

10 - Image Enhancement

(Bagian 3)

IF4073 Interpretasi dan Pengolahan Citra

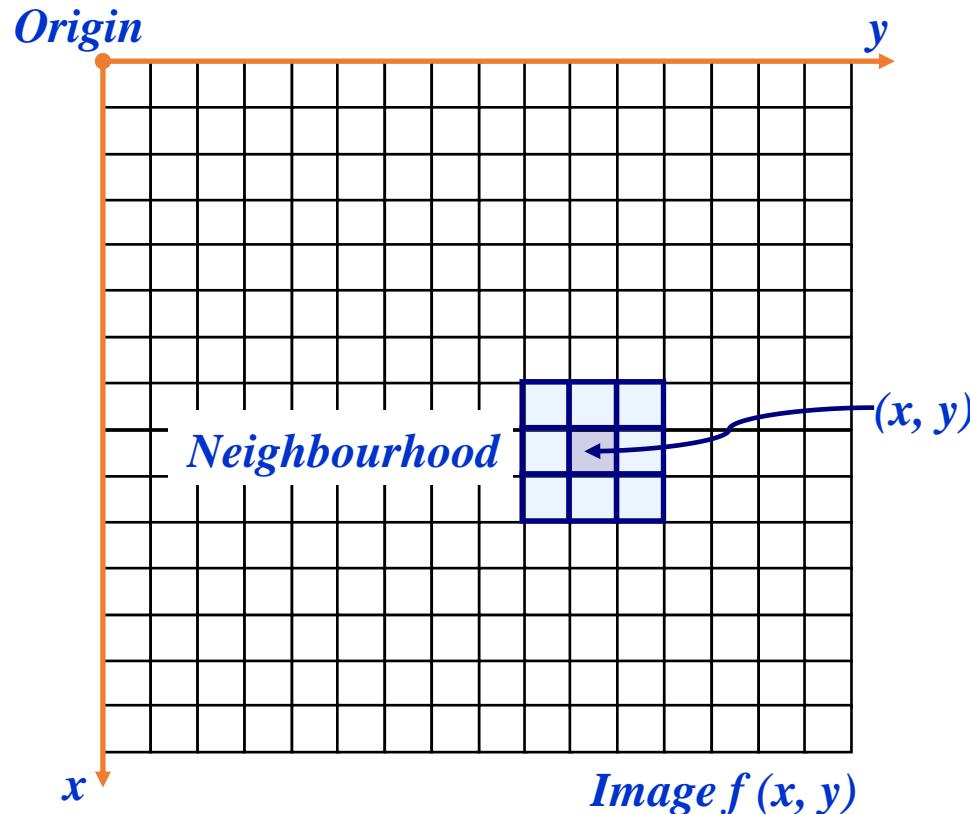
Oleh: Rinaldi Munir



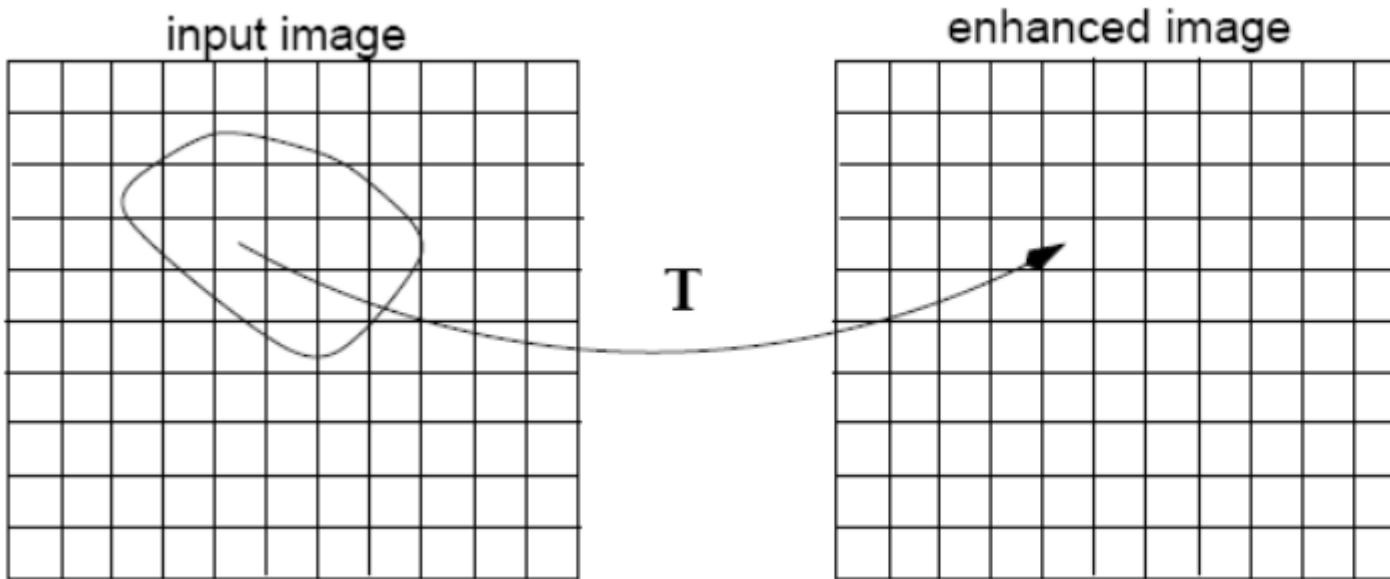
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2022

Perbaikan Citra dengan cara penapisan (*image filtering*)

- Penapisan citra berarti memodifikasi *pixel-pixel* di dalam citra berdasarkan transformasi terhadap nilai-nilai *pixel* tetangganya.



Sumber gambar: Ali Javed, Digital Image Processing,
Chapter # 3, *Image Enhancement in Spatial Domain*



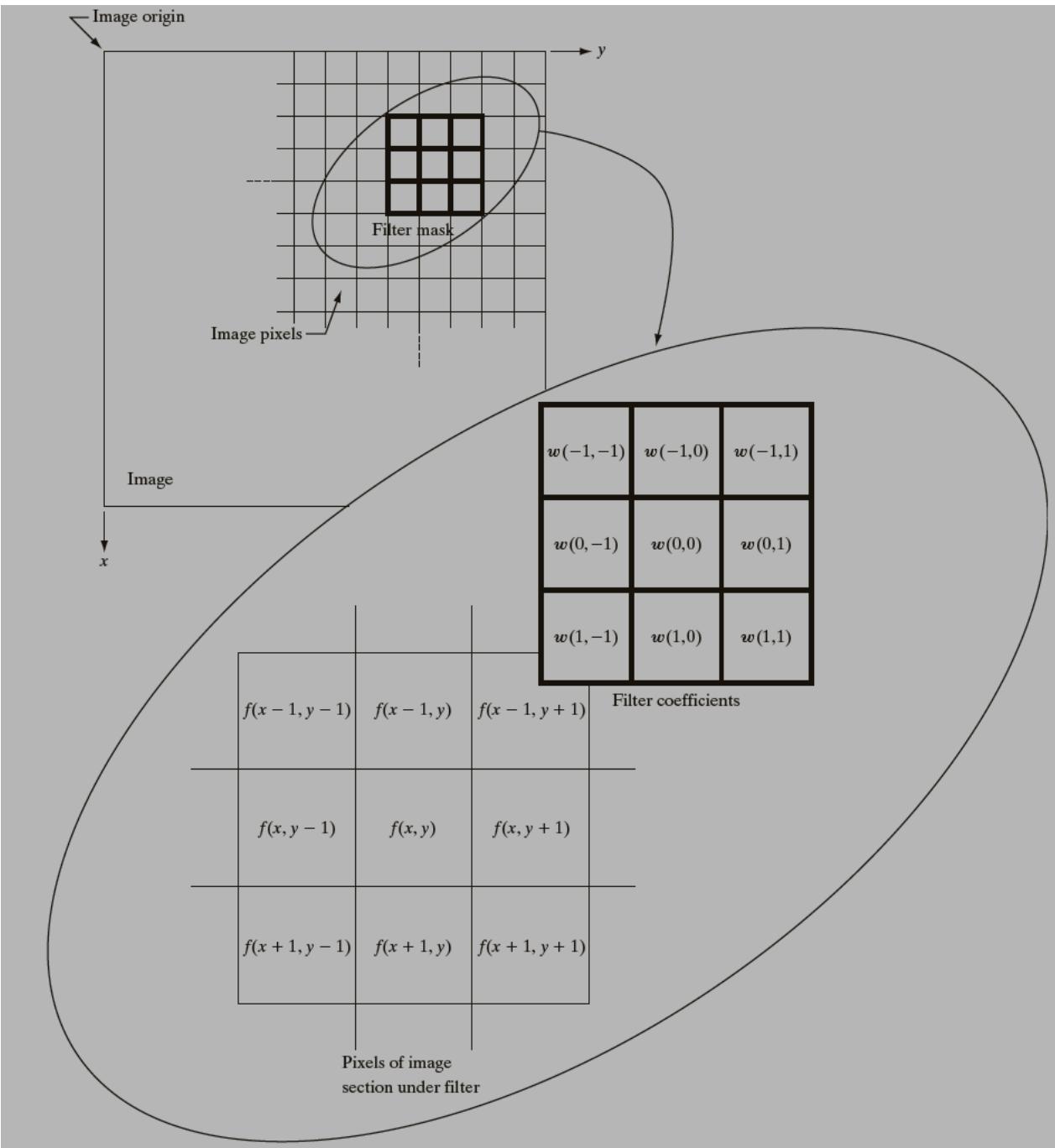
$$g(x,y) = T[f(x,y)]$$

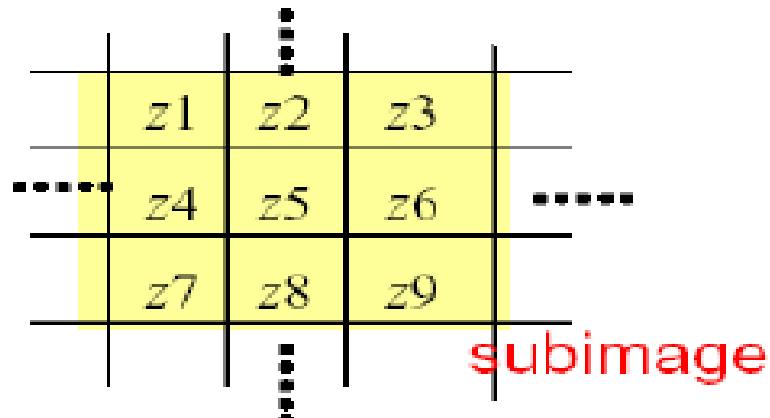
T operates on a neighborhood of pixels

- Penapisan citra dilakukan dengan operator **konvolusi**.
- Citra masukan dikonvolusi dengan sebuah **mask** (nama lainnya **filter**, **template**, **window**, atau **kernel**).

w1	w2	w3
w4	w5	w6
w7	w8	w9

mask





$$z_5 \leftarrow z = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{i=1}^9 w_i z_i$$

w1	w2	w3
w4	w5	w6
w7	w8	w9

Mask coefficients

1. Pelembutan Citra (*image smoothing*)

- Pelembutan citra (*image smoothing*) bertujuan untuk menghasilkan efek *blur* dan untuk mengurangi derau (*noise*) pada citra.



