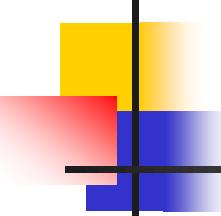


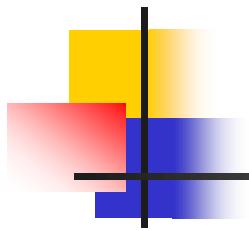


# Segmentasi Citra



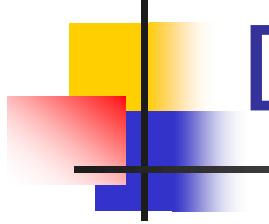
# Pendahuluan

- Segmentasi adalah proses mempartisi citra menjadi beberapa daerah atau objek, berdasarkan sifat-sifat tertentu dari citra.
- Algoritma segmentasi didasarkan pada salah satu dari sifat dasar nilai intensitas berikut :
  - ❖ Discontinuity : pembagian citra berdasarkan perubahan besar nilai intensitasnya seperti tepi
  - ❖ Similarity : pembagian ke dalam region-region yang serupa sesuai kriteria yang diberikan



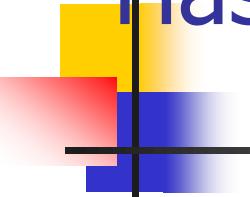
# Deteksi Discontinuity

- Titik
- Garis
- Tepi



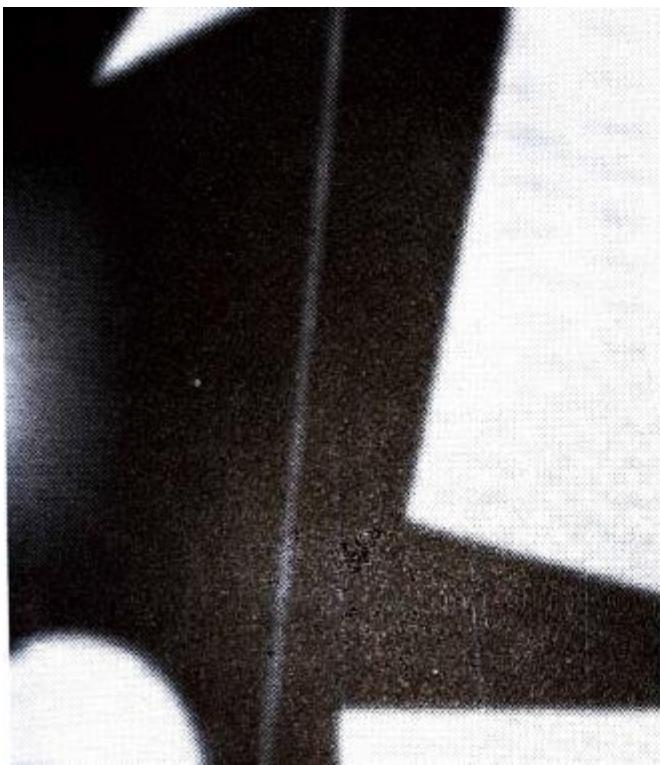
# Deteksi Titik

- Isolated point adalah sebuah titik yang gray-levelnya sangat berbeda dari backgroundnya dan ada di daerah homogen atau hampir homogen.
- Deteksi titik dapat dilakukan dengan operasi konvolusi pada mask yang jumlah koefisiennya 0 , mengindikasikan bahwa respons mask akan bernilai 0 pada area gray-level yang konstan.

