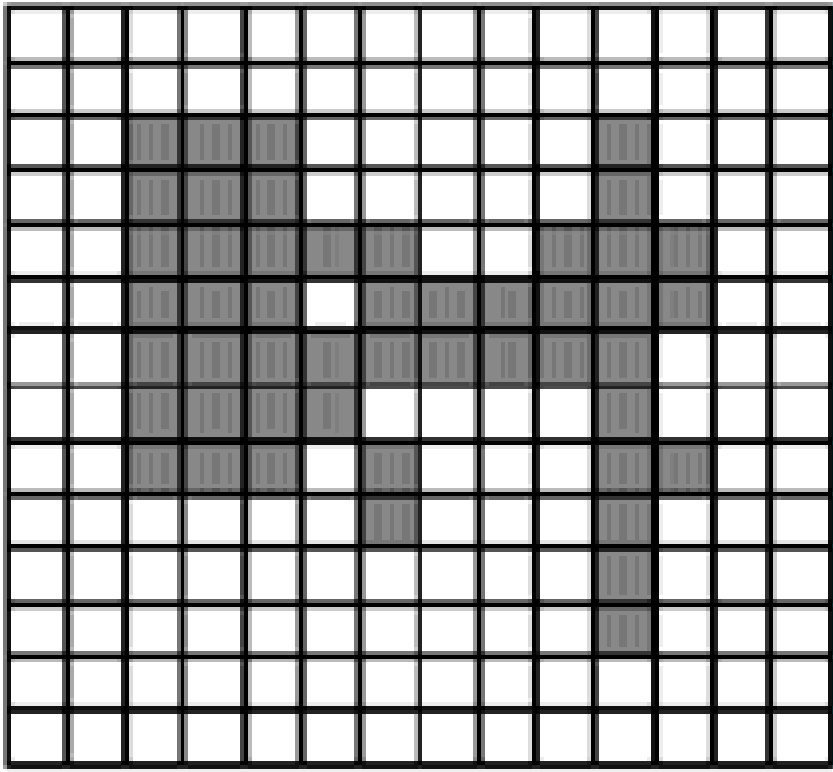
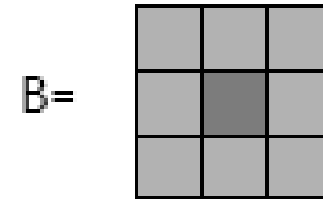
FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

structuring element $B = 13 \times 13$ pixels of gray level 1

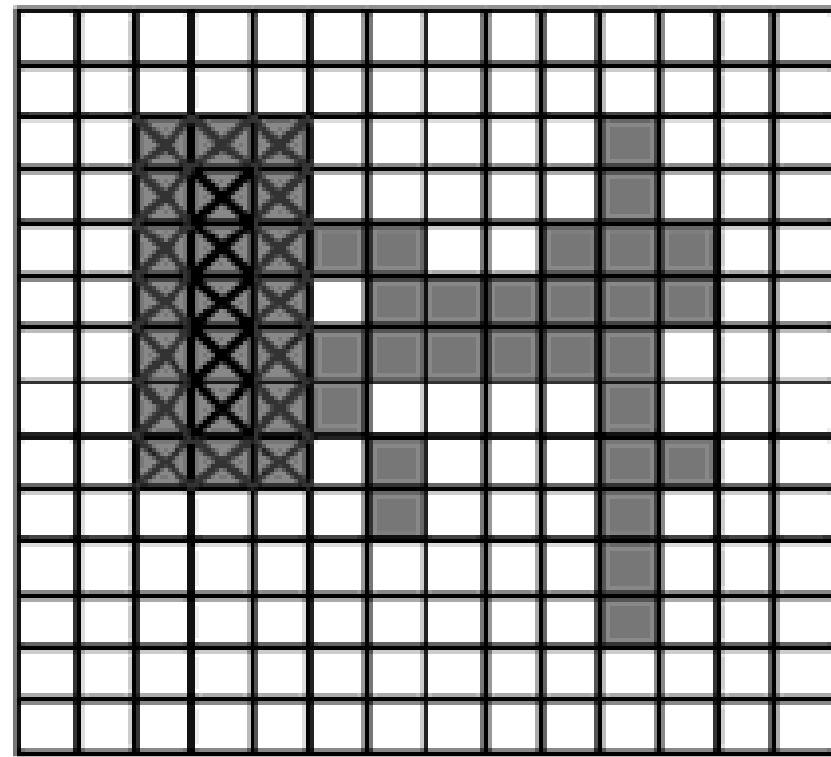
Pembukaan (*Opening*)

