

3. Relasi Antar Piksel

Outline

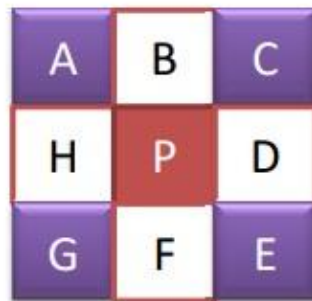
- Konsep Operasi Ketetanggaan
- Aplikasi Operasi Ketetanggaan pada Filtering
 - Filter batas
 - Filter mean
 - Filter Median

Konsep Piksel Tetangga

- Kernel menggunakan konsep piksel tetangga (*neighbouring pixels*), di mana matriks kernel dibuat dengan asumsi bahwa nilai sebuah piksel bisa dipengaruhi oleh piksel-piksel tetangganya
- Piksel tetangga adalah sejumlah piksel yang bersebelahan langsung (*adjacent*) dengan sebuah piksel pusat.



4-tetangga +



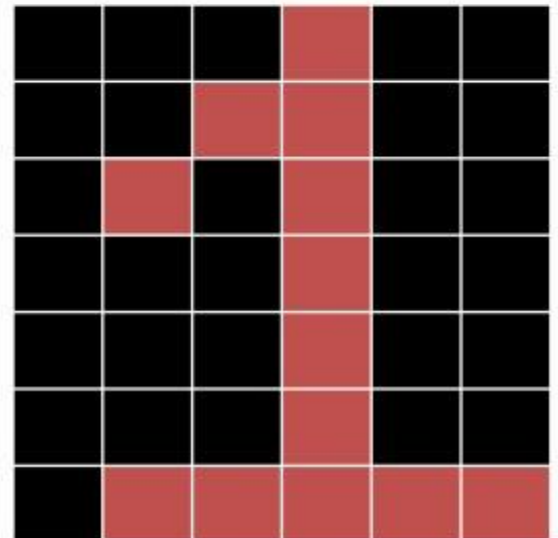
4-tetangga x



8-tetangga

Konsep Operasi Ketetanggaan

- Setiap piksel pada citra saling terkait
 - Dibuat algoritma mengolah citra berdasar tetangganya
- Operasi ketetanggaan piksel adalah operasi pengolahan citra untuk mendapatkan nilai suatu piksel yang melibatkan nilai piksel-piksel tetangganya.



Konvolusi Diskret

- Konvolusi diskret dapat didefinisikan sebagai operasi “geser dan kalikan” (*‘shift and multiply’*), di mana kita menggeser kernel sepanjang permukaan citra dan mengalikan elemennya dengan nilai intensitas piksel ybs. pada citra
- Untuk kernel persegi dengan ukuran $M \times M$, kita dapat menghitung citra outputnya dengan rumus berikut:

$$g(i, j) = \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \sum_{n=-\frac{M}{2}}^{\frac{M}{2}} h(m, n) f(i - m, j - n)$$

Konvolusi Diskret - Contoh

- Citra input:

5	5	6	6
5	4	4	7
0	0	2	2
0	1	1	3

 Kernel:

$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
- Maka hasil perhitungan konvolusi daerah di atas adalah:
$$g(2,2) = 5 \times \frac{1}{4} + 5 \times \frac{1}{4} + 6 \times \frac{1}{4} + 5 \times \frac{1}{4} + 4 \times \frac{1}{4} + 4 \times \frac{1}{4} + 0 \times \frac{1}{4} + 0 \times \frac{1}{4} + 2 \times \frac{1}{4}$$

Konvolusi Diskret - Contoh

- Konvolusi dilakukan sampai semua piksel citra input terkena perhitungan konvolusi

5	5	6	6
5	4	4	7
0	0	2	2
0	1	1	3

5	5	6	6
5	4	4	7
0	0	2	2
0	1	1	3

5	5	6	6
5	4	4	7
0	0	2	2
0	1	1	3

5	5	6	6
5	4	4	7
0	0	2	2
0	1	1	3

- Piksel yang berubah pada citra output:

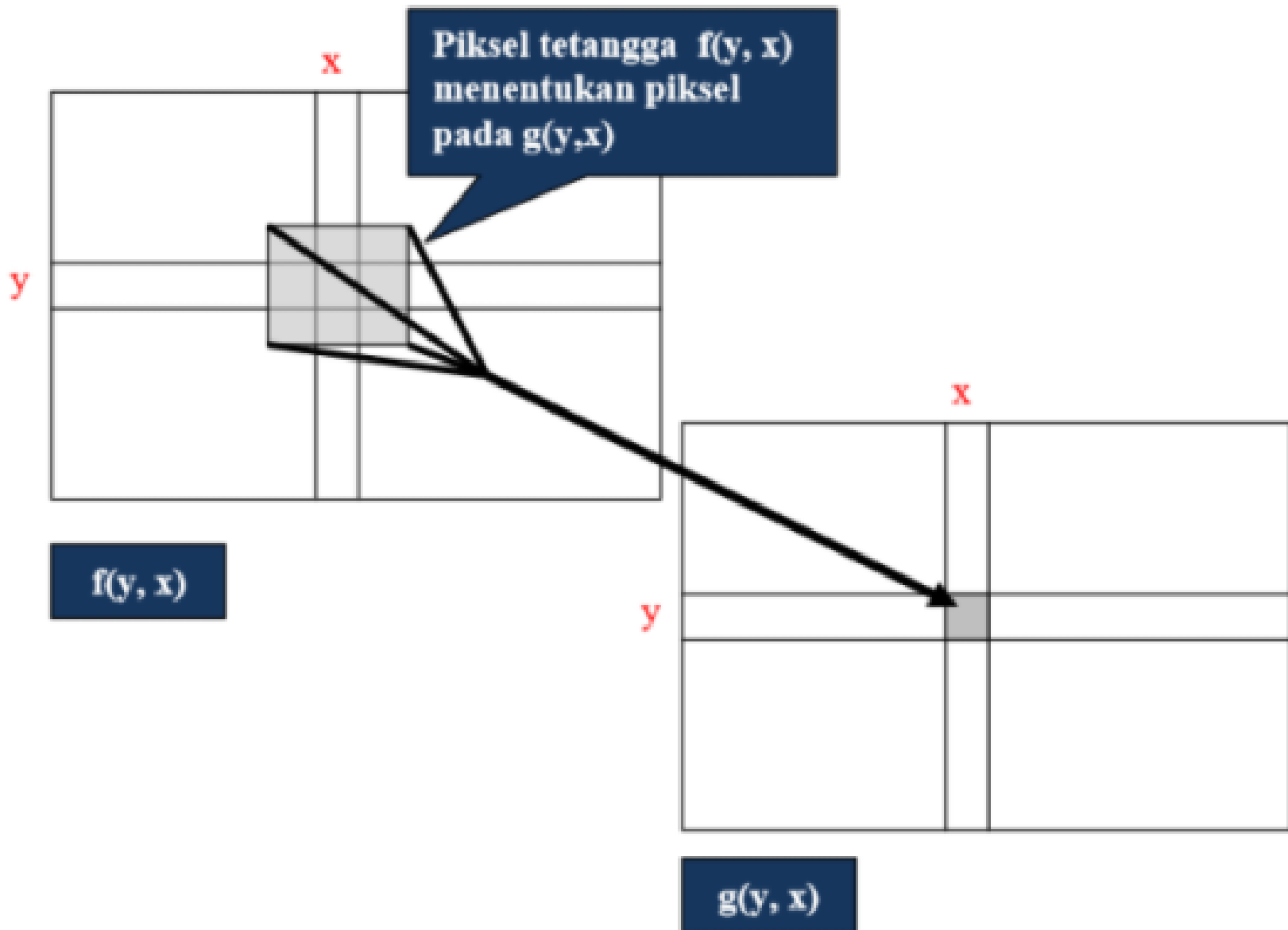
5	5	6	6
5	?	?	7
0	?	?	2
0	1	1	3

Operasi Ketetanggaan Piksel

PENGERTIAN OPERASI KETETANGGAAN PIKSEL

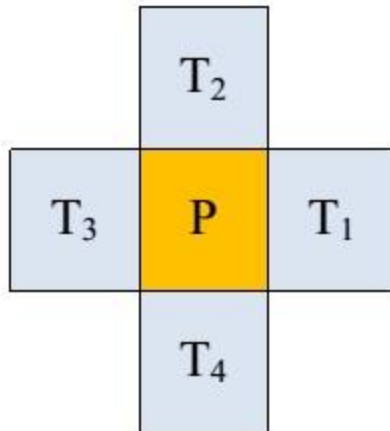
- Operasi ketetanggaan piksel adalah operasi pengolahan citra untuk mendapatkan nilai suatu piksel yang melibatkan nilai piksel-piksel tetangganya.
- Hal ini didasarkan kenyataan bahwa setiap piksel pada umumnya tidak berdiri sendiri, melainkan terkait dengan piksel tetangga, karena merupakan bagian suatu objek tertentu di dalam citra. Sifat inilah yang kemudian mendasari timbulnya algoritma untuk mengolah setiap piksel citra melalui piksel-piksel tetangga.
- Sebagai contoh, suatu citra yang berderau dapat dihaluskan melalui pererataan atas piksel-piksel tetangga.
- Gambar 1 memberikan ilustrasi operasi ketetanggaan piksel. Delapan piksel tetangga terdekat dengan piksel $f(y,x)$ digunakan untuk memperbaikinya menjadi $g(y,x)$ di tempat yang sama.

