

# Image Smoothing

---

- Biasa dilakukan untuk menghilangkan efek pada citra digital yang disebabkan oleh keterbatasan sistem pencuplikan atau kanal transmisi
- Teknik penghalusan:
  - Domain spasial, contoh: mean, median, dan modus filtering
  - Domain frekwensi, contoh: lowpass filtering
- Efek samping: citra menjadi blur

# Mean filtering (neighborhood averaging)

---

- Diberikan (bagian) citra NxN pixel:  $f(x,y)$
- Citra hasil:  $g(x,y) \rightarrow$  merata-ratakan nilai gray level pixel-pixel pada  $f(x,y)$  yang termasuk dalam area (neighborhood) tertentu

$$g(x,y) = \frac{1}{M} \sum_{(n,m) \in S} f(n,m); \quad x, y = 0, 1, \dots, N-1$$

**S:** himpunan titik koordinat yang merupakan tetangga (neighbor) dari titik  $(x,y)$ , termasuk  $(x,y)$  itu sendiri

**M:** Jumlah total titik dalam neighborhood {neighborhood tidak selalu berbentuk bujur sangkar}

# Matriks ketetanggaan

---

- ◻ Biasanya N bilangan ganjil → titik  $(x,y)$  bisa berada di tengah matriks

1	2	3
4	T	5
6	7	8

$3 \times 3$

1	2	3	4	5
6	7	8	9	10
11	12	T	13	14
15	16	17	18	19
20	21	22	23	24

$5 \times 5$

# Matriks Mask

---

- Ukuran sama dengan matriks ketetanggaan
- Menyimpan operasi yang akan dikenakan terhadap matriks ketetanggaan
  - Isi matriks mask menentukan operasi terhadap matriks ketetanggaan
  - Untuk averaging diisi dengan  $1/M$
- Operasi secara skalar:

$$G(x, y) = \sum_{(n,m) \in S} Mask(n, m) \times Neighborhood(n, m)$$

---

# Contoh

CITRA AWAL


CITRA HASIL


90	3	81
9	180	45
27	18	9

x

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

=

10	3/9	9
1	20	5
3	2	1

51,3 ≈ 51 +

NEIGHBOR

MASK

# Shortcut

---

- Karena yang sebenarnya dilakukan adalah mencari rata-rata, maka dapat langsung dilakukan penjumlahan isi matriks *neighborhood* baru kemudian membaginya dengan  $(N \times N)$  → tidak perlu mengalikan satu per satu baru kemudian dijumlahkan
-

# Contoh mean filtering

---



→

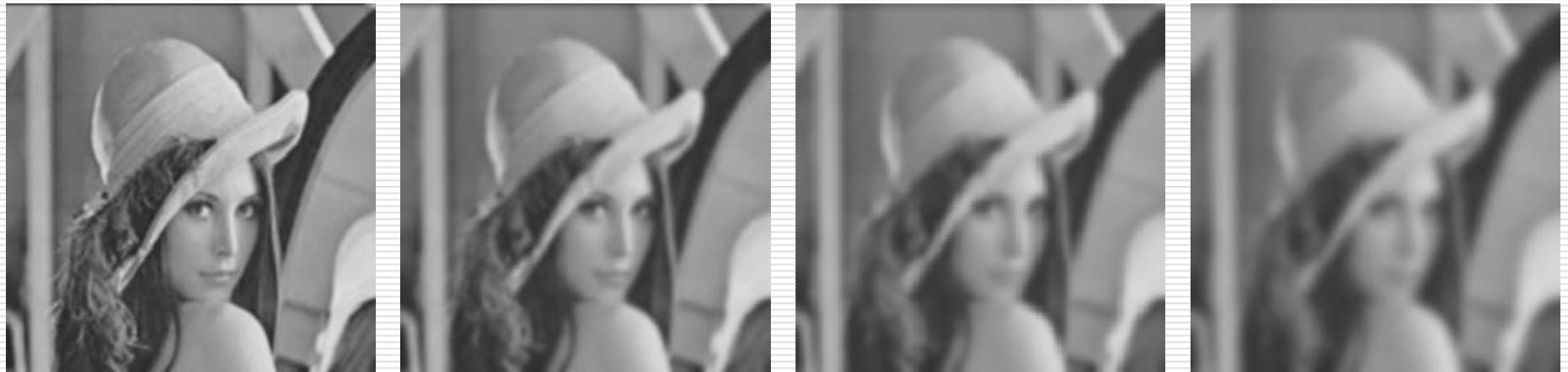
0,11	0,11	0,11
0,11	0,11	0,11
0,11	0,11	0,11