- 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

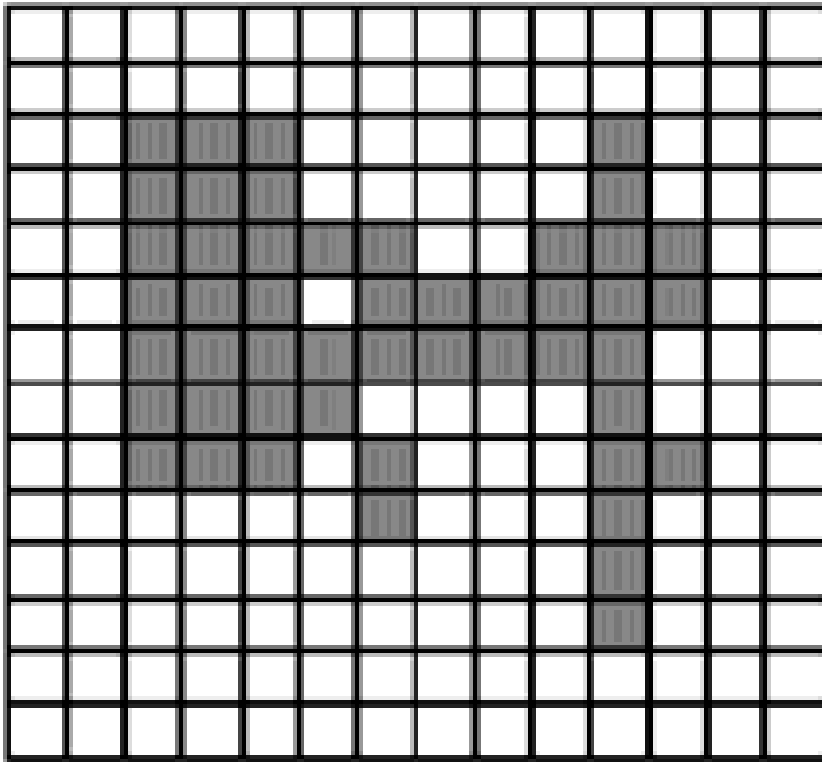
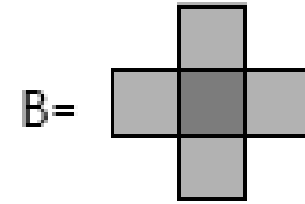


A

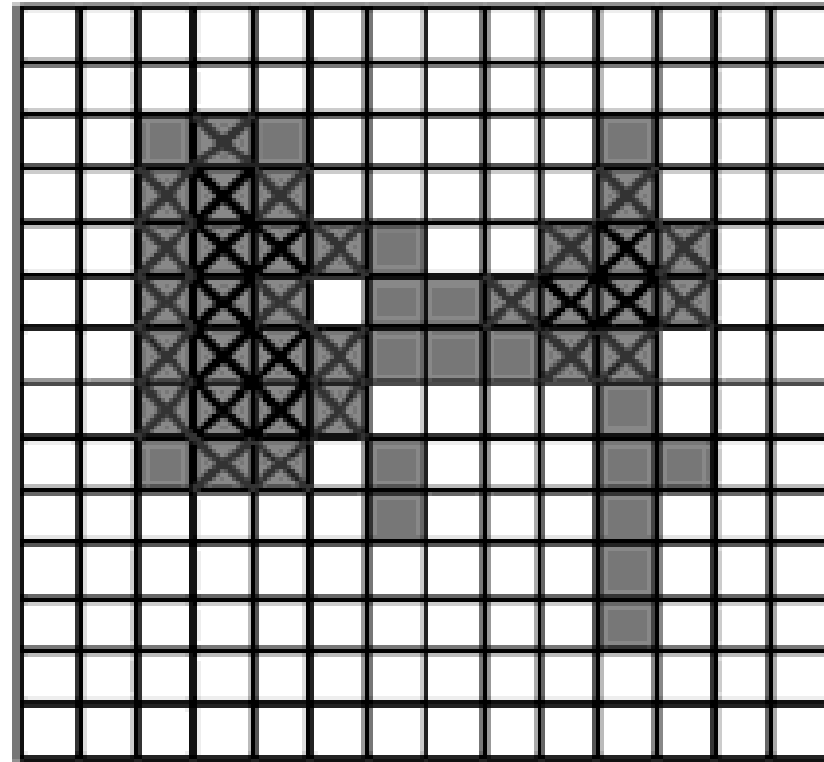


$A \ominus B$ $A \circ B$

Opening



A



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


```
%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');
```



