

PENGOLAHAN CITRA DIGITAL

IMAGES ENHANCED (SPATIAL DOMAIN)

GKV - IF 2020

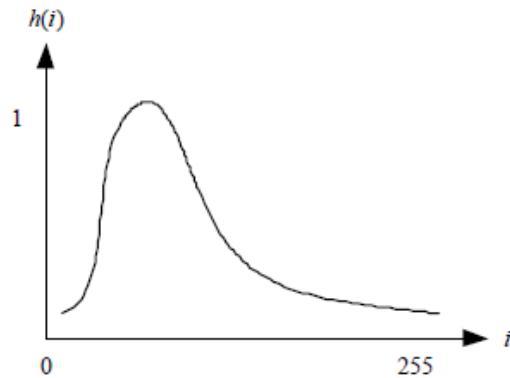
OUTLINE

- HISTOGRAM
- KONVOLUSI
- PELEMBUTAN GAMBAR
- PENAJAMAN GAMBAR
- PEREGANGAN KONTRAS

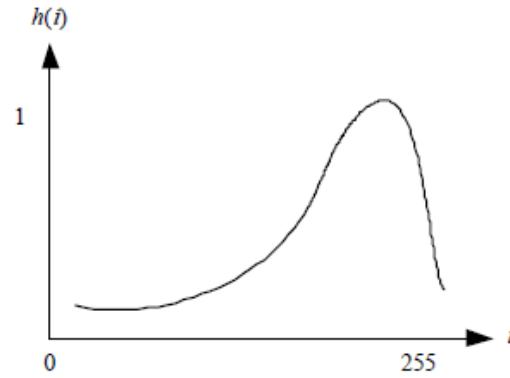
HISTOGRAM-1

- Histogram sebuah citra digital merupakan grafik yang memvisualisasikan distribusi nilai intensitas pixel dari citra digital.
- Histogram merupakan sebuah tool yang sangat berguna untuk mengetahui kualitas sebuah citra digital.
- Berdasarkan penyebaran atau pengelompokan pixel dari histogram akan dapat diketahui apakah sebuah citra terlalu terang atau terlalu gelap atau dalam keadaan normal.

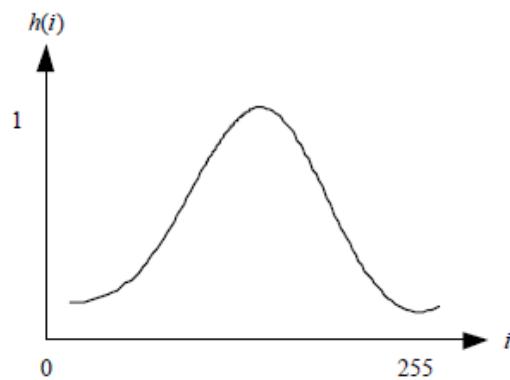
HISTOGRAM-2



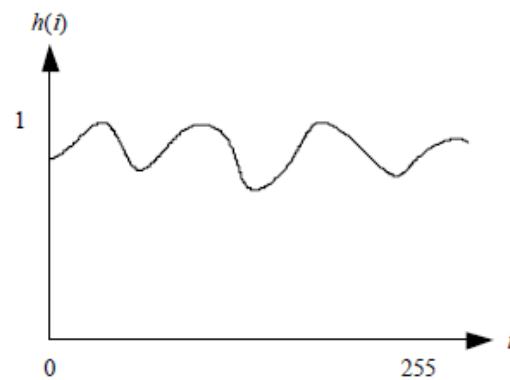
(a)



(b)



(c)



(d)

(a) citra gelap, (b) citra terang, (c) citra normal dan (d) citra normal dengan kontras tinggi

MEMBANGUN HISTOGRAM-1

- Histogram citra digital dengan derajat keabuan $[0, L-1]$ merupakan sebuah fungsi diskrit :

$$h(r_k) = n_k$$

dimana r_k merupakan derajat keabuan ke k dan n_k merupakan jumlah pixel dalam citra yang memiliki derajat keabuan r_k .

- Nilai L bergantung pada kedalaman bit yang digunakan. Untuk citra digital dengan kedalaman 8 bit maka nilai L memiliki rentang 0 – 255.

MEMBANGUN HISTOGRAM-2

- Untuk menormalisasi histogram maka dilakukan dengan cara membagi masing – masing nilai n_k dengan jumlah pixel pada citra tersebut dengan rumus :

$$p(r_k) = n_k / n$$

dimana n merupakan jumlah pixel dari citra.

MEMBANGUN HISTOGRAM-3

- Andaikan terdapat sebuah citra dengan kedalaman 3 bit, maka derajat keabuannya memiliki rentang nilai 0 – 7 dengan ukuran 4×4 pixel dengan matriks :

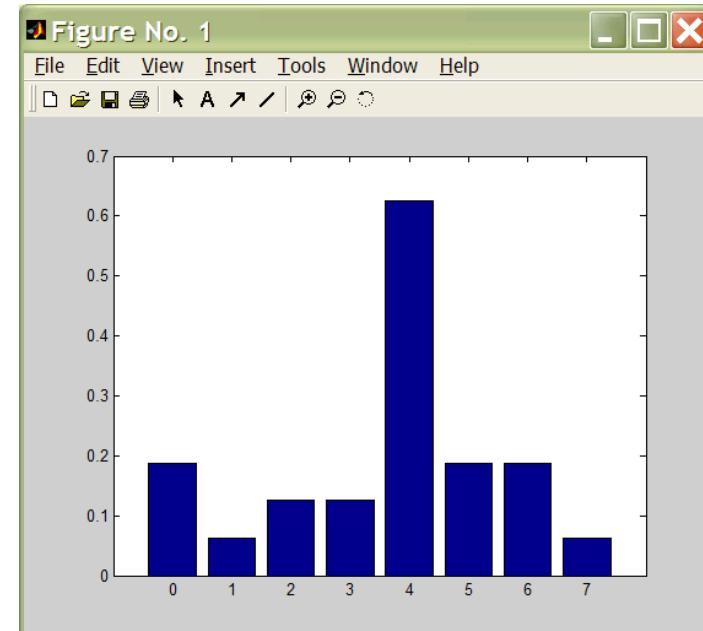
$$\begin{bmatrix} 1 & 7 & 0 & 5 \\ 6 & 6 & 0 & 2 \\ 0 & 5 & 6 & 5 \\ 3 & 2 & 4 & 3 \end{bmatrix}$$

MEMBANGUN HISTOGRAM-4

Menghitung frekuensi kemunculan pixel dan mentabulasikannya

Derajat keabuan	Jumlah	$p(r_k) = n_k / n$
0	3	0,1875
1	1	0,0625
2	2	0,125
3	2	0,125
4	1	0,0625
5	3	0,1875
6	3	0,1875
7	1	0,0625