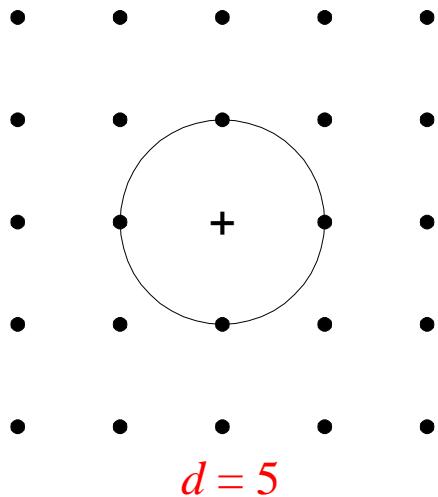
Derau pada citra umumnya berupa variasi intensitas suatu *pixel* yang tidak berkorelasi dengan *pixel-pixel* tetangganya

Pixel yang mengalami gangguan umumnya memiliki frekuensi tinggi.

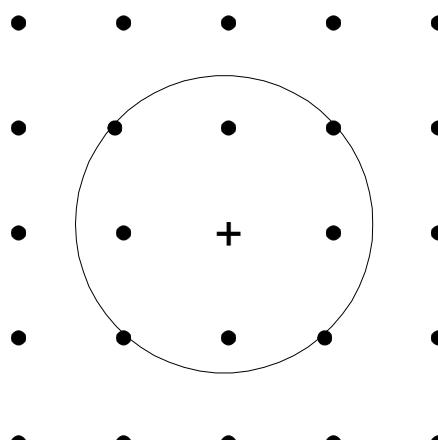
Komponen citra berfrekuensi rendah umumnya mempunyai *pixel* bernilai konstan atau berubah sangat lambat.

Pelembutan citra dilakukan untuk menekan komponen yang berfrekuensi tinggi dan meloloskan komponen yang berfrekuensi rendah.

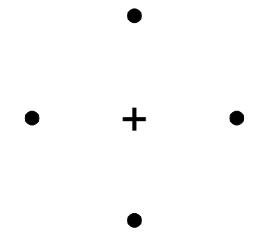
- Cara pelembutan citra: nilai sebuah *pixel* diganti dengan nilai rata-rata pixel tersebut beserta *pixel-pixel* tetangganya.



Tetangga *pixel* +

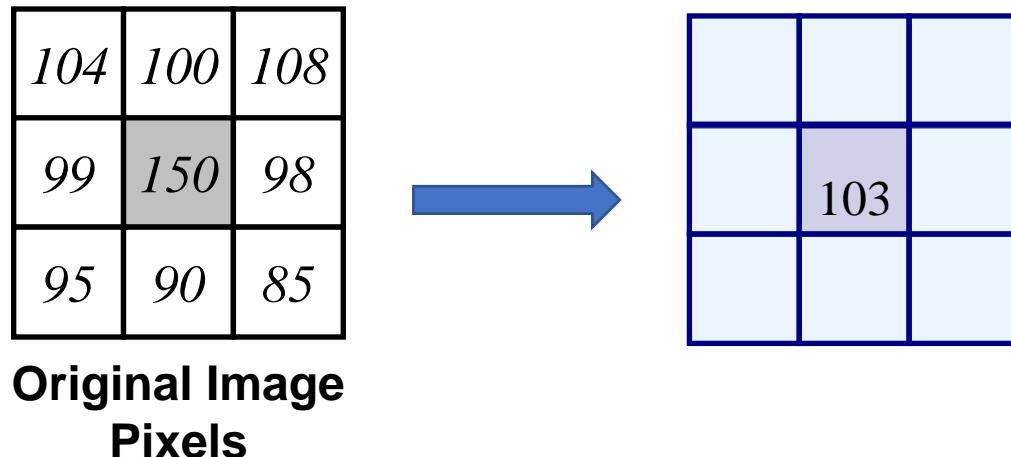


Tetangga *pixel* +



$$g(x, y) = \frac{1}{d} \sum_{r=m_1}^{m_2} \sum_{s=n_1}^{n_2} f(x+r, y+s)$$

d adalah jumlah *pixel* yang terlibat dalam perhitungan rata-rata



$$\begin{aligned}
 f(x,y) &= (104 + 100 + 108 + 99 + 150 + 98 + 95 + 90 + 85)/9 \\
 &= 103,22 \approx 103
 \end{aligned}$$

Nilai 150 direduksi menjadi 103

- Operasi perata-rataan di atas dapat dipandang sebagai konvolusi antara citra $f(x,y)$ dengan penapis $h(x,y)$:

$$g(x,y) = f(x,y) * h(x,y)$$

Penapis h disebut **penapis rerata** (*mean filter* atau *averaging filter*).

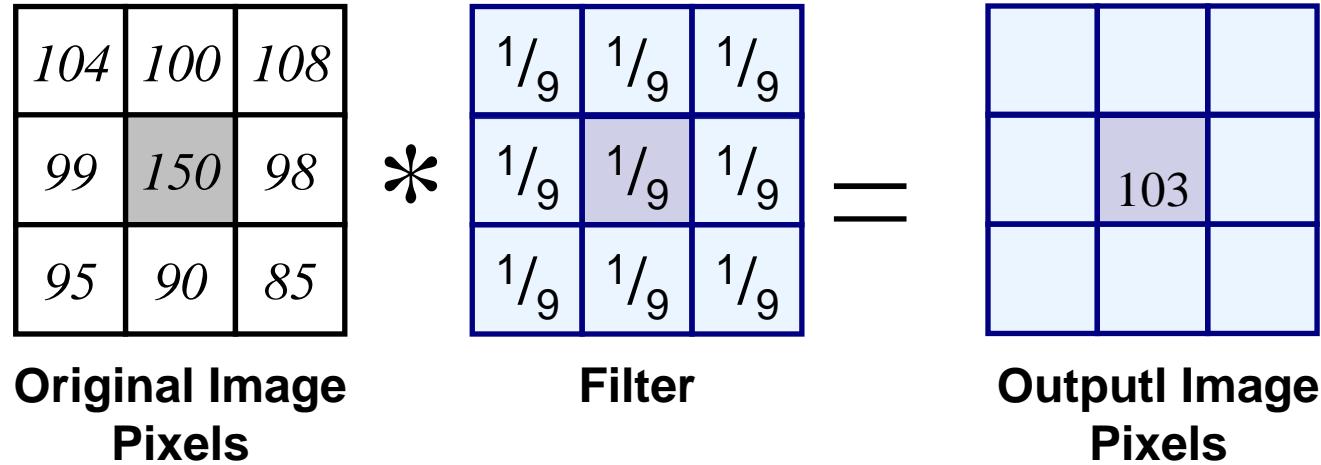
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

atau

1	1	1
1	1	1
1	1	1

$\frac{1}{9} \times$

$$f(x,y) * h(x,y) = g(x,y)$$



$$\begin{aligned}
 f(x,y) &= \frac{1}{9} * 104 + \frac{1}{9} * 100 + \frac{1}{9} * 108 + \\
 &\quad \frac{1}{9} * 99 + \frac{1}{9} * 150 + \frac{1}{9} * 98 + \\
 &\quad \frac{1}{9} * 95 + \frac{1}{9} * 90 + \frac{1}{9} * 85 \\
 &= 103.22
 \end{aligned}$$

```

I = imread('peppers512.bmp');
Inoise = imnoise(I, 'salt & pepper', 0.1);
imshow(I);
figure; imshow(Inoise);
G = [1/9 1/9 1/9; 1/9 1/9 1/9; 1/9 1/9 1/9];
Ifiltered = uint8(convn(double(Inoise), double(G)));
figure; imshow(Ifiltered)

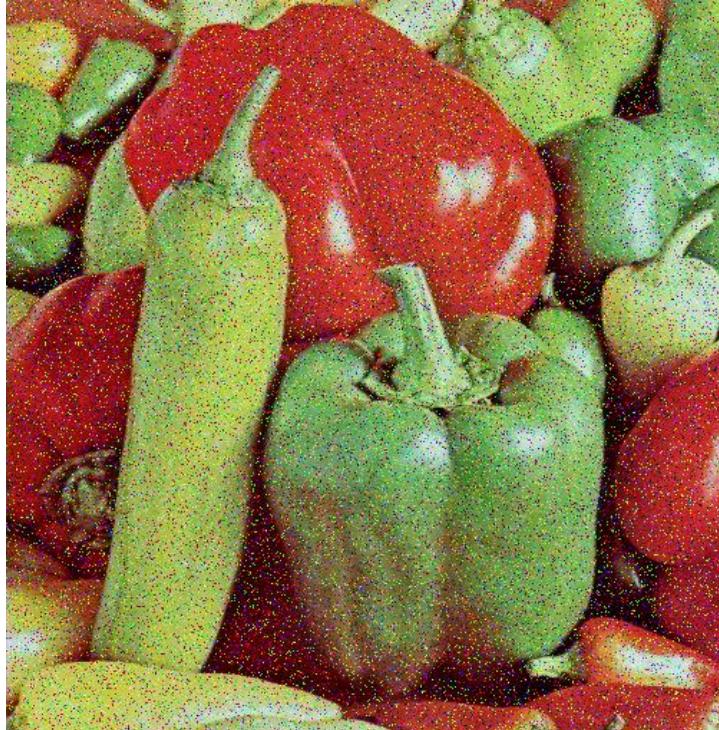
```

Filter

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$



Original image



Noisy image



Filtered image



- Pelembutan citra dengan penapis rerata (*mean filter*) menghasilkan efek *blurring*.
- Efek *blurring* terjadi karena pemerataan derajat keabuan (*graylavel*).
- Jadi, penapis rerata selain digunakan untuk mereduksi derau, ia juga dapat digunakan untuk menghasilkan efek *blurring*.