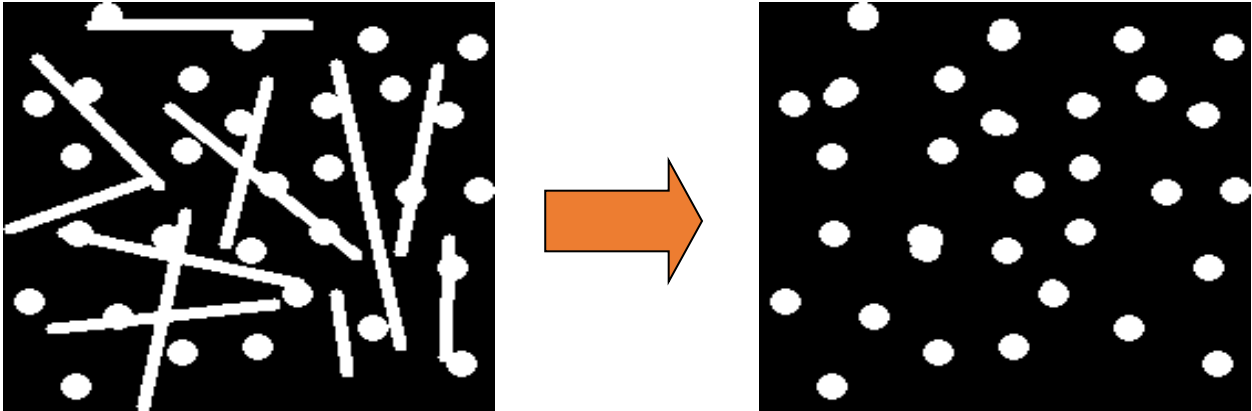
original image

11 x 11 square

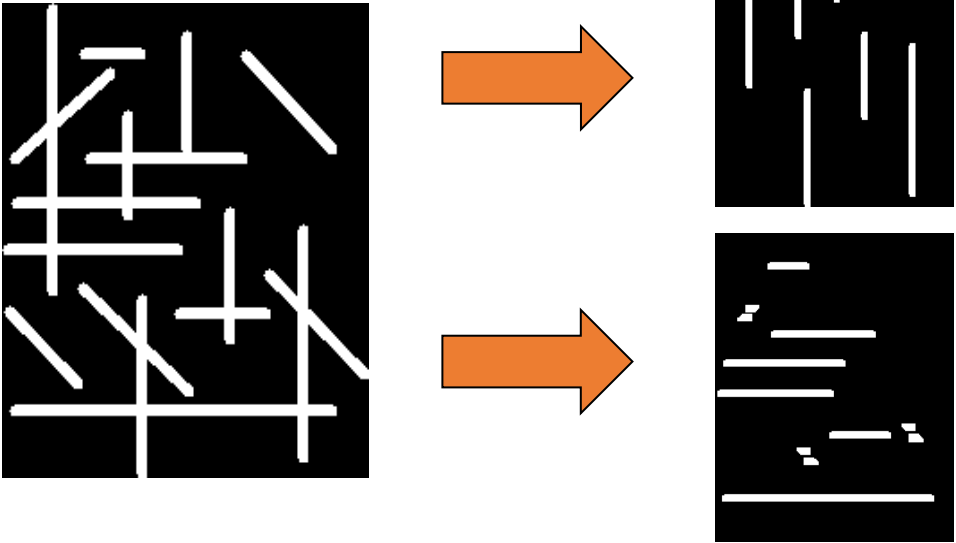


Image after opening

Opening with a 11 pixel diameter disc:



3x9 and 9x3 Structuring Element



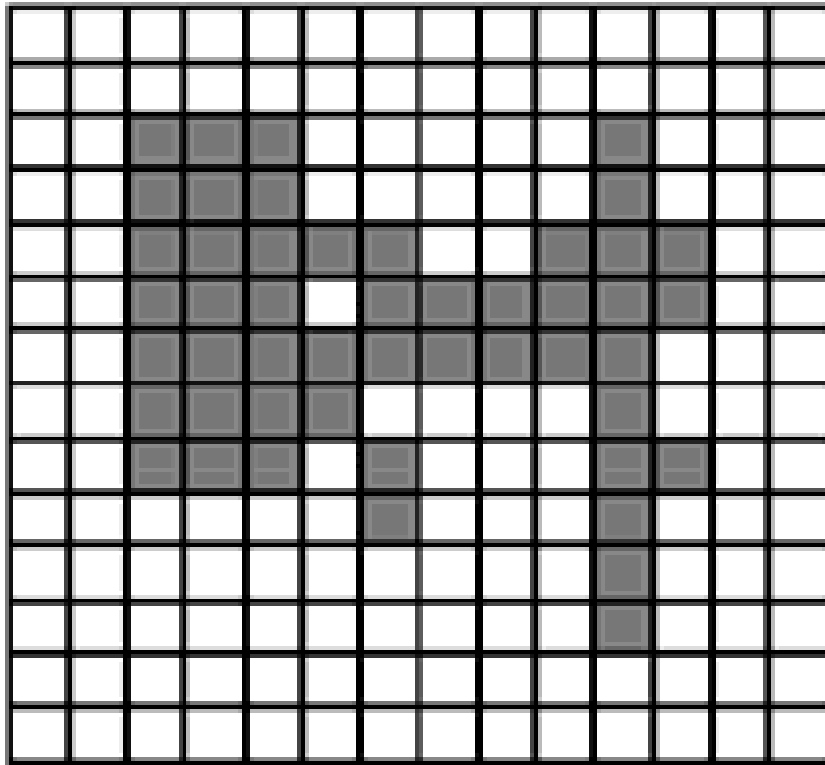
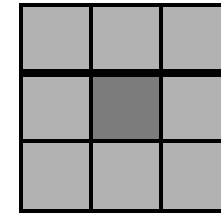
Penutupan (*closing*)

- Penutupan terdiri dari dilatasi yang diikuti erosi dan menghubungkan objek-objek yang berdekatan.
- Dapat digunakan untuk mengisi lubang dan celah kecil.
- Notasi: $A \bullet B = (A \oplus B) \ominus B$