MEMBANGUN HISTOGRAM-5

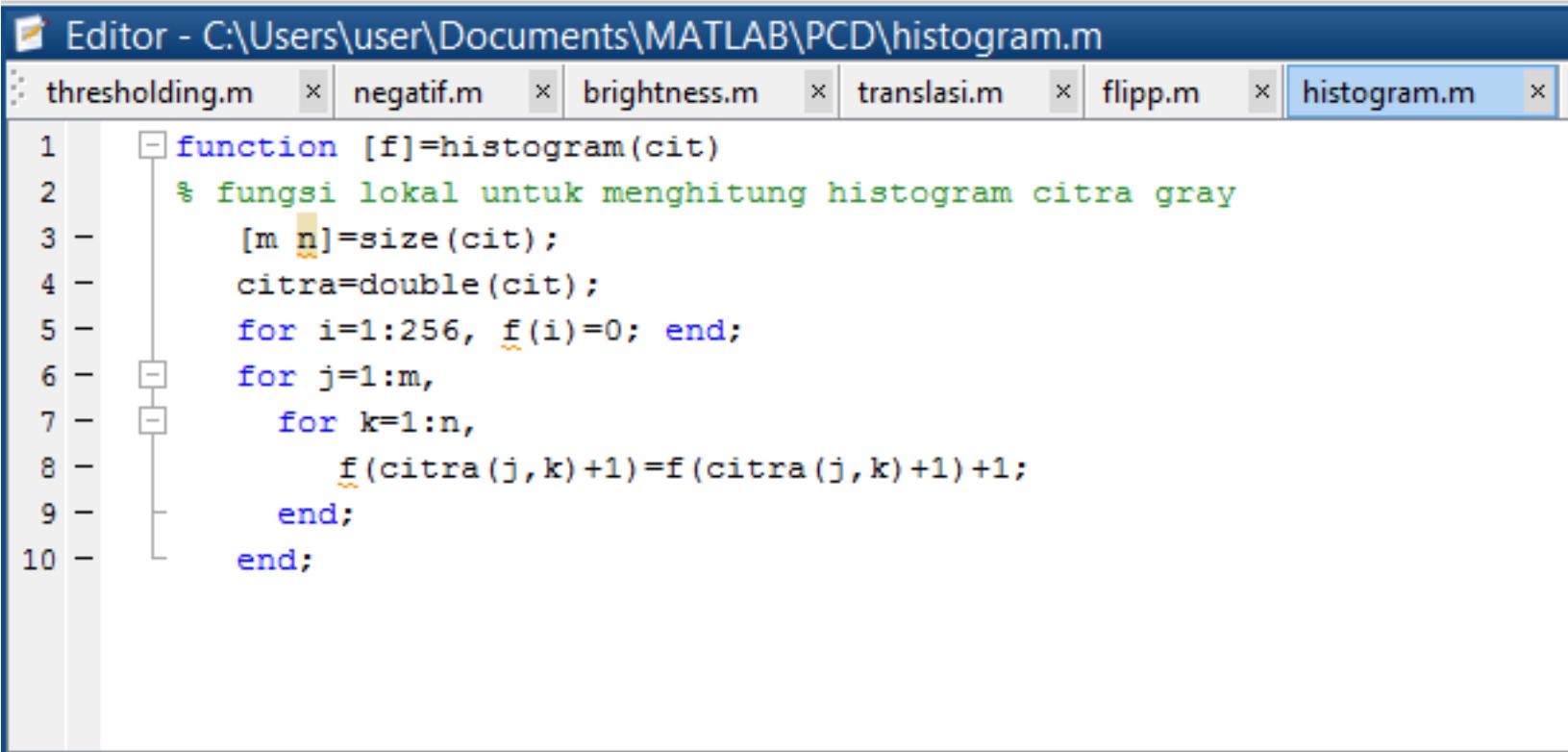
$$\begin{bmatrix} 1 & 7 & 0 & 5 \\ 6 & 6 & 0 & 2 \\ 0 & 5 & 6 & 5 \\ 3 & 2 & 4 & 3 \end{bmatrix}$$


MEMBANGUN HISTOGRAM-6

Tahapan :

- Baca citra
- Hitung ukuran citra (M baris dan N kolom)
- Untuk i dari 1 sampai M
 - Untuk j dari 1 sampai N
 - Hitung kemunculan tiap-tiap derajat keabuan (Turus)

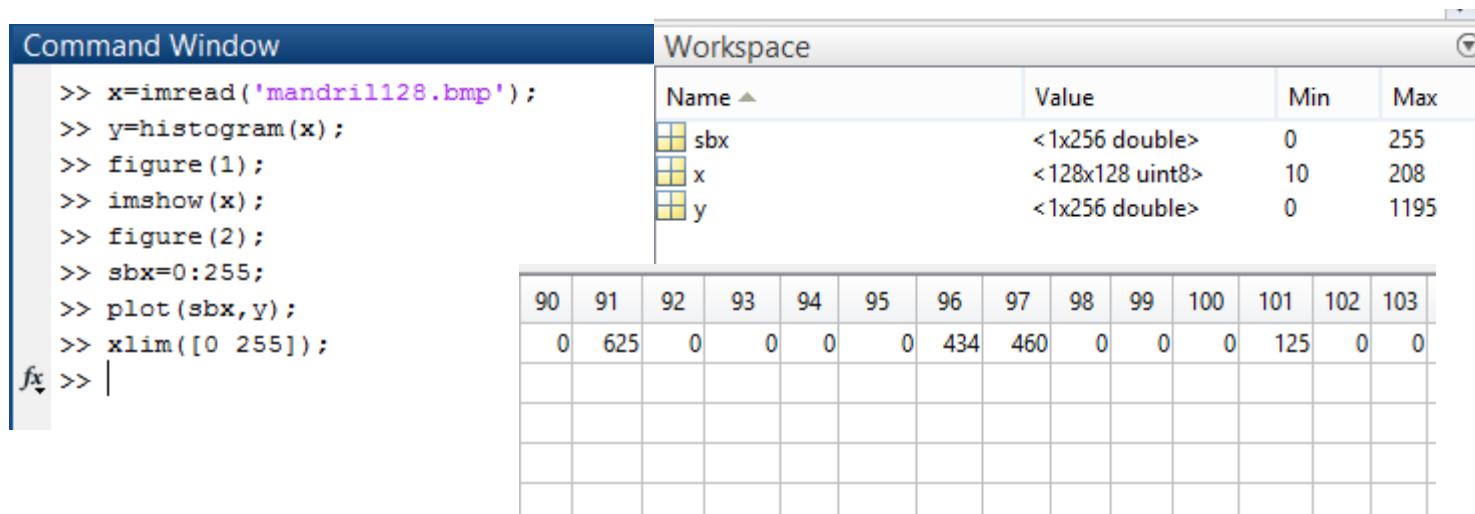
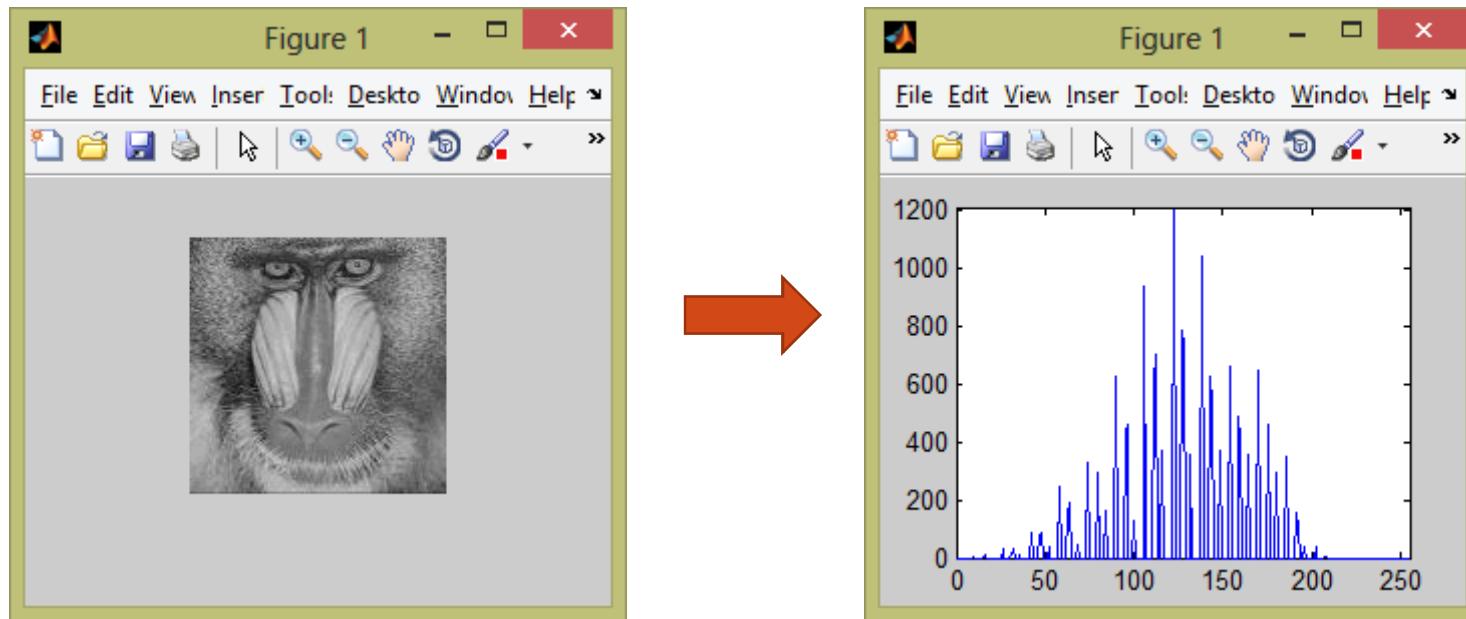
MEMBANGUN HISTOGRAM-7