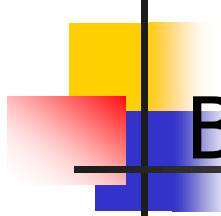


# Hasil deteksi titik pada mask

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

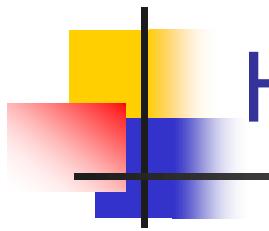


# Deteksi Garis

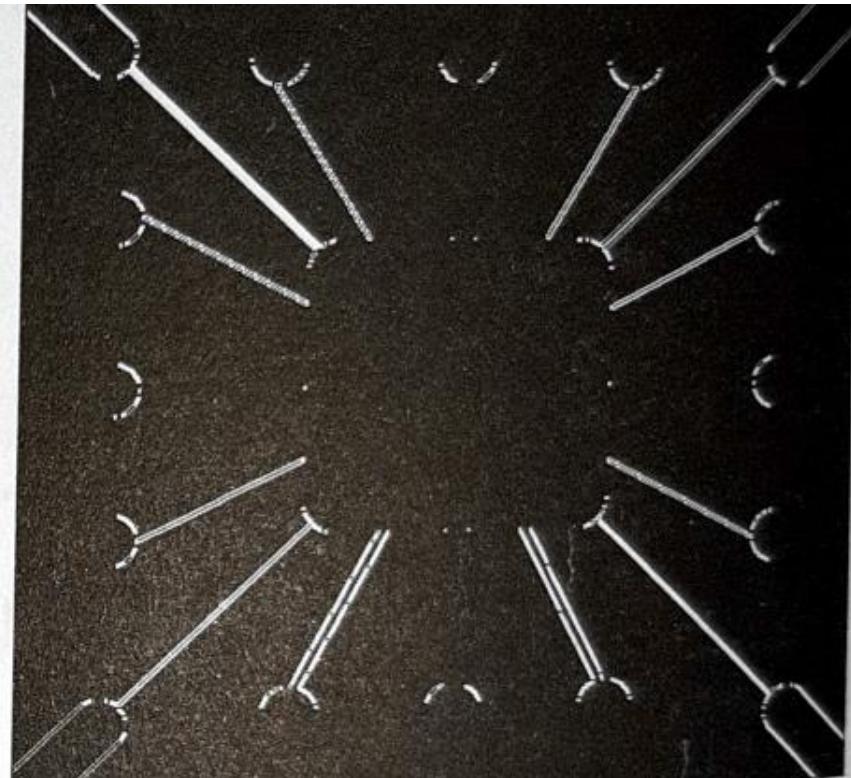
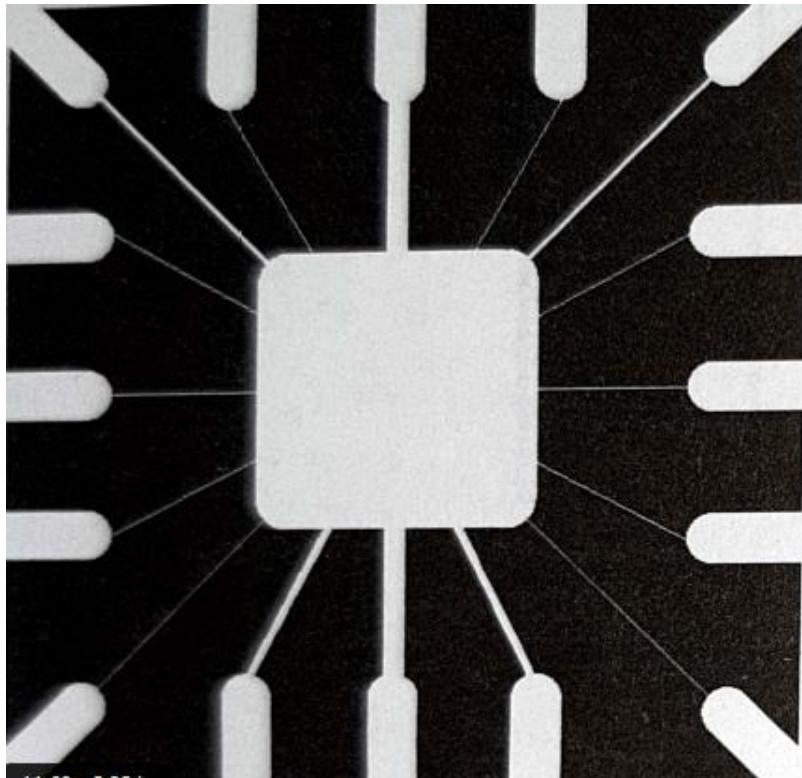


Beberapa mask untuk deteksi garis :

- Garis horisontal  $\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$
- Garis arah  $+45^\circ$   $\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$
- Garis vertikal  $\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$
- Garis arah  $-45^\circ$   $\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$

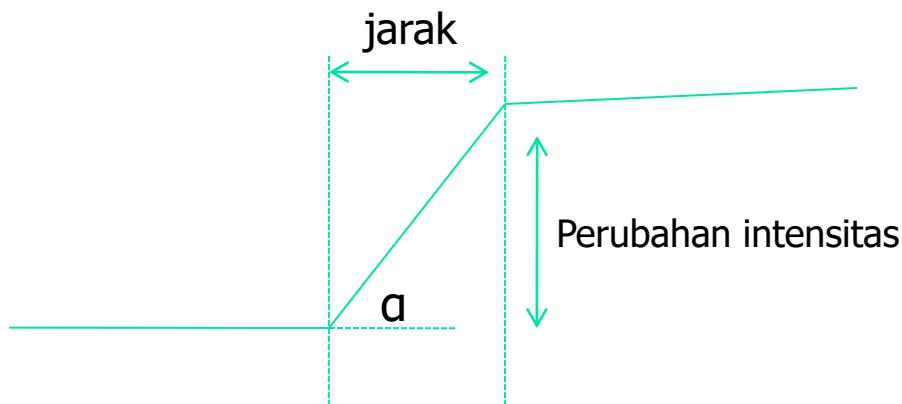


# Hasil Deteksi garis arah -45°

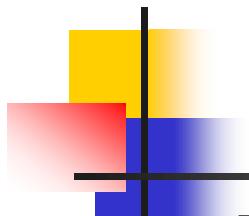


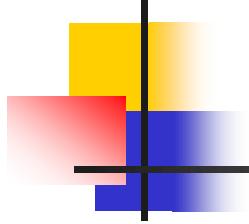
# Deteksi Tepi

Tepi (edge) adalah perubahan nilai derajat keabuan yang mendadak (besar) dalam jarak yang singkat.