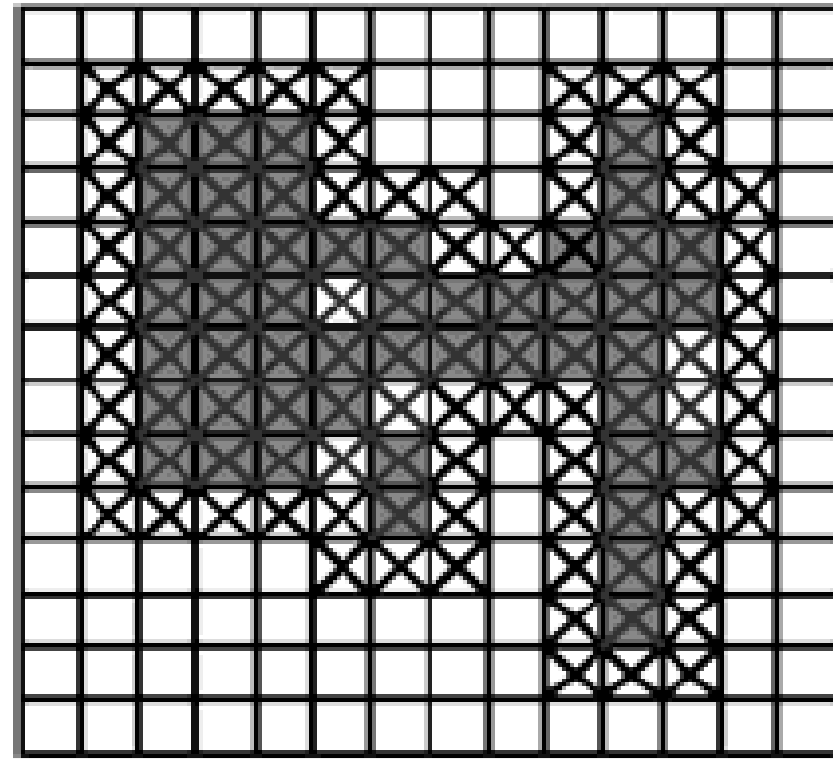
Closing



B=

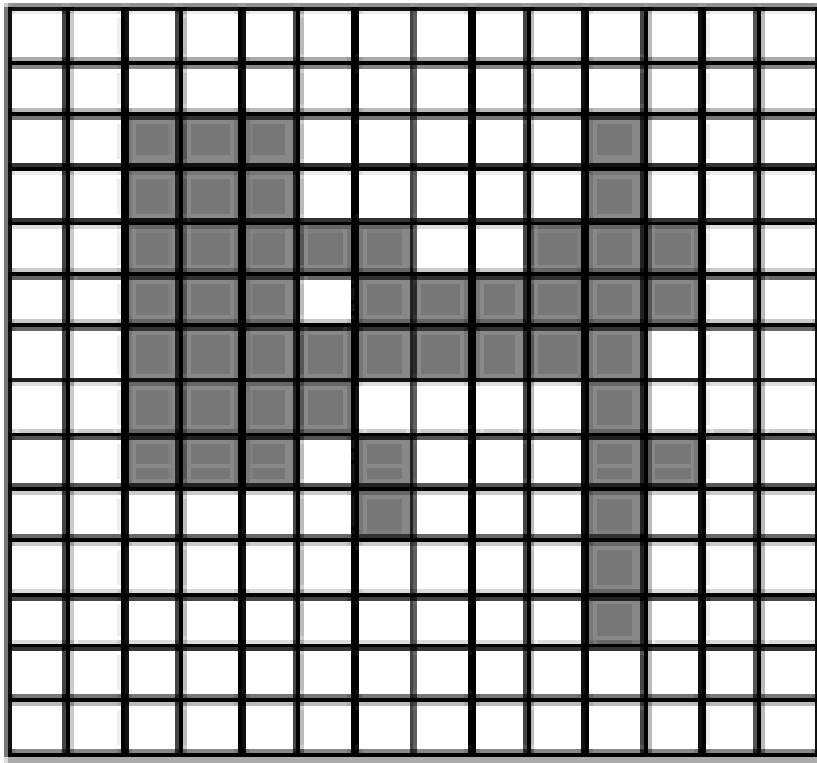
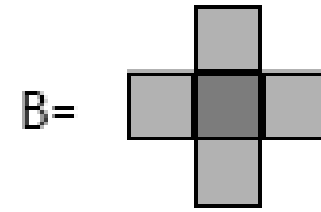


A

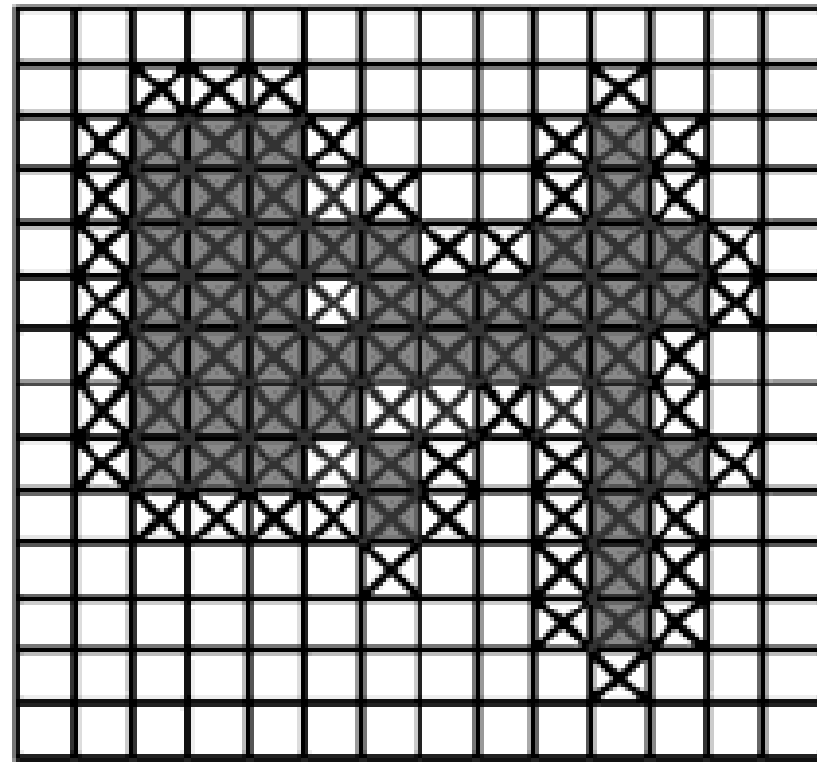


$A \oplus B$ $A \bullet B$

Closing



A



$A \oplus B$ $A \bullet B$

```
%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');
```