The screenshot shows the MATLAB Editor window with the title bar "Editor - C:\Users\user\Documents\MATLAB\PCD\histogram.m". The current tab is "histogram.m", which is highlighted in blue. The editor displays the following MATLAB code:

```
1 function [f]=histogram(cit)
2 % fungsi lokal untuk menghitung histogram citra gray
3 [m n]=size(cit);
4 citra=double(cit);
5 for i=1:256, f(i)=0; end;
6 for j=1:m,
7     for k=1:n,
8         f(citra(j,k)+1)=f(citra(j,k)+1)+1;
9     end;
10 end;
```

MEMBANGUN HISTOGRAM-8



KONVOLUSI-1

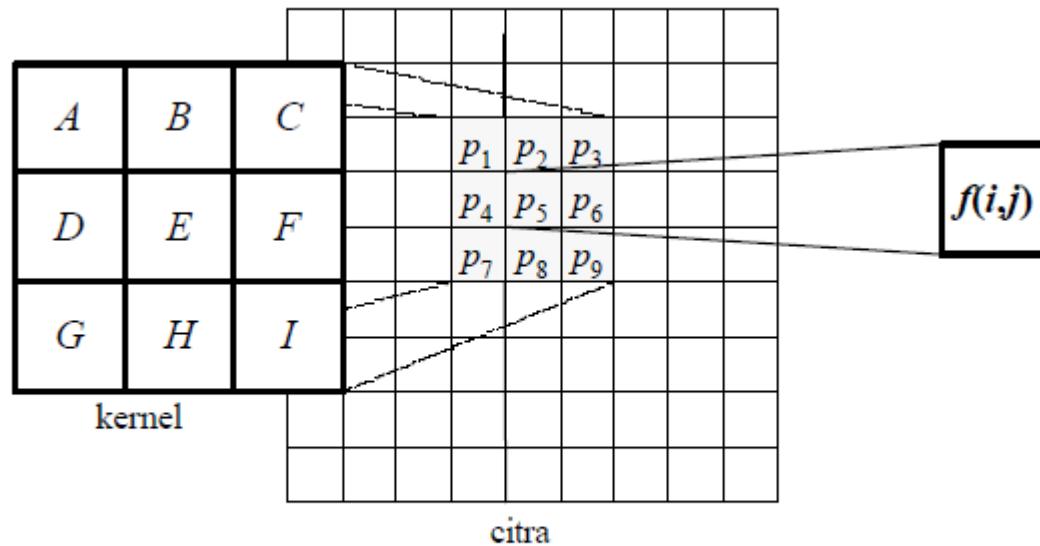
- Konvolusi merupakan operasi pengolahan citra yang mengalikan sebuah citra dengan sebuah *mask atau kernel*.
- Fungsi Konvolusi kontinu dan diskrit adalah :

$$h(x, y) = f(x, y) * g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b)g(x-a, y-b) da db$$

$$h(x, y) = f(x, y) * g(x, y) = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f(a, b)g(x-a, y-b)$$

- Dimana $f(x)$ adalah citra dan $g(x)$ merupakan kernel atau filter.

KONVOLUSI-2



$$f(i,j) = A p_1 + B p_2 + C p_3 + D p_4 + E p_5 + F p_6 + G p_7 + H p_8 + I p_9$$

- Kernel dapat berupa matriks berukuran 2x2, 3x3, 5x5 maupun yang lain dan umumnya ukuran kernel lebih kecil dibandingkan dengan ukuran citranya.

KONVOLUIS-3

Citra (C)

Kernel (K)

K(1,1)	K(1,2)	K(1,3)
K(2,1)	K(2,2)	K(2,3)
K(3,1)	K(3,2)	K(3,3)

(1,1)	(1,2)	(1,3)	(1,N)
(2,1)	(2,2)	(2,3)	(2,N)
(3,1)	(3,2)	(3,3)	(3,N)
:	:	:	:	:	:
:	:	:	:	:	:
(M,1)	(M,2)	(M,3)	(M,N)

for i = 2 : M-1,

for j = 2 : N-1,

$$f(i,j) = C(i-1,j-1)*K(1,1) + C(i-1,j)*K(1,2) + C(i-1,j+1)*K(1,3) + \\ C(i,j-1)*K(2,1) + C(i,j)*K(2,2) + C(i,j+1)*K(2,3) + \\ C(i+1,j-1)*K(3,1) + C(i+1,j)*K(3,2) + C(i+1,j+1)*K(3,3)$$

end

end

KONVOLUIS-4

$$f(x, y) = \begin{bmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 6 & 5 & 5 & 2 \\ 5 & 6 & 6 & 6 & 2 \\ 6 & 7 & 5 & 5 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{bmatrix} \quad g(x, y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & \bullet 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

4	4	3	5	4
6	6	5	5	2
5	6	6	6	2
6	7	5	5	3
3	5	2	4	4

———

		3		

$$(0 \times 4) + (-1 \times 4) + (0 \times 3) + (-1 \times 6) + (4 \times 6) + (-1 \times 5) + (0 \times 5) + (-1 \times 6) + (0 \times 6) = 3$$

KONVOLUIS-5

4	4	3	5	4
6	6	5	5	2
5	6	6	6	2
6	7	5	5	3
3	5	2	4	4

	3	0		

4	4	3	5	4
6	6	5	5	2
5	6	6	6	2
6	7	5	5	3
3	5	2	4	4

	3	0	2	

KONVOLUIS-6

4	4	3	5	4
6	6	5	5	2
5	6	6	6	2
6	7	5	5	3
3	5	2	4	4

3	0	2		
0				

4	4	3	5	4
6	6	5	5	2
5	6	6	6	2
6	7	5	5	3
3	5	2	4	4

4	0	8		
0	2			

KONVOLUSI-7

4	4	3	5	4
6	6	5	5	2
5	6	6	6	2
6	7	5	5	3
3	5	2	4	4

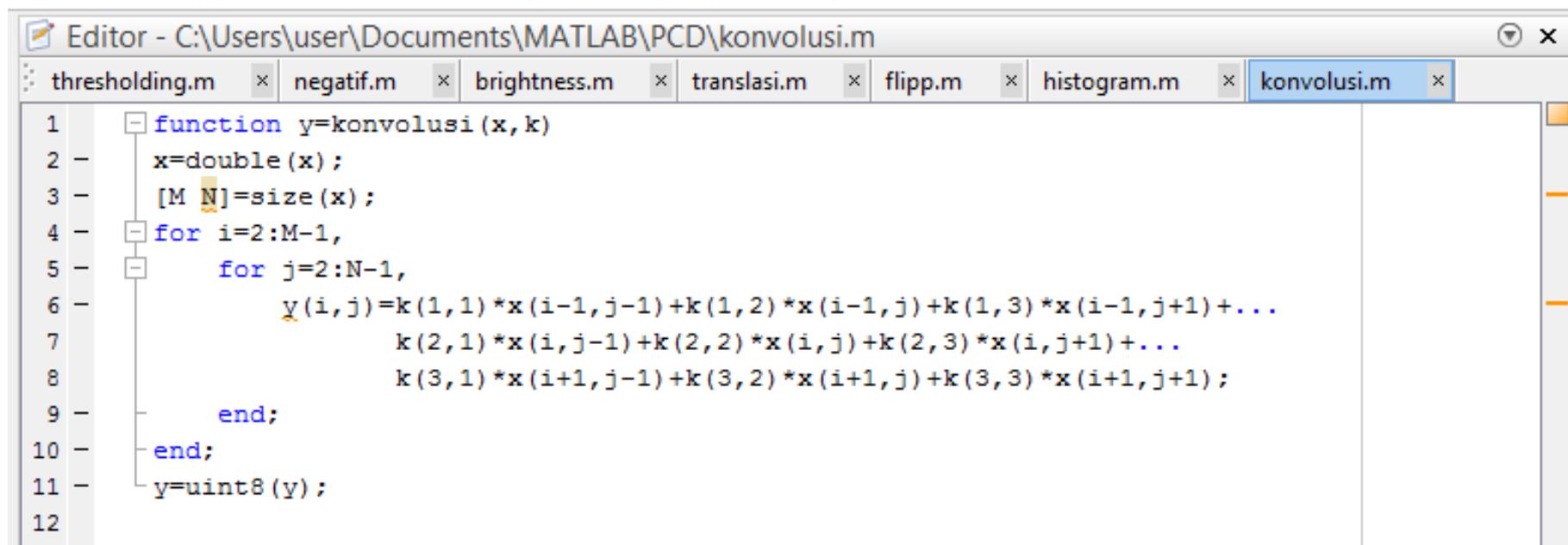
—————

4	0	8		
0	2	6		

Dan seterusnya, dengan cara yang sama akhirnya diperoleh hasil sebagai berikut :