# Tingkat *blurring*

---

- Tingkat *blurring* yang didapat pada citra hasil sebanding dengan ukuran matriks ketetanggaan yang digunakan



---

Ukuran matriks ketetanggaan semakin besar

# *Thresholding*

---

- ❑ Mengurangi efek blurring pada pixel sisi

$$g(x, y) = \begin{cases} \frac{1}{M} \sum_{(m,n) \in S} f(m, n) & \text{if } \left| f(x, y) - \frac{1}{M} \sum_{(m,n) \in S} f(m, n) \right| < T \\ f(x, y) & \text{otherwise} \end{cases}$$

---

# Masalah pada mean filtering

---

- Masalah dengan mean filtering: cara mempertahankan sisi atau detil tepi
  - Alternatif solusi: penggunaan threshold
  - Masalah baru: penentuan threshold
    - Mungkin perlu trial and error
  - Alternatif lain: median filtering
-

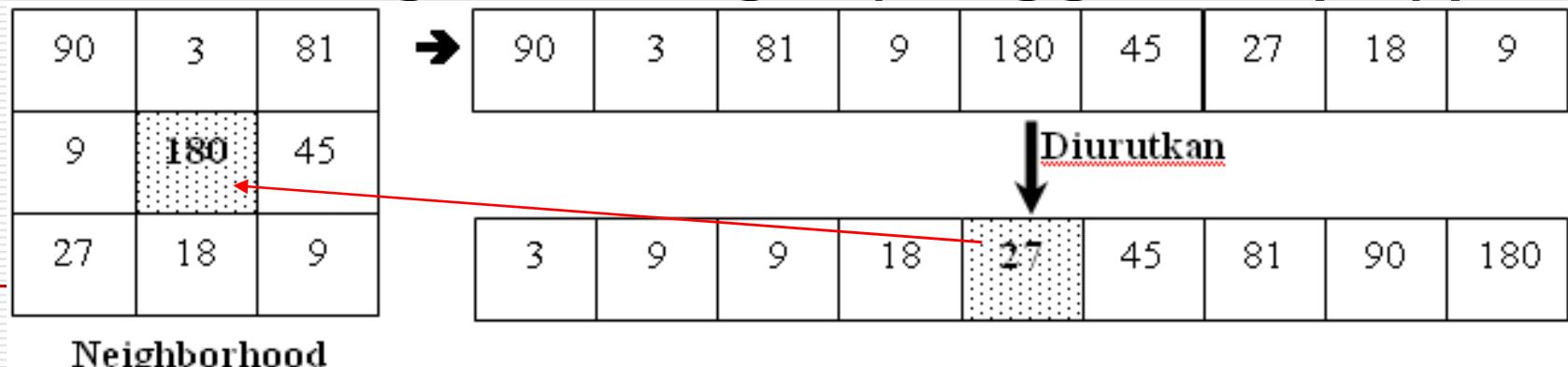
# Median filtering

---

- Ide: nilai median dari pixel-pixel dalam matriks ketetanggaan digunakan sebagai warna pixel  $f(x,y)$
  - Metode ini tepat untuk menghilangkan noise yang bersifat spike sementara diinginkan untuk tetap mempertahankan ketajaman sisi
-

# Mencari median

- Masukkan nilai-nilai dalam matriks neighborhood dalam matriks satu dimensi
- Urutkan nilai dalam matriks 1 dimensi tsb
- Nilai tengah sebagai pengganti  $f(x,y)$



# Contoh median filtering

---



Median filtering  
dgn mask  $3 \times 3$



---

# Modus filtering

---

- ☐ Ide: warna yang paling banyak muncul dalam matriks ketetanggaan digunakan sebagai warna  $f(x,y)$