Penapis rerata lainnya:

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

(a)

$$\frac{1}{25} \times \begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

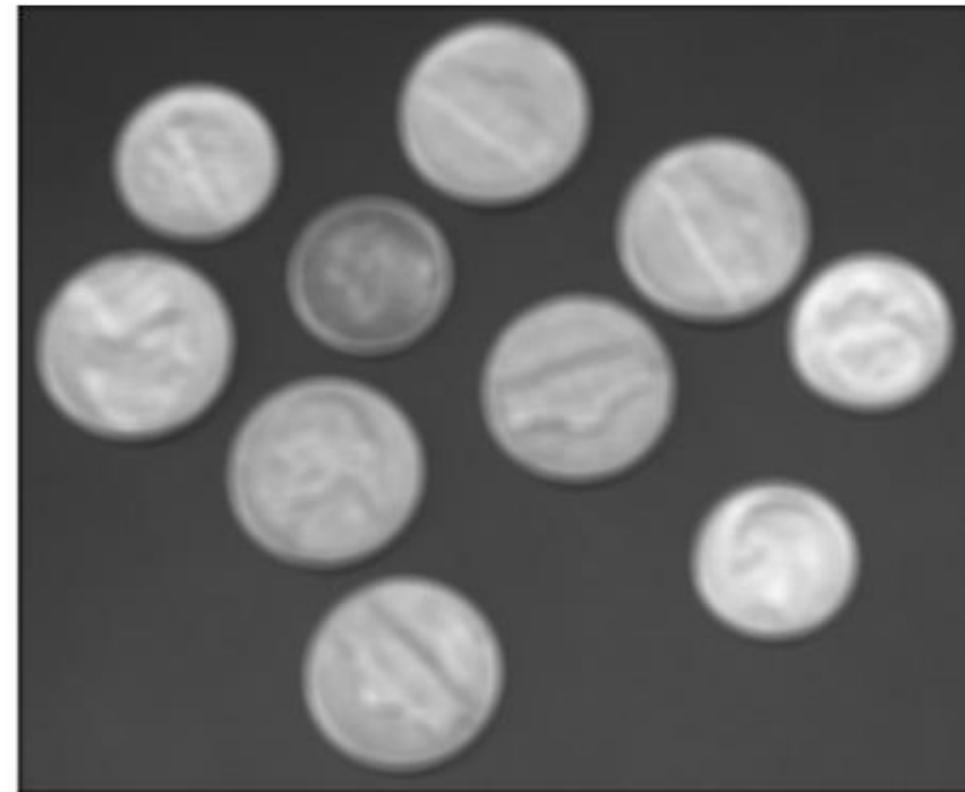
$$\frac{1}{49} \times \begin{array}{|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

(c)

Menghasilkan efek *blurring* pada gambar dengan penapis rerata



Original Image



Filtered Image

Program: Penapis rerata 5 x 5

```
I = imread('coins.bmp');  
imshow(I);  
G = 1/25 * [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];  
Ifiltered = uint8(convn(double(I), double(G)));  
figure; imshow(Ifiltered)
```



Original image



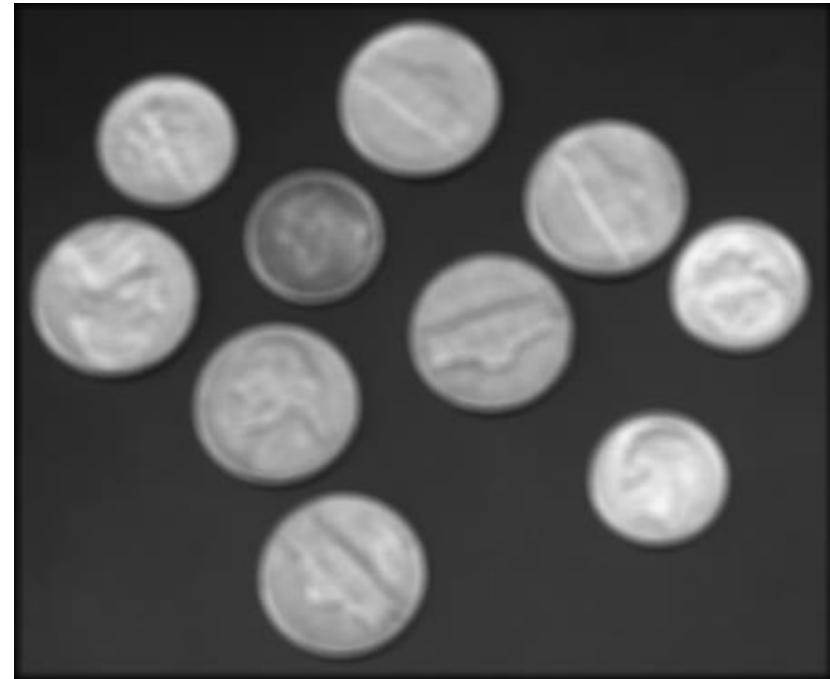
Filtered image

Program: Penapis rerata 7 x 7:

```
I = imread('coins.bmp');
imshow(I);
G = 1/49 * [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 1 1];
Ifiltered = uint8(convn(double(I), double(G)));
figure; imshow(Ifiltered)
```