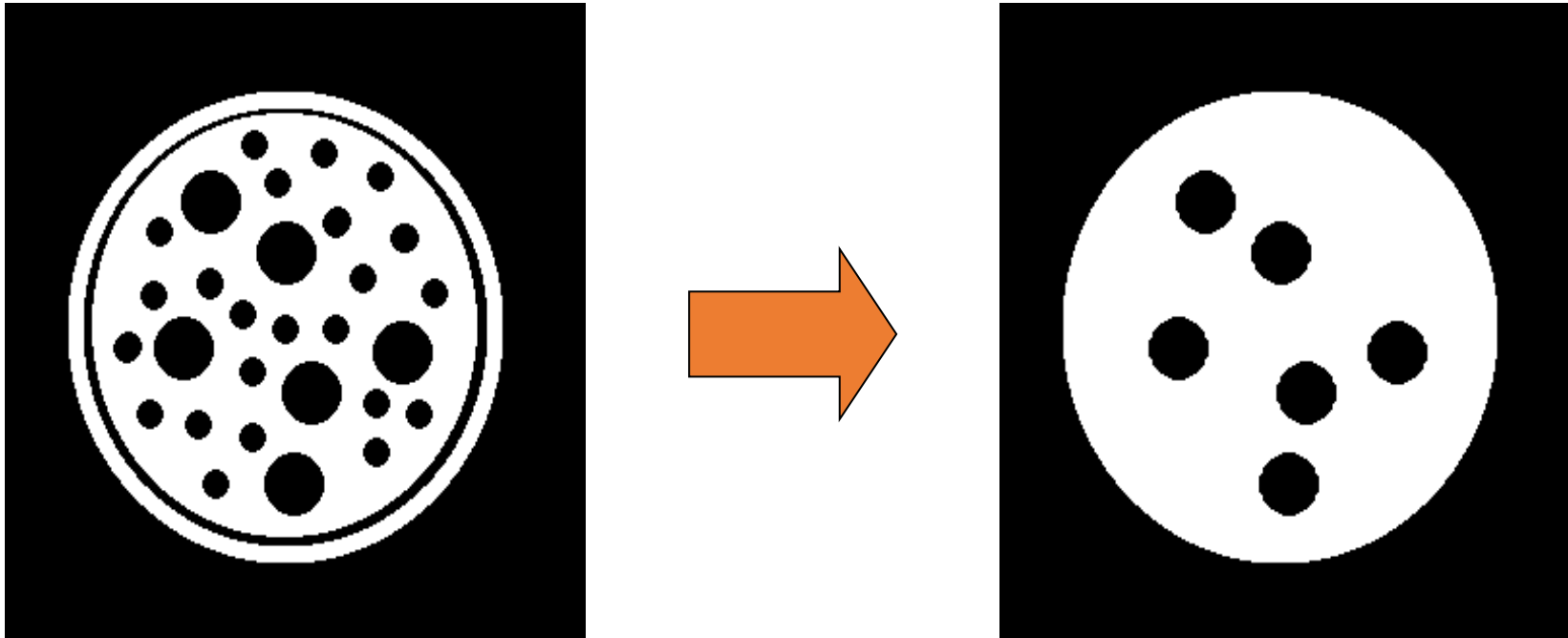
original image

11x11 square

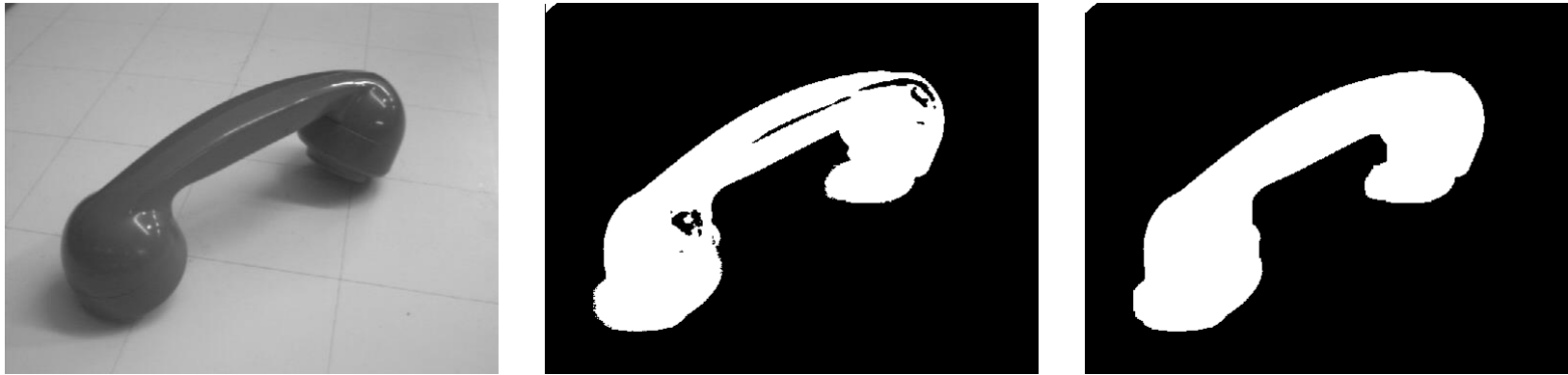


Image after closing

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



Threshold, closing with disc of size 20.