90	3	81
9	180	45
27	18	9

Neighborhood

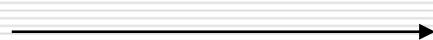
---

→ Nilai yang paling sering muncul = 9

→ Warna  $f(x,y)$  diubah dari 180 menjadi 9

# Contoh modus filtering

---



Modus filtering  
dgn mask  $5 \times 5$



---

**3 x 3**



**5 x 5**



**7 x 7**



**Mean  
filtering**



**Median  
filtering**

**Citra asli**



**Modus  
filtering**

**3 x 3**



**5 x 5**



**7 x 7**

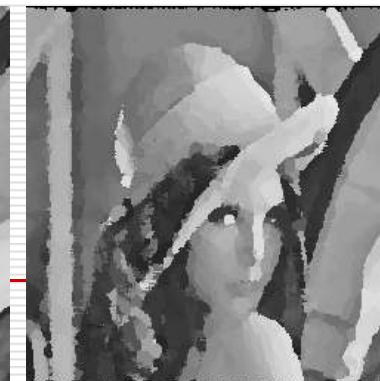


**Mean  
filtering**



**Median  
filtering**

**Citra asli**



**Modus  
filtering**

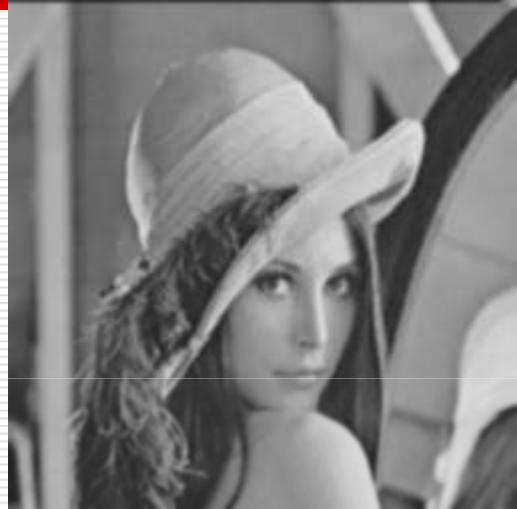
# Low-pass filtering:

Blurring pada domain frekwensi

---

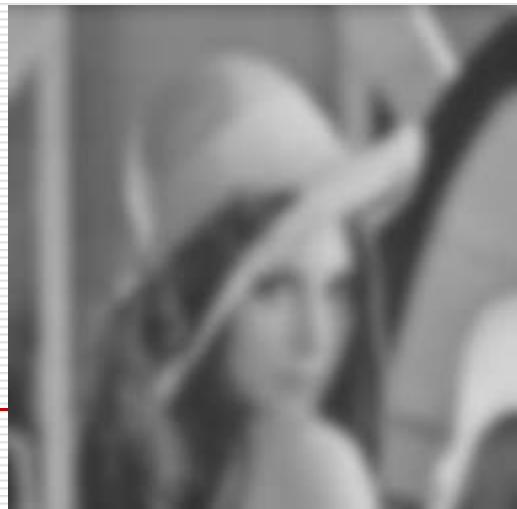
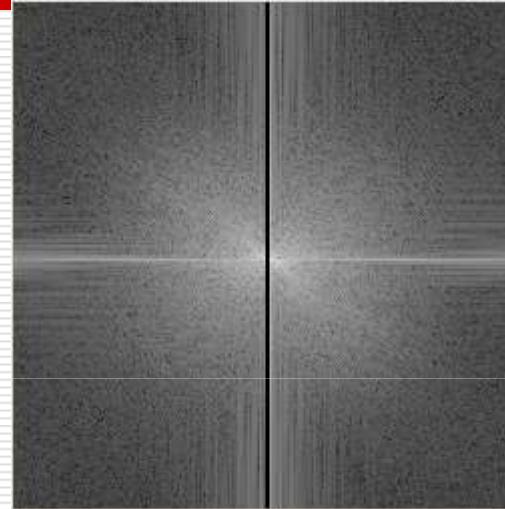
- Sisi dan transisi tajam lain (misal: noise) pada gray level dari suatu citra berkontribusi terhadap frekwensi tinggi pada transformasi Fourier
  - Blurring dapat dilakukan dengan 'menyaring' (menghilangkan) frekwensi tinggi
-