Original image

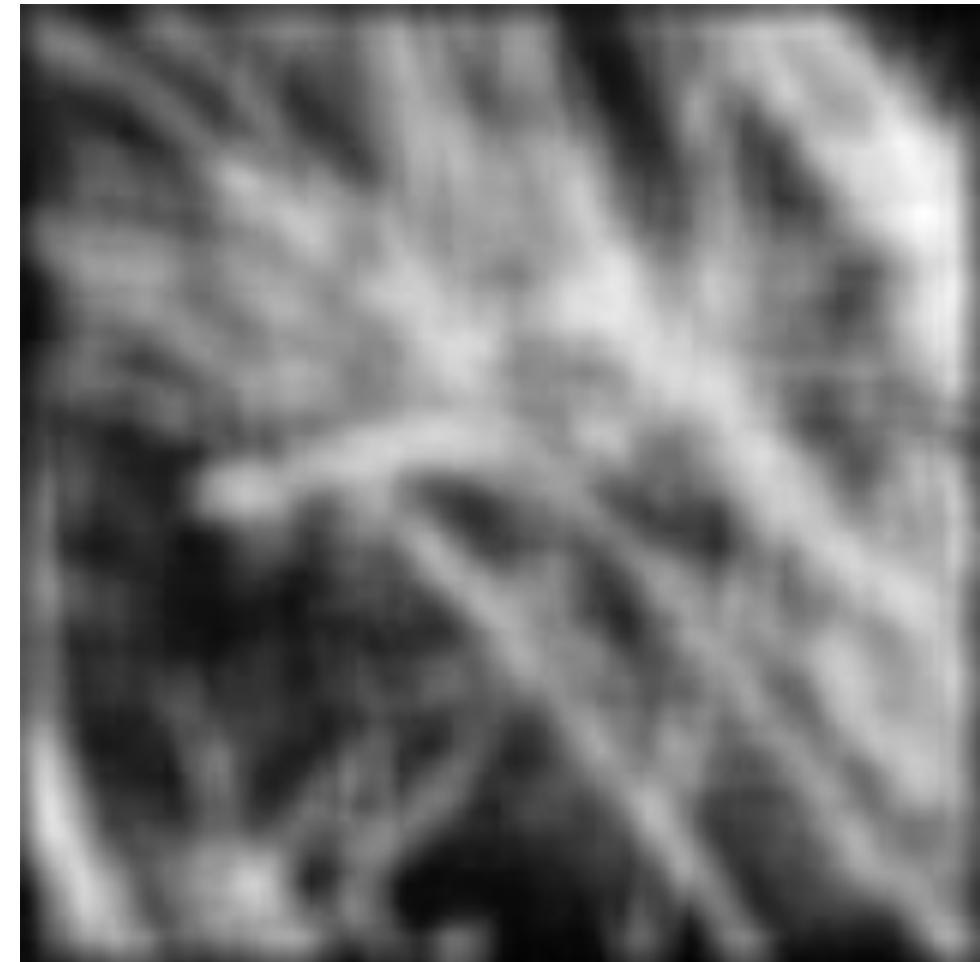


Filtered image

Contoh lain menghasilkan efek *blurring* pada gambar dengan penapis rerata

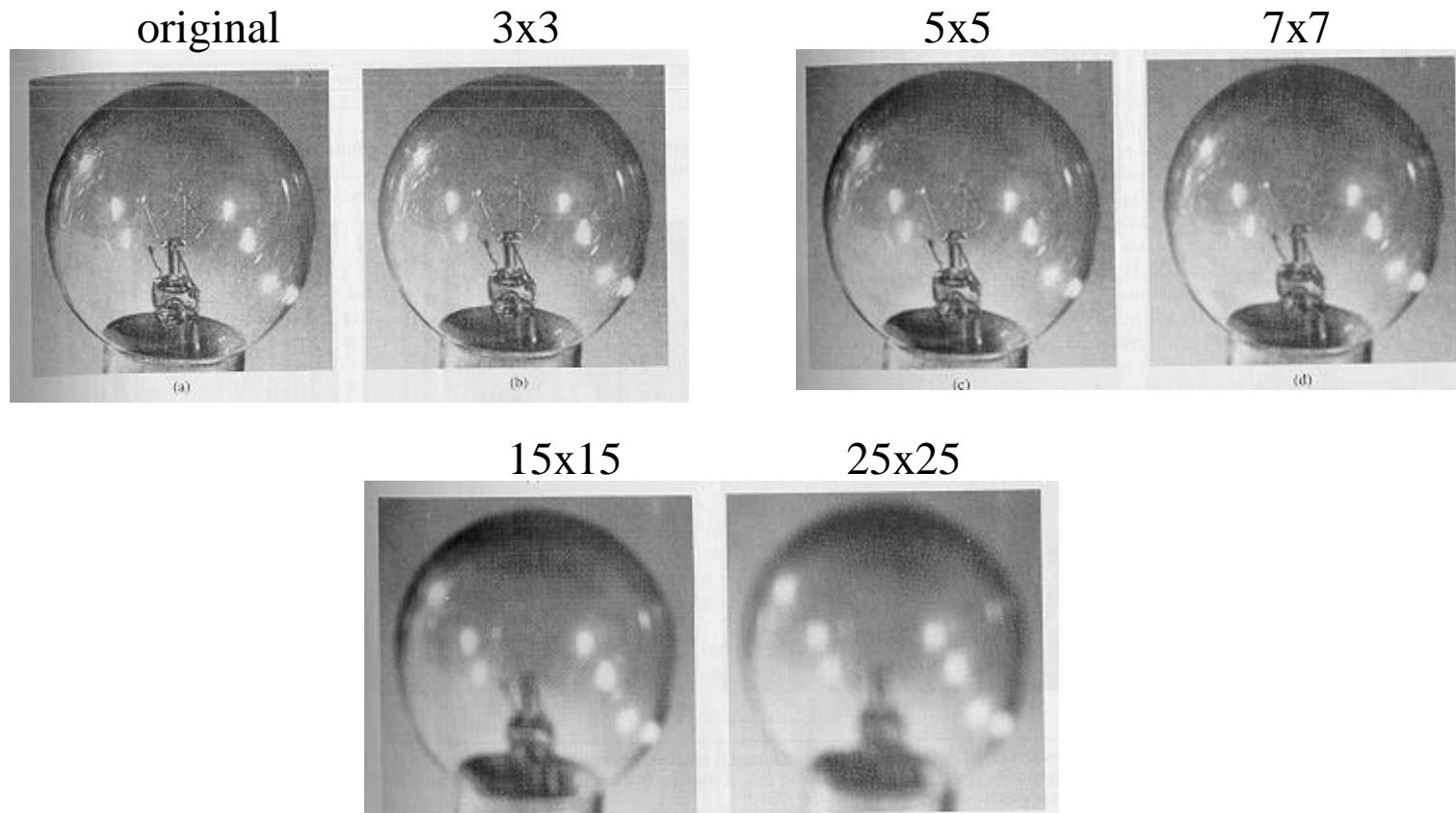


Original Image

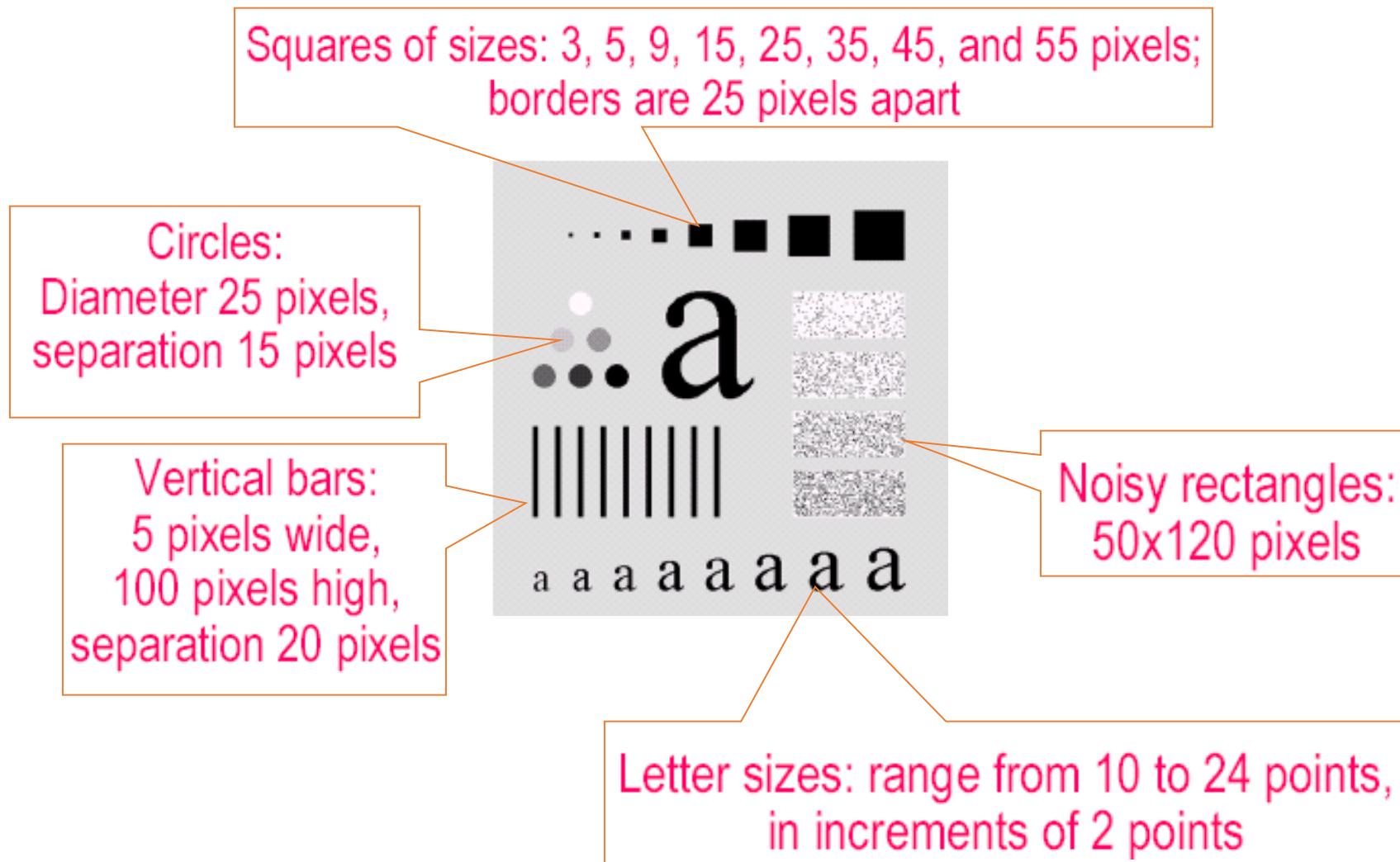


Filtered Image

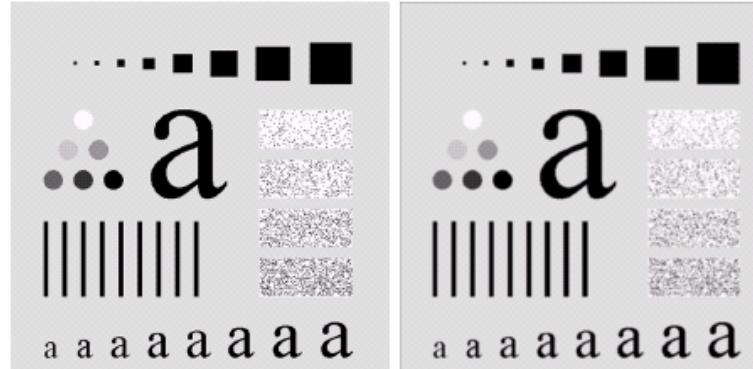
Trade-off: noise vs blurring dan kehilangan detil (*mean filter n x n*)



Trade-off: noise vs blurring dan kehilangan detil

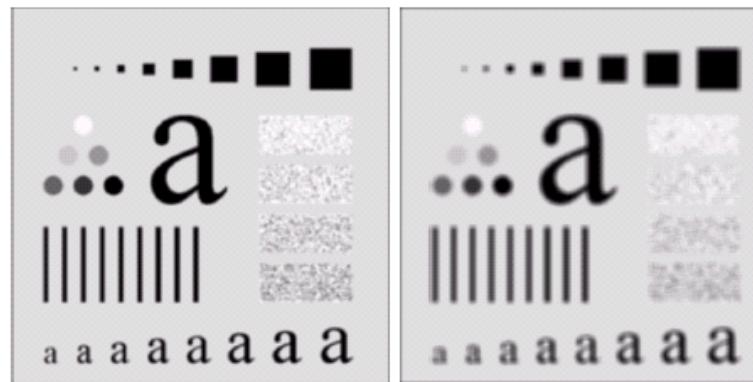


Original image
size: 500 x 500



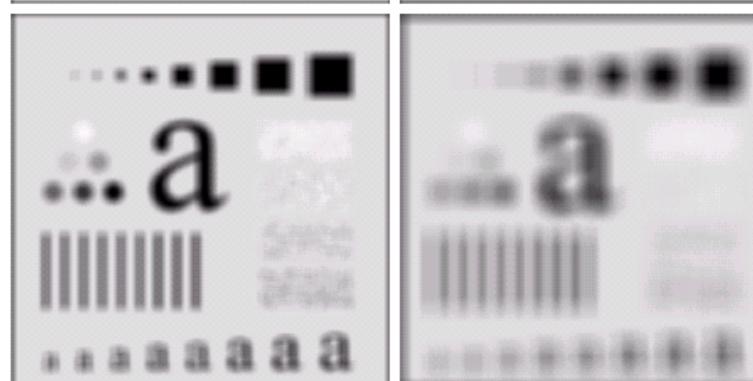
Smoothed by
3 x 3 mean filter

Smoothed by
5 x 5 mean filter



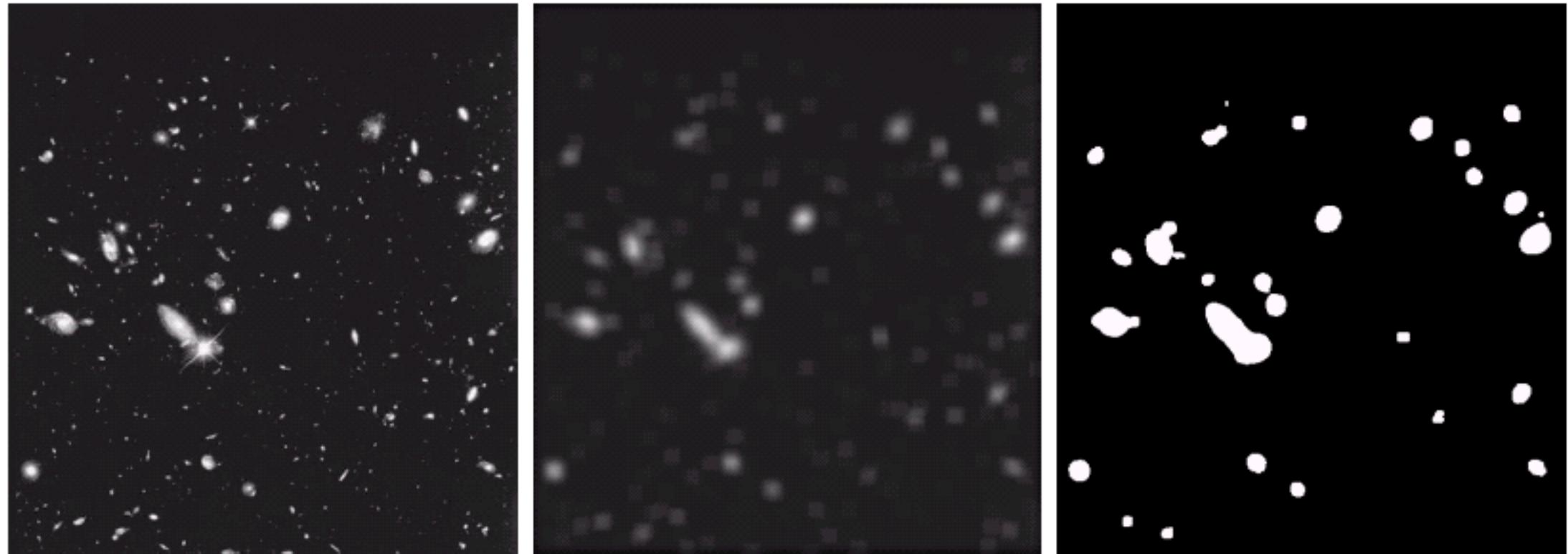
Smoothed by
9 x 9 box filter

Smoothed by
15 x 15 mean filter



Smoothed by
35 x 35 mean filter

Trade-off: noise vs blurring dan kehilangan detil



a b c

FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15×15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Penapis Lolos-Rendah (*Low-pass filter*)

- Penapis rerata disebut juga **penapis lolos-rendah** (*low-pass filter*), karena penapis tersebut menekan komponen yang berfrekuensi tinggi (misalnya *pixel* derau, *pixel* tepi) dan meloloskan komponen yang berfrekuensi rendah.
- Penapis lolos-rendah memiliki aturan sebagai berikut:
 - Semua bobot di dalam penapis harus positif
 - Jumlah semua bobot sama dengan 1

- Jika jumlah semua bobot lebih besar dari 1, maka konvolusi menghasilkan penguatan pada derau (tidak diinginkan).
- Jika jumlah semua bobot kurang dari 1, maka yang dihasilkan adalah penurunan, dan nilai setiap *pixel* di seluruh bagian citra berkurang. Akibatnya, citra hasil pelembutan tampak lebih gelap.
- Penapis lolos-rendah lainnya:

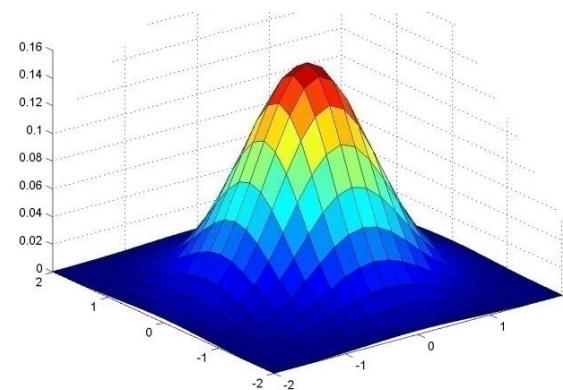
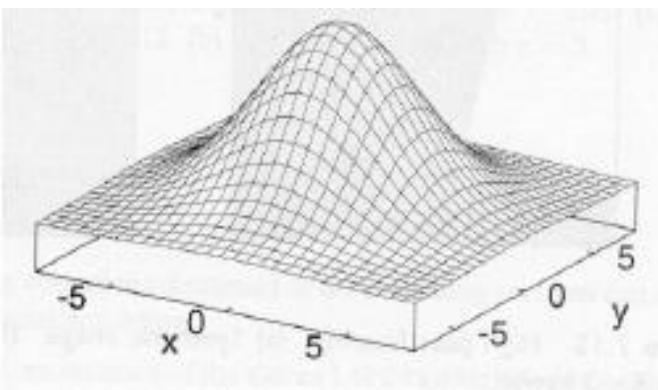
$$\begin{bmatrix} 1/10 & 1/10 & 1/10 \\ 1/10 & 1/5 & 1/10 \\ 1/10 & 1/10 & 1/10 \end{bmatrix}$$

Jumlah seluruh bobot = $1/10 + 1/10 + 1/10 + 1/10 + 1/5 + 1/10 + 1/10 + 1/10 + 1/10 = 8/10 + 1/5 = 8/10 + 2/10 = 10/10 = 1$

Gaussian Smoothing

- Bobot di dalam penapis dihitung dengan penerokan (*sampling*) pada fungsi Gaussian

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} \exp -\frac{x^2 + y^2}{2\sigma^2}$$



3 x 3 Gaussian mask

1	2	1
2	4	2
1	2	1

$$\frac{1}{16} \times$$

7 x 7 Gaussian mask

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

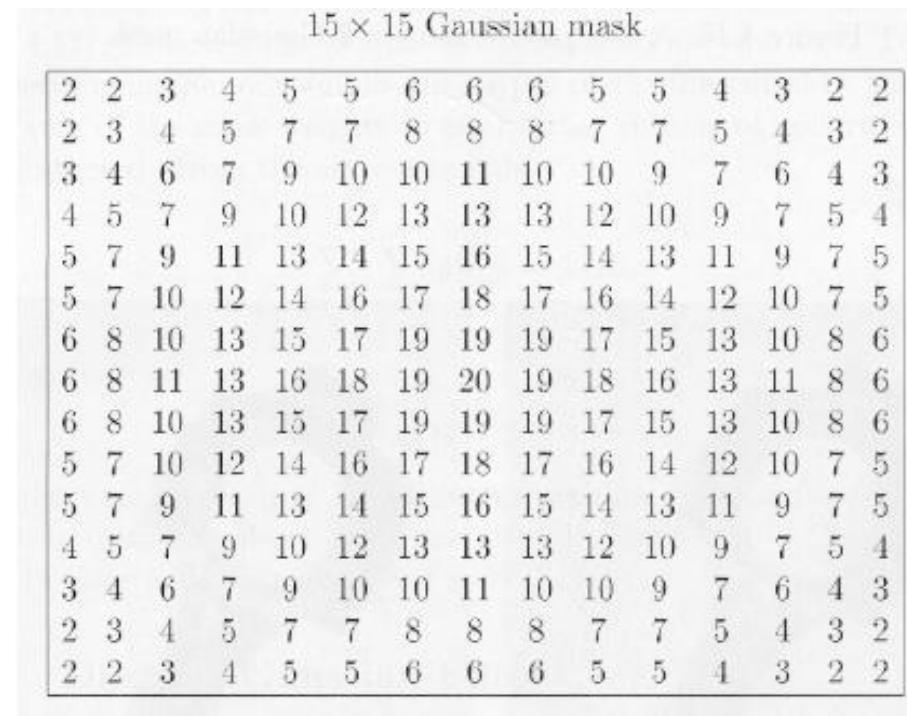
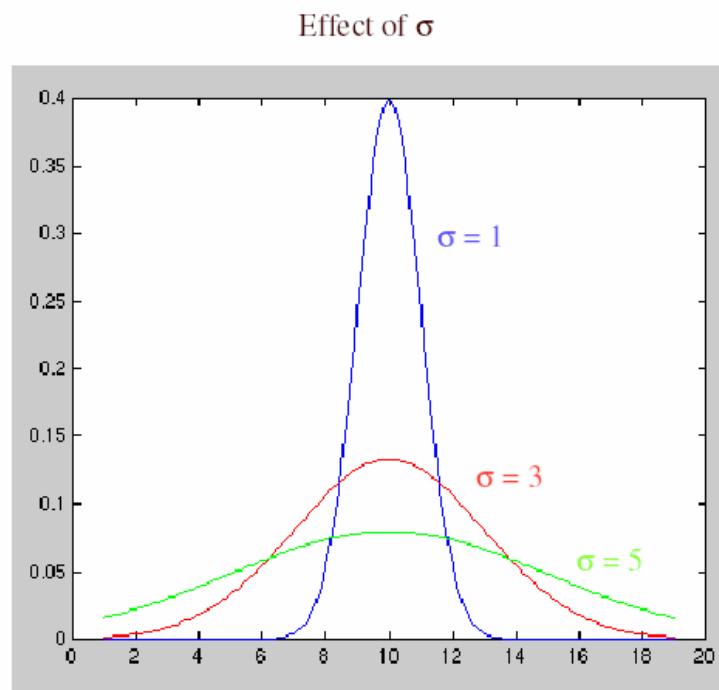
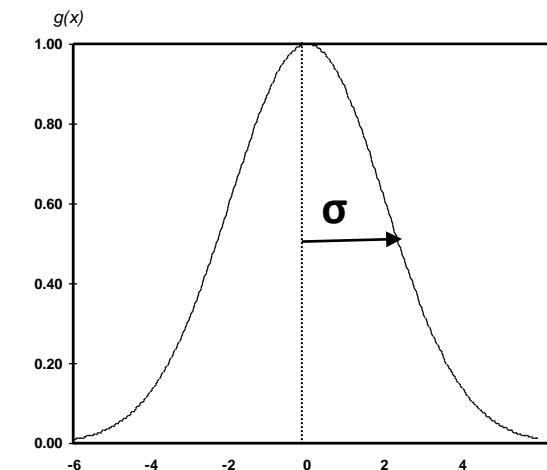
$$\frac{1}{140} \times$$

Note: weight values decrease with distance from mask center!

- Ukuran penapis bergantung pada σ

$height = width = 5\sigma$ (subtends 98.76% of the area)

- σ menentukan derajat pelembutan!
Makin besar σ makin lembut, makin blur



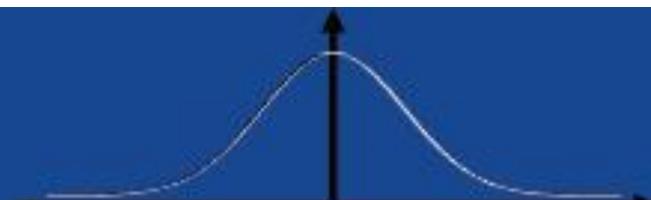
$$\sigma = 3$$



small σ



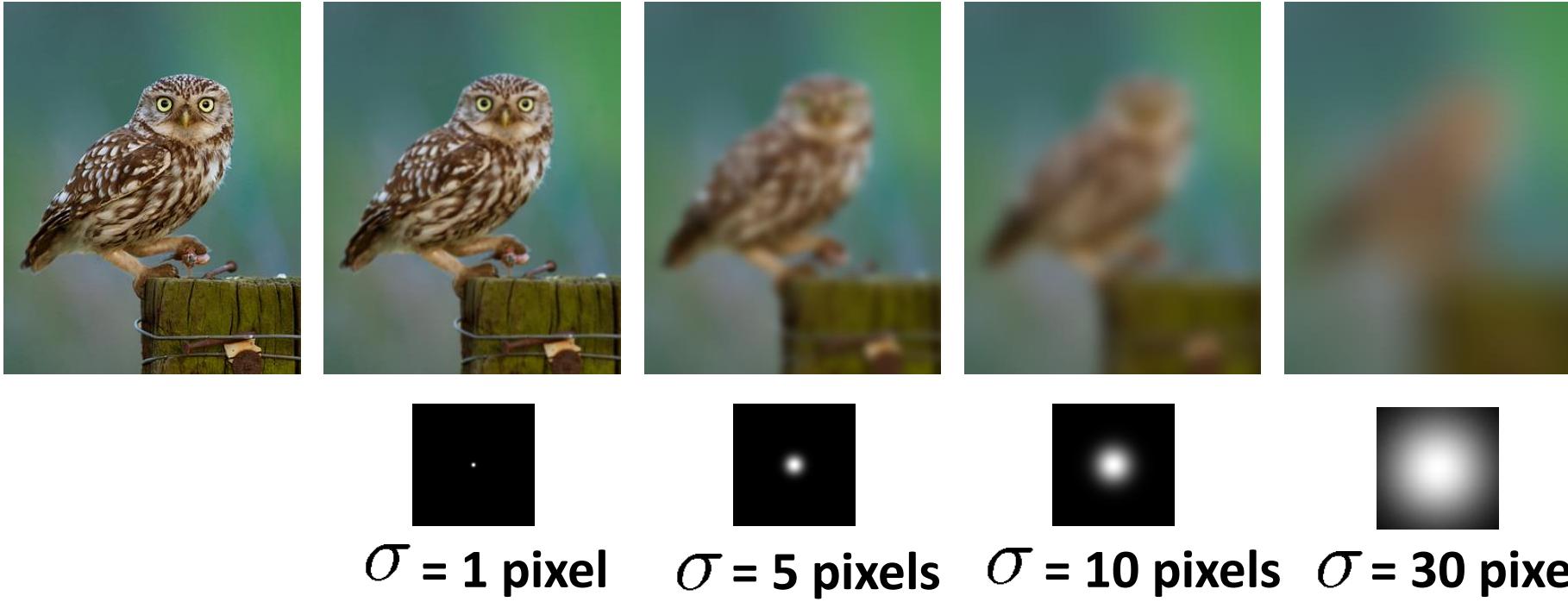
limited smoothing



large σ



strong smoothing



Program: Gaussian Smoothing 7 x 7:

```
I = imread('cameraman.bmp');
imshow(I);
G = 1/140 * [1 1 2 2 2 1 1; 1 2 2 4 2 2 1; 2 2 4 8 4 2 2; 2 4 8 16 8 4 2;
               1 1 2 2 2 1 1; 1 2 2 4 2 2 1; 2 2 4 8 4 2 2];
Ifiltered = uint8(convn(double(I), double(G)));
figure; imshow(Ifiltered)
```



Original image



Filtered image

- Di dalam Matlab sudah tersedia fungsi *gaussian smoothing* dengan parameter simpangan baku (σ), `imgaussfilt(I, sigma)`

```
I = imread('cameraman.bmp'); imshow(I);
Iblur1 = imgaussfilt(I,2);
Iblur2 = imgaussfilt(I,4);
Iblur3 = imgaussfilt(I,8);
figure
imshow(I)
title('Original image')

figure
imshow(Iblur1)
title('Smoothed image, \sigma = 2')

figure
imshow(Iblur2)
title('Smoothed image, \sigma = 4')

figure
imshow(Iblur3)
title('Smoothed image, \sigma = 8')
```

Original image



Smoothed image, $\sigma = 2$



Smoothed image, $\sigma = 4$



Smoothed image, $\sigma = 8$



Averaging vs Gaussian Smoothing



input



Averaging

Gaussian

Penapis Nirlanjar (*Non-linear Filter*)

- Penapis rerata (*mean filter*) adalah salah satu contoh penapis lolos-rendah.
- Penapis lolos-rendah merupakan penapis lanjar (*linear filter*).
- Operasi pelembutan dapat juga dilakukan dengan menggunakan penapis nirlanjar.
- Pada penapis nirlanjar, sebuah “jendela” (*window*) berukuran $n \times n$ memuat sejumlah *pixel* (ganjil) digeser titik demi titik pada seluruh daerah citra.
- Tiga macam penapis nirlanjar:
 1. Penapis minimum (*min filter*) – mengganti nilai *pixel* di tengah *window* dengan nilai minimum di dalam *window*.
 2. Penapis maksimum (*max filter*) – mengganti nilai *pixel* di tengah *window* dengan nilai maksimum dalam *window*.
 3. Penapis median (*median filter*) – mengganti nilai *pixel* di tengah *window* dengan nilai median di dalam *window*.