Gambar 1. Operasi ketetanggaan piksel. Sejumlah tetangga menentukan nilai sebuah piksel

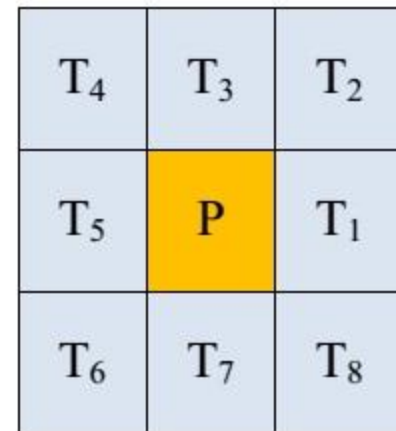


Jenis Ketetanggaan

4-ketetanggaan



8-ketetanggaan



Bila P mempunyai koordinat (b, k) dengan b baris dan k kolom

Pada 4-ketetanggaan

$$T_1 = (b, k + 1), T_2 = (b - 1, k), T_3 = (b, k - 1), T_4 = (b + 1, k)$$

Pada 8-ketetanggaan

$$T_1 = (b, k + 1), T_2 = (b - 1, k - 1),$$

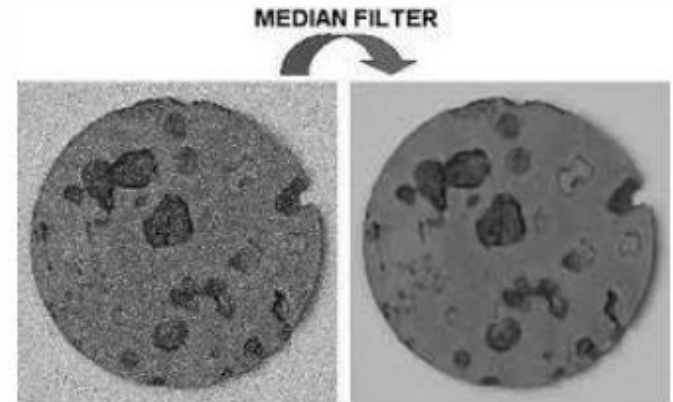
$$T_3 = (b, k - 1), T_4 = (b - 1, k - 1)$$

$$T_5 = (b, k - 1), T_6 = (b + 1, k - 1),$$

$$T_7 = (b + 1, k - 1), T_8 = (b + 1, k + 1)$$

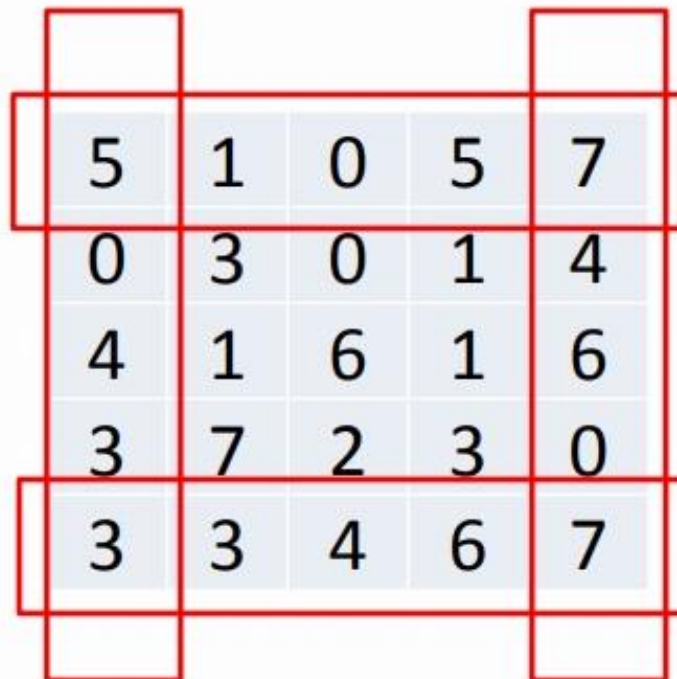
Penggunaan Operasi Ketetanggaan

- Proses Filtering
 - Mengurangi noise pada citra
- Beberapa jenis filtering
 - Filter batas
 - Filter mean
 - Filter Median



Permasalahan Pada Operasi Ketetanggaan

- Bagaimana menentukan tetangga untuk piksel yang berada di bagian tepi?
- Contoh citra 5x5 berikut



	5	1	0	5	7	
	0	3	0	1	4	
	4	1	6	1	6	
	3	7	2	3	0	
	3	3	4	6	7	

Permasalahan Pada Operasi



Ketetanggaan

Solusi:

1. Tidak merubah nilai pada kolom/baris tepi
2. Menambahkan satu kolom/baris dengan salah satu dari 2 nilai berikut:
 - Kolom/baris berisi nilai 0
 - Kolom/baris berisi nilai yang sama dengan kolom/baris tepi