# Contoh FFT citra asli & citra blur



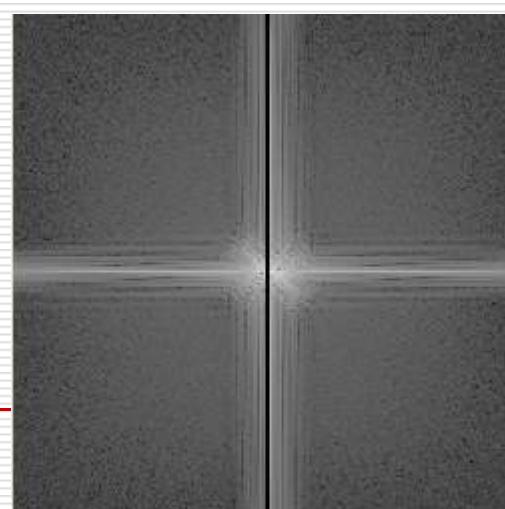
→

FFT



→

FFT



# Image Sharpening

---

- Teknik sharpening biasa digunakan untuk memperjelas sisi pada citra
  - Teknik sharpening
    - Di domain spasial (contoh: differentiation)
    - Di domain frekwensi (contoh: high-pass filter)
-

# Sharpening dengan differentiation

---

- Averaging → integrasi; sharpening → turunan (differentiation)
- Metode differentiation yang sering digunakan: *gradient*
- Diberikan fungsi  $f(x,y)$ , gradient dari  $f$  pada  $(x,y)$  didefinisikan dengan vektor **G**:

$$\mathbf{G}[f(x,y)] = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad G[f(x,y)] = mag[\mathbf{G}] = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

---

# Salah satu pendekatan gradien untuk proses digital

---

$$G[f(x, y)] \approx$$

$$\left\{ [f(x, y) - f(x + 1, y)]^2 + [f(x, y) - f(x, y + 1)]^2 \right\}^{1/2}$$

atau

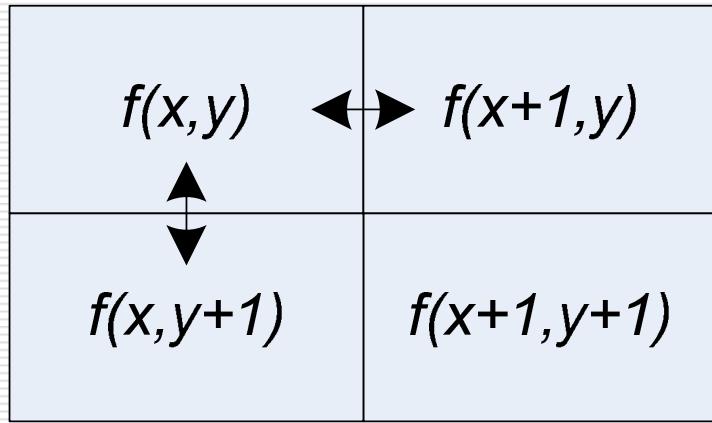
$$G[f(x, y)] \approx$$

$$|f(x, y) - f(x + 1, y)| + |f(x, y) - f(x, y + 1)|$$

---

# Ilustrasi & kelemahan

---



- ◻ Untuk citra  $N \times N$  pixel, tidak mungkin didapat gradien untuk pixel-pixel pada baris maupun kolom terakhir
-