4	0	8		
0	2	6		
6	0	2		

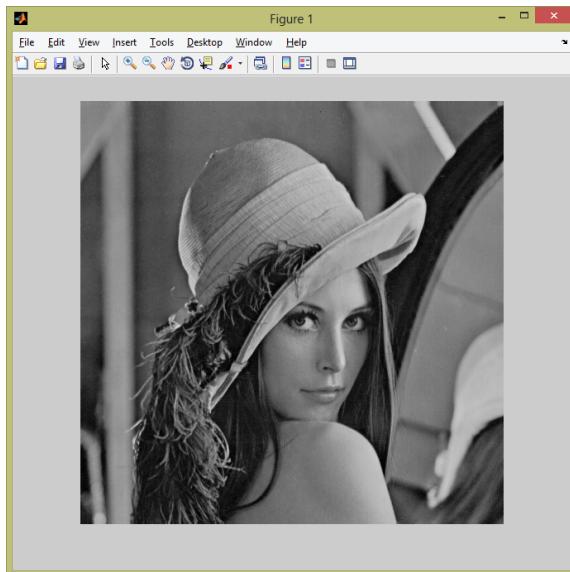
KONVOLUSI-8



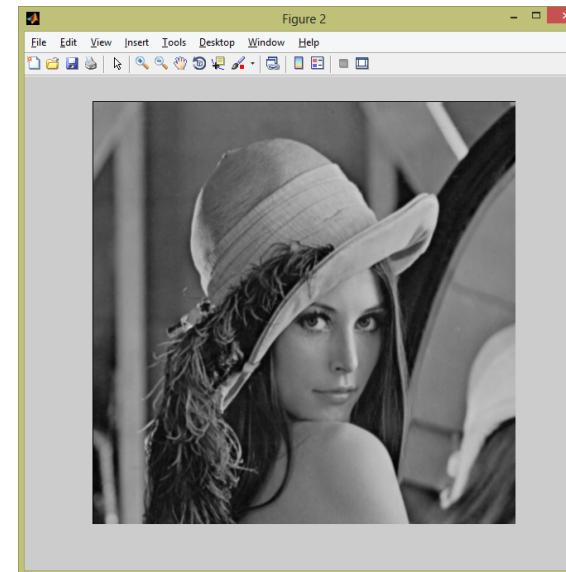
The screenshot shows the MATLAB Editor window with the title "Editor - C:\Users\user\Documents\MATLAB\PCD\konvolusi.m". The tab bar also lists other files: thresholding.m, negatif.m, brightness.m, translasi.m, flipp.m, histogram.m, and konvolusi.m (which is currently selected). The code in the editor is as follows:

```
1 function y=konvolusi(x,k)
2 - x=double(x);
3 - [M N]=size(x);
4 - for i=2:M-1,
5 -     for j=2:N-1,
6 -         y(i,j)=k(1,1)*x(i-1,j-1)+k(1,2)*x(i-1,j)+k(1,3)*x(i-1,j+1)+...
7 -                 k(2,1)*x(i,j-1)+k(2,2)*x(i,j)+k(2,3)*x(i,j+1)+...
8 -                 k(3,1)*x(i+1,j-1)+k(3,2)*x(i+1,j)+k(3,3)*x(i+1,j+1);
9 -     end;
10 -    end;
11 -    y=uint8(y);
12
```

KONVOLUSI-9




$$\begin{bmatrix} 0.0625 & 0.1250 & 0.0625 \\ 0.1250 & 0.2500 & 0.1250 \\ 0.0625 & 0.1250 & 0.0625 \end{bmatrix} =$$



Command Window

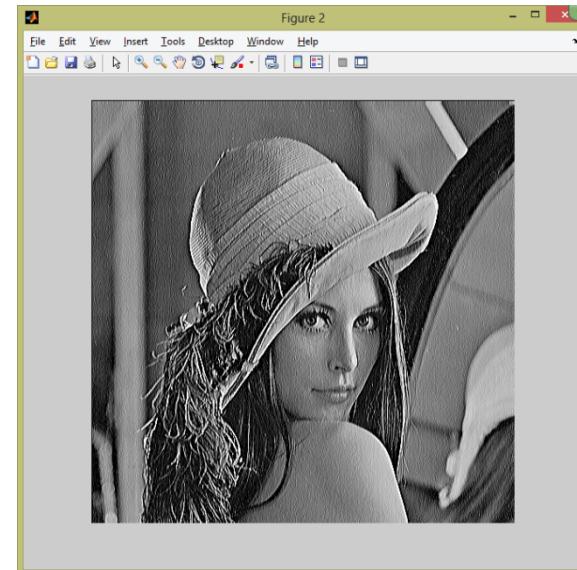
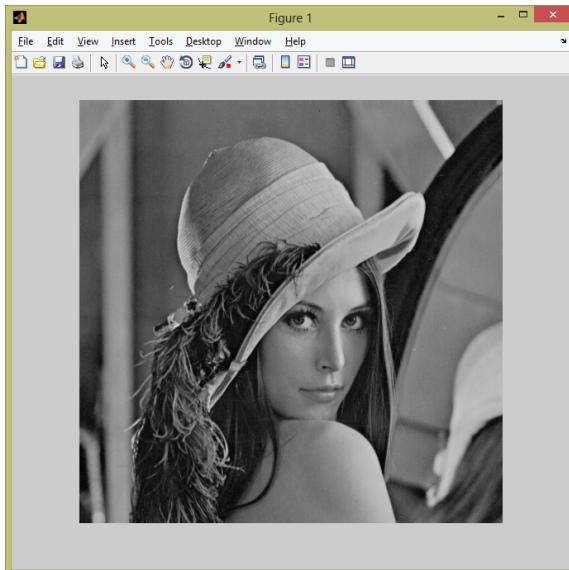
```
>> x=imread('lena.bmp');  
>> k=[1/16 1/8 1/16;1/8 1/4 1/8;1/16 1/8 1/16]
```

k =

```
0.0625 0.1250 0.0625  
0.1250 0.2500 0.1250  
0.0625 0.1250 0.0625
```

```
>> y=konvolusi(x,k);  
>> figure(1);  
>> imshow(x);  
>> figure(2);  
>> imshow(y);  
fx >> |
```

KONVOLUSI-10



✗ $G = \begin{bmatrix} -1 & 2 & -1 \\ -2 & 5 & -2 \\ -1 & 2 & -1 \end{bmatrix} =$

