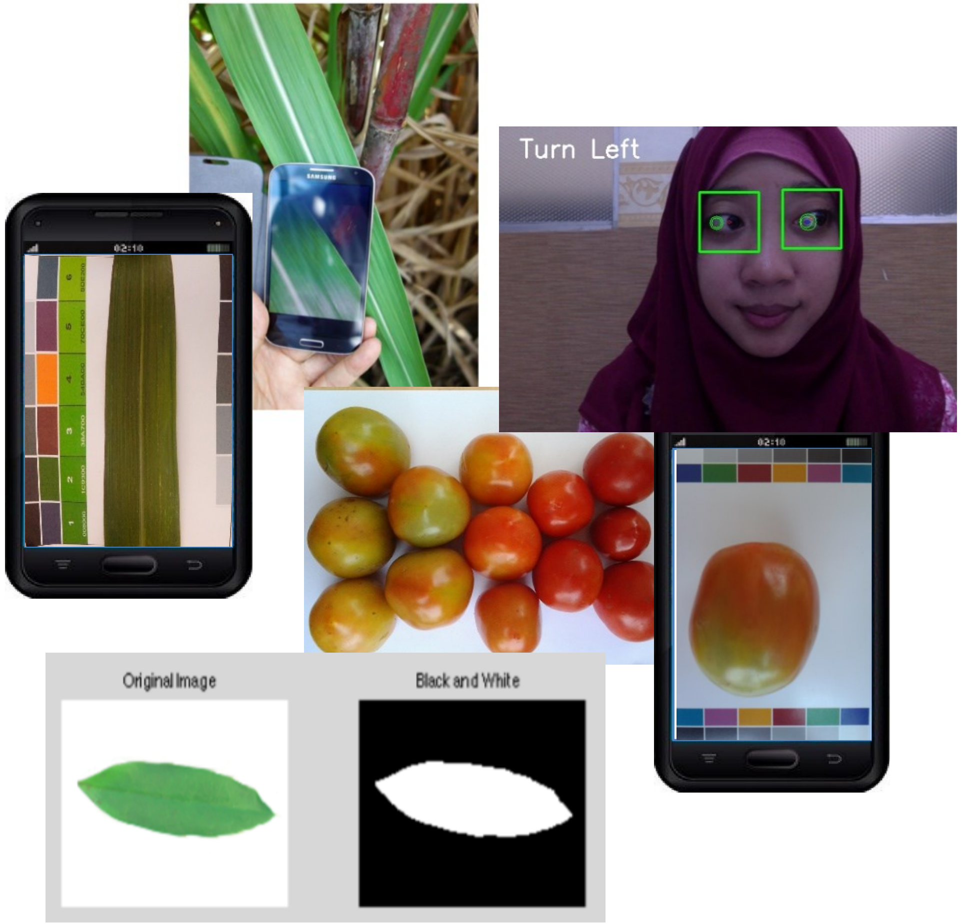
Visi Komputer

MORFOLOGI CITRA



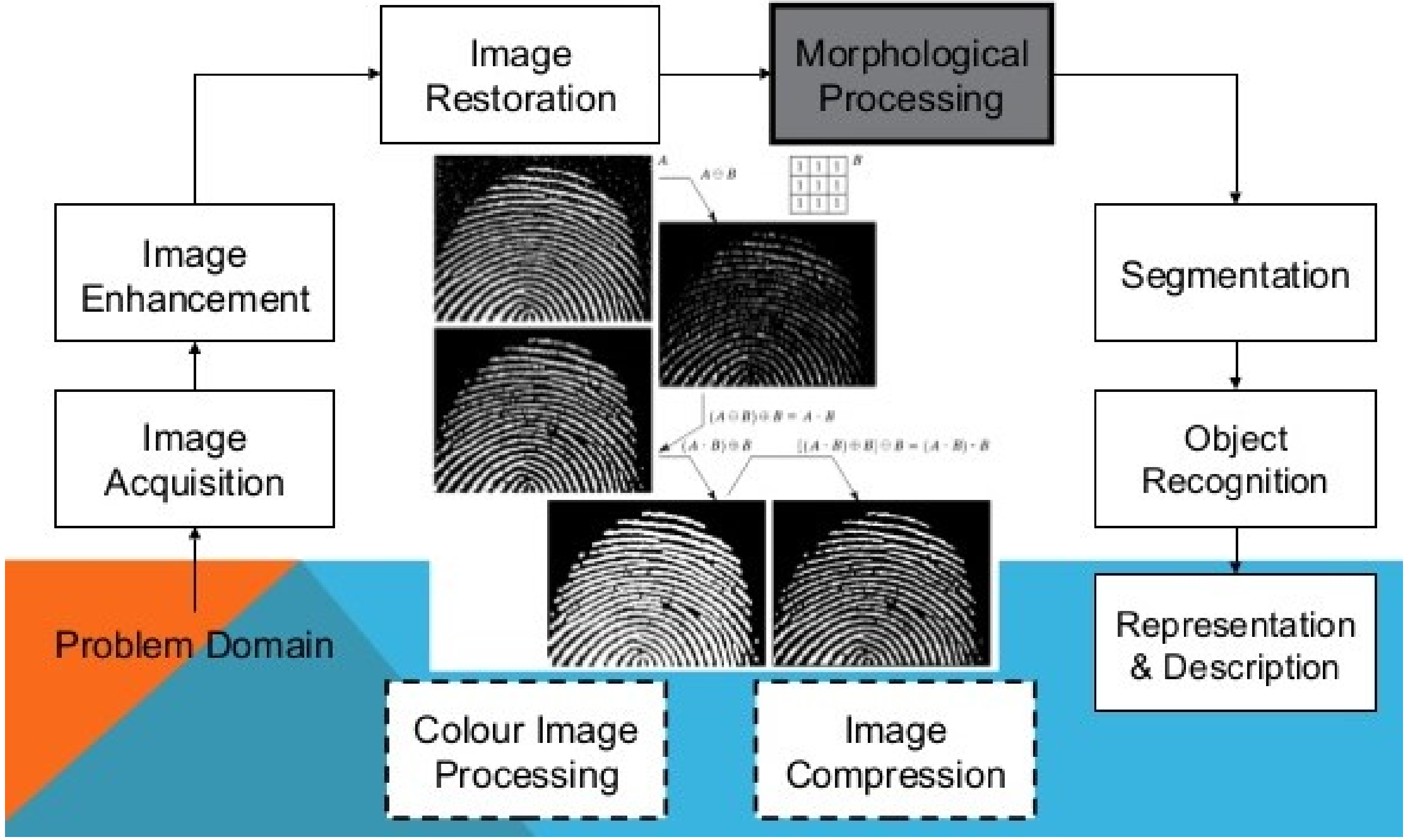
Team Teaching :

1. Dr. Eng. Fitri Utaminingrum, S.T, M.T

- [ ] Pegertian Operasi Morfologi
- [ ] Teori Himpunan
- [ ] Operasi Logika pada Citra Biner
- [ ] Dasar Proses Morfologi
- [ ] Structuring Element (SE)
- [ ] Dilasi dan Erosi
- [ ] Opening dan Closing
- [ ] Transformasi Hit or Miss
- [ ] Region Filling
- [ ] Boundary Extraction
- [ ] Belajar Mandiri

Operasi Morfologi

- Operasi morfologi merupakan operasi yang umumnya digunakan pada citra biner untuk mengubah struktur bentuk objek yang terkandung di dalam citra.
- Perbedaan antara pemrosesan citra morfologi dengan pemrosesan citra yang telah kita pelajari adalah:
  - Dulu kita memandang sebuah citra adalah suatu fungsi intensitas terhadap posisi(x,y)
  - **Dengan pendekatan morfologi, kita memandang citra sebagai sebuah himpunan.**
- Contoh aplikasi morfologi:
  - Membentuk filter spasial
  - Memperoleh skeleton(rangka) objek.
  - Menentukan letak objek di dalam citra
  - Memperoleh bentuk struktur objek



Teori Himpunan

- Jika a adalah elemen dari A, maka dituliskan a $\in$ A.
- Jika a bukan elemen A, dituliskan a $\notin$ A.

- Himpunan dispesifikasikan dengan tanda kurung {.} yang didalamnya berisi elemen-elemen himpunan.

- Jika tiap elemen dari himpunan A adalah juga elemen dari himpunan B, maka A adalah subset dari B, dan dituliskan $A \subseteq B$.

- **Union** himpunan A dan B, dinyatakan dengan $C = A \cup B$, adalah himpunan dari semua elemen anggota A, B, atau keduanya.

- **Irisan** A dan B, dinyatakan dengan $D = A \cap B$, adalah himpunan yang anggotanya merupakan anggota persekutuan dari dua himpunan A dan B.

Teori Himpunan

Dua himpunan A dan B disebut disjoint atau mutually exclusive jika kedua himpunan tersebut tidak memiliki elemen bersama.

Dalam kasus ini, $A \cap B = \emptyset$.

Complement himpunan A adalah himpunan elemen yang bukan anggota A :

$$A^c = \{w \mid w \notin A\}$$

Selisih dua himpunan A dan B, dinyatakan dengan A-B, memiliki definisi :

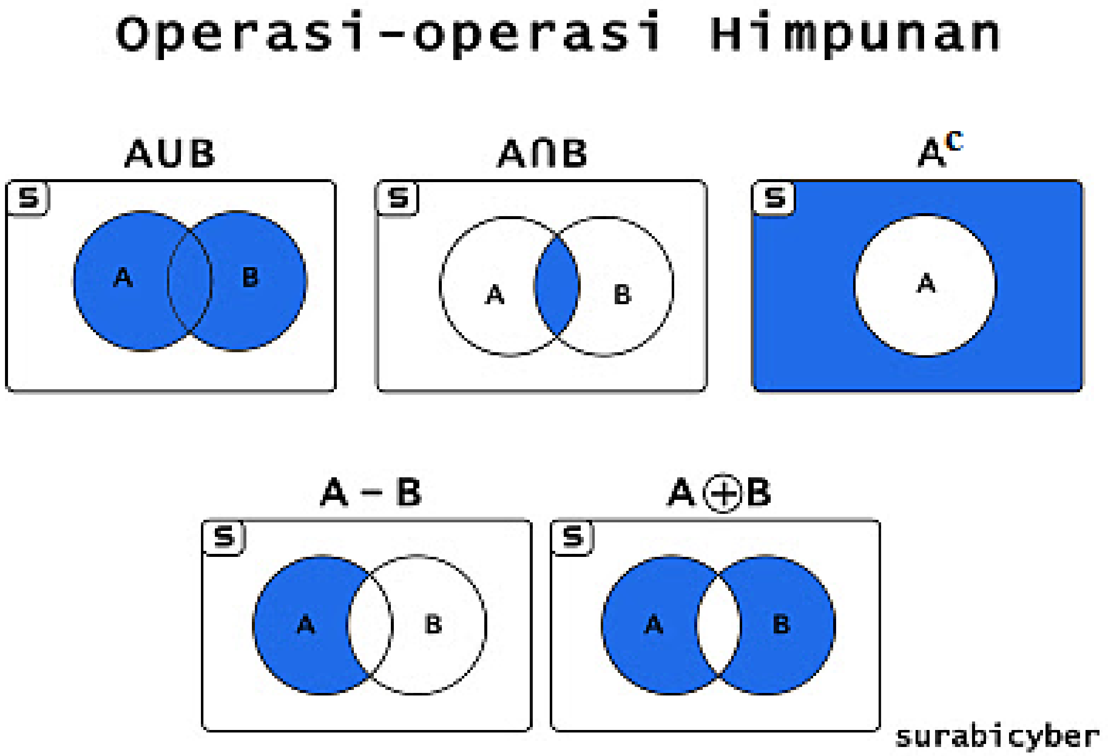
$$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$$

Refleksi dari himpunan B, dinyatakan dengan denoted , memiliki definisi :

