

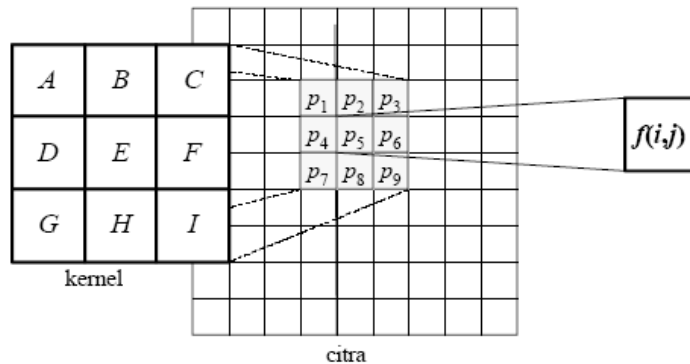
## D. MENINGKATKAN KUALITAS CITRA-OPERASI SPASIAL

### Judul PRAKTEK D1- KONVOLUSI 2D

#### Deskripsi

Konvolusi 2D merupakan suatu proses untuk memperoleh suatu piksel berdasarkan pada nilai piksel itu sendiri dan ketetanggannya dengan melibatkan suatu matriks kernel yang mempresentasikan pembobotan (**Kadir & Susanto, 2013**).

Proses konvolusi matrik piksel citra dengan matriks kernel dapat diilustrasikan sebagai berikut ini (**Munir, 2004**):



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

Gambar 20. Ilustrasi Konvolusi

Berbeda dengan koordinat piksel citra yang dimulai (0,0) dari ujung kiri atas, koordinat kernel ditunjukkan pada Gambar di bawah ini

|            |          |            |
|------------|----------|------------|
| $i-1, j-1$ | $i-1, j$ | $i-1, j+1$ |
| $i, j-1$   | $i, j$   | $i, j+1$   |
| $i+1, j-1$ | $i+1, j$ | $i+1, j+1$ |

Gambar 21. Koordinat kernel

Estimasi waktu 20 menit

#### Prerequisite

Siapkan 5 buah citra yang telah diedit menggunakan tools editing image seperti Adobe Photoshop, atur sehingga citra tersebut memiliki noise seperti salt & papper, Spike dll

## Alur Proses

Masukkan :

X: Citra yang akan dikonvolusi

F: Kernel Konvolusi

Keluaran:

Out: citra hasil keluaran

1. Baca ukuran tinggi dan lebar citra
2. Baca ukuran tinggi dan lebar kernel
3.  $H = \text{ukuran tinggi kernel} / 2$
4.  $W = \text{ukuran lebar kernel} / 2$
5. For i:  $H+1$  to  $\text{ukuran\_tinggi\_citra} - H$   
    For j:  $W+1$  to  $\text{ukuran\_lebar\_citra} - W$   
        #lakukan konvolusi  
        Sum=0  
        For k:  $-H$  to  $H$   
            For l:  $-W$  to  $W$   
                 $a = X[i+k, j+l]$   
                 $w = F[H+k, W+l]$   
                 $\text{sum} = \text{sum} + (w * a)$   
            End for  
        End for  
         $\text{out}[i, j] = \text{sum}$   
    End For  
End For

- 
1. Untuk memanggil fungsi konvolusi di GUI buatlah menu Filtering dibawah struktur menu Operasi Spasial dengan objectname 'actionFilter'
  2. Inisialisasi menu filtering di main.python
  3. Panggil fungsi konvolusi dengan cara import function
  4. Buat prosedur menu filtering

```
def filteringClicked(self):  
    ubah citra masukan menjadi grayscale  
    kernel = array piksel kernel  
  
    img_out = fungsi_konvolusi(citra masukan, kernel)  
  
    plt.imshow(img_out, cmap='gray', interpolation='bicubic')  
    plt.xticks([], plt.yticks([]))  
    plt.show()
```

---

## Tugas

1. Implementasikan pseudo code diatas menjadi fungsi Konvolusi yang terpisah file python dan disimpan dalam satu direktori dengan project utama
2. Ujikan fungsi konvolusi diatas dengan memanggil fungsi tersebut dengan kernel sbb:
  - a)  $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
  - b)  $\begin{bmatrix} 6 & 0 & -6 \\ 6 & 1 & -6 \\ 6 & 0 & -6 \end{bmatrix}$
3. Analisis hasil pengujian diatas
4. Analisis citra dari perubahan piksel citra sebelum dikonvolusi dan citra yang telah dikonvolusi!