Contoh penapis median (*median filter*):

13	10	15	14	18
12	10	10	10	15
11	11	35	10	10
13	9	12	10	12
13	12	9	8	10

(a) *Pixel* bernilai 35 terkena derau

13	10	15	14	18
12	10	10	10	15
11	11	10	10	10
13	9	12	10	12
13	12	9	8	10

(b) 35 diganti dengan median dari kelompok 3×3 *pixel*

Misalkan *pixel* di tengah, 35, akan diproses.

Urutkan *pixel-pixel* tersebut:

9 10 10 10 **10** 10 11 12 35

Median dari kelompok tersebut adalah 10 (dicetak tebal, warna biru).

Titik tengah dari jendela (35) sekarang diganti dengan nilai median (10).

```
I = imread('zelda.bmp');  
Inoise = imnoise(I, 'salt & pepper', 0.1);  
figure; imshow(I);  
figure; imshow(Inoise);  
Ifiltered = medfilt2(Inoise, [3 3]);  
figure; imshow(Ifiltered)
```

Penapis Median



Original image



Noisy image



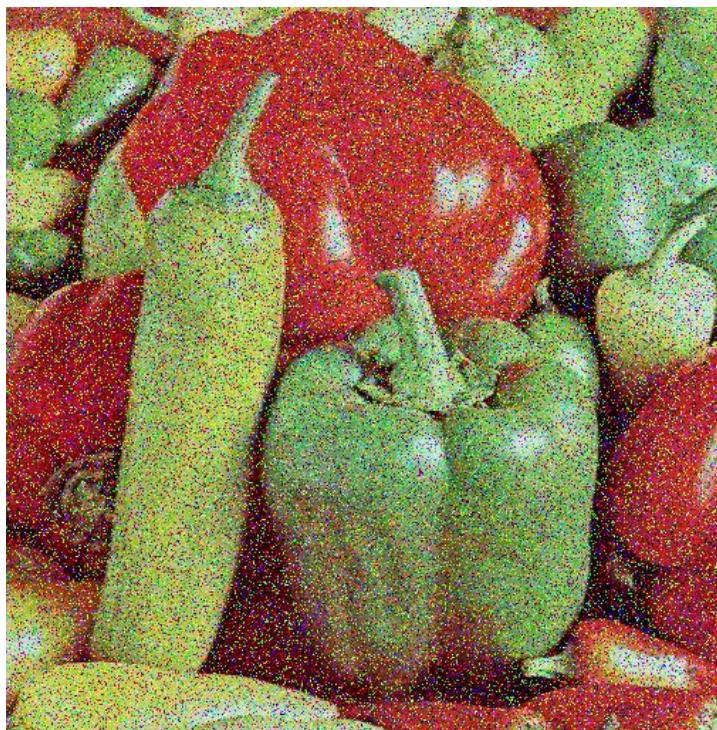
Filtered image

```
I = imread('peppers512.bmp');
Inoise = imnoise(I, 'salt & pepper', 0.2);
figure; imshow(I); figure; imshow(Inoise);
r = Inoise(:,:,1); g = Inoise(:,:,2); b = Inoise(:,:,3);
Ifiltered_r = medfilt2(r, [3 3]);
Ifiltered_g = medfilt2(g, [3 3]);
Ifiltered_b = medfilt2(b, [3 3]);
Ifiltered = cat(3, Ifiltered_r, Ifiltered_g, Ifiltered_b);
figure; imshow(Ifiltered)
```

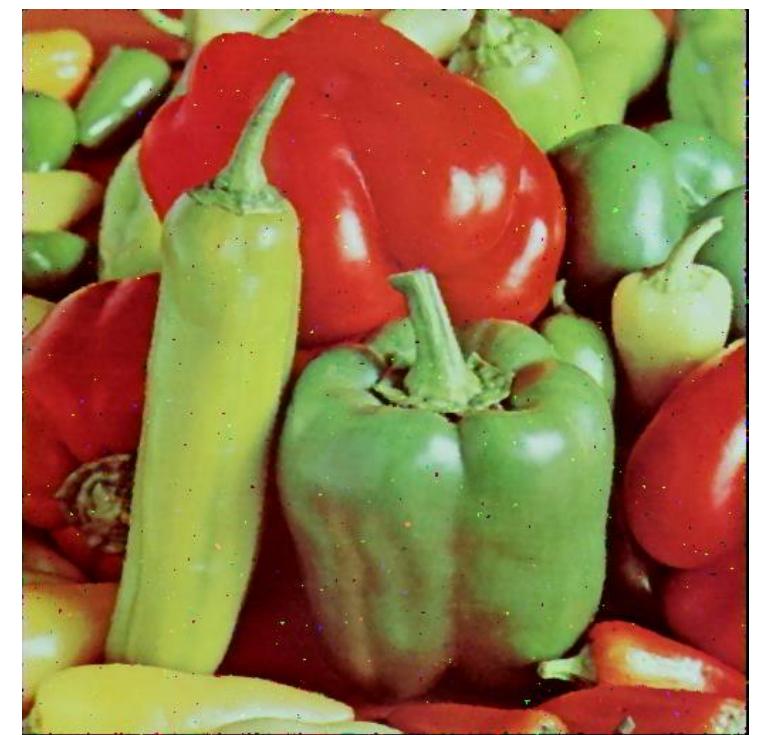
Penapis Median



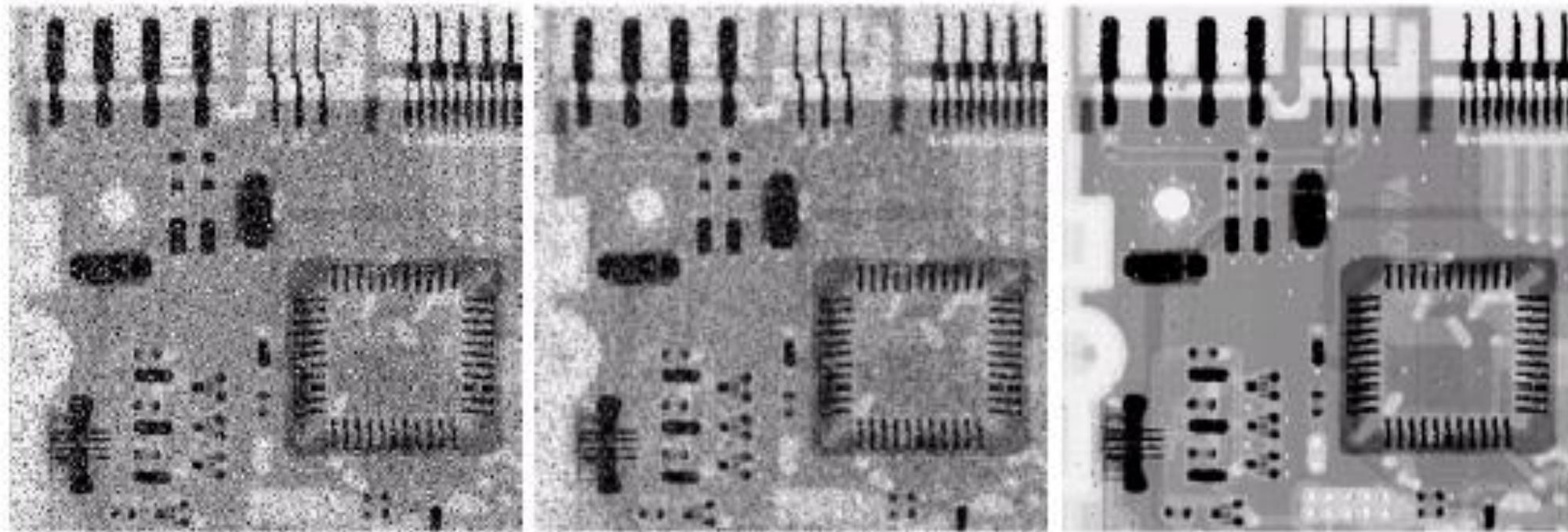
Original image



Noisy image



Filtered image



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

2. Penajaman Citra (*image sharpening*)

- Tujuan: memperjelas **tepi** (*edge*) objek di dalam citra.
- Penajaman citra merupakan kebalikan dari operasi pelembutan citra karena operasi ini menghilangkan bagian citra yang lembut.



Gambar Kiri: Citra Lena semula, **Kanan:** Citra Lena setelah penajaman

Di mana tepi?