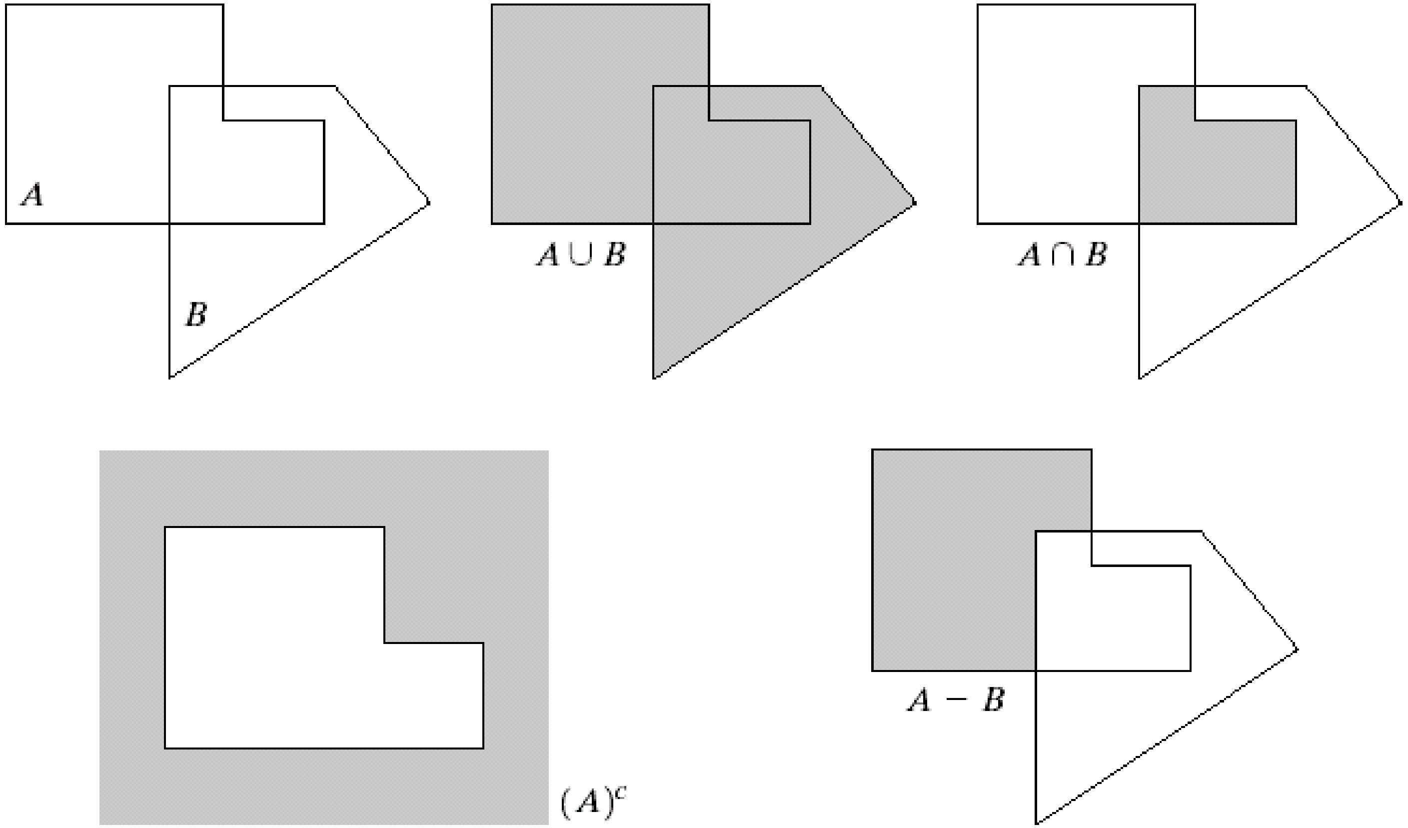
$$\hat{B} = \{w \mid w = -b, \text{ for } b \in B\}$$

Translasi dari himpunan A dengan titik $z = (z_1, z_2)$, dinyatakan dengan $(A)_z$, memiliki definisi :

$$(A)_z = \{c \mid c = a + z, \text{ for } a \in A\}$$



Teori Himpunan



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.

Operasi Logika pada Citra Biner

•Prinsip-prinsip operasi logika yang digunakan dalam image processing adalah **AND, OR, NOT (COMPLEMENT)**.

•Operasi logika dilakukan dari pixel ke pixel dengan pixel-pexel yang bersesuaian.

•Operasi logika penting lainnya adalah: **XOR (exclusive OR), NAND (NOT-AND)**

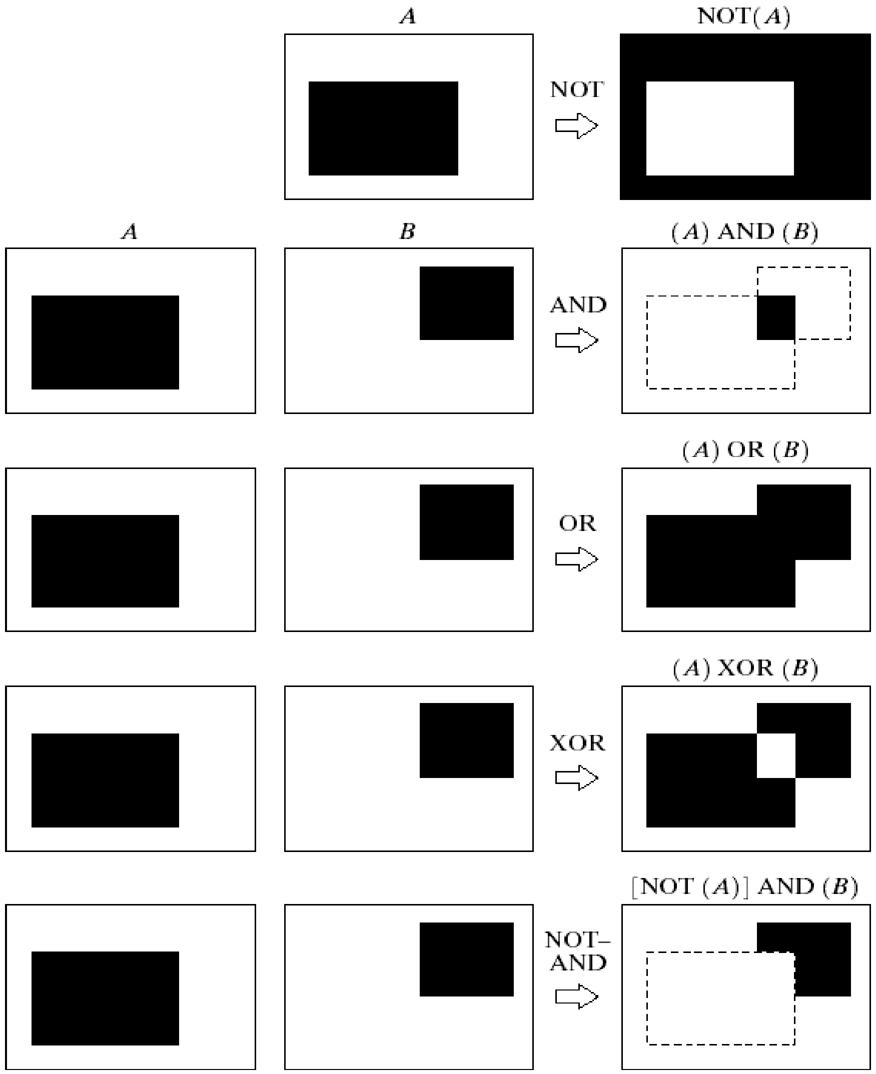
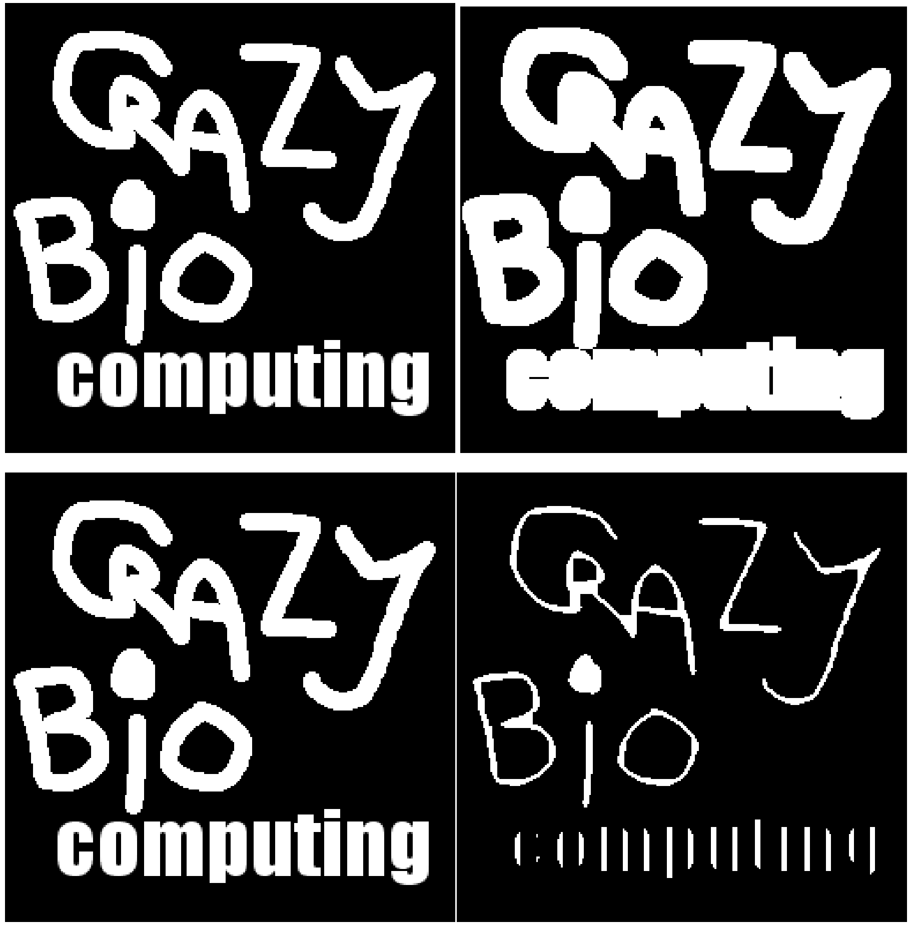


FIGURE 9.3 Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

Dasar Proses Morfologi

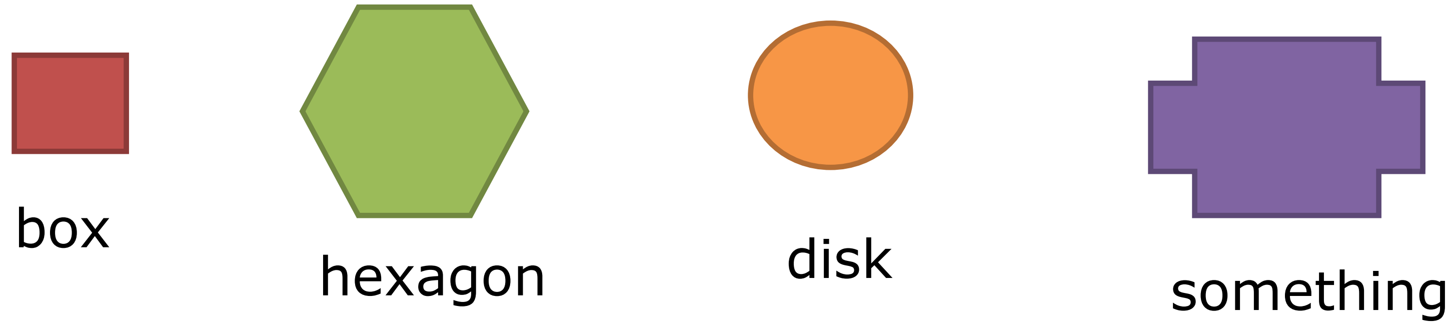
Dilasi
- Membesar
- Meluas
- Melebar

Erosi
- Pengikisan
- Memperkecil
- Mempersempit



Structuring Element (SE)

- A **structuring element** is a shape mask used in the basic morphological operations.
- They can be any shape and size that is digitally representable, and each has an origin.



box(length,width)　　　　disk(diameter)

- Shape and size must be adapted to geometric properties for the objects.