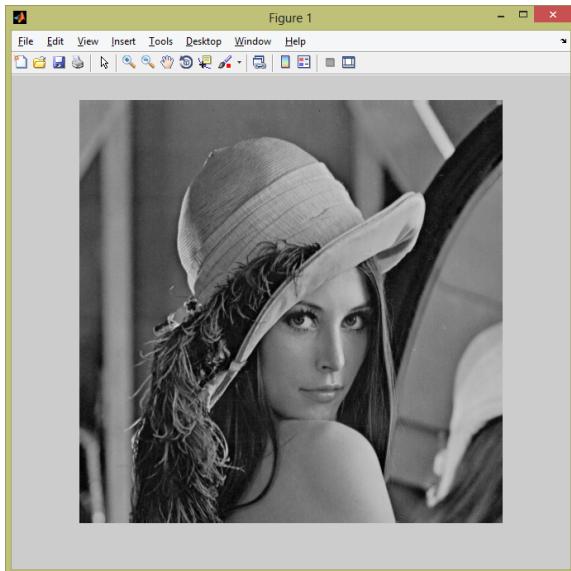
Command Window

```
>> x=imread('lena.bmp');
>> g=[-1 2 -1;-2 5 -2;-1 2 -1]

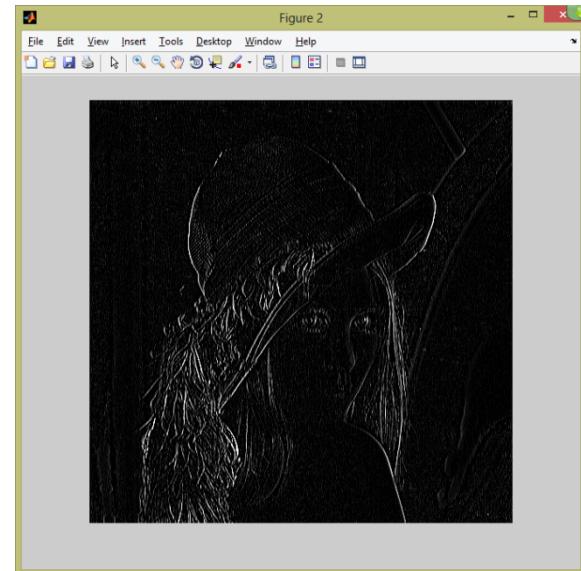
g =
    -1     2     -1
    -2     5     -2
    -1     2     -1

>> y=konvolusi(x,g);
>> figure(1);
>> imshow(x);
>> figure(2);
>> imshow(y);
fx >> |
```

KONVOLUSI-11



✗ $G = \begin{bmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{bmatrix} =$



Command Window

```
>> x=imread('lena.bmp');
>> g=[-1 2 -1;-2 4 -2;-1 2 -1]

g =

-1      2      -1
-2      4      -2
-1      2      -1

>> y=konvolusi(x,g);
>> figure(1);
>> imshow(x);
>> figure(2);
>> imshow(y);

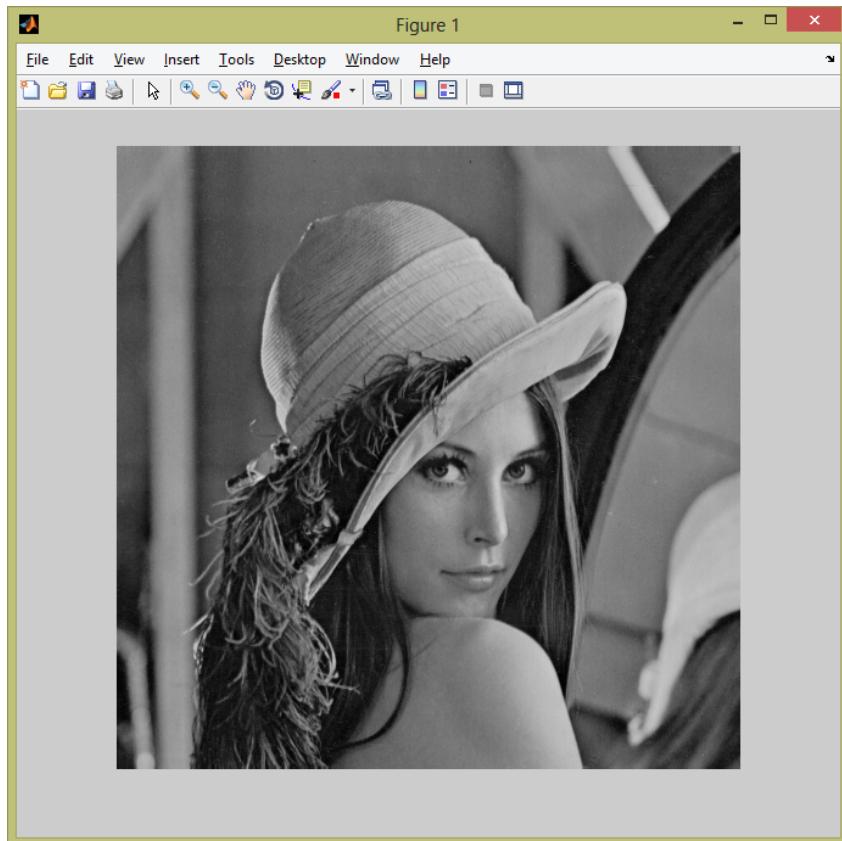
fx >>
```

$G - \begin{vmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{vmatrix}$?

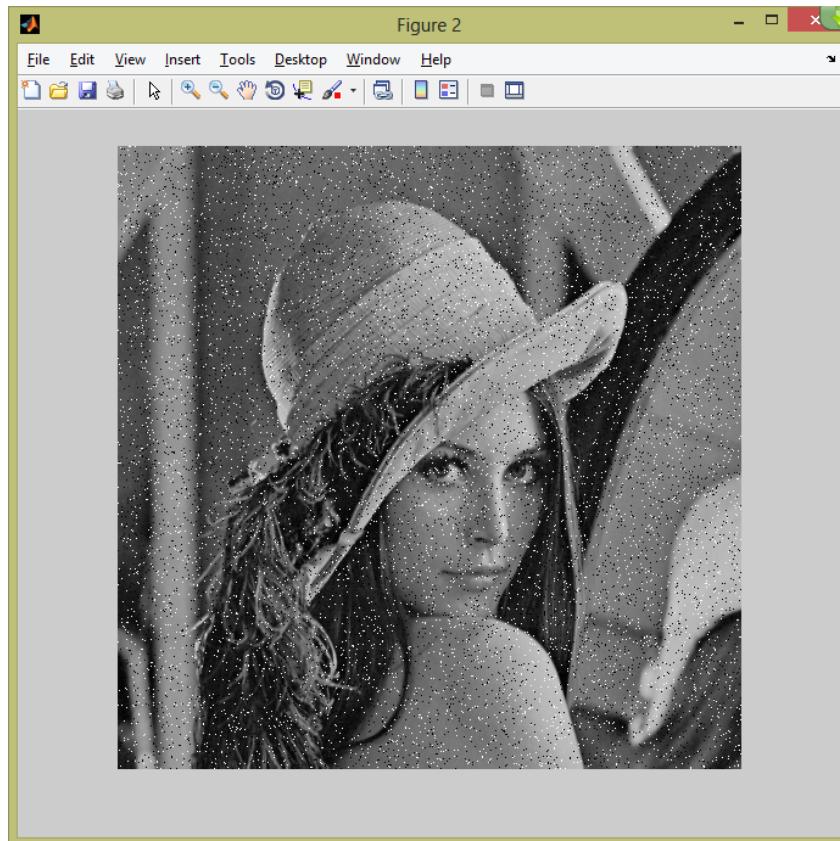
PELEMBUTAN (SMOOTHING)-1

- Pelembutan citra (*image smoothing*) bertujuan untuk menekan gangguan (*noise*) pada citra.
- Gangguan tersebut biasanya muncul sebagai akibat dari hasil pengambilan gambar yang tidak bagus (*sensor noise, photographic grain noise*) atau akibat saluran transmisi (pada pengiriman data).
- Pada domain spasial, operasi pelembutan dilakukan dengan mengganti intensitas suatu *pixel* dengan rata-rata dari nilai *pixel* tersebut dengan nilai *pixel/pixel* tetangganya.