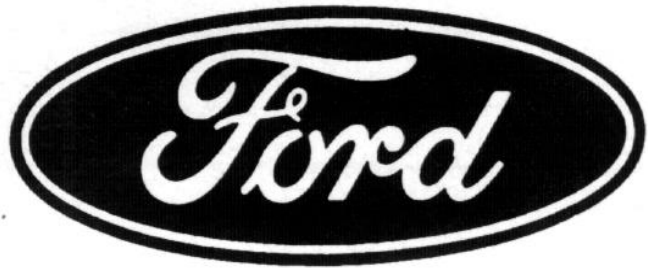


Note that opening is the **dual** of closing i.e. opening the foreground pixels with a particular structuring element is equivalent to closing the background pixels with the same element.

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



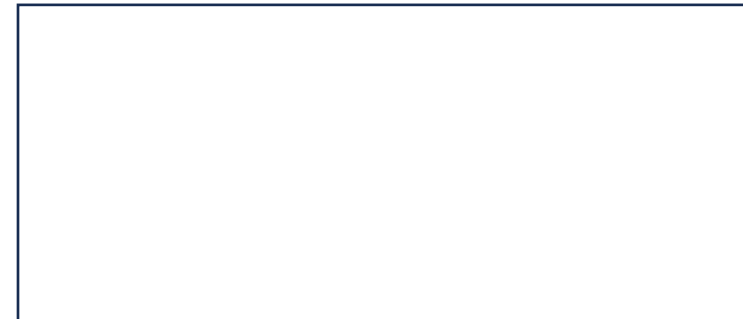
Original image



after 1 dilation

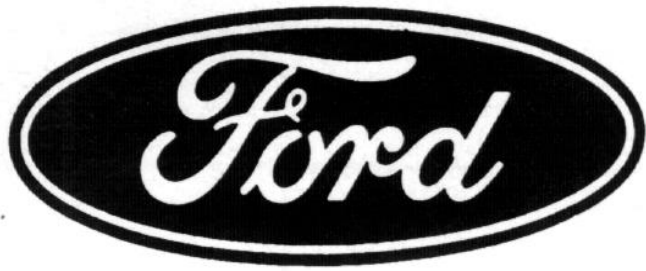


after 5th dilations



after inf dilations

Hasil program Matlab operasi erosi berulang kali



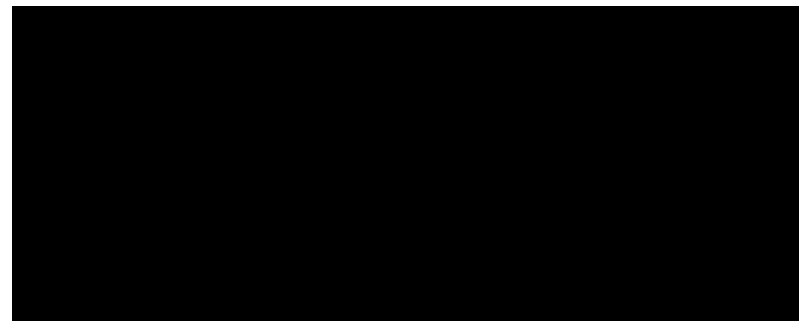
Original image



after 1 erosion



after 5th erosions

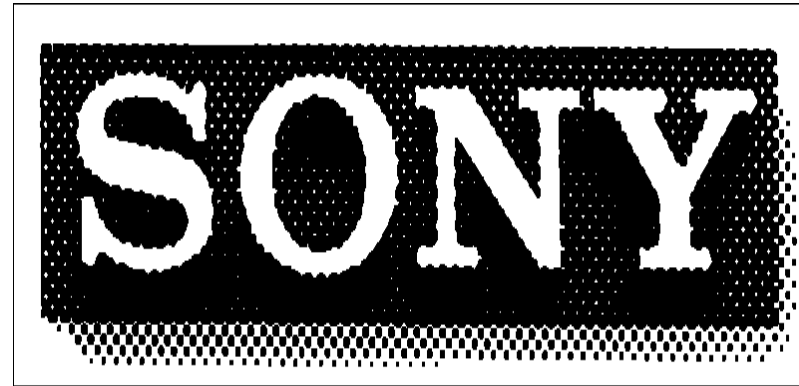


after inf erosions

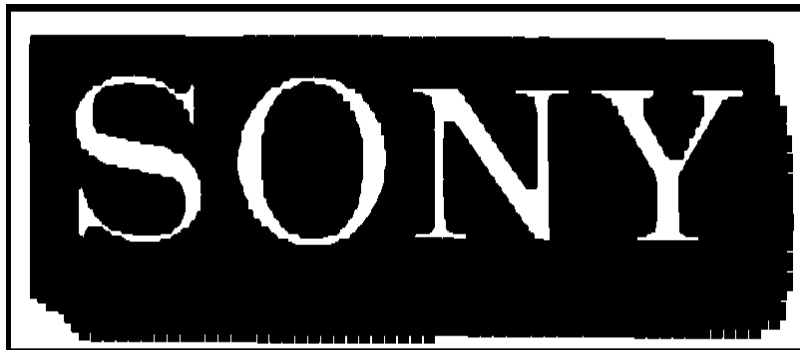
Hasil program Matlab operasi erosi berulang kali