Model tepi satu matra

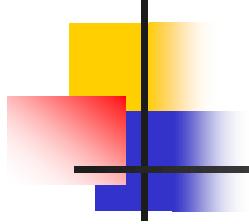
- 
- Perubahan intensitas yang besar dalam jarak yang singkat dipandang sebagai fungsi yang memiliki kemiringan yang besar.
  - Kemiringan fungsi biasanya diperoleh dengan menghitung turunan pertama (*gradient*).

- 
- Gradient citra  $f(x, y)$  sebagai berikut :

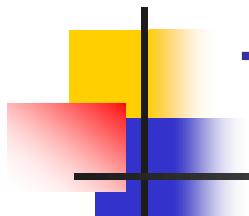
$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- Magnitude dari vektor ini :

$$mag(\nabla f) = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

- 
- Keputusan apakah suatu pixel merupakan tepi atau bukan tepi dinyatakan dengan operasi thresholding berikut :

$$g(x, y) = \begin{cases} 1, & \text{jika } \text{mag}(x, y) \geq T \\ 0, & \text{lainnya} \end{cases}$$



# Beberapa Teknik untuk Mendeteksi Tepi

1. Operator Sobel
2. Operator Prewitt
3. Operator Roberts

# 1. Operator Sobel

Tinjau pengaturan pixel di sekitar pixel  $(x,y)$  :

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Dengan turunan parsial sebagai berikut :

$$s_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$s_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$

- Operator Sobel adalah magnitudo dari gradien yang dihitung dengan :

$$M = \sqrt{s_x^2 + s_y^2}$$

dimana

$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

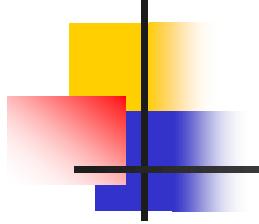
# Contoh

$$\begin{bmatrix} 3 & 4 & 2 & 5 & 1 \\ 2 & 1 & 6 & 4 & 2 \\ 3 & 5 & 7 & 1 & 3 \\ 4 & 2 & 5 & 7 & 1 \\ 2 & 5 & 1 & 3 & 2 \end{bmatrix}$$

(i) Citra semula

$$\begin{bmatrix} * & * & * & * & * \\ * & 18 & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix}$$

(ii) Hasil konvolusi



Nilai 18 pada citra hasil konvolusi diperoleh dengan perhitungan berikut :

$$S_x = (3)(-1) + (2)(-2) + (3)(-1) + (2)(1) + (6)(2) + (7)(1) = 11$$

$$S_y = (3)(1) + (4)(2) + (2)(1) + (3)(-1) + (5)(-2) + (7)(-1) = -7$$

$$M = \sqrt{S_x^2 + S_y^2} = \sqrt{11^2 + (-7)^2} \cong |S_x| + |S_y| = |11| + |-7| = 18$$

# Hasil deteksi tepi dengan Operator Sobel

citra asal



Deteksi tepi dg op.sobel



## 2. Operator Prewitt

Persamaan gradien pada operator Prewitt sama seperti operator Sobel, tetapi menggunakan  $c = 1$

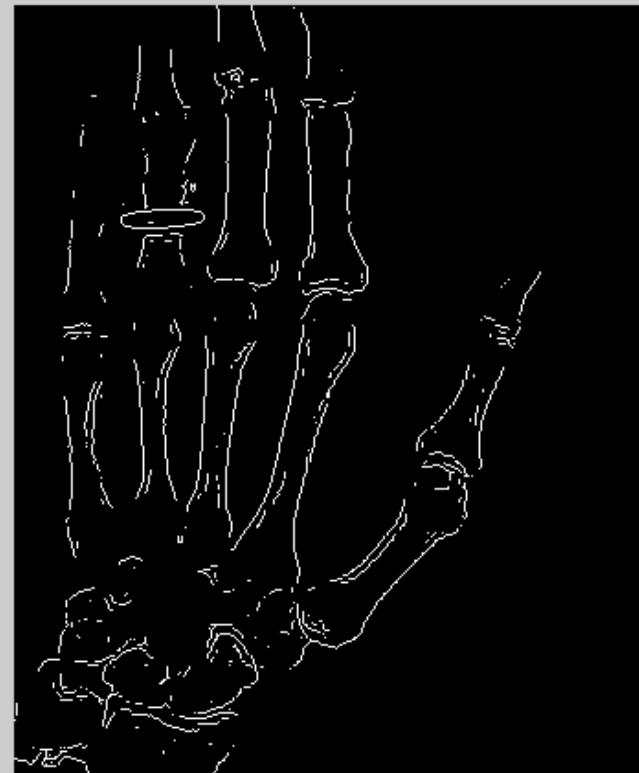
$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