Dilasi

- Dilasi digunakan untuk mengembagkan/ekspan sebuah elemen A menggunakan *structuring element* B.
- Dilasi A oleh B dapat didefinisikan dengan persamaan berikut:

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \quad (1)$$

- Dilasi A oleh B adalah set dari seluruh pergeseran z, seperti dan A tumpang tindih dengan setidaknya satu elemen. Sehingga Pers. (1) dapat dituliskan menjadi:

$$A \oplus B = \{z \mid [(\hat{B})_z \cap A] \subseteq A\} \quad (2)$$

Dalam hal ini,

a) $\hat{B} = \{w \mid w = -b, untuk\ b \in B\}$

b) $(B)_z = \{c \mid c = a + z, untuk\ a \in A\}$

c) z=(z1, z2)

Dilasi

- Burger & Burge (2008) mendefinisikan operasi dilasi seperti dalam persamaan berikut: $A \oplus B = \{z \mid z = a + b, \text{ dengan } a \in A \text{ dan } b \in B\}$

Contoh Dilasi:

$A = \{ (2,2), (2,3), (2,4), (3,2), (3,3), (3,4), (4,3) \}$

$B = \{ (-1, 0), (0,0), (1,0) \}$

	-1	0	1
-1	0	1	0
0	0	1	0
1	0	1	0

Structuring Element

	1	2	3	4	5
1	0	0	0	0	0
2	0	1	1	1	0
3	0	1	1	1	0
4	0	0	1	0	0
5	0	0	0	0	0

Dilasi

$$
\begin{aligned}
A \oplus B = \{\ & (2,2)+(-1,0),\ (2,2)+(0,0),\ (2,2)+(1,0), \\
& (2,3)+(-1,0),\ (2,3)+(0,0),\ (2,3)+(1,0), \\
& (2,4)+(-1,0),\ (2,4)+(0,0),\ (2,4)+(1,0), \\
& (3,2)+(-1,0),\ (3,2)+(0,0),\ (3,2)+(1,0), \\
& (3,3)+(-1,0),\ (3,3)+(0,0),\ (3,3)+(1,0), \\
& (3,4)+(-1,0),\ (3,4)+(0,0),\ (3,4)+(1,0), \\
& (4,3)+(-1,0),\ (4,3)+(0,0),\ (4,3)+(1,0)\ \}
\end{aligned}
$$

$$
\begin{aligned}
= \{\ & (1,2), (2,2), (3,2),\ (1,3), (2,3), (3,3), \\
& (1,4), (2,4), (3,3),\ (2,2), (3,2), (4,2), \\
& (2,3), (3,3), (4,3),\ (2,4), (3,4), (4,4), \\
& (3,3), (4,3), (5,3)\ \}
\end{aligned}
$$

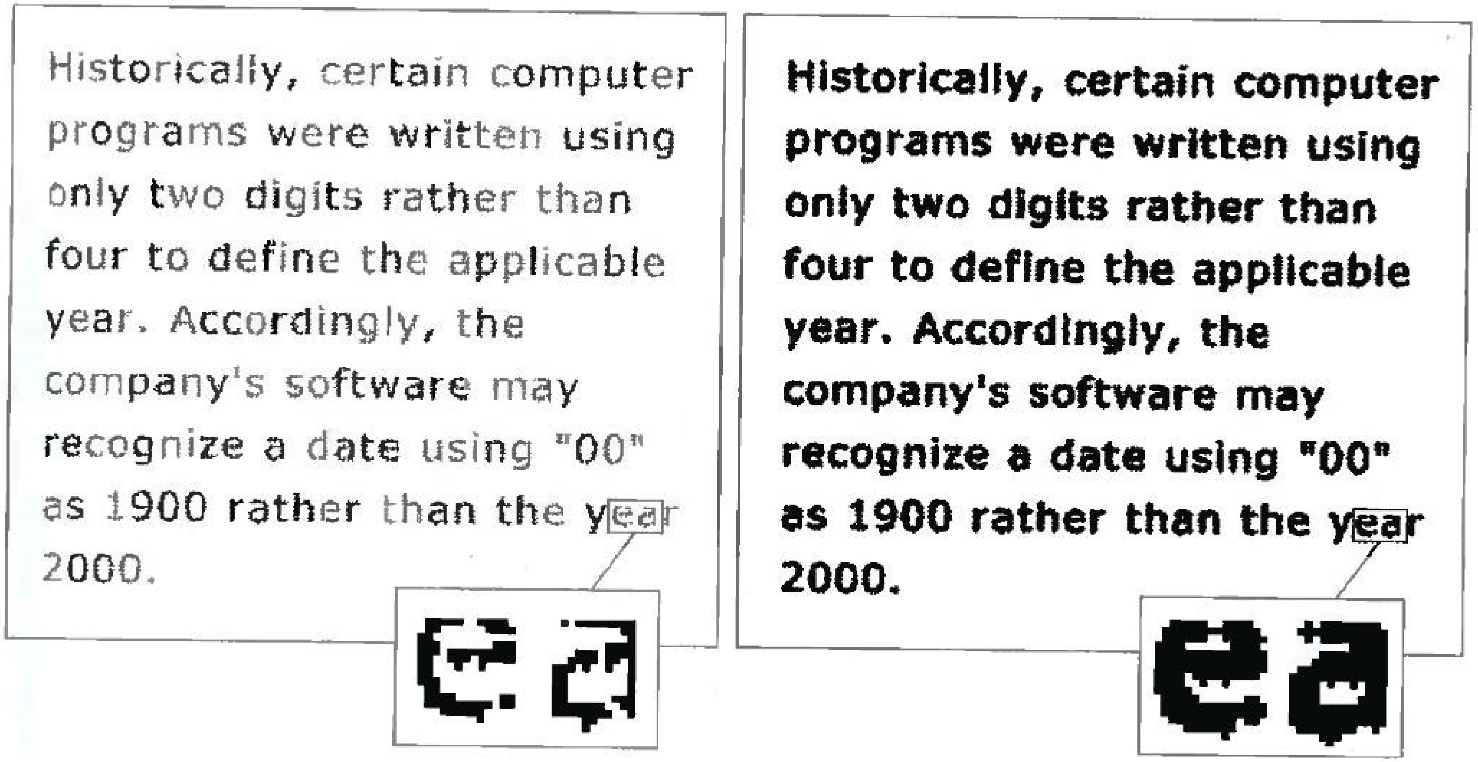
$$
\begin{aligned}
= \{\ & (1,2), (1,3), (1,4), (2,2), (2,3), (2,4), \\
& (3,2), (3,3), (3,4), (4,2), (4,3), (4,4), (5,3)\ \}
\end{aligned}
$$

	1	2	3	4	5
1	0	0	0	0	0
2	0	1	1	1	0
3	0	1	1	1	0
4	0	0	1	0	0
5	0	0	0	0	0

	-1	0	1
-1	0	1	0
0	0	1	0
1	0	1	0

	1	2	3	4	5
1	0	1	1	1	0
2	0	1	1	1	0
3	0	1	1	1	0
4	0	1	1	1	0
5	0	0	1	0	0

Dilasi

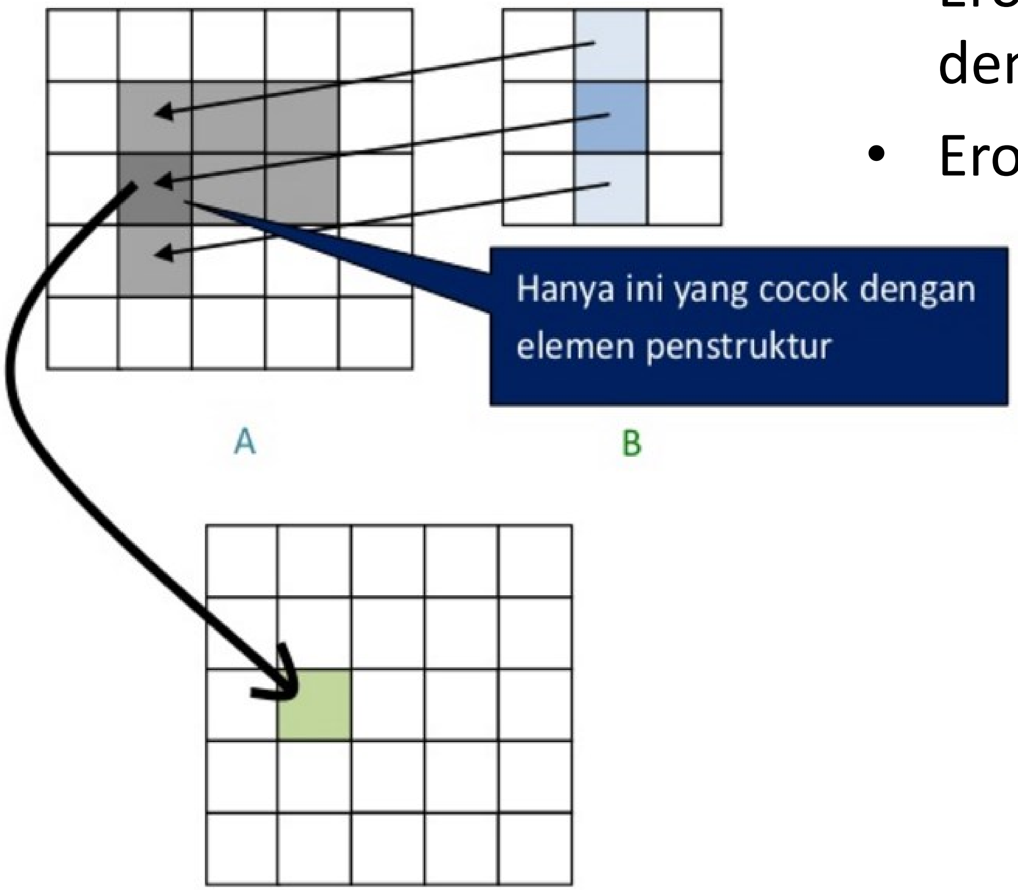


0	1	0
1	1	1
0	1	0

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



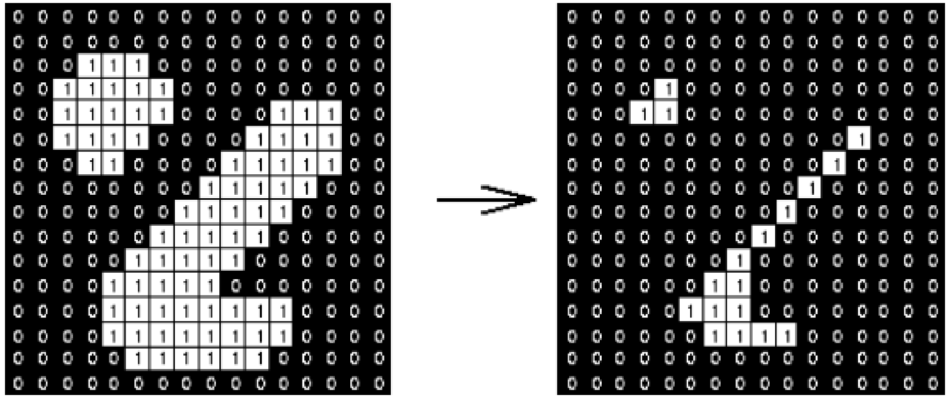
- Erosi digunakan untuk menyusutnya elemen A dengan menggunakan elemen B
- Erosi A dengan B data dituliskan:

$$A \ominus B$$

$$g(x, y) = f(x, y) \Theta SE$$

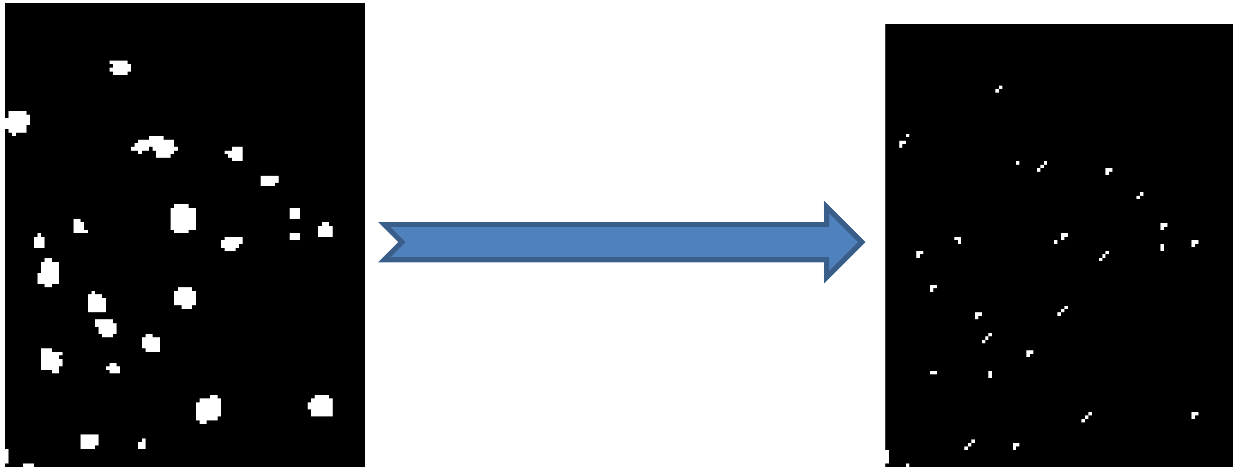
Structuring Element

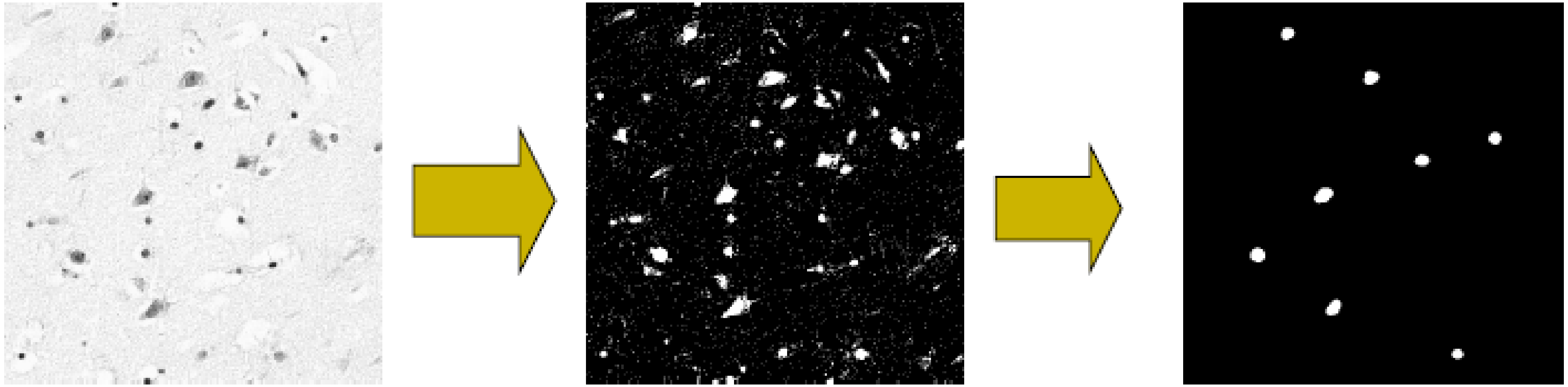
1	1	1
1	1	1
1	1	1



- Objects menjadi lebih kecil

Erosi





Dilasi dan Erosi di Phyton

Dilasi

```
dilation = cv2.dilate(img,kernel,iterations = 1)
```

Erosi

```
import cv2
import numpy as np

img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
erosion = cv2.erode(img,kernel,iterations = 1)
```