Contoh Solusi 2.1

Menambahkan kolom/baris berisi nilai 0

5	1	0	5	7
0	3	0	1	4
4	1	6	1	6
3	7	2	3	0
3	3	4	6	7



0	0	0	0	0	0	0
0	5	1	0	5	7	0
0	0	3	0	1	4	0
0	4	1	6	1	6	0
0	3	7	2	3	0	0
0	3	3	4	6	7	0
0	0	0	0	0	0	0

Contoh Solusi 2.2

Kolom/baris berisi nilai yang sama

5	1	0	5	7
0	3	0	1	4
4	1	6	1	6
3	7	2	3	0
3	3	4	6	7



5	5	1	0	5	7	7
5	5	1	0	5	7	7
0	0	3	0	1	4	4
4	4	1	6	1	6	6
3	3	7	2	3	0	0
3	3	3	4	6	7	7
3	3	3	4	6	7	7

Aplikasi Operasi Ketetanggaan pada Filtering

Filter Batas

- Ide dasar: mencegah piksel yang intensitasnya di luar intensitas piksel-piksel tetangga

Algoritma Filter Batas

Masukan:

$f(y, x)$: Piksel pada posisi (y, x)

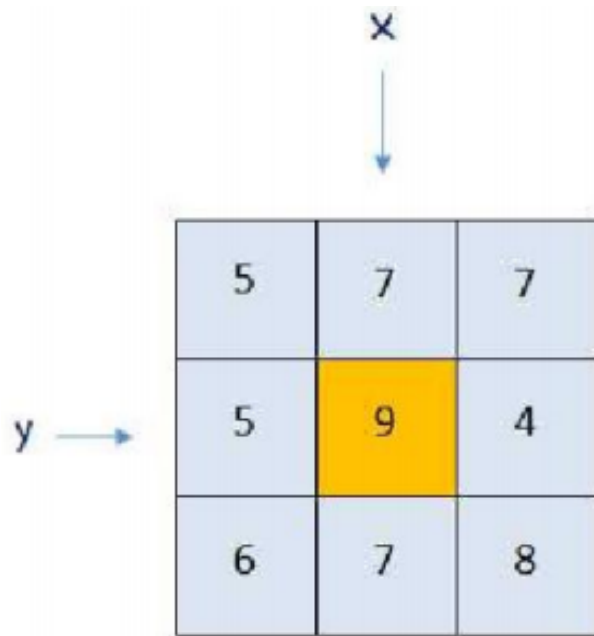
Keluaran:

$g(y, x)$: Nilai intensitas untuk piksel pada citra g pada posisi (y, x)

Algoritma:

1. Carilah nilai intensitas terkecil pada tetangga $f(y, x)$ dengan menggunakan 8-ketetanggan dan simpan pada minInt .
2. Carilah nilai intensitas terbesar pada tetangga $f(y, x)$ dengan menggunakan 8-ketetanggan dan simpan pada maksInt .
3. IF $f(y, x) < \text{minInt}$
4. $g(y, x) \leftarrow \text{minInt}$
5. ELSE IF $f(y, x) > \text{maksInt}$
6. $g(y, x) \leftarrow \text{maksInt}$
7. ELSE
8. $g(y, x) \leftarrow f(y, x)$
9. END-IF

Contoh



	5	7	7
y →	5	9	4
	6	7	8

- $\text{minInt} = \text{minimum}(5, 7, 7, 5, 4, 6, 7, 8) = 4$;
- $\text{maksInt} = \text{maksimum}(5, 7, 7, 5, 4, 6, 7, 8) = 8$;
- mengingat $f(y, x)$ bernilai 9 dan lebih besar daripada 8 (maksInt) maka $g(y, x)$ bernilai 8;
- seandainya $f(y, x)$ pada keadaan di atas bernilai 2 (bukan 9), $g(y, x)$ akan bernilai 4.

Contoh



(a) Citra mobil yang telah diberi bintik-bintik putih



(b) Hasil pemfilteran gambar (a)



Filter Mean

- Diberikan (bagian) citra $N \times N$ pixel: $f(x,y) \rightarrow$ frame
- Citra hasil: $g(x,y) \rightarrow$ merata-ratakan nilai gray level pixel-pixel pada $f(x,y)$ yang termasuk dalam area ketetanggaan (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 \rightarrow titik (x,y) bisa berada di tengah matriks

1	2	3
4	T	5
6	7	8

3x3

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

5x5

19

Contoh Mean Filtering

- Diberikan citra 5x5 piksel berikut, hitung nilai mean filtering dengan $M = 3 \times 3$