PELEMBUTAN (SMOOTHING)-2

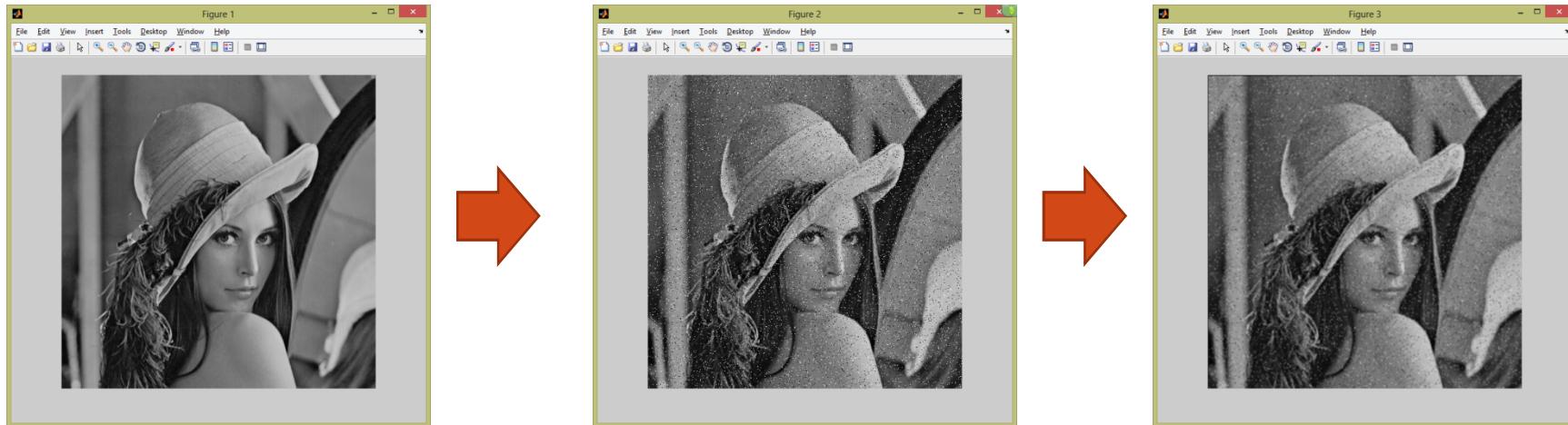


Lena-Normal



Lena-Noisy-'salt & pepper'

PELEMBUTAN (SMOOTHING)-3



Command Window

```
>> x=imread('lena.bmp');
>> x_noise=imnoise(x,'salt & pepper');
>> g=[1/9 1/9 1/9;1/9 1/9 1/9;1/9 1/9 1/9]

g =

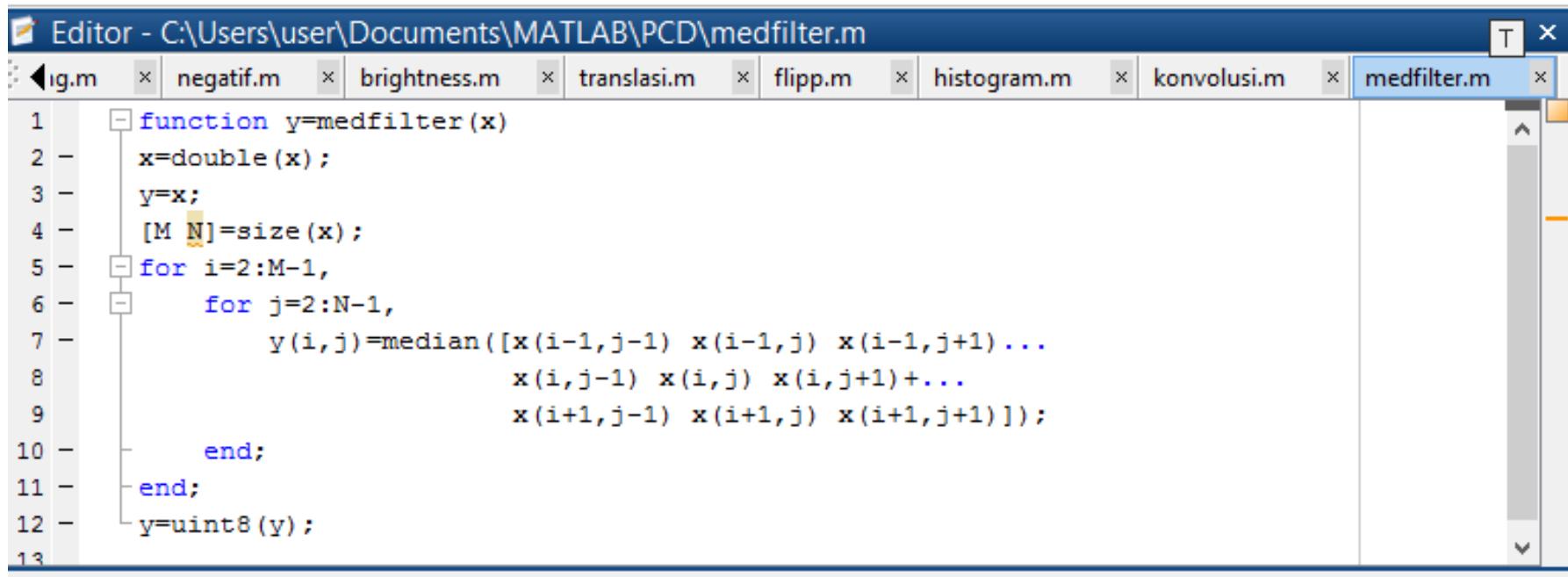
    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111
    0.1111    0.1111    0.1111

>> y=konvolusi(x_noise,g);
>> figure(1);
>> imshow(x);
>> figure(2);
>> imshow(x_noise);
>> figure(3);
>> imshow(y);
fx >>
```

Kernel rata-rata

PELEMBUTAN (SMOOTHING)-4

- Selain menggunakan teknik averaging pelembutan dapat juga dilakukan dengan menggunakan teknik median filtering.
- Teknik ini dilakukan dengan mengganti nilai pusat sliding window dengan menggunakan nilai median dari piksel-piksel tetanggannya.



The screenshot shows the MATLAB Editor window with the title bar "Editor - C:\Users\user\Documents\MATLAB\PCD\medfilter.m". The tab bar below the title bar lists several other files: ig.m, negatif.m, brightness.m, translasi.m, flipp.m, histogram.m, konvolusi.m, and medfilter.m. The "medfilter.m" tab is currently selected. The main workspace displays the MATLAB code for the "medfilter" function:

```
1 function y=medfilter(x)
2 x=double(x);
3 y=x;
4 [M N]=size(x);
5 for i=2:M-1,
6 for j=2:N-1,
7 y(i,j)=median([x(i-1,j-1) x(i-1,j) x(i-1,j+1) ...
8 x(i,j-1) x(i,j) x(i,j+1)+...
9 x(i+1,j-1) x(i+1,j) x(i+1,j+1)]);
10 end;
11 end;
12 y=uint8(y);
```

PELEMBUTAN (SMOOTHING)-5



Averaging



Median Filtering



Lena-Noisy-'salt & pepper'

PENAJAMAN GAMBAR-1

- Operasi penajaman citra bertujuan memperjelas tepi pada objek di dalam citra.
- Penajaman citra merupakan kebalikan dari operasi pelembutan citra karena operasi ini menghilangkan bagian citra yang lembut.
- Operasi penajaman dilakukan dengan melewatkannya citra pada **filter lolos-tinggi** (*high-pass filter*).

PENAJAMAN GAMBAR-2

- *Penapis lolos-tinggi akan meloloskan (atau memperkuat) komponen yang berfrekuensi tinggi (misalnya tepi atau pinggiran objek) dan akan menurunkan komponen berfrekuensi rendah. Akibatnya, pinggiran objek telihat lebih tajam dibandingkan sekitarnya.*
- Karena penajaman citra lebih berpengaruh pada tepi (*edge*) objek, maka penajaman citra sering disebut juga **penajaman tepi (*edge sharpening*) atau** peningkatan kualitas tepi (*edge enhancement*).

PENAJAMAN GAMBAR-3

Kriteria filter lolos tinggi :

1. koefisien filter boleh positif, negatif, atau nol
2. jumlah semua koefisien adalah 0 atau 1
 - Jika jumlah koefisien = 0, maka komponen berfrekuensi rendah akan turun nilainya, sedangkan jika jumlah koefisien sama dengan 1, maka komponen berfrekuensi rendah akan tetap sama dengan nilai semula.