Dilasi dan Erosi

- Dilasi dan erosi adalah bersifat duals satu sama lain berkaitan dengan komplemen dan refleksi. Yaitu: $(A \ominus B)^c = A^c \oplus \hat{B}$

PROVING

- Starting with the definition of erosion

$$(A\Theta B)^c = \{z \mid (B)_z \subseteq A\}^c$$

- If set $(B)_z$ is contained in set A, then

$$(B)_z \cap A^c = \phi$$

thus

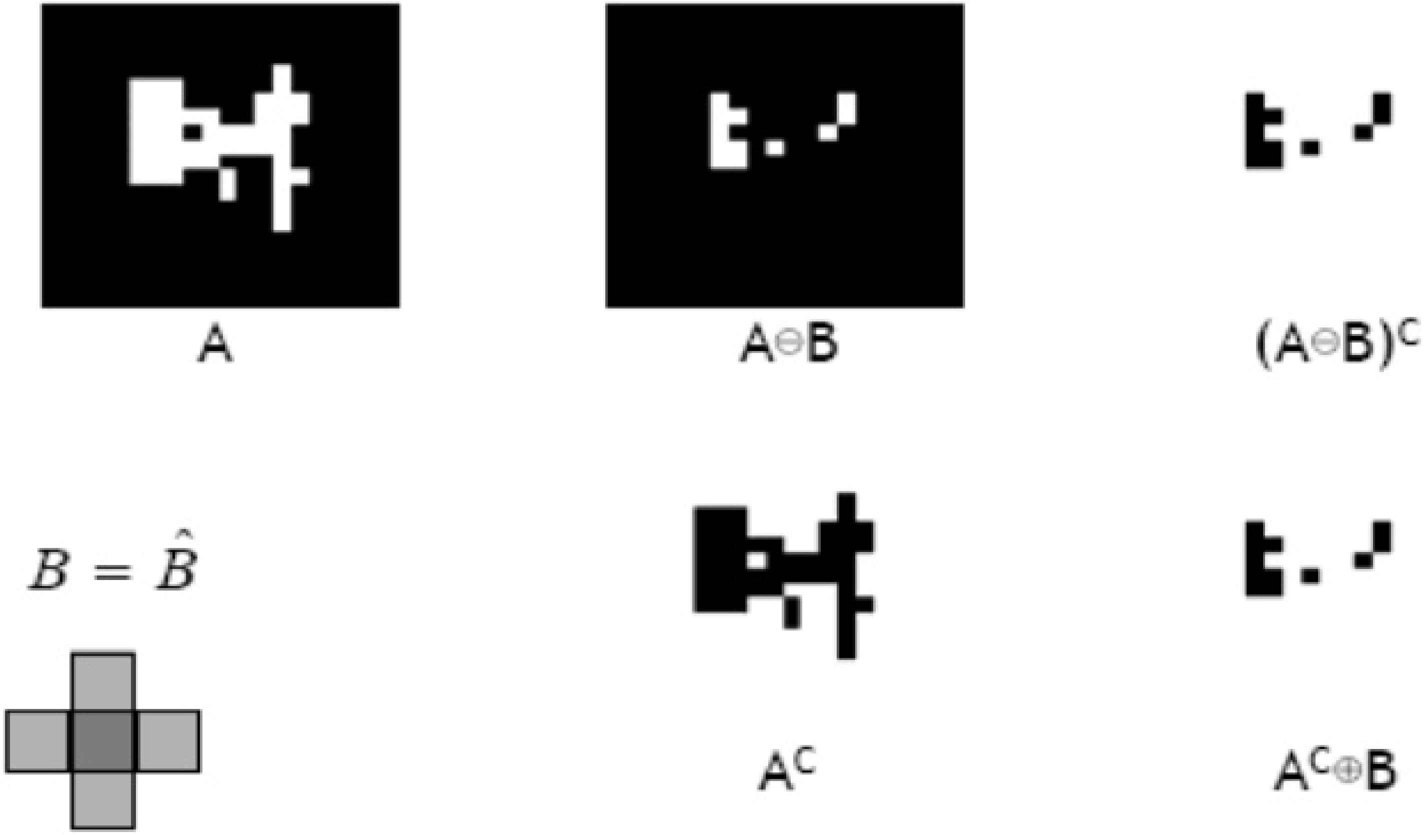
$$(AB)^c = \left\{z \mid (B)_z \cap A^c = \phi\right\}$$

$$= A^c \oplus \hat{B}$$

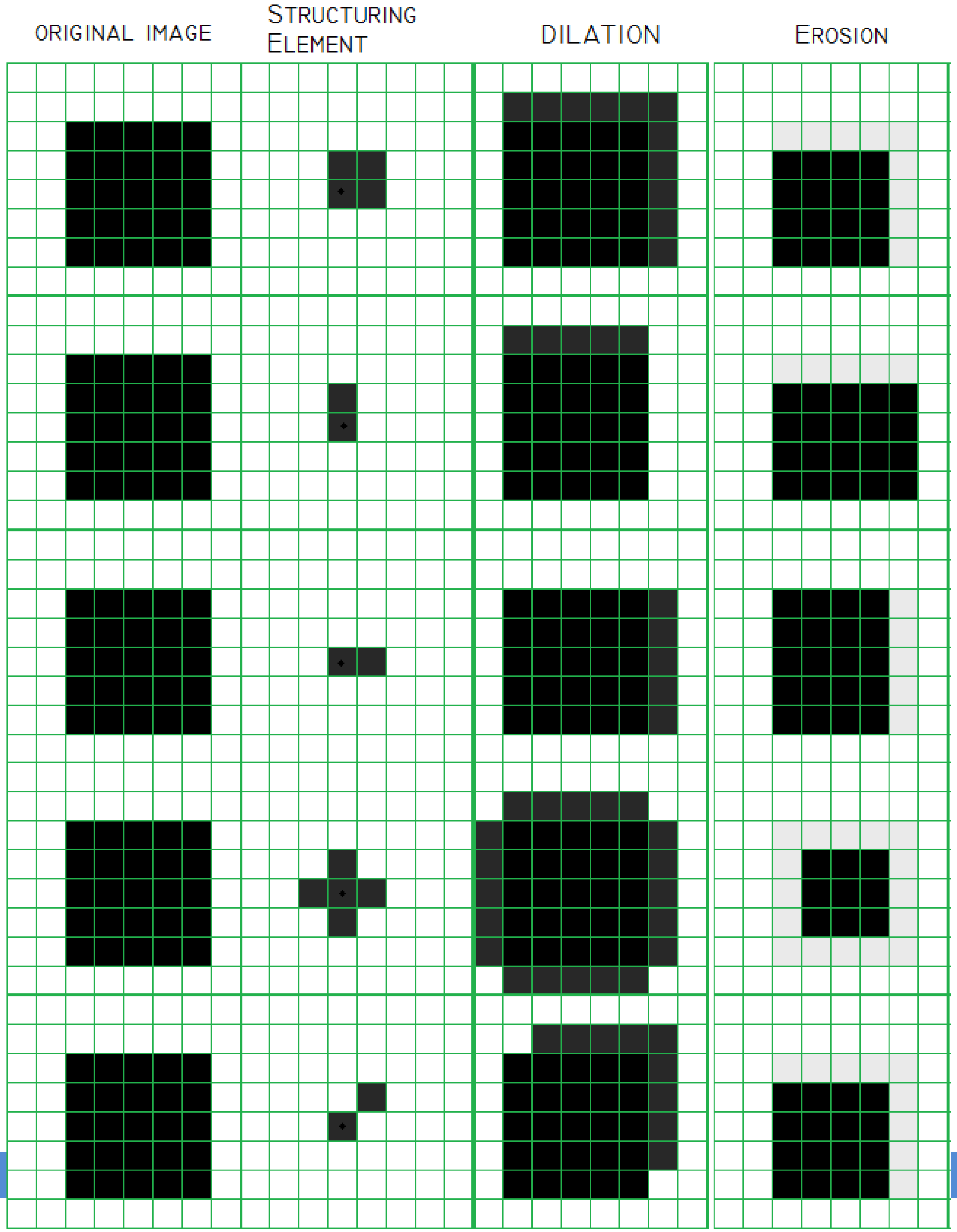
Dilasi dan Erosi Bersifat Duals

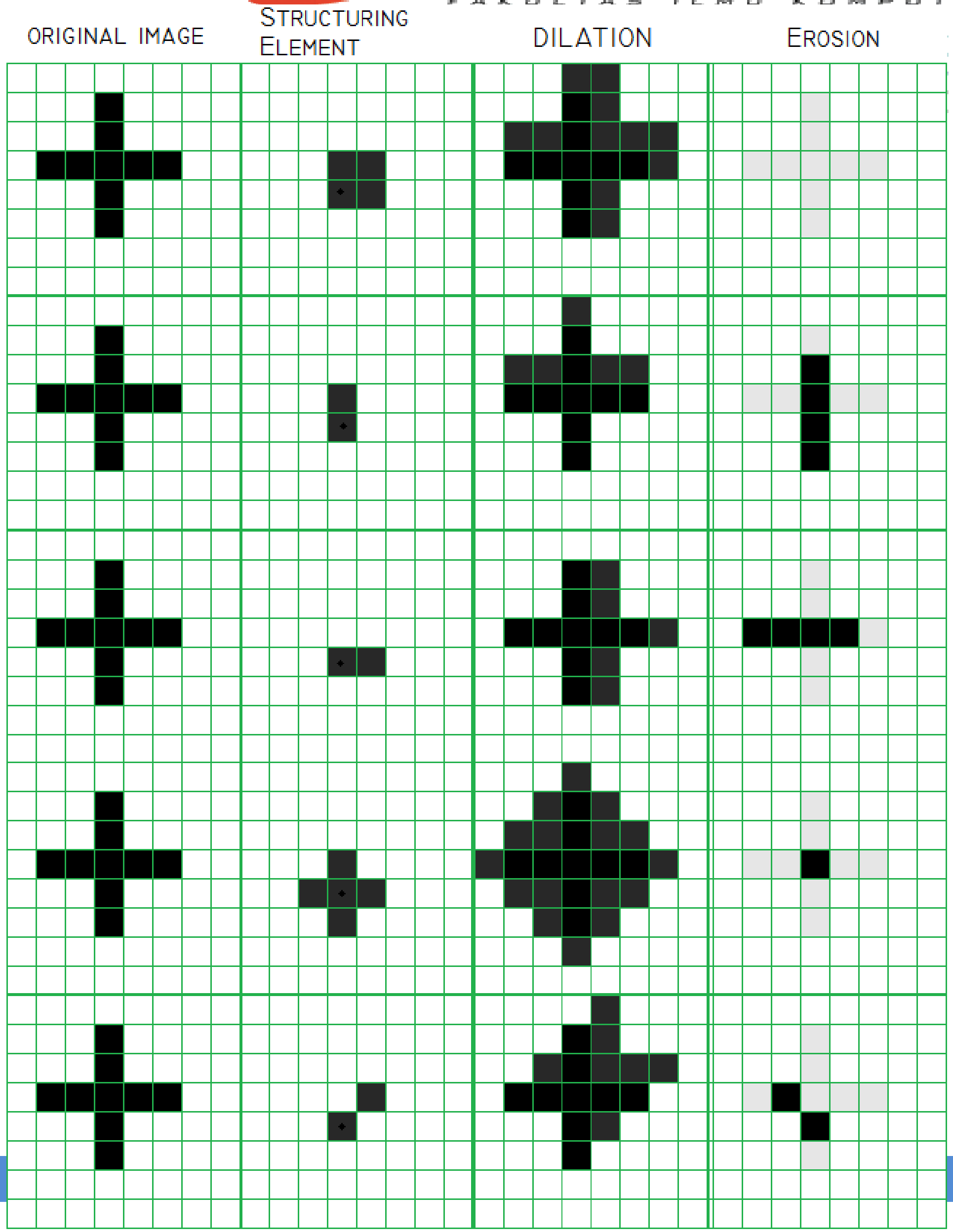
Dilasi dan erosi adalah bersifat duals satu sama lain berkaitan dengan komplemen dan refleksi. Yaitu:

$$(A \ominus B)^c = A^c \oplus \hat{B}$$



Dilasi dan Erosi





Structuring Element in Phyton

Fungsi Phyton untuk Structuring Element **cv2.getStructuringElement()**

```
# Rectangular Kernel
>>> cv2.getStructuringElement(cv2.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
>>> cv2.getStructuringElement(cv2.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
>>> cv2.getStructuringElement(cv2.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)
```