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

Contoh Mean Filtering

- Solusi: tentukan titik tengah dan rubah dengan nilai mean dari tetangga2nya

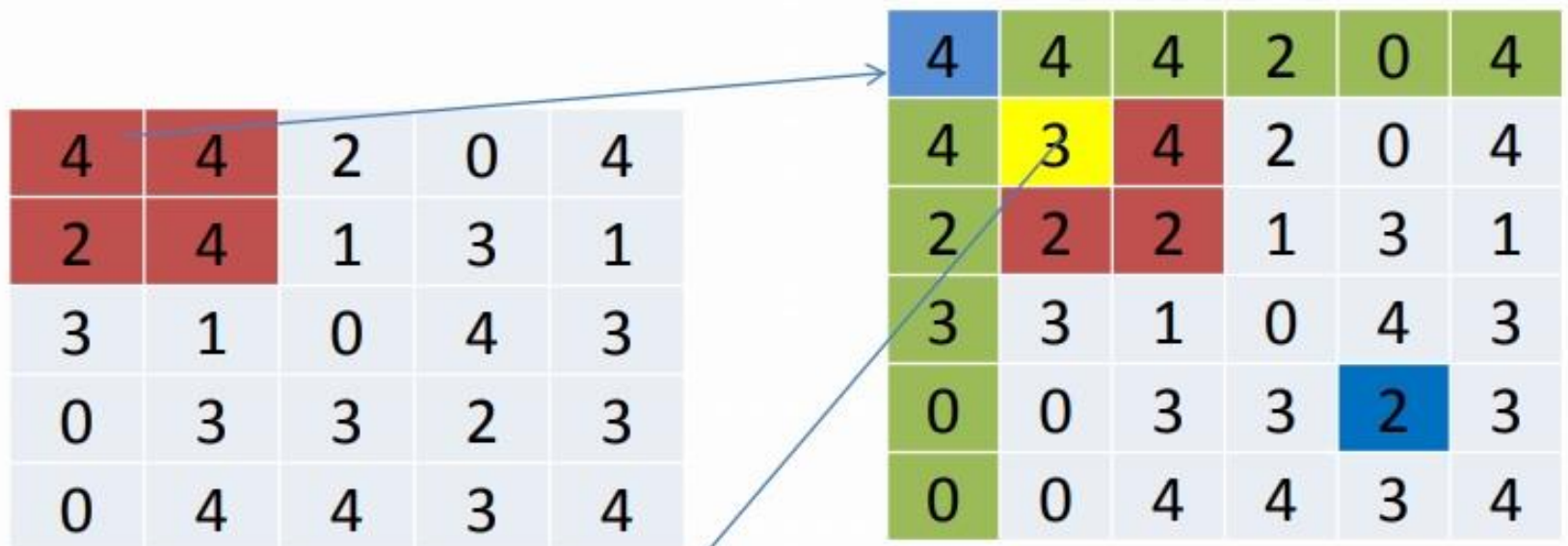
4	4	2	0	4
2	4	1	3	1
3	1	0	4	3
0	3	3	2	3
0	4	4	3	4

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

$$g(1,1) = (4+4+2+2+4+1+3+1+0)/9 = 2$$

Contoh Mean Filtering

- Contoh menghitung nilai border (titik $g(0,0)$) dengan menambahkan nilai border