PENAJAMAN GAMBAR-4

- Filter lolos tinggi

$$(i) \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$
$$\Sigma = 0$$

$$(ii) \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$
$$\Sigma = 1$$

$$(iii) \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$
$$\Sigma = 1$$

$$(iv) \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$
$$\Sigma = 1$$

$$(v) \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$
$$\Sigma = 0$$

$$(vi) \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$
$$\Sigma = 0$$

PENAJAMAN GAMBAR-5



X
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$



```
Command Window
>> x=imread('mandril128.bmp');
>> K=[-1 -1 -1;-1 9 -1;-1 -1 -1]

K =
-1     -1     -1
-1      9     -1
-1     -1     -1

>> y=konvolusi(x,K);
>> figure(1)
>> imshow(x);
>> figure(2)
>> imshow(y);

fx >>
```

PENAJAMAN GAMBAR-6



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



```
Command Window
>> x=imread('mandrill128.bmp');
>> K2=[-1 -1 -1;-1 8 -1;-1 -1 -1]

K2 =
-1     -1     -1
-1      8     -1
-1     -1     -1

>> y=konvolusi(x,K2);
>> figure(1)
>> imshow(x);
>> figure(2)
>> imshow(y);
fx >> |
```

PEREGANGAN KONTRAS-1

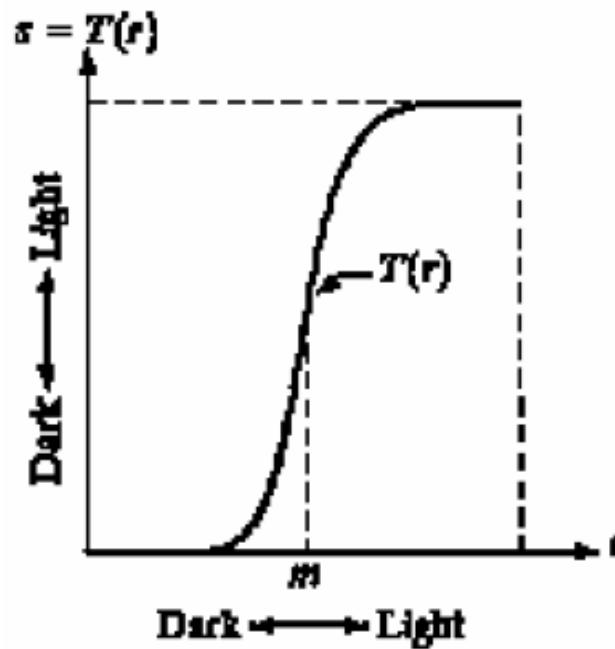
- Kontras menyatakan sebaran terang (*lightness*) dan gelap (*darkness*) *di dalam* sebuah gambar.
- Citra dapat dikelompokkan ke dalam tiga kategori kontras: citra kontras-rendah (*low contrast*), *citra kontras-bagus (good contrast atau normal contrast)*, dan *citra kontras-tinggi (high contrast)*.

PEREGANGAN KONTRAS-2

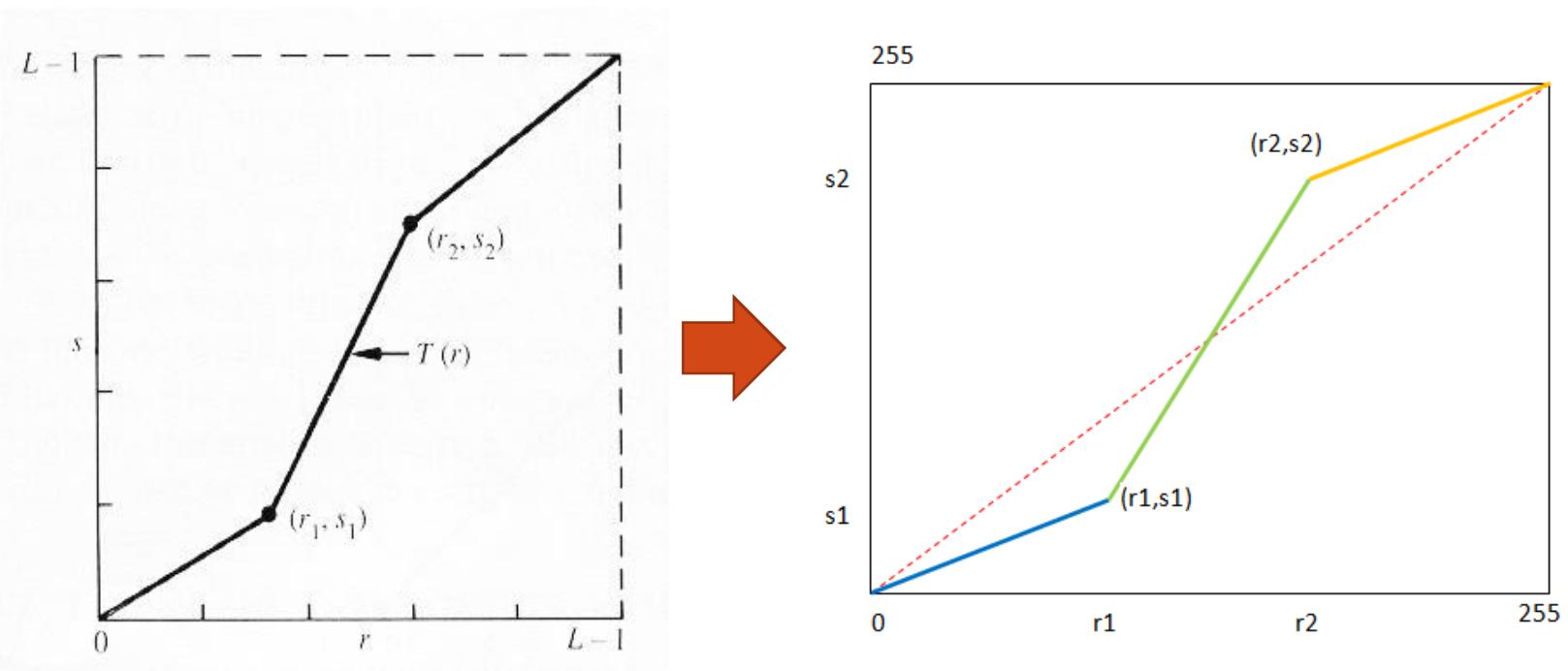
- Citra kontras-rendah dicirikan dengan sebagian besar komposisi citranya adalah terang atau sebagian besar gelap.
- Citra kontras-bagus memperlihatkan jangkauan nilai keabuan yang lebar tanpa ada suatu nilai keabuan yang mendominasi. Histogram citranya memperlihatkan sebaran nilai keabuan yang relatif seragam.
- Citra kontras-tinggi, seperti halnya citra kontras bagus, memiliki jangkauan nilai keabuan yang lebar, tetapi terdapat area yang lebar yang didominasi oleh warna gelap dan area yang lebar yang didominasi oleh warna terang.

PEREGANGAN KONTRAS-3

- Menghasilkan nilai contrast yang lebih besar dari image original, dengan cara :
 - Menggelapkan (darkening) level dibawah m dari image asli.
 - Mencerahkan (Brightening) level atas m dari image asli.

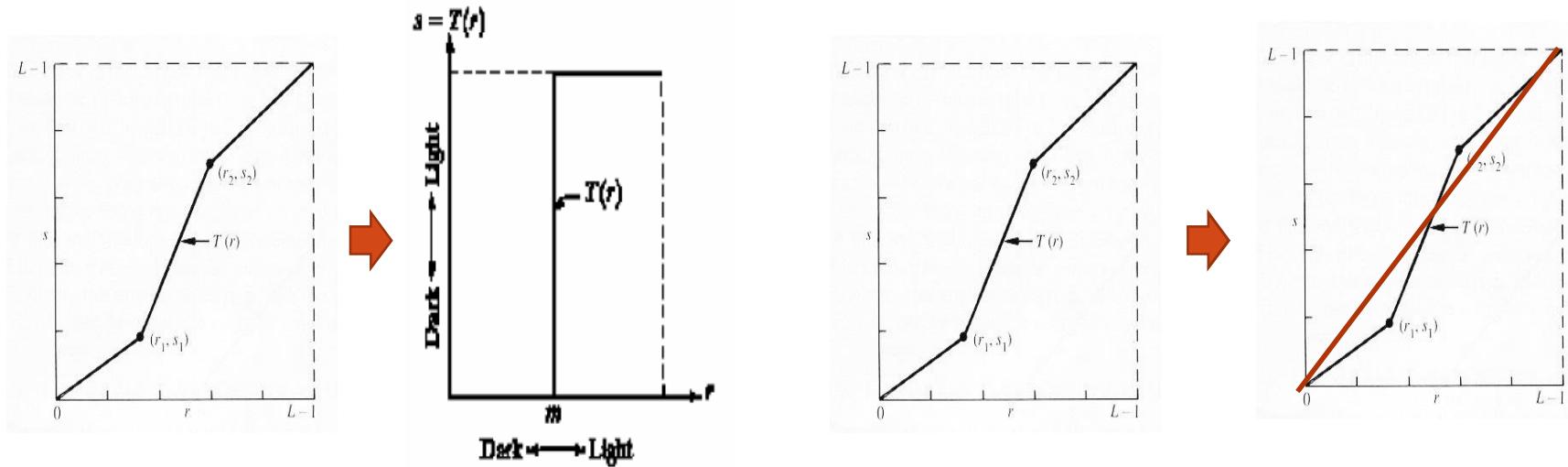


PEREGANGAN KONTRAS-4



Lokasi (r_1, s_1) dan (r_2, s_2) menjadi kontrol dari bentuk fungsi transformasi.