Structuring Element in Matlab

- `SE = strel('diamond',R)` → creates a diamond-shaped structuring element, where R specifies the distance from the structuring element origin to the points of the diamond.
- `SE = strel('disk',R,N)` → creates a disk-shaped structuring element, where R specifies the radius.
- `SE = strel('line',len,deg)` → creates a linear structuring element that is symmetric with respect to the neighborhood center.
- `SE = strel('rectangle',MN)` → creates a rectangular structuring element, where MN specifies the size.
- `SE = strel('square',W)` → creates a square structuring element whose width is W pixels.
- `SE = strel('cube',W)` → creates a cubic structuring element whose width is W pixels. W must be a nonnegative integer scalar.
- `SE = strel('cuboid',XYZ)` creates a cuboidal structuring element of size XYZ.
- `SE = strel('sphere',R)` creates a spherical structuring element whose radius is R pixels.

https://www.mathworks.com/help/images/ref/strel-class.html

Opening dan Closing

- **Opening**
  - menghaluskan batas (contour) objek,
  - mematahkan hubungan/jembatan yang sempit,
  - menghilangkan tonjolan yang tipis.

- **Closing**
  - juga menghaluskan contour objek, tetapi kebalikan dari opening
  - closing menggabungkan jembatan yang sempit dan jurang sempit yang panjang,
  - menghilangkan lubang-lubang kecil dan mengisi celah di dalam contour.

Opening dan Closing

- Opening dari himpunan A dengan “structuring element” B didefinisikan :

$$A \circ B = (A \ominus B) \oplus B$$

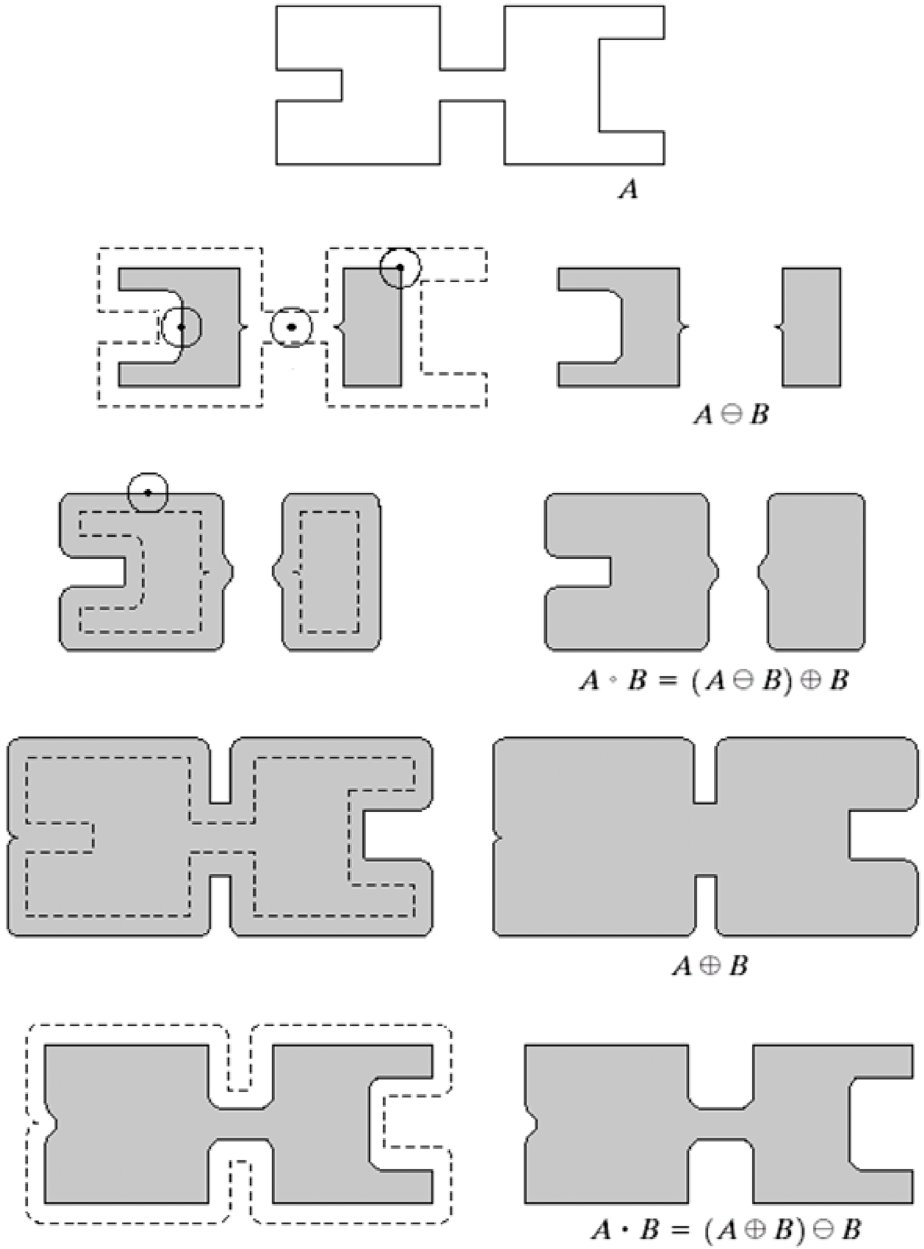
  – Pertama - mengikis A oleh B, dan kemudian melebarkan hasilnya dengan B

- Closing dari himpunan A dengan “structuring element” B didefinisikan :

$$A \bullet B = (A \oplus B) \ominus B$$

a
b c
d e
f g
h i

FIGURE 9.10 Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The dark dot is the center of the structuring element.



Operasi opening

- Erosi yang diikuti dengan dilasi
- Operasi ini berguna untuk menghaluskan kontur objek dan menghilangkan seluruh piksel di area yang terlalu kecil untuk ditempati oleh elemen penstruktur. Kemudian penghalusan dilakukan melaui dilasi.

Operasi Closing

- Dilasi yang diikuti dengan erosi
- Berguna untuk menghaluskan kontus dan menghilangkan lubang-lubang kecil.

Closing

A

opening of A
→removal of small protrusions, thin connections, ...

closing of A
→ removal of holes

Opening dan Closing with Phyton

Opening

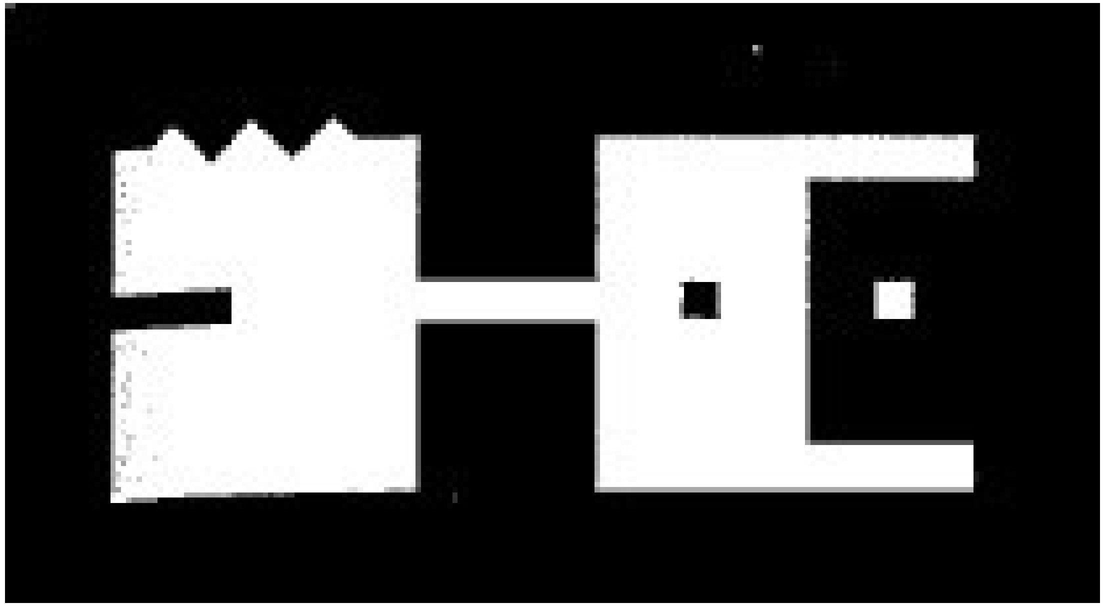
```
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
```

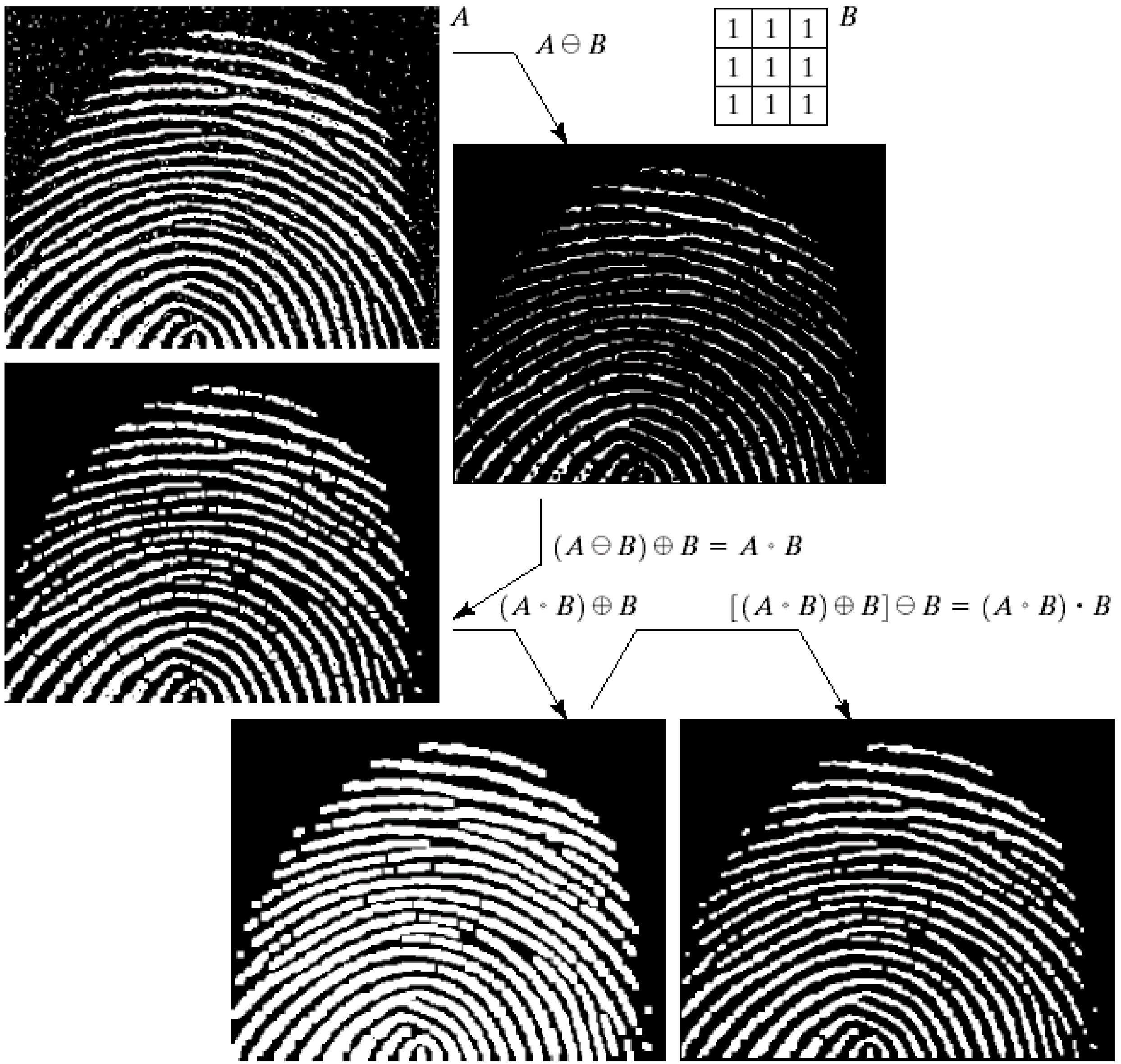
Closing

```
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
```

Opening dan Closing with Mathlab

```
I=imread(‘ im.jpg‘);
I1=im2bw(I);
S=strel(‘square’,13);
I2=imopen(I1,S);
I2=imclose(I2,S);
Subplot(1,2,1);imshow(I);
Subplot(1,2,2);imshow(I1);
```





a b c d e f

FIGURE 9.11
(a) Noisy image.
(c) Eroded image.
(d) Opening of A.
(d) Dilation of the opening.
(e) Closing of the opening. (Original image for this example courtesy of the National Institute of Standards and Technology.)