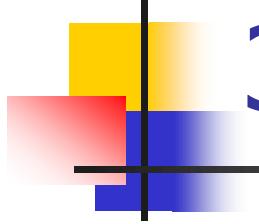
# Hasil deteksi tepi dengan Operator Prewitt

citra asal



Deteksi tepi dg op.prewitt



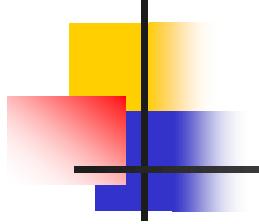


### 3. Operator Robert

$$R_+ = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$R_- = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

# Contoh:


$$\begin{bmatrix} 3 & 4 & 2 & 5 & 1 \\ 2 & 1 & 6 & 4 & 2 \\ 3 & 5 & 7 & 1 & 3 \\ 4 & 2 & 5 & 7 & 1 \\ 2 & 5 & 1 & 3 & 2 \end{bmatrix}$$

Citra asal

$$\begin{bmatrix} 4 & 3 & 3 & 6 & * \\ 5 & 7 & 8 & 2 & * \\ 2 & 5 & 4 & 4 & * \\ 1 & 1 & 8 & 7 & * \\ * & * & * & * & * \end{bmatrix}$$

Hasil konvolusi

Nilai 4 pada pojok kiri atas diperoleh dg perhitungan sbb:

$$R_+ = |3 - 1| = 2 \quad R_- = |4 - 2| = 2$$

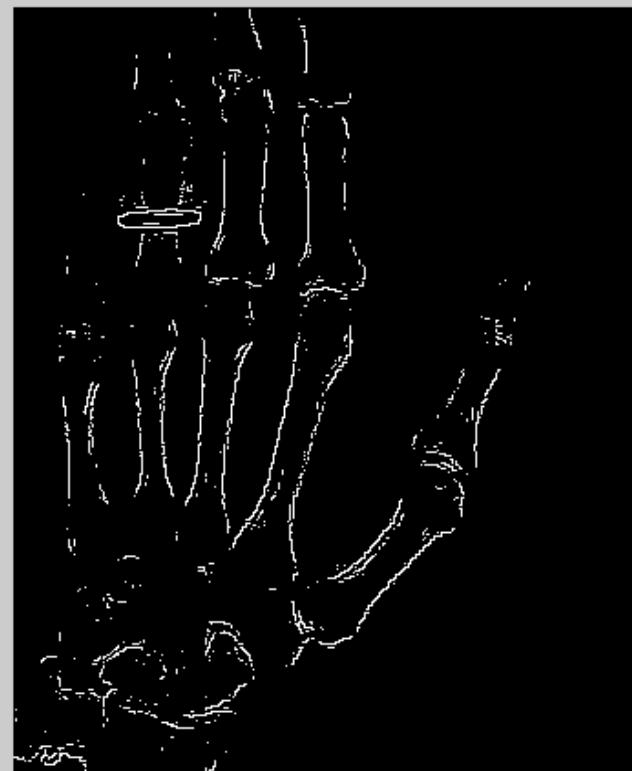
$$M = |R_+| + |R_y| = 4$$

# Hasil deteksi tepi dengan Operator Robert

citra asal



Deteksi tepi dg op.robert



citra asal



Deteksi tepi dg op.sobel

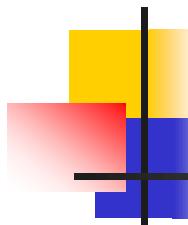


Deteksi tepi dg op.prewitt



Deteksi tepi dg op.robert





citra asal



Deteksi tepi dg op.sobel

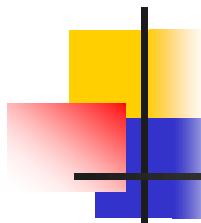


Deteksi tepi dg op.prewitt



Deteksi tepi dg op.robert





citra asal



Deteksi tepi dg op.sobel



Deteksi tepi dg op.prewitt



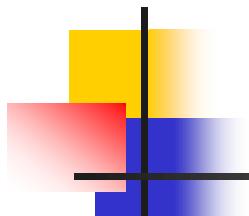
Deteksi tepi dg op.robert



## 4. Operator Laplace

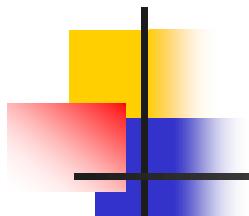
- $$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$
 