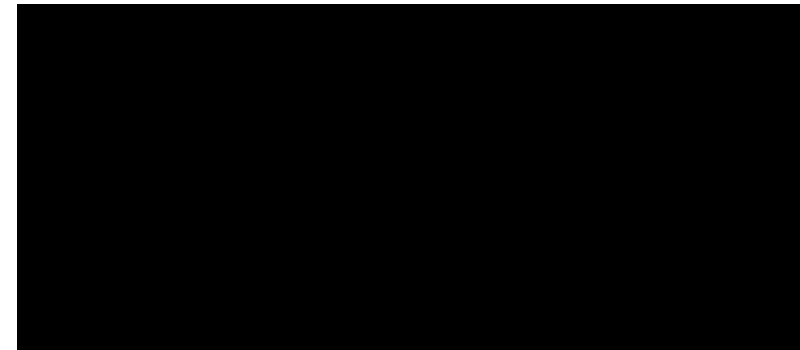
Original image



after 1 erosion

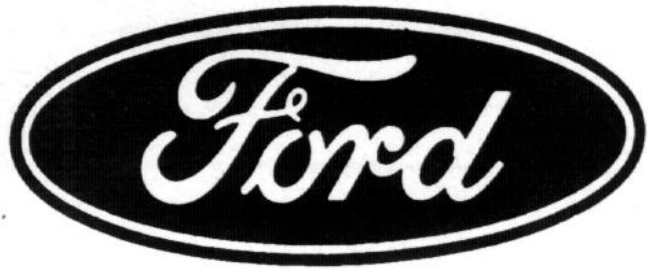


after 5th erosions



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



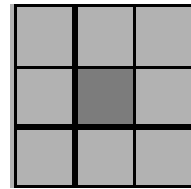
after 5th closings



after inf closnings

Application: 1. filtering

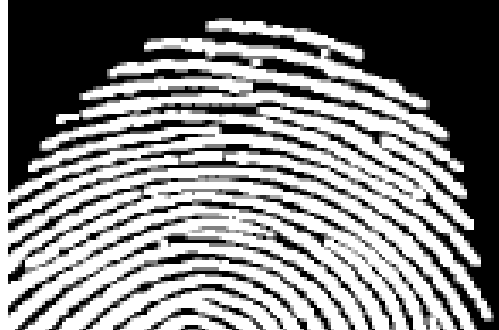
Application:
filtering



1. erode
 $A \ominus B$



2. dilate
 $(A \oplus B) \ominus B = A \circ B$



3. dilate
 $(A \circ B) \oplus B$

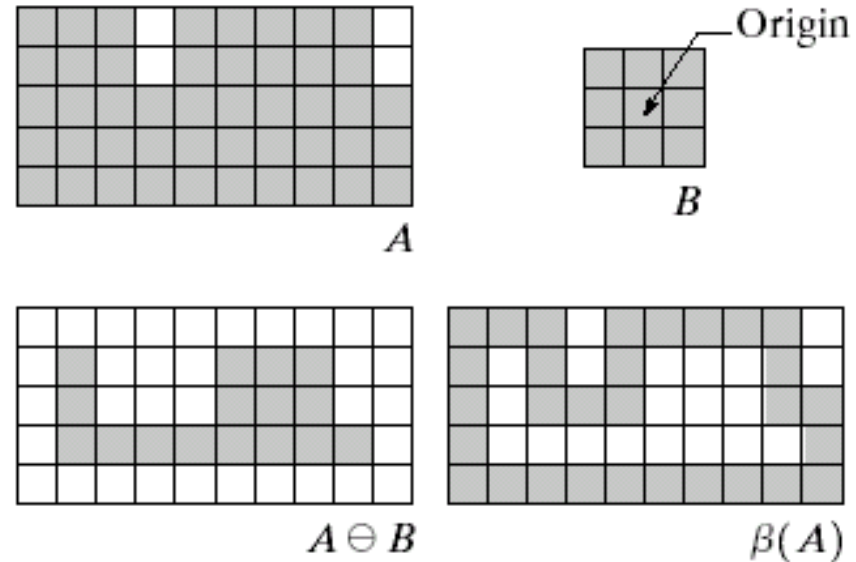


4. erode
 $((A \circ B) \oplus B) \ominus B = (A \circ B) \bullet B$

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



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*