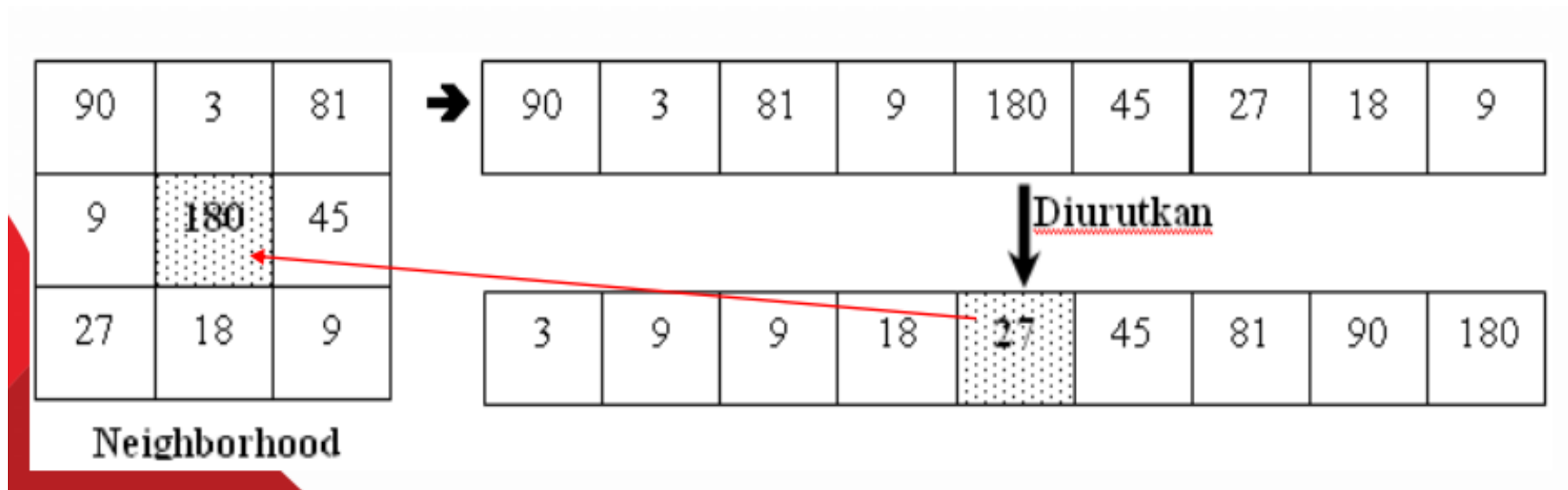
$$g(1,1) = (4+4+4+4+3+4+2+2+2)/9 = 3$$

Median filtering

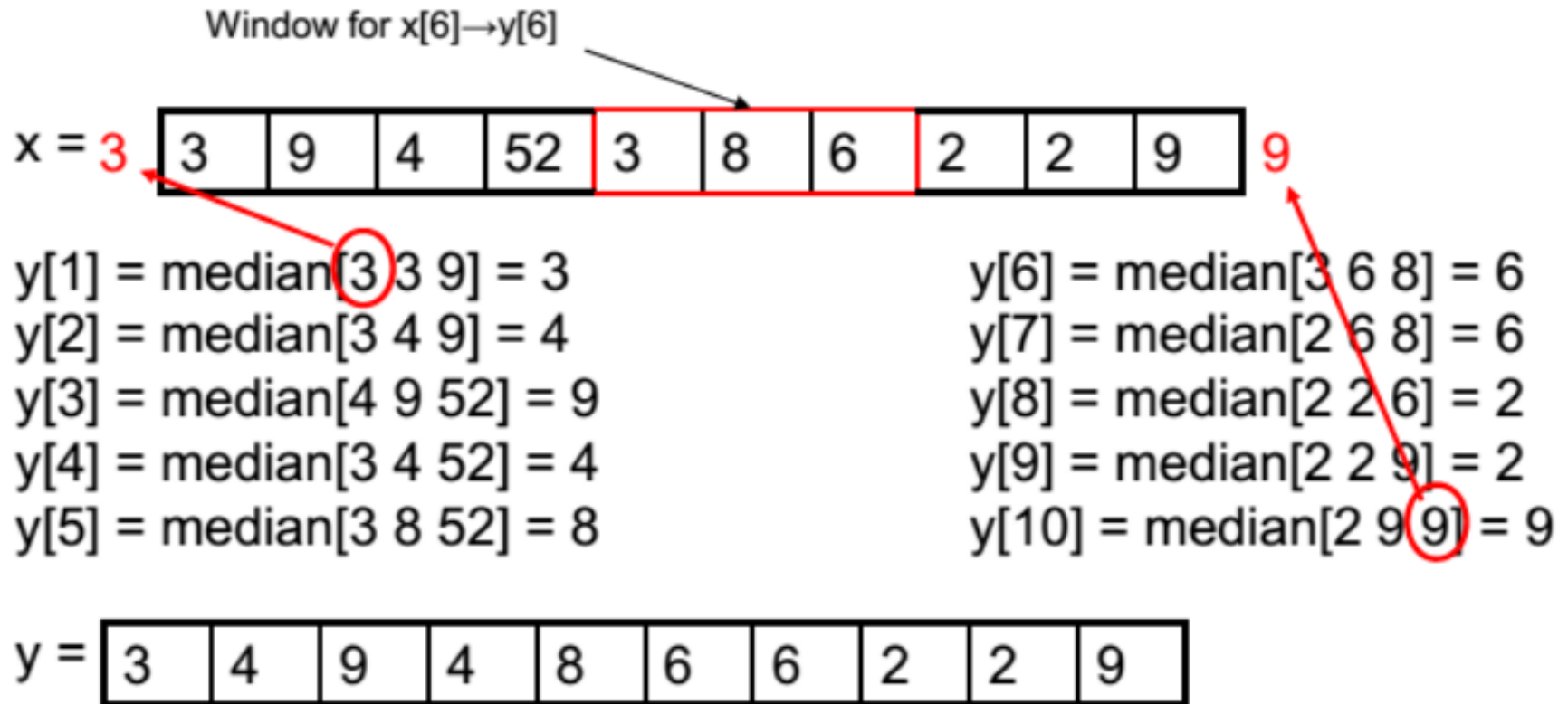
- Ide: nilai median dari pixel-pixel dalam matriks ketetanggaan digunakan sebagai nilai pixel $f(x,y)$
- Metode ini tepat untuk menghilangkan noise yang bersifat spike sementara – tetap mempertahankan ketajaman sisi

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 pada 1D



Median Filtering pada 2D

2D Median filtering example using a 3 x 3 sampling window:

Keeping border values unchanged

Sorted: 0,0,1,1,1,2,2,4,4

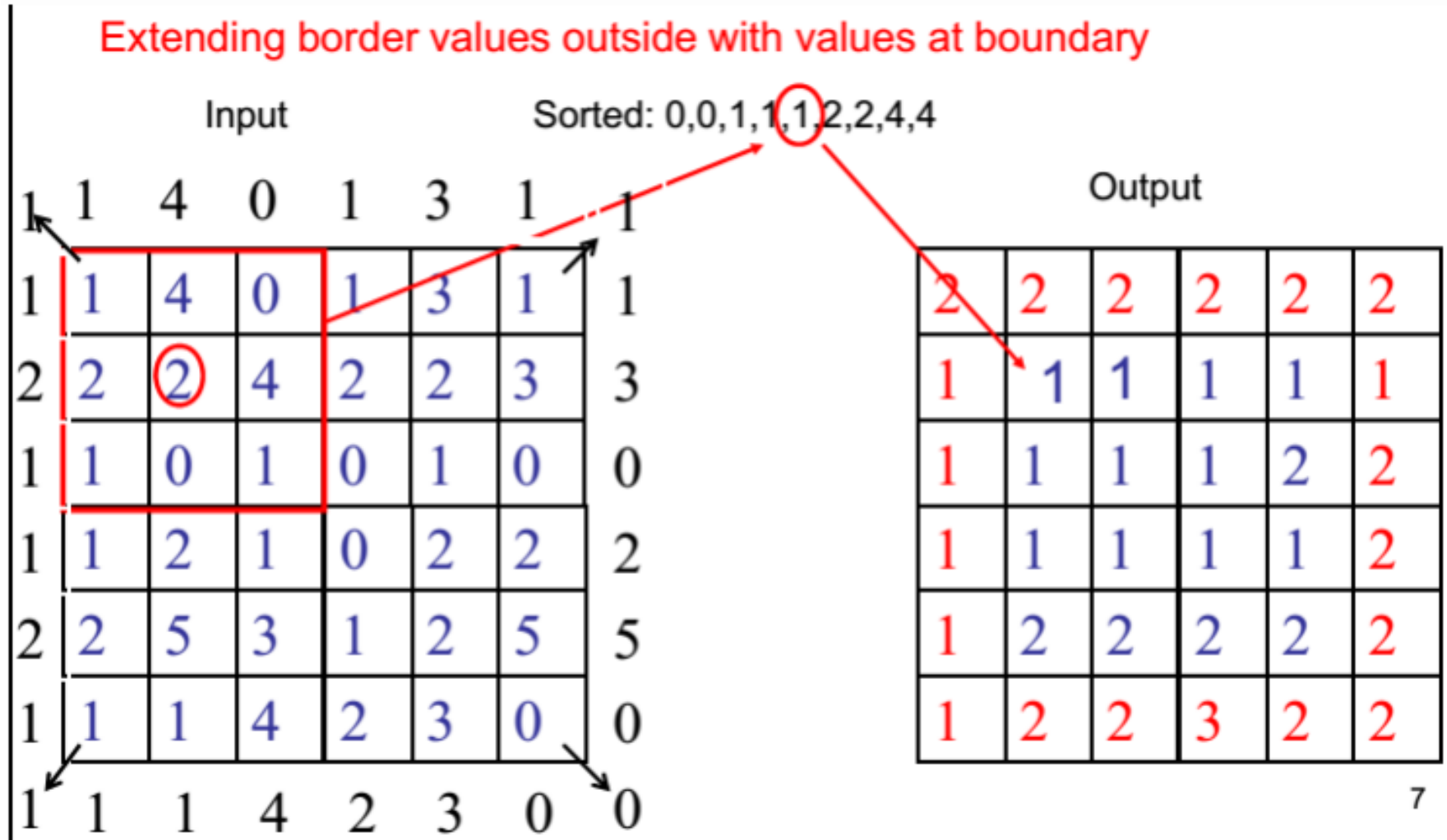
Input

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

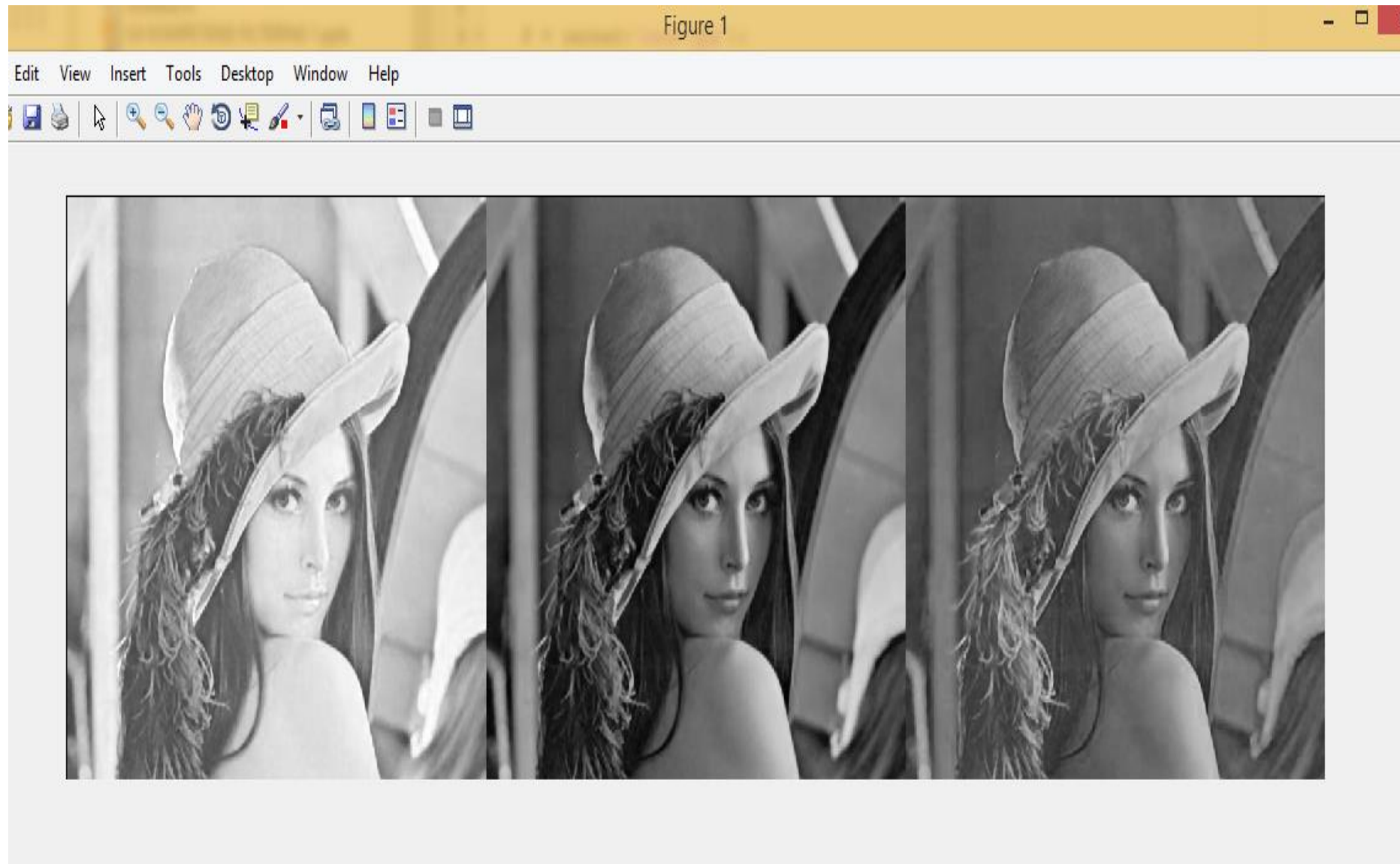
Output

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

Median Filtering pada 2D



Contoh Median Filtering



Modus filtering

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

90	3	81
9	180	45
27	18	9

Neighborhood

→ Nilai yang paling sering muncul = 9

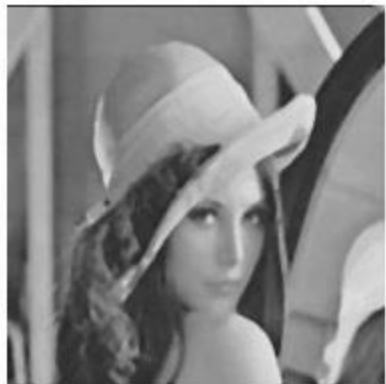
→ Warna $f(x,y)$ diubah dari 180 menjadi 9



Citra asli



**Mean
filtering**



**Median
filtering**



**Modus
filtering**

Latihan

Hitunglah nilai-nilai piksel yang diberi tanda merah dengan menggunakan:

- Filter Batas
- Filter Mean (Ukuran $M = 3 \times 3$)
- Filter Median (Ukuran $M = 3 \times 3$)

Untuk piksel tepi, gunakan ketiga metode yang telah disebutkan di atas

index	0	1	2	3	4	5	6	7	8
0	7	10	2	13	10	11	5	8	15
1	14	6	1	11	6	9	13	6	14
2	2	1	5	11	1	15	15	6	5
3	7	11	15	9	4	8	7	15	14
4	11	1	13	14	9	5	8	12	9
5	5	6	1	3	3	7	5	12	9
6	10	7	5	14	8	4	6	15	3
7	2	6	5	4	11	12	6	14	10
8	11	5	7	1	6	1	14	15	9