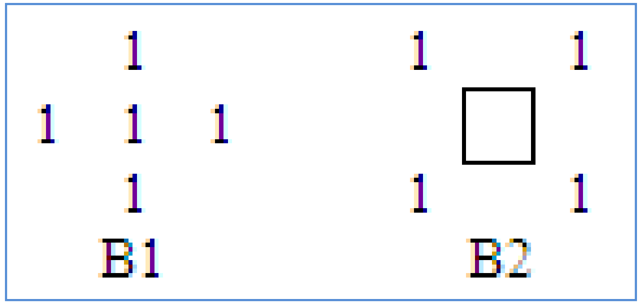
Transformasi hit-or-miss

Transformasi hit-or-miss A oleh B dinyatakan oleh $A \otimes B$

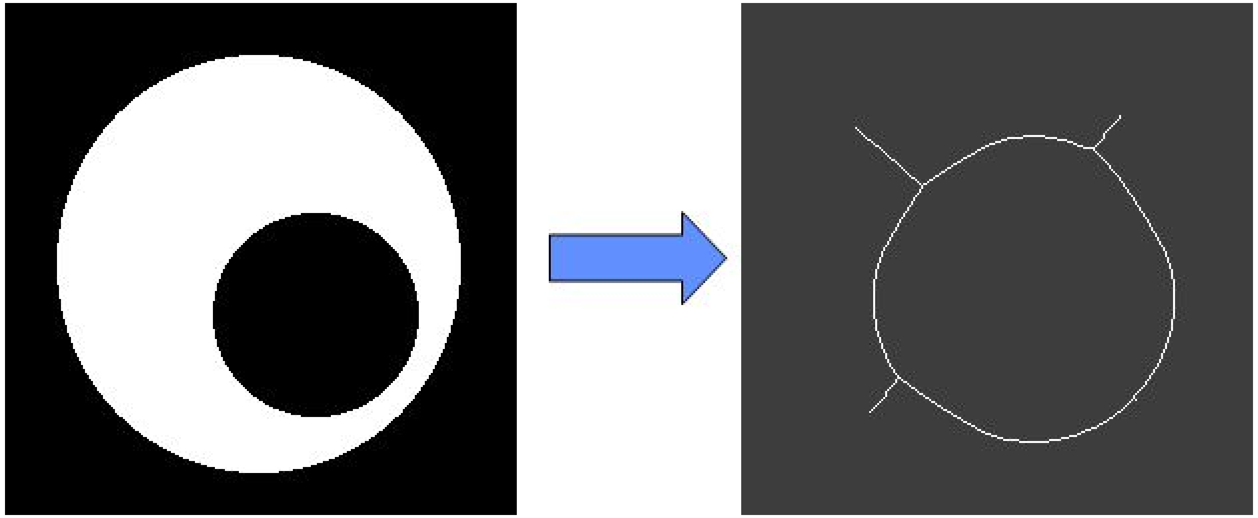
$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

Kita bisa men-generalisasi notasi dengan menyatakan B=(B1, B2),

Transformasi hit-or-miss didefinisikan dengan dua strel/konstruktor seperti:



Contoh pasangan strel



Original image

Skeleton image

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

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

a. Piksel citra asli A

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

b. Erosi A oleh B1

	1	
1	1	1
	1	

B1

1		1
	□	
1		1

B2

Pasangan strel yang digunakan

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

c. Komplemen citra A (Ac)

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

d. Erosi Ac oleh B2

Toolbox di MATLAB:

```
>> C = bwhitmiss(A, B1, B2)
```

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	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	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

e. Irisan b dan d

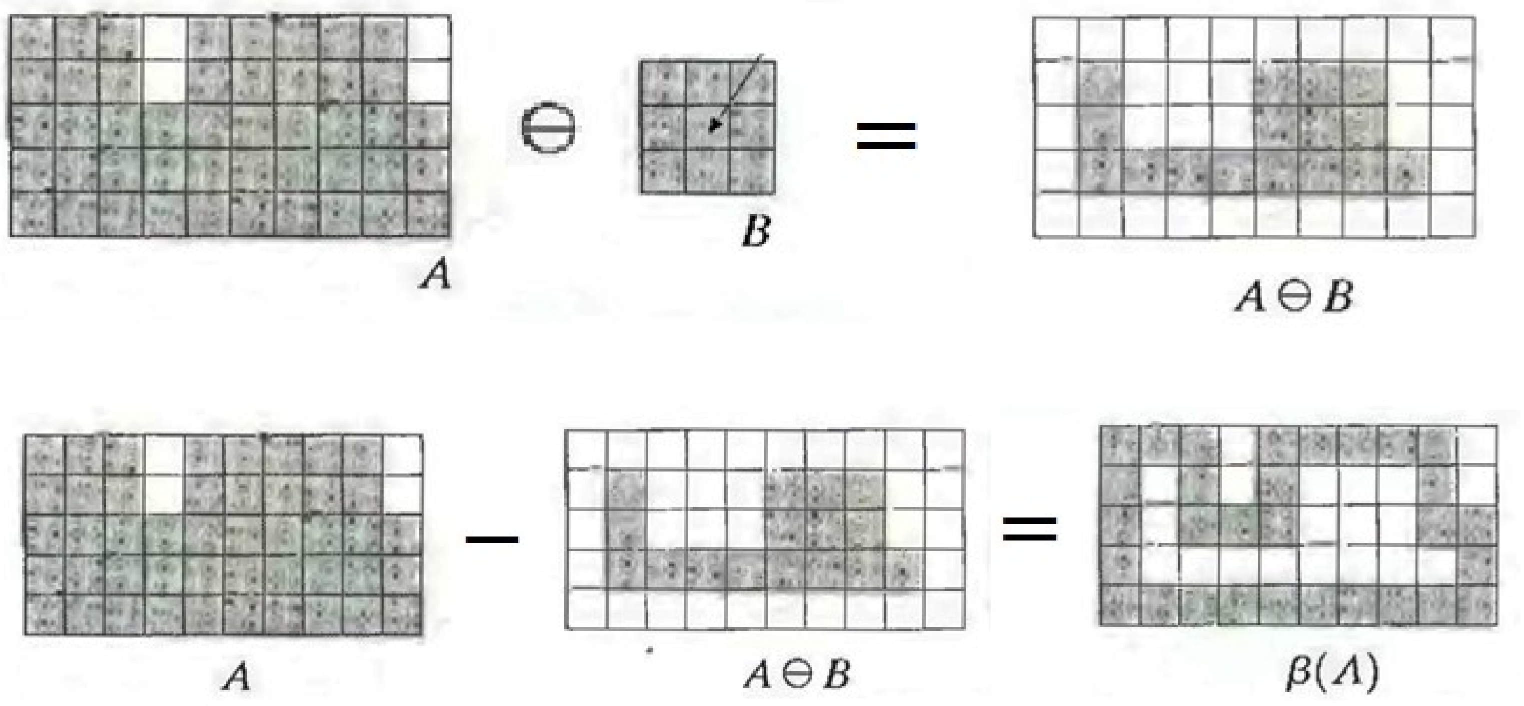
← Hasil transformasi Hit or Miss

Transformasi hit-or-miss

```
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/highgui.hpp>

using namespace cv;

int main(){
    Mat input_image = (Mat_<uchar>(8, 8) <<
        0, 0, 0, 0, 0, 0, 0, 0,
        0, 255, 255, 255, 0, 0, 0, 255,
        0, 255, 255, 255, 0, 0, 0, 0,
        0, 255, 255, 255, 0, 255, 0, 0,
        0, 0, 255, 0, 0, 0, 0, 0,
        0, 0, 255, 0, 0, 255, 255, 0,
        0, 255, 0, 255, 0, 0, 255, 0,
        0, 255, 255, 255, 0, 0, 0, 0);

    Mat kernel = (Mat_<int>(3, 3) <<
        0, 1, 0,
        1, -1, 1,
        0, 1, 0);

    Mat output_image;
    morphologyEx(input_image, output_image, MORPH_HITMISS, kernel);

    const int rate = 50;
    kernel = (kernel + 1) * 127;
    kernel.convertTo(kernel, CV_8U);

    resize(kernel, kernel, Size(), rate, rate, INTER_NEAREST);
    imshow("kernel", kernel);
    moveWindow("kernel", 0, 0);

    resize(input_image, input_image, Size(), rate, rate, INTER_NEAREST);
    imshow("Original", input_image);
    moveWindow("Original", 0, 200);

    resize(output_image, output_image, Size(), rate, rate, INTER_NEAREST);
    imshow("Hit or Miss", output_image);
    moveWindow("Hit or Miss", 500, 200);

    waitKey(0);
    return 0;
}
```

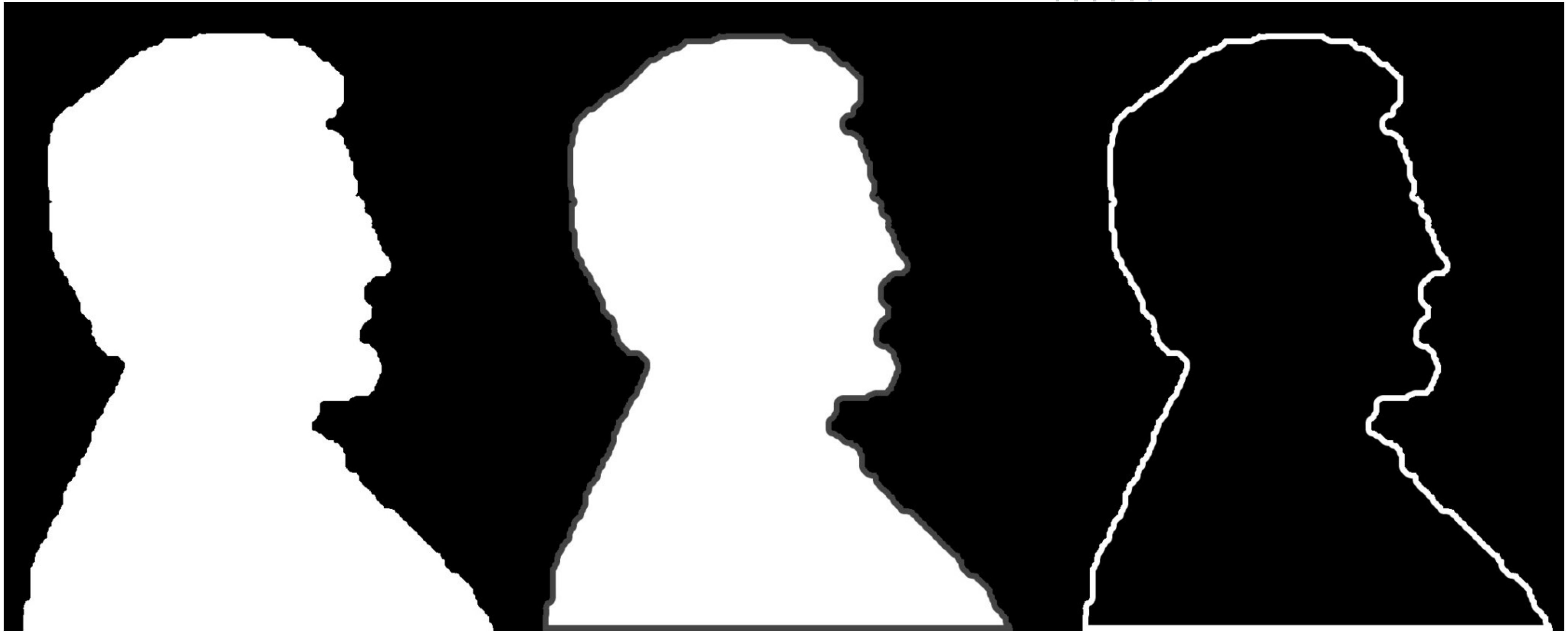
Boundary Extraction

- Pertama, erosi A dengan B, kemudian buat perbedaan set antara A dan Erosi
- Ketebalan kontur tergantung pada ukuran dari constructing object – B