$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$
 
$$\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$



# Deteksi Similarity

- Beberapa teknik segmentasi citra :
  - ✓ Thresholding (global thresholding dan lokal adaptif thresholding)
  - ✓ Connected Component labelling
  - ✓ Segmentasi Berbasis Clustering (Iterasi, K-means, fuzzy C-means, SOM)



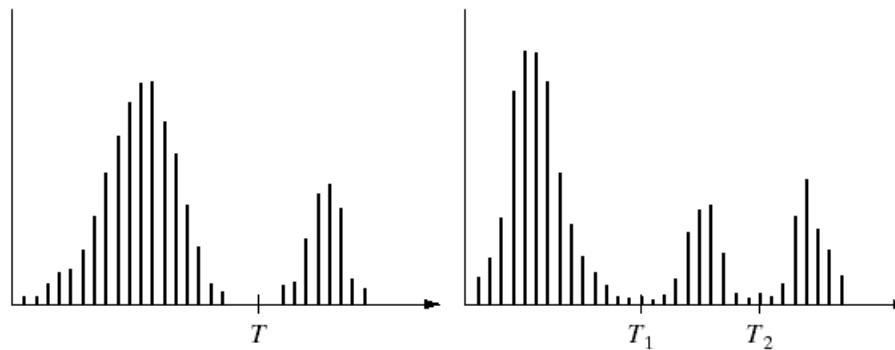
# Pengambangan(Thresholding)

- Proses pengambangan akan menghasilkan citra biner, yaitu citra hitam dan putih.
- Secara umum proses pengambangan citra grayscale adalah sbb :

$$g(x, y) = \begin{cases} 1, & f(x, y) \geq T \\ 0, & f(x, y) < T \end{cases}$$

# Global Thresholding : Metode Histogram

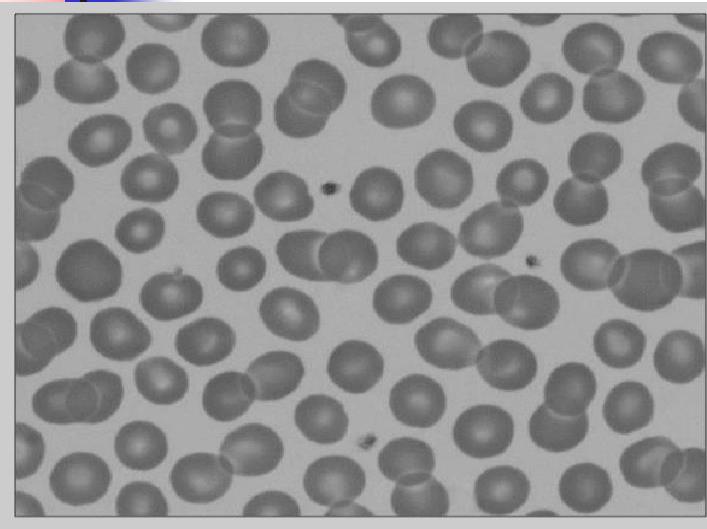
- Nilai  $T$  ditentukan berdasarkan histogram dari citra yang akan diambil.