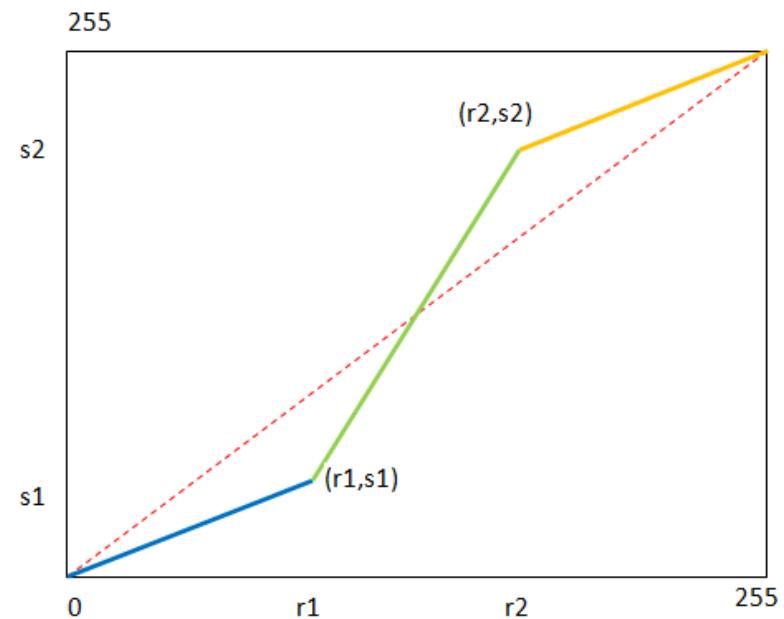
PEREGANGAN KONTRAS-5



- Lokasi dari (r_1, s_1) dan (r_2, s_2) menjadi kontrol dari bentuk fungsi transformasi.
 - Bila $r_1 = s_1$ dan $r_2 = s_2$ transformasi adalah fungsi linear dan hasilnya adalah tidak ada perubahan image.
 - Bila $r_1 = r_2$, $s_1 = 0$ dan $s_2 = L-1$, transformasi akan berubah menjadi sebuah fungsi thresholding yang menghasilkan sebuah binary image.

PEREGANGAN KONTRAS-6

- Untuk menyelesaikan peregangan kontras berdasarkan grafik disamping, maka perlu dicari 3 persamaan garis.
 - Garis biru, melalui titik $(0,0)$ dan (r_1, s_1)
 - Garis hijau, melalui titik (r_1, s_1) dan (r_2, s_2)
 - Garis kuning, melalui titik (r_2, s_2) dan $(255, 255)$



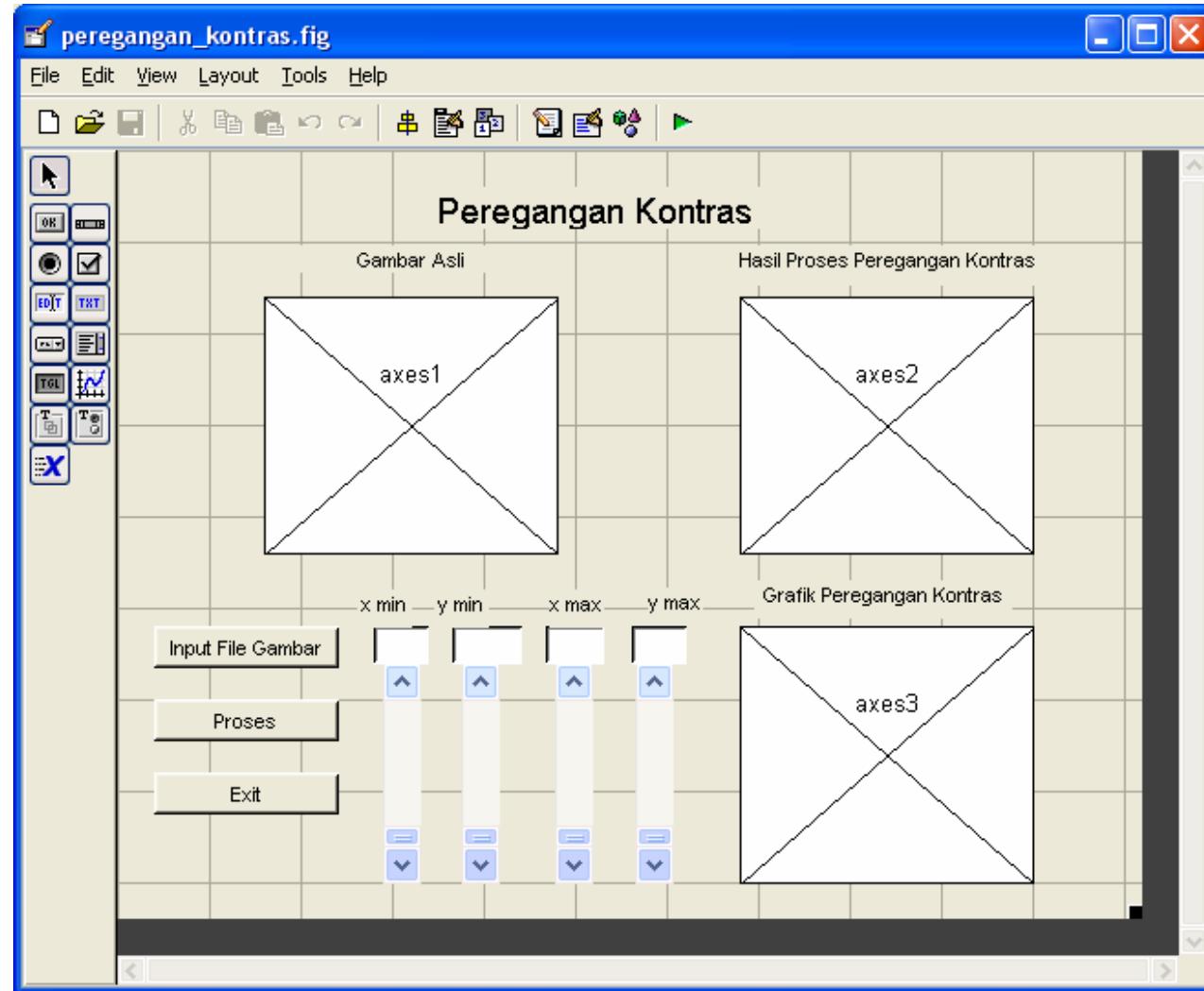
Solusinya hanya persamaan garis !
$$(y-y_1)/(y_2-y_1)=(x-x_1)/(x_2-x_1)$$

PEREGANGAN KONTRAS-7

Algorithma :

- Input citra
- Tentukan ukuran citra
- Untuk tiap-tiap piksel citra
 - Jika piksel citra $< r1$ maka gunakan persamaan garis 1 (biru)
 - Jika piksel citra $> r2$ maka gunakan persamaan garis 3 (kuning)
 - Selain itu gunakan persamaan garis 2 (hijau)

PEREGANGAN KONTRAS-8



PEREGANGAN KONTRAS-9