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

Boundary Extraction



Original image, after dilation and boundary extraction with the help of dilation

Boundary Extraction

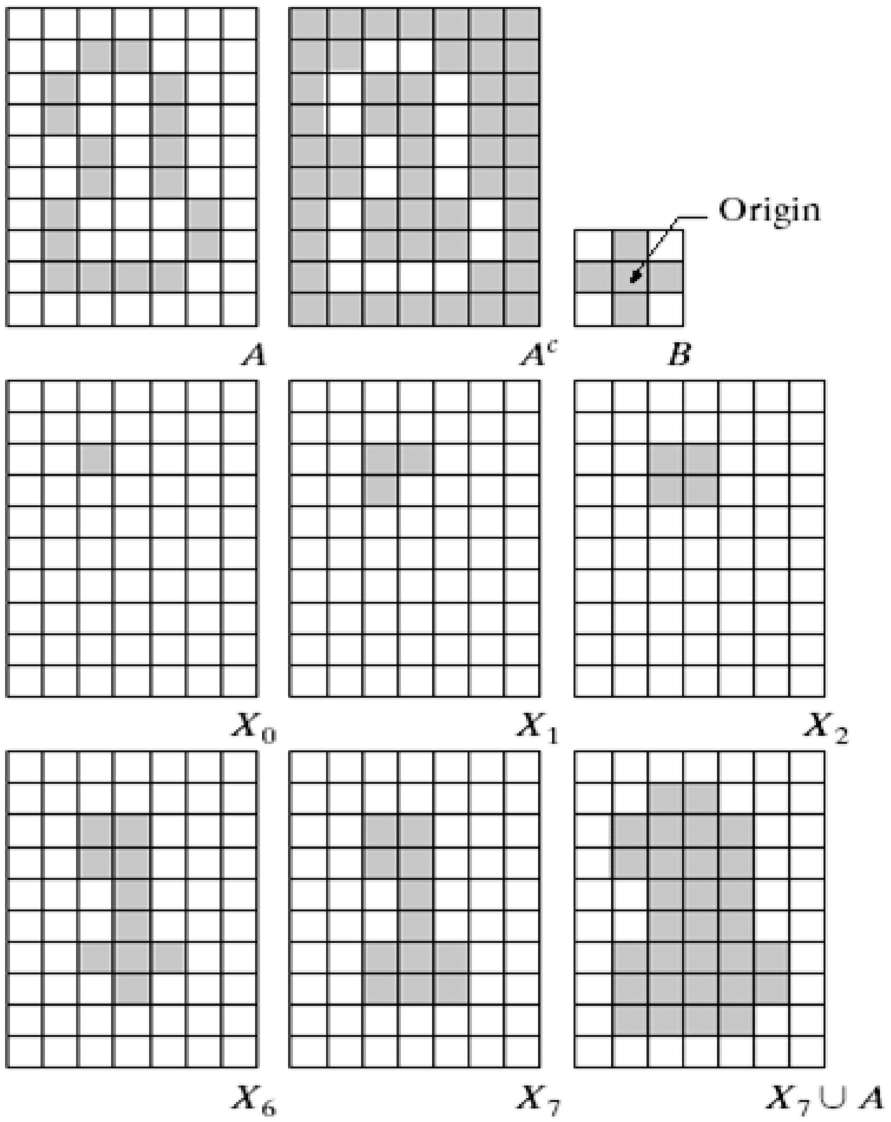
- Algoritma ini didasarkan pada set dari dilasi, komplemen dan intersections
- p is the point inside the boundary, with the value of 1
- X(k) = (X(k-1) xor B) conjunction denga complemented A, atau:

$$X_k = (X_{k-1} \oplus B) \cap A^c$$

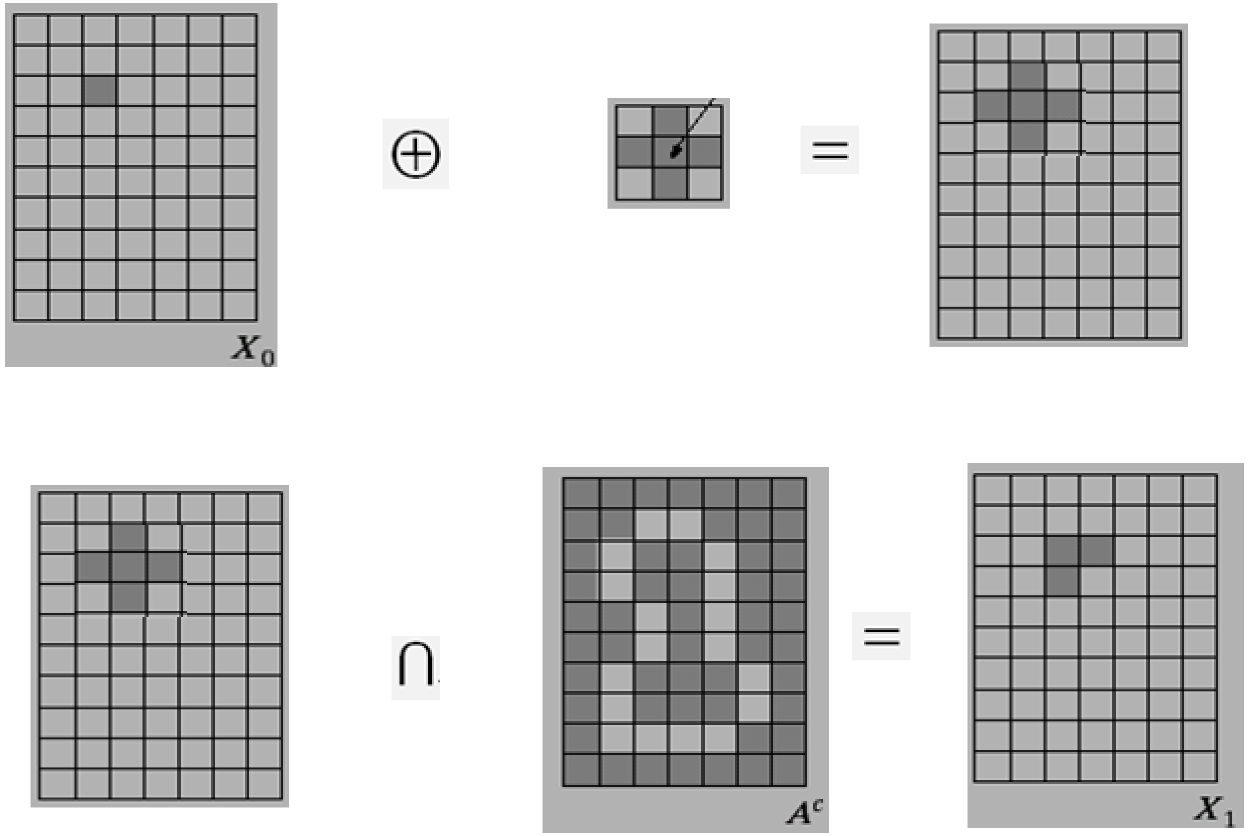
- Proses akan berhenti jika X(k) = X(k-1)
- Hasil diberikan oleh union dari A dan X(k), adalah sebuah set yang berisi isi-set dan tepi.

a	b	c
d	e	f
g	h	i

FIGURE 9.15 Region filling. (a) Set A. (b) Complement of A. (c) Structuring element B. (d) Initial point inside the boundary. (e)–(h) Various steps of Eq. (9.5-2). (i) Final result [union of (a) and (h)].