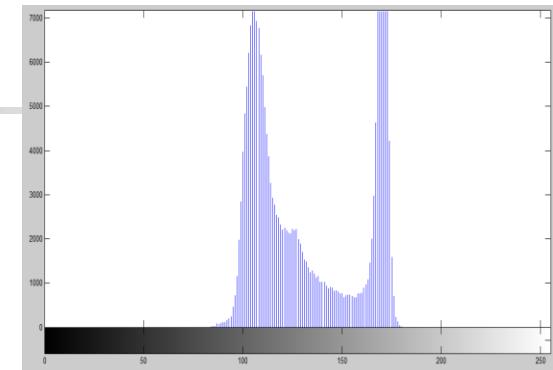
a b

**FIGURE 10.26** (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

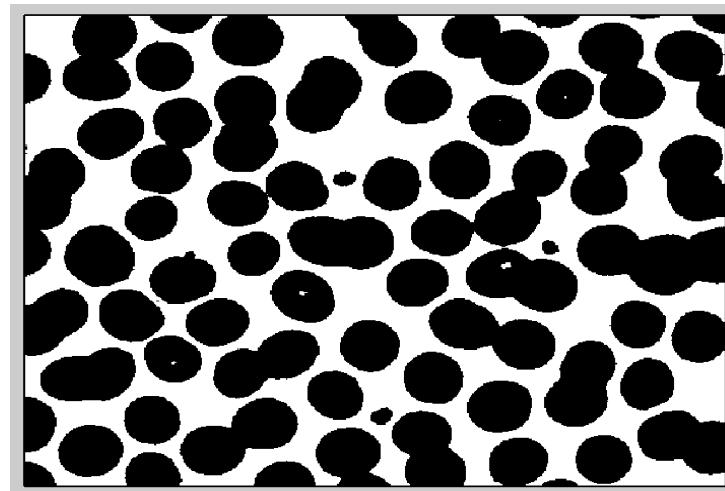
# Contoh: Single Thresholding



Citra Sel darah merah

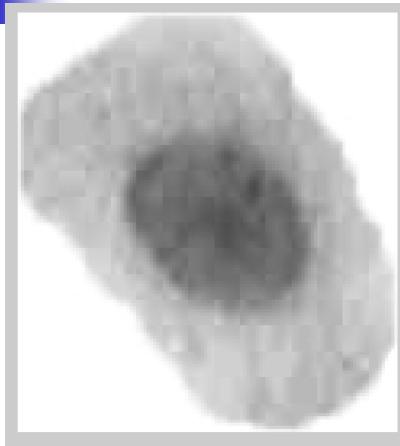


Histogram Citra

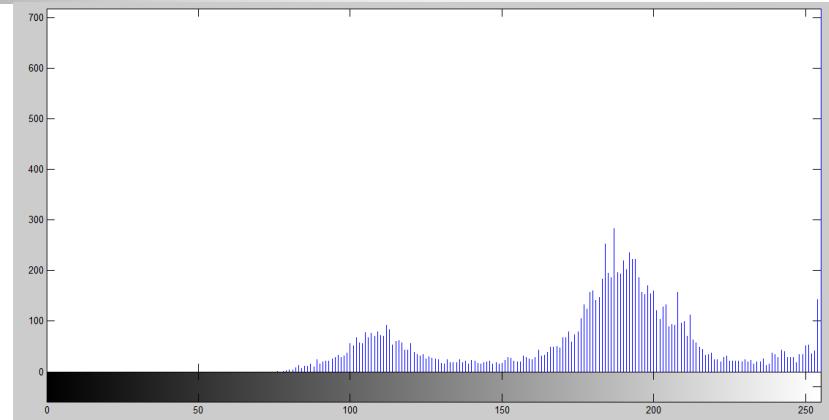


Hasil Segmentasi dg  $T = 160$

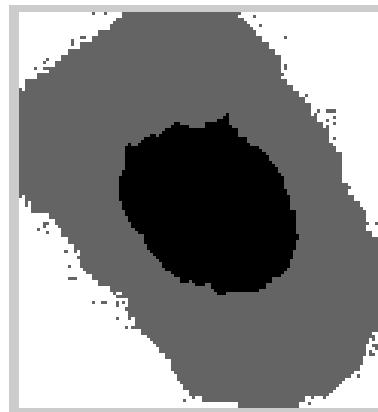
# Contoh: Multiple Thresholding



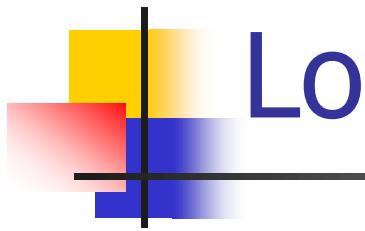
Citra sel serviks



Histogram Citra

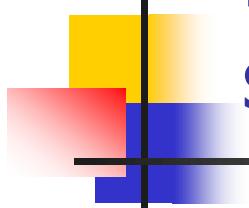


Hasil Multiple Thresholding



# Lokal Adaptif Thresholding

- Motivasi : Citra dengan lembah histogram yang tinggi/tanpa lembah.
- Prinsip dasar :  
Citra dibagi menjadi blok-blok kecil dan kemudian dilakukan pengambangan lokal pada setiap blok dengan nilai  $T$  yang berbeda.



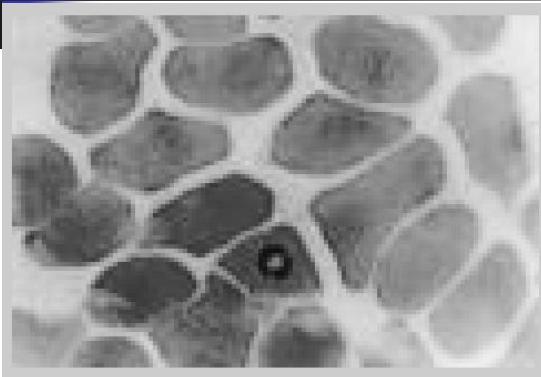
Nilai ambang lokal dapat dihitung dengan salah satu dari tiga cara berikut :