Ini tepi

- Operasi penajaman dilakukan dengan melewatkannya pada **penapis lolos-tinggi** (*high-pass filter*).
- Penapis lolos-tinggi akan meloloskan (sekaligus memperkuat) komponen yang berfrekuensi tinggi (misalnya tepi atau pinggiran objek) dan akan menurunkan komponen berfrekuensi rendah. Akibatnya, pinggiran objek terlihat 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*)

Penapis Lolos-Tinggi (*high-pass filter*)

Aturan penapis lolos-tinggi:

1. bobot di dalam penapis boleh positif, negatif, atau nol
2. jumlah semua bobot adalah 0 atau 1

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

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

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

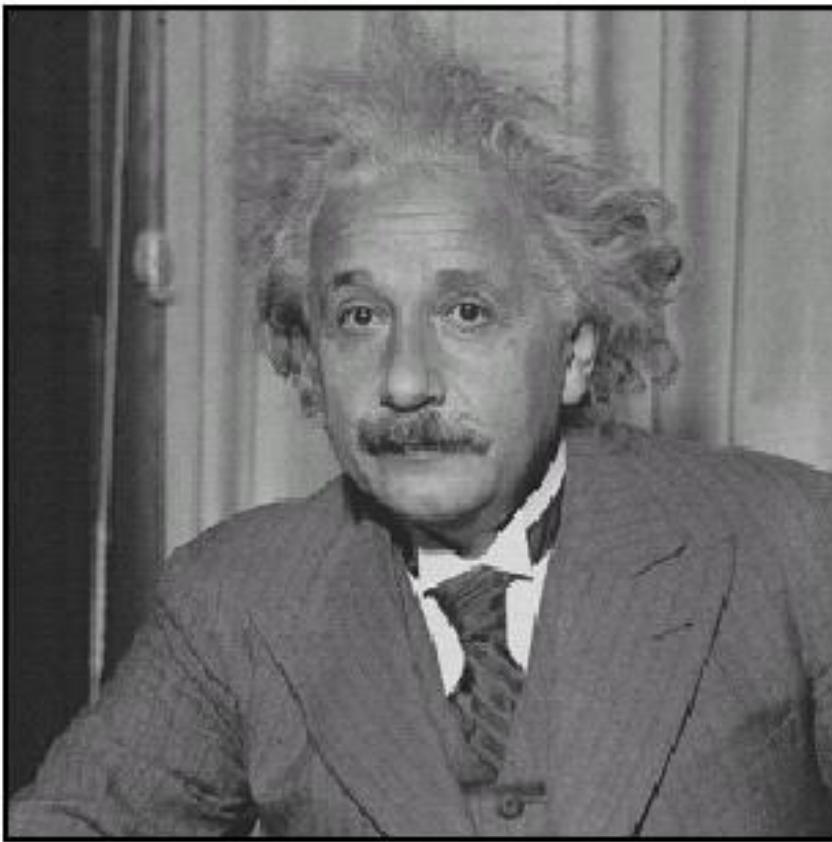
(iv)
$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$
$$\sum = 1$$

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

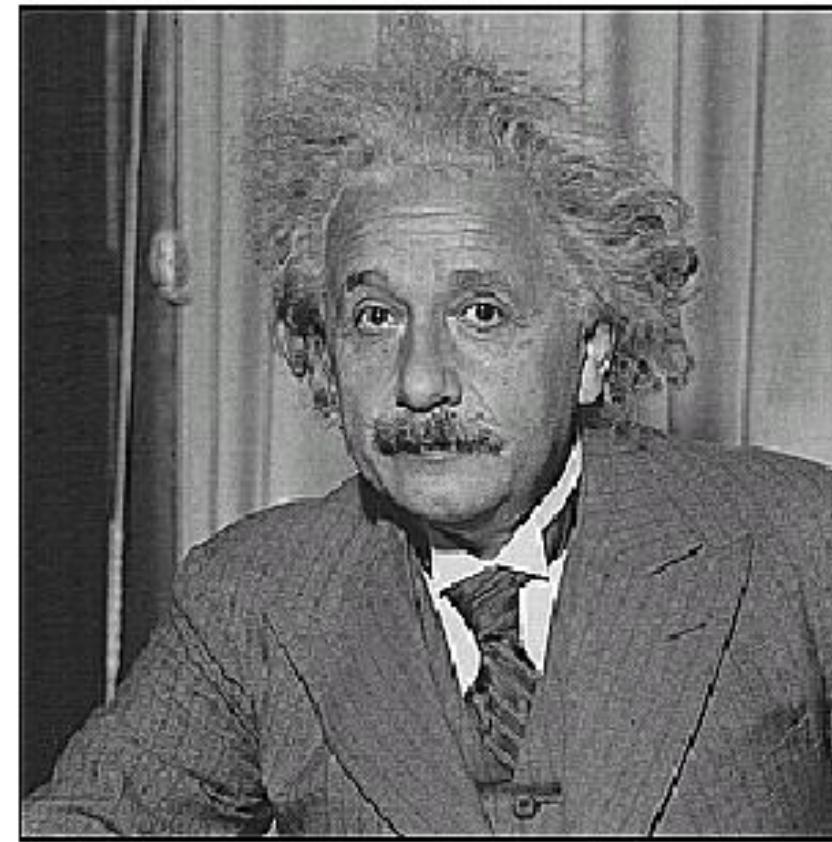
(vi)
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$
$$\sum = 0$$

Jika jumlah bobot = 0, maka komponen berfrekuensi rendah akan turun nilainya

Jika jumlah bobot sama dengan 1, maka komponen berfrekuensi rendah akan tetap sama dengan nilai semula. 41



before



after

```

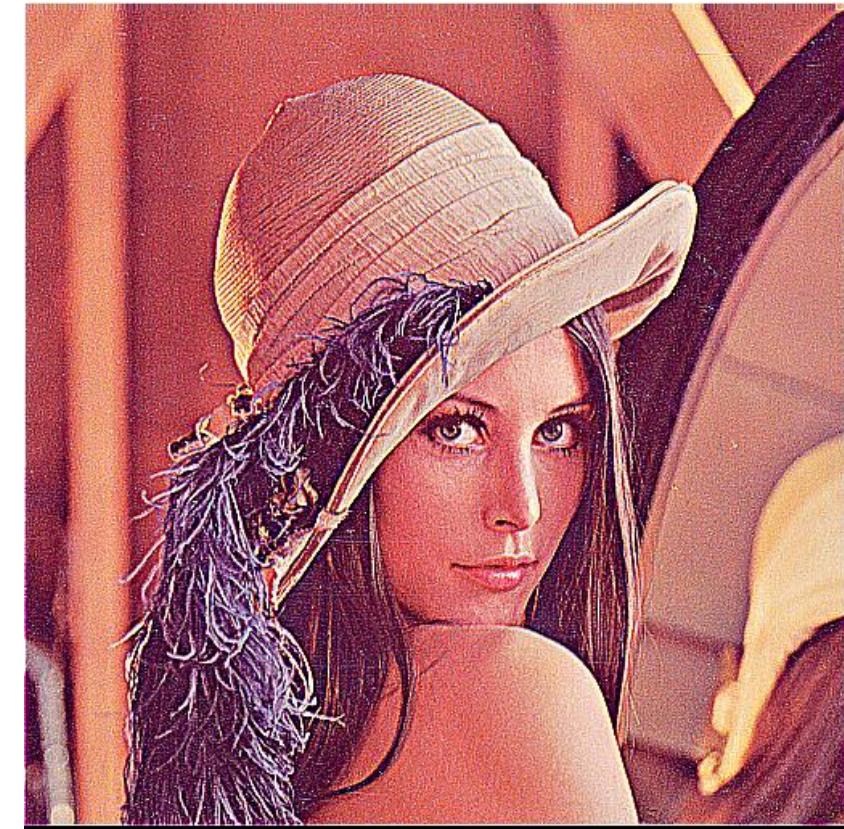
I = imread('lena.bmp');
figure, imshow(I);
G = [-1 -1 -1; -1 9 -1; -1 -1 -1];
Isharp = uint8(convn(double(I), double(G)));
figure; imshow(Isharp)

```



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

$$\sum = 1$$

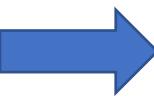


Gambar Kiri: Citra Lena semula, **Kanan:** Citra Lena setelah penajaman

```
I = imread('lena.bmp');  
figure, imshow(I);  
G = [0 -1 0; -1 5 -1; 0 -1 0];  
Isharp = uint8(convn(double(I), double(G)));  
figure; imshow(Isharp)
```



$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$



$$\sum = 1$$



```
I = imread('lena.bmp');  
figure, imshow(I);  
G = [-1 -1 -1; -1 8 -1; -1 -1 -1];  
Isharp = uint8(convn(double(I),  
double(G)));  
figure; imshow(Isharp)
```



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

$$\Sigma = 0$$



Konsep dasar untuk *edge detection*

```
I = imread('lena.bmp');  
figure, imshow(I);  
G = [0 1 0; 1 -4 1; 0 1 0];  
Isharp = uint8(convn(double(I),  
double(G)));  
figure; imshow(Isharp)
```



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

$$\Sigma = 0$$



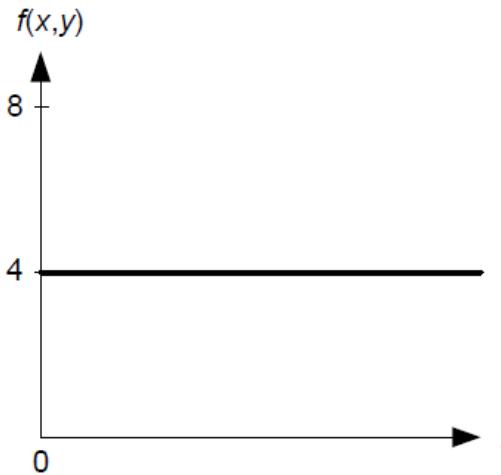
Konsep dasar untuk *edge detection*

Jika jumlah bobot = 0, maka komponen berfrekuensi rendah akan turun nilainya, area bernilai konstan menjadi 0
Jika jumlah bobot sama dengan 1, maka komponen berfrekuensi rendah akan tetap sama dengan nilai semula.

Citra semula:

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

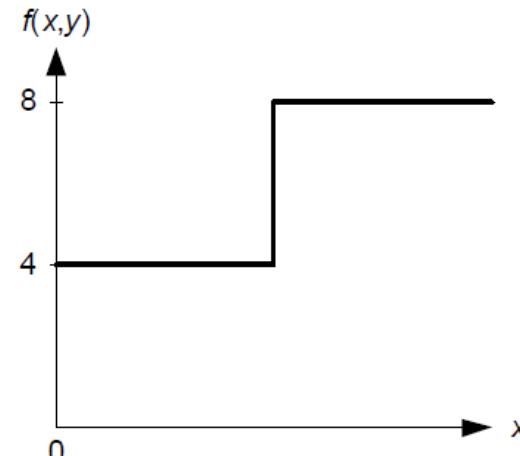
Kurva yang merepresentasikan citra:



Citra semula:

$$\begin{bmatrix} 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 & 8 \end{bmatrix}$$

Kurva yang merepresentasikan citra:



Catatan: hasil penapisan mungkin menghasilkan nilai negatif, petakan kembali ke dalam [0, 255].

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

Hasil konvolusi dengan penapis (i):

$$\begin{bmatrix} x & x & x & x & x & x & x \\ x & 0 & 0 & 0 & 0 & 0 & x \\ x & 0 & 0 & 0 & 0 & 0 & x \\ x & 0 & 0 & 0 & 0 & 0 & x \\ x & x & x & x & x & x & x \end{bmatrix}$$

Hasil konvolusi dengan penapis (ii):

$$\begin{bmatrix} x & x & x & x & x & x & x \\ x & 4 & 4 & 4 & 4 & 4 & x \\ x & 4 & 4 & 4 & 4 & 4 & x \\ x & 4 & 4 & 4 & 4 & 4 & x \\ x & x & x & x & x & x & x \end{bmatrix}$$

(a)

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

Hasil konvolusi dengan penapis (i):

$$\begin{bmatrix} x & x & x & x & x & x & x \\ x & 0 & -12 & +12 & 0 & 0 & x \\ x & 0 & -12 & +12 & 0 & 0 & x \\ x & 0 & -12 & +12 & 0 & 0 & x \\ x & x & x & x & x & x & x \end{bmatrix}$$

Hasil konvolusi dengan penapis (ii):

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

(b)

- Nilai bobot yang besar di titik pusat penapis memainkan peranan kunci dalam proses konvolusi.
- Pada komponen citra dengan frekuensi tinggi (yang berarti perubahan yang besar pada nilai intensitasnya), nilai tengah ini dikalikan dengan nilai *pixel* yang dihitung.
- Bobot negatif yang lebih kecil di sekitar titik tengah penapis bekerja untuk mengurangi faktor pembobotan yang besar.
- Efek nettonya adalah, *pixel-pixel* yang bernilai besar diperkuat, sedangkan area citra dengan intensitas *pixel* konstan tidak berubah nilanya.