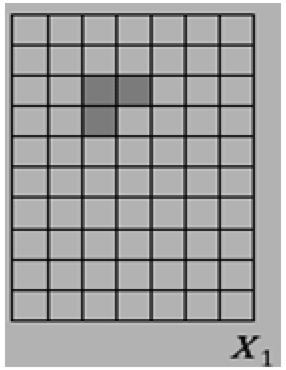
Boundary Extraction



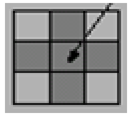
$$(X_{k-1} \oplus B)$$

$$X_k = (X_{k-1} \oplus B) \cap A^c$$

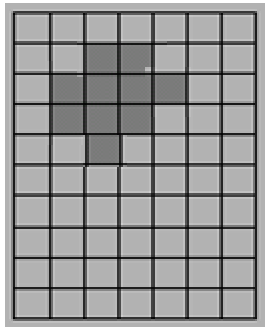
$$X_2 \rightarrow X_{k-1} = X_1$$



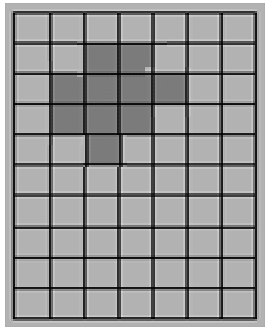
$\oplus$



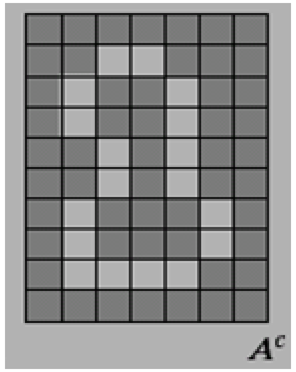
$=$



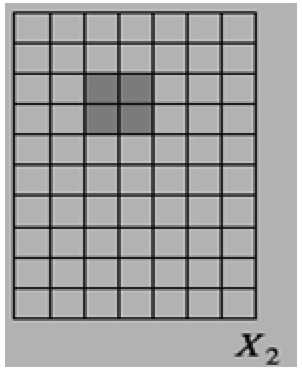
$$(X_{k-1} \oplus B)$$



$\cap$

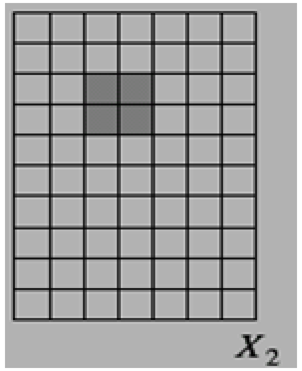


$=$

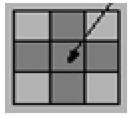


$$X_k = (X_{k-1} \oplus B) \cap A^c$$

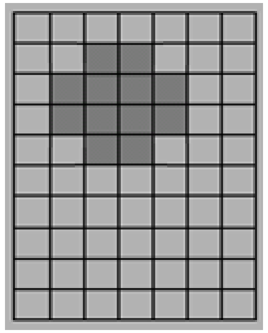
$$X_3 \rightarrow X_{k-1} = X_2$$



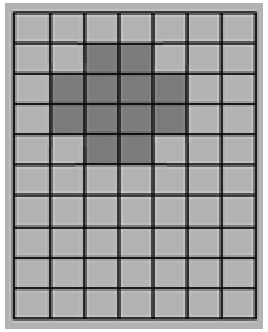
$\oplus$



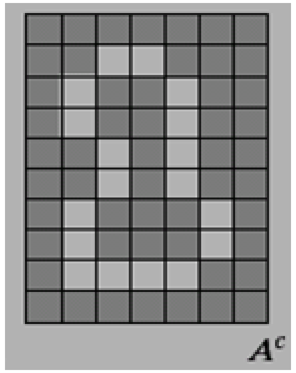
$=$



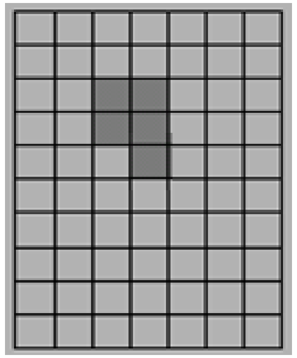
$(X_{k-1} \oplus B)$



$\cap$



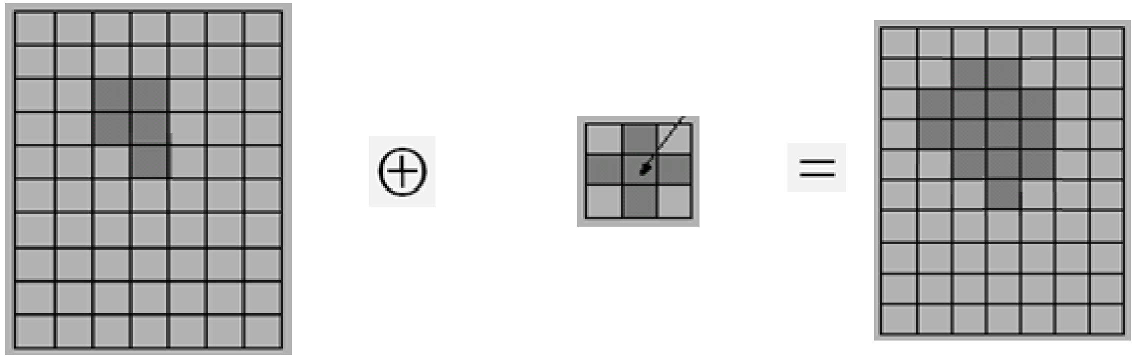
$=$



X_3

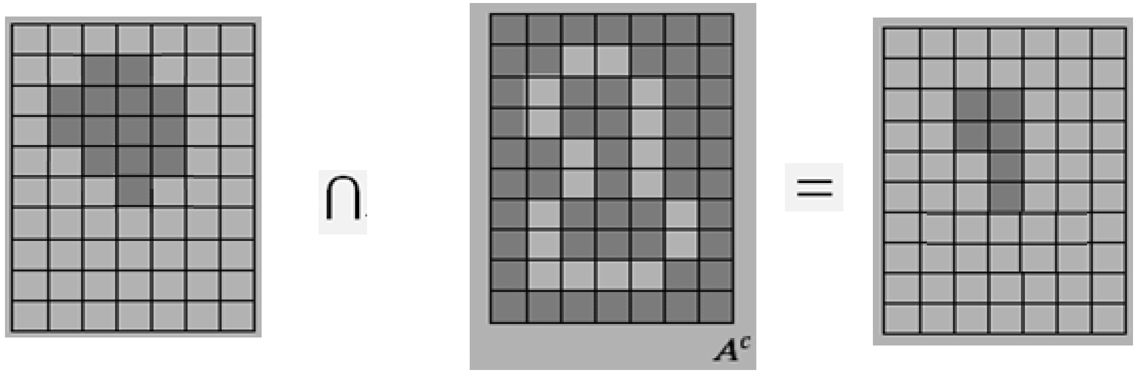
$$X_k = (X_{k-1} \oplus B) \cap A^c$$

$$X_4 \rightarrow X_{k-1} = X_3$$



X_3

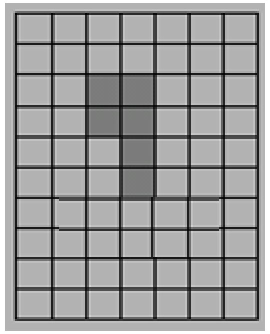
$(X_{k-1} \oplus B)$



X_4

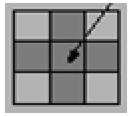
$$X_k = (X_{k-1} \oplus B) \cap A^c$$

$$X_5 \rightarrow X_{k-1} = X_4$$

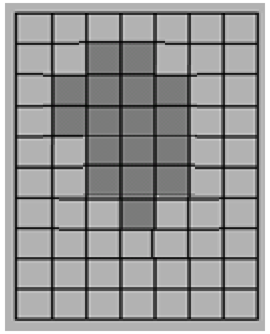


X_4

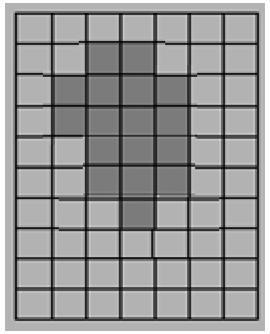
$\oplus$



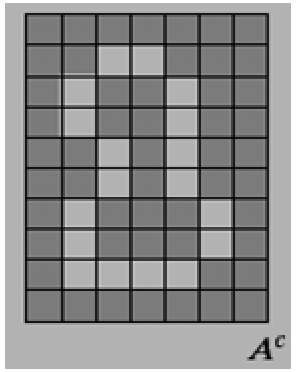
$=$



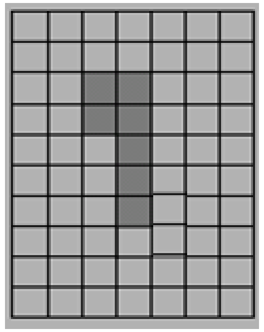
$$(X_{k-1} \oplus B)$$



$\cap$



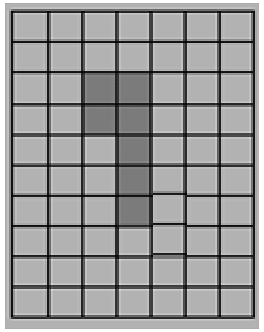
$=$



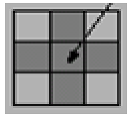
X_5

$$X_k = (X_{k-1} \oplus B) \cap A^c$$

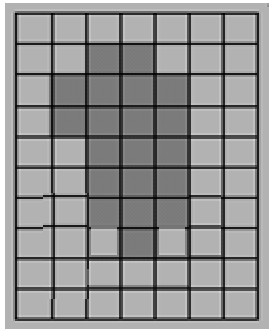
$$X_6 \rightarrow X_{k-1} = X_5$$



$\oplus$

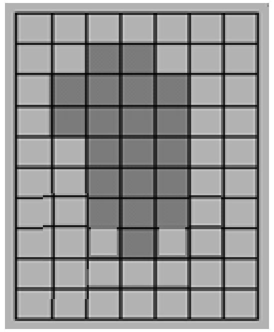


$=$

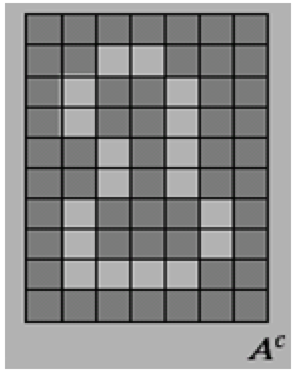


$(X_{k-1} \oplus B)$

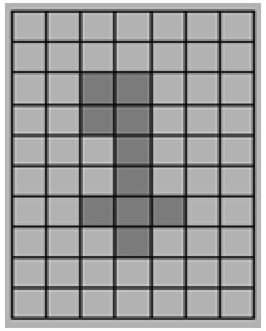
X_5



$\cap$



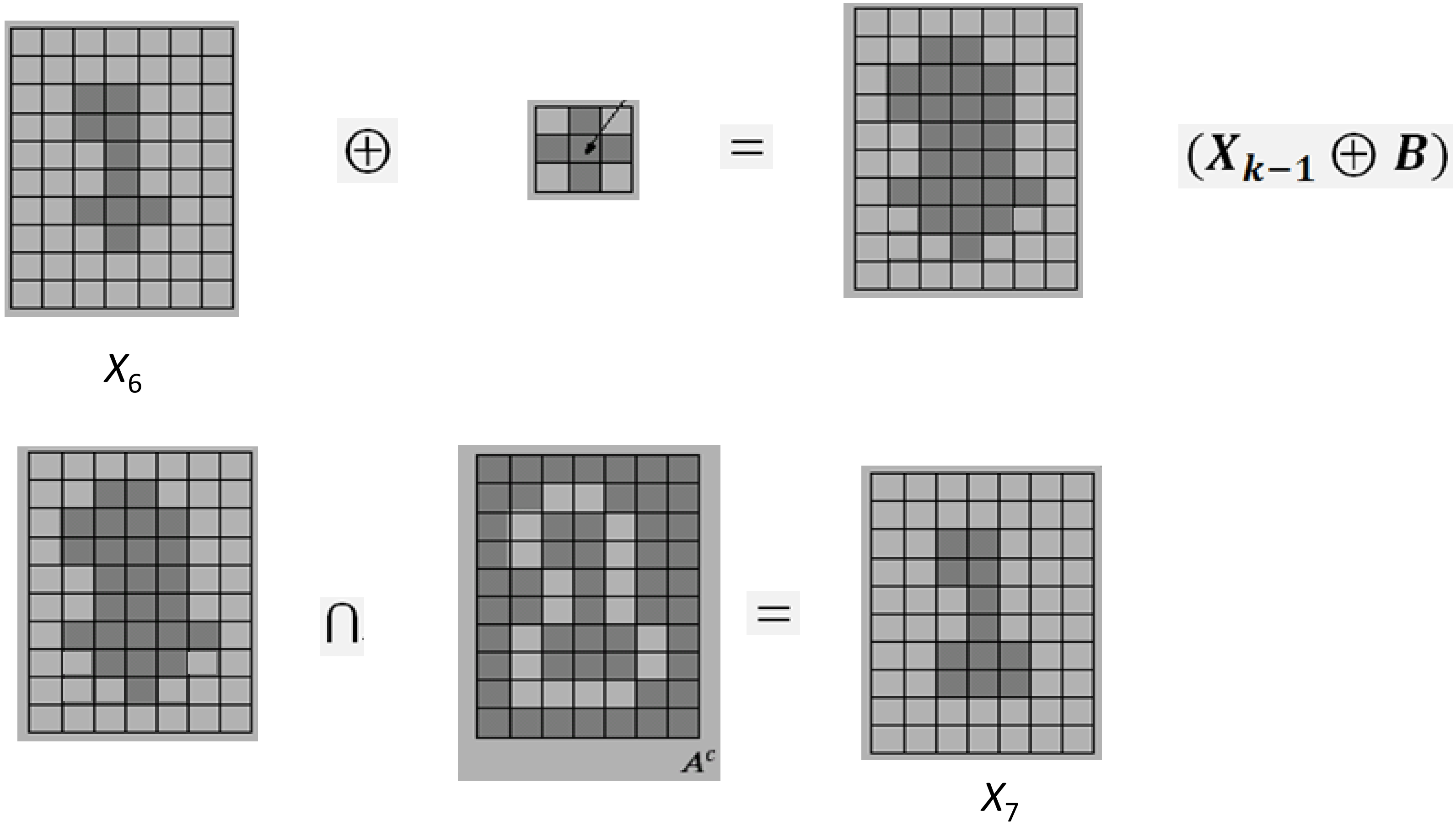
$=$



$$X_k = (X_{k-1} \oplus B) \cap A^c$$

X_6

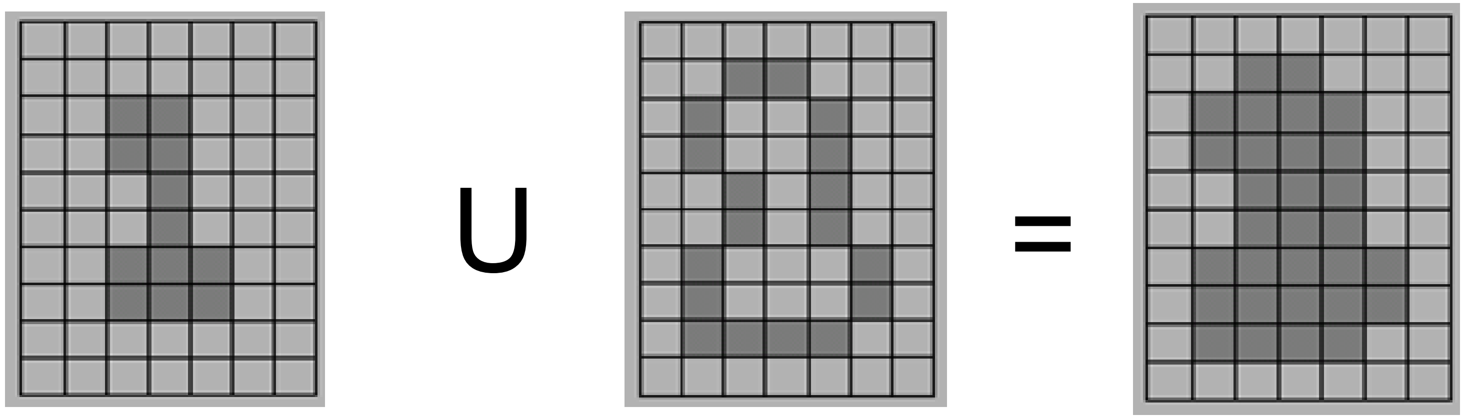
$$X_7 \rightarrow X_{k-1} = X_6$$



$$X_8 \rightarrow X_{k-1} = X_7$$

Dalam hal ini $X_8 = X_7$ sehingga proses dihentikan

- Dan hasil akhir adalah $X_7 \cup A$



- Algoritma ini mengekstrak sebuah komponenn dengan memilih sebuah titik pada binary object A
- Kerjanya mirip dengan region filling, namun disini menggunakan conjunction object A, instead of it’s complement

$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1,2,3,\ldots$$

- Proses iterasi akan berakhir jika $X_k = X_{k-1}$

- Belajar mandiri terkait:
  - Skeleton
  - Convex Hull
  - Thickening

Selamat Belajar