$$T = \frac{\sum_{(x,y) \in W} f(x, y)}{N_W} - C$$

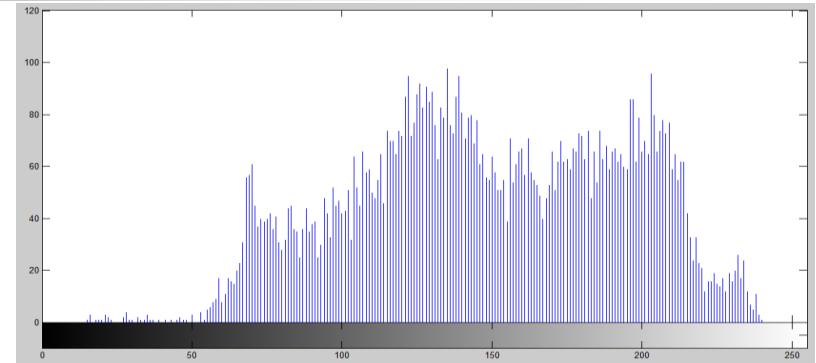
$$T = median\{f(x, y), (x, y) \in W\}$$

$$T = \frac{\max\{f(x, y), (x, y) \in W\} + \min\{f(x, y), (x, y) \in W\}}{2}$$

# Contoh: Lokal Adaptif Thresholding



Citra grayscale



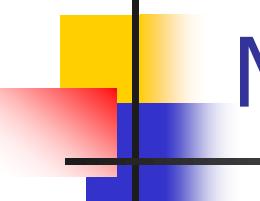
Histogram Citra



Hasil Global Thresholding



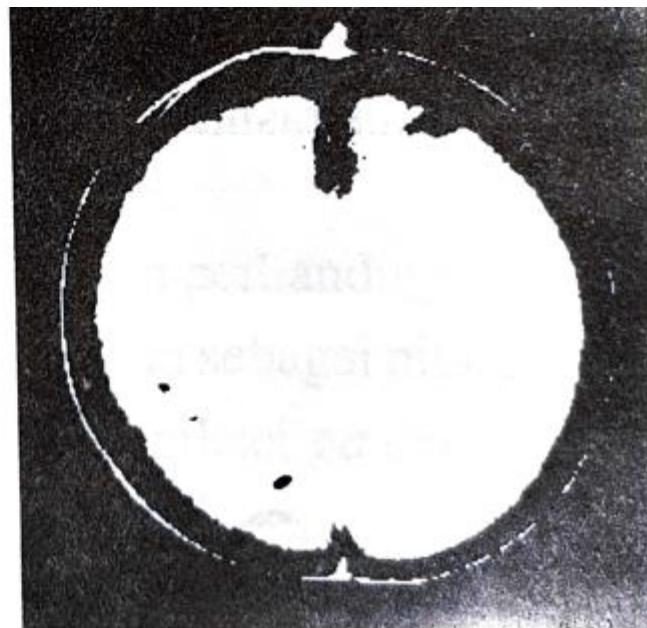
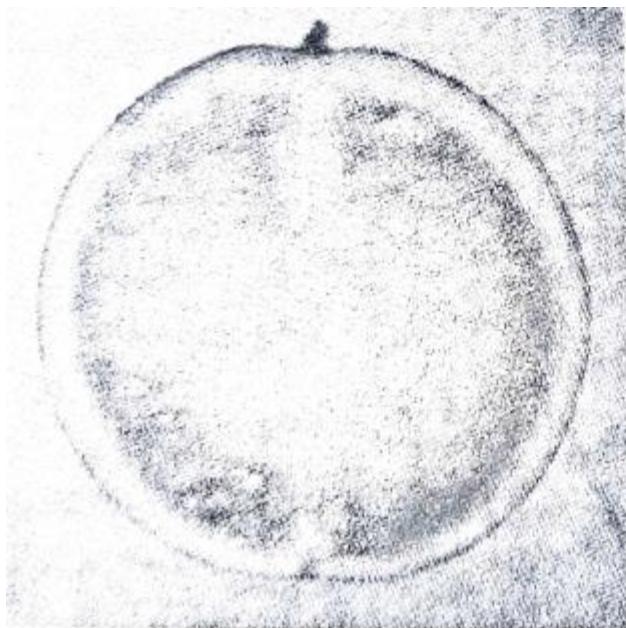
Hasil Lokal Adaptif Thresholding



# Segmentasi Berbasis Clustering : Metode Iterasi

1. Pilih nilai T awal, biasanya dipakai nilai rata-rata dari intensitas citra
2. Segmentasi citra menjadi dua daerah, misalnya R1 dan R2 dengan menggunakan nilai T awal sebelumnya.
3. Hitung nilai rata-rata intensitas pada daerah R1 dan R2. Kedua nilai rata-rata tersebut berturut-turut disebut  $r_1$  dan  $r_2$
4. Hitung nilai T baru dengan rumus  $T = (r_1 + r_2)/2$
5. Ulangi langkah 2 sampai 4 sampai nilai T tercapai. Nilai T dikatakan telah tercapai bila nilai T tidak mengalami perubahan nilai T lagi.