# Pendekatan lain: *Roberts gradient*

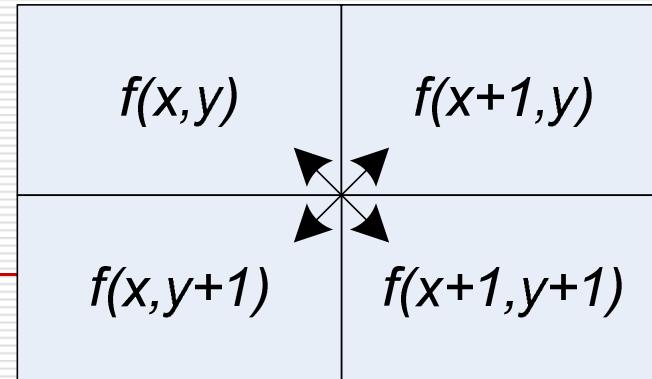
$$G[f(x, y)] \cong$$

$$\left\{ [f(x, y) - f(x+1, y+1)]^2 + [f(x+1, y) - f(x, y+1)]^2 \right\}^{1/2}$$

atau

$$G[f(x, y)] \cong$$

$$|f(x, y) - f(x+1, y+1)| + |f(x+1, y) - f(x, y+1)|$$



# Nilai gradien

---

- Proporsional dengan perbedaan nilai gray level antar pixel yang bertetangga
  - Nilai tinggi untuk sisi (warna berbeda dengan tajam)
  - Nilai kecil untuk daerah yang relatif sama warnanya



Citra asli



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

# Variasi pendekatan untuk $g(x,y)$

$$g(x,y) = \begin{cases} G[f(x,y)]; & G[f(x,y)] \geq T \\ f(x,y); & G[f(x,y)] < T \end{cases}$$

**Masalah:** penentuan nilai T yang tepat shg tepi dapat dipertajam tanpa merusak pixel-pixel non-tepi

$$g(x,y) = \begin{cases} L_G; & G[f(x,y)] \geq T \\ f(x,y); & G[f(x,y)] < T \end{cases}$$

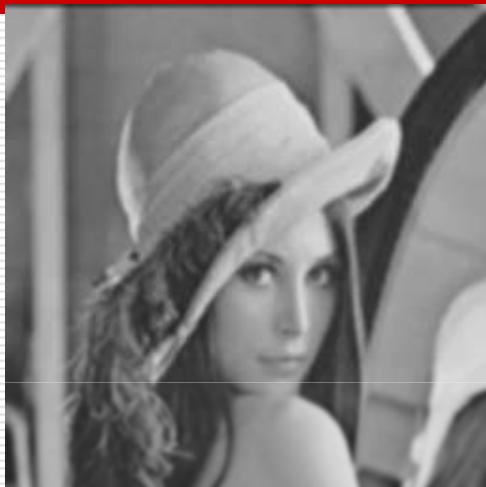
**L<sub>G</sub>:** Nilai gray level tertentu untuk mewakili pixel-pixel tepi

$$g(x,y) = \begin{cases} G[f(x,y)]; & G[f(x,y)] \geq T \\ L_B; & G[f(x,y)] < T \end{cases}$$

**L<sub>B</sub>:** Nilai gray level tertentu untuk mewakili pixel-pixel non-tepi

$$g(x,y) = \begin{cases} L_G; & G[f(x,y)] \geq T \\ L_B; & G[f(x,y)] < T \end{cases}$$

# Contoh sharpening



# High-pass filtering:

Sharpening pada domain frekwensi

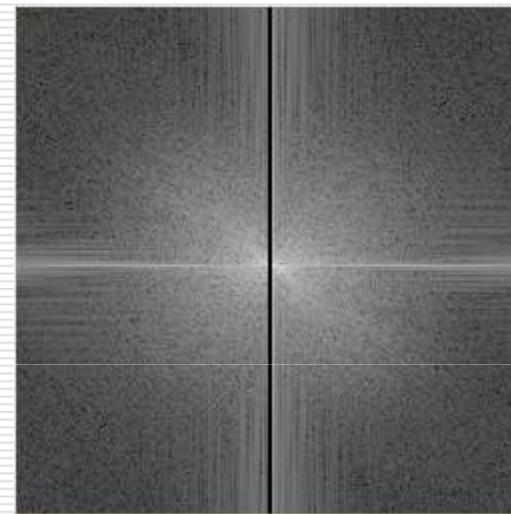
---

- Sisi dan transisi tajam lain (misal: noise) pada gray level dari suatu citra berkontribusi terhadap frekwensi tinggi pada transformasi Fourier
  - Sharpening dapat dilakukan dengan 'menyaring' (menghilangkan) frekwensi rendah
-

# Contoh FFT citra asli & citra sharpened



FFT



FFT