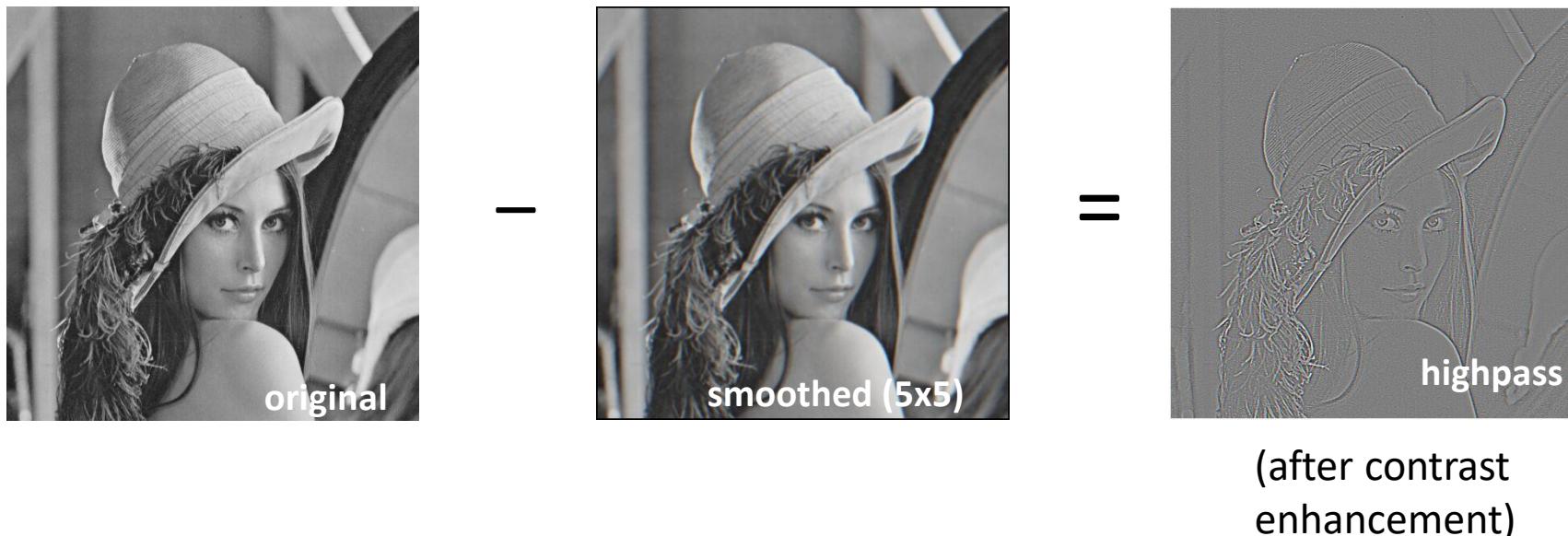
Penapis lain untuk penajaman citra

- *Unsharp masking*
- *High Boost filter*
- *Gradient (1st derivative)*
- *Laplacian (2nd derivative)*
- Dua penapis terbawah akan dibahas dalam materi Pendektsian Tepi (*Edge Detection*)

Sharpening Filters: Unsharp Masking

- Menghasilkan citra tajam dengan cara mengurangkan citra hasil pelembutan (*smoothed image* atau citra hasil *low-pass filter*) dari citra semula (*original image*).

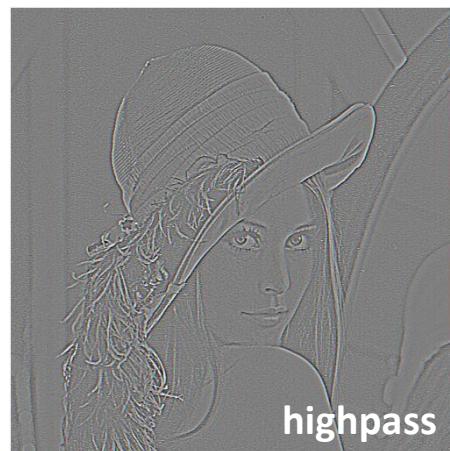
$$\text{Highpass} = \text{Original} - \text{Lowpass}$$



Original + highpass = sharp image



+



=



Source: S. Lazebnik

```

Original = imread('lena.bmp');
G = [1/25 1/25 1/25 1/25 1/25;
      1/25 1/25 1/25 1/25 1/25;
      1/25 1/25 1/25 1/25 1/25;
      1/25 1/25 1/25 1/25 1/25;
      1/25 1/25 1/25 1/25 1/25];
Lowpass = uint8(convn(double(Original),
                      double(G), 'same'));
Highpass = Original - Lowpass;
Isharp = Original + Highpass;
figure, imshow(Original);
figure, imshow(Lowpass);
figure, imshow(Highpass);
figure, imshow(Isharp);

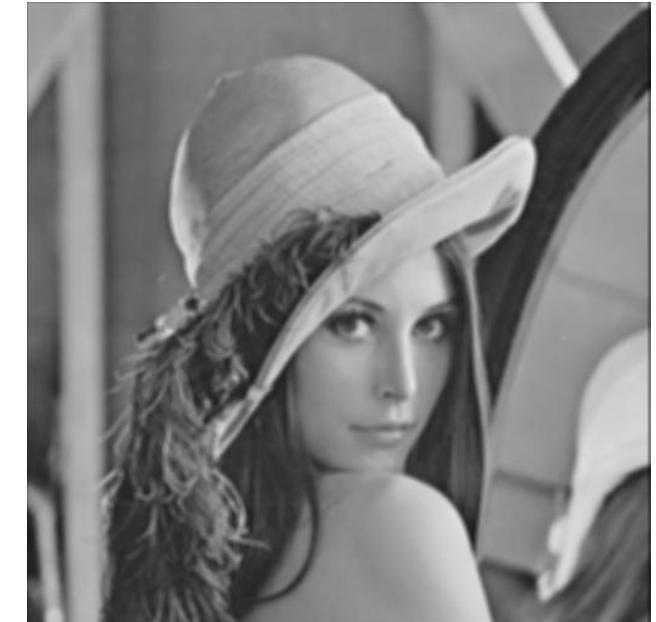
```



Original image



Highpass image



Lowpass image (smooth)



Sharp image

Sharpening Filters: High Boost

- *Boost filtering* digunakan bilamana citra masukan lebih gelap daripada citra yang diinginkan.
- *High-boost filter* menghasilkan citra menjadi lebih terang dan lebih alami.
- **High boost filter:** amplifikasi citra original, lalu kurangkan dengan *lowpass image*.

$$\begin{aligned} \text{Highboost} &= \alpha \text{ Original} - \text{Lowpass} \\ &= (\alpha - 1) \text{ Original} + \text{Original} - \text{Lowpass} \\ &= (\alpha - 1) \text{ Original} + \text{Highpass} \end{aligned}$$



- Jika $\alpha = 1$, kita mendapatkan *unsharp masking*.
- Jika $\alpha > 1$, bagian citra original ditambahkan kembali ke citra hasil *high pass filter*.

$$Highboost = (\alpha - 1) \text{ Original} + \text{Highpass}$$

Satu cara untuk *mengimplementasikan high boost filtering* adalah dengan menggunakan *mask* berikut:

$A >= 1$			$A = 2$		
$w = 9A - 1$			$w = 17$		
-1	-1	-1	-1	-1	-1
-1	w	-1	-1	17	-1
-1	-1	-1	-1	-1	-1

$$\alpha = 2.4$$

```
A = 2.4;  
Original = imread('lena.bmp');  
G = [1/25 1/25 1/25 1/25 1/25;  
      1/25 1/25 1/25 1/25 1/25;  
      1/25 1/25 1/25 1/25 1/25;  
      1/25 1/25 1/25 1/25 1/25;  
      1/25 1/25 1/25 1/25 1/25];  
Lowpass =  
uint8(convn(double(Original),  
          double(G), 'same'));  
Highpass = Original - Lowpass;  
Isharp = (A - 1)*Original + Highpass;  
figure, imshow(Original);  
figure, imshow(Lowpass);  
figure, imshow(Highpass);  
figure, imshow(Isharp);
```



Original image



Highpass image



Lowpass image (smooth)



Sharp image

$$\alpha = 2.9$$

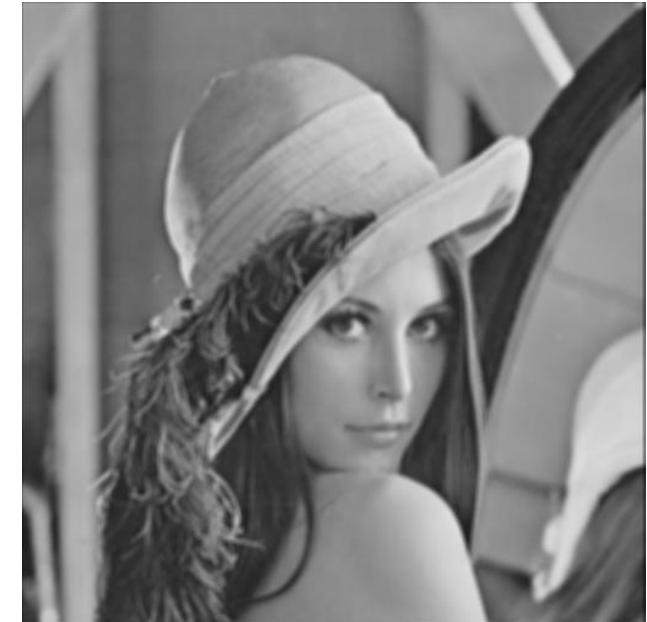
```
A = 2.9;  
Original = imread('lena-gray.bmp');  
G = [1/25 1/25 1/25 1/25 1/25;  
      1/25 1/25 1/25 1/25 1/25;  
      1/25 1/25 1/25 1/25 1/25;  
      1/25 1/25 1/25 1/25 1/25;  
      1/25 1/25 1/25 1/25 1/25];  
Lowpass =  
uint8(convn(double(Original),  
          double(G), 'same'));  
Highpass = Original - Lowpass;  
Isharp = (A - 1)*Original + Highpass;  
figure, imshow(Original);  
figure, imshow(Lowpass);  
figure, imshow(Highpass);  
figure, imshow(Isharp);
```



Original image



Highpass image



Lowpass image (smooth)



Sharp image