# Hasil Segmentasi metode iterasi



# Segmentasi Berbasis Clustering : SOM/Kohonen

Algoritma pengelompokan pola jaringan kohonen adalah sebagai berikut:

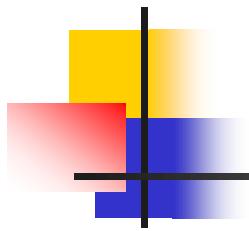
1. Inisialisasi
  - i. Bobot  $w_{ij}$  (acak)
  - ii. Laju pemahaman awal dan faktor penurunannya
  - iii. Bentuk dan jari-jari ( $R$ ) topologi sekitarnya
2. Melakukan langkah 3-8 selama kondisi penghentian bernilai salah
3. Melakukan langkah 4-6 untuk setiap vektor input  $x$
4. Menghitung jarak *Euclidean* untuk semua  $j$

$$D(j) = \sum_i (w_{ij} - x_i)^2 \quad \dots \dots \dots (2.16)$$

5. Menentukan indeks  $J$  sedemikian hingga  $D(J)$  minimum
6. Memodifikasi bobot untuk setiap unit  $j$  di sekitar  $J$

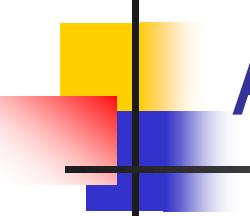
$$w_{ji}^{baru} = w_{ji}^{lama} + a (x_i - w_{ji}^{lama}) \quad \dots \dots \dots (2.17)$$

7. Memodifikasi laju pemahaman
8. Menguji kondisi penghentian



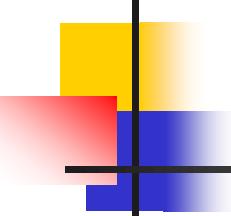
# Hasil segmentasi SOM





# Segmentasi Berbasis Clustering : Algoritma K-Means

1. Tentukan jumlah kelompok
2. Alokasikan data ke dalam kelompok secara acak
3. Hitung pusat cluster (centroid/rata-rata) dari data yang ada di masing-masing cluster
4. Alokasikan masing-masing data ke centroid/rata-rata terdekat
5. Kembali ke Langkah 3,
  - apabila masih ada data yang berpindah cluster,
  - atau apabila perubahan nilai centroid ada yang di atas nilai threshold yang ditentukan,
  - atau apabila perubahan nilai pada fungsi obyektif yang digunakan masih di atas nilai threshold yang ditentukan



# Algoritma K-Means

Jika  $M$  menyatakan jumlah data dalam sebuah cluster,  $i$  menyatakan fitur ke- $i$  dalam sebuah cluster dan  $p$  menyatakan dimensi data, maka untuk menghitung centroid fitur ke- $i$  digunakan formula:

$$C_i = \frac{1}{M} \sum_{j=1}^M x_j$$

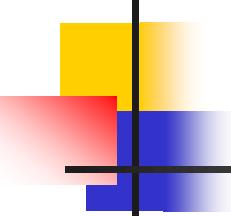
- dilakukan sebanyak  $p$  dimensi, sehingga  $i$  mulai 1 sampai  $p$ .
- Cara mengukur jarak data ke pusat cluster, diantaranya : Euclidean (Bezdek, 1981), Manhattan/City Block (Miyamoto dan Agusta, 1995), dan Minkowsky (Miyamoto dan Agusta, 1995)

Euclidean

$$D(x_2, x_1) = \|x_2 - x_1\|_2 = \sqrt{\sum_{j=1}^p |x_{2j} - x_{1j}|^2}$$

Manhattan/City Block

$$D(x_2, x_1) = \|x_2 - x_1\|_1 = \sum_{j=1}^p |x_{2j} - x_{1j}|$$



# Algoritma K-Means

- Pengalokasian data ke cluster dapat dirumuskan sebagai berikut (MacQueen, 1967):

$$a_{il} = \begin{cases} 1 & d = \min\{D(x_i, C_l)\} \\ 0 & \text{lainnya} \end{cases}$$

- Dimana  $a_{il}$  adalah nilai keanggotaan titik  $x_i$  ke pusat cluster  $C_l$ ,  $d$  adalah jarak terpendek dari data  $x_i$  ke  $K$  cluster setelah dibandingkan, dan  $C_l$  centroid (pusat cluster) ke- $l$ .
- Fungsi objektif berdasarkan jarak dan nilai keanggotaan data dalam cluster

$$J = \sum_{i=1}^N \sum_{l=1}^K a_{ic} D(x_i, C_l)^2$$

- Dimana  $N$  adalah jumlah data,  $K$  adalah jumlah cluster,  $a_{il}$  adalah nilai keanggotaan titik data  $x_i$  ke pusat cluster  $C_l$ ,  $C_l$  adalah pusat cluster ke- $l$ ,  $D(x_i, C_l)$  adalah jarak titik  $x_i$  ke cluster  $C_l$  yang diikuti.
- Untuk  $a$  mempunyai nilai 0 atau 1. Apabila suatu data merupakan anggota suatu kelompok maka nilai  $a_{il} = 1$ , jika tidak, akan maka nilai  $a_{il} = 0$

# Hasil Segmentasi menggunakan Fuzzy C-Means