## Judul

## PRAKTEK D2- IMAGE SMOOTHING USING MEAN FILTER

## Deskripsi

Operasi pelembutan dilakukan dengan mengganti intensitas suatu *pixel* dengan rata-rata dari nilai *pixel* tersebut dengan nilai *pixel-pixel* tetangganya (**Munir, 2004**).

Contoh penapis rerata yang berukuran 3 x 3 dan 2 x 2 adalah seperti di bawah ini

$$(i) \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad (ii) \begin{bmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{bmatrix}$$

Gambar 22. Kernel mean filter

## Estimasi waktu

15 menit

## Prerequisite

1. Asumsi telah membuat fungsi konvolusi
2. Citra noise sebanyak minimal 5 buah citra

## Tugas

1. Buat prosedur untuk menu pelembutan citra menggunakan mean filter
2. Analisis hasil program untuk kernel (i) dan kernel (ii)
3. Jelaskan mengapa proses mean filter memberikan efek blurring? Dapat dijelaskan menggunakan studi kasus proses konvolusi untuk melembutkan citra pada ukuran citra 6 x 6 dengan derajat keabuan 8 bit!
4. Analisis piksel citra berdasarkan aplikasi yang dibuat berdasarkan citra sebelum dikonvolusi dan yang setelah dikonvolusi!

|                       |   |
|-----------------------|---|
| <b>Judul</b>          | <b>PRAKTEK D3- IMAGE SMOOTHING USING GAUSSIAN FILTER</b>  |
| <b>Deskripsi</b>      | <p>Gaussian Filter digunakan untuk proses penghalusan citra, pengaburan, menghilangkan detail, menghilangkan noise (<b>Munir, 2004</b>).</p> $G_{(x,y)} = \frac{1}{2\pi\sigma^2} \exp(-(x^2 + y^2)/2\sigma^2) \quad (8)$ <p>Dimana <math>\sigma</math> adalah standar deviasi dan distribusi biasanya diatur <math>\sigma=1</math><br/> <math>X,y</math> adalah posisi koordinat pada kernel</p>                                      |
| <b>Estimasi waktu</b> | 20 menit  |
| <b>Prerequisite</b>   | <ol style="list-style-type: none"> <li>1. Asumsi telah membuat fungsi konvolusi</li> <li>2. Citra noise sebanyak minimal 5 buah citra</li> </ol>  |
| <b>Tugas</b>          | <ol style="list-style-type: none"> <li>1. Hitung bagaimana membentuk kernel Gaussian menggunakan Persamaan (8)</li> <li>2. Buat prosedur untuk menu pelembutan citra menggunakan Gaussian filter</li> <li>3. Analisis hasil program untuk beberapa jenis gambar yang memiliki noise</li> <li>4. Analisis piksel citra berdasarkan aplikasi yang dibuat berdasarkan citra sebelum dikonvolusi dan yang setelah dikonvolusi!</li> </ol> |

|                       |   |
|-----------------------|---|
| <b>Judul</b>          | <b>PRAKTEK D4- IMAGE SHARPENING</b>   |
| <b>Deskripsi</b>      | <p>Operasi penajaman citra bertujuan memperjelas tepi pada objek di dalam citra atau menghilangkan bagian citra yang lembut. Operasi penajaman dilakukan dengan melewati citra pada <b>penapis lolos-tinggi</b> (<i>high-pass filter</i>).</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;"> <math display="block">(i) \begin{bmatrix} -1 &amp; -1 &amp; -1 \\ -1 &amp; 8 &amp; -1 \\ -1 &amp; -1 &amp; -1 \end{bmatrix}</math> <math display="block">\Sigma = 0</math> </div> <div style="text-align: center;"> <math display="block">(ii) \begin{bmatrix} -1 &amp; -1 &amp; -1 \\ -1 &amp; 9 &amp; -1 \\ -1 &amp; -1 &amp; -1 \end{bmatrix}</math> <math display="block">\Sigma = 1</math> </div> <div style="text-align: center;"> <math display="block">(iii) \begin{bmatrix} 0 &amp; -1 &amp; 0 \\ -1 &amp; 5 &amp; -1 \\ 0 &amp; -1 &amp; 0 \end{bmatrix}</math> <math display="block">\Sigma = 1</math> </div> <div style="text-align: center;"> <math display="block">(iv) \begin{bmatrix} 1 &amp; -2 &amp; 1 \\ -2 &amp; 5 &amp; -2 \\ 1 &amp; -2 &amp; 1 \end{bmatrix}</math> <math display="block">\Sigma = 1</math> </div> <div style="text-align: center;"> <math display="block">(v) \begin{bmatrix} 1 &amp; -2 &amp; 1 \\ -2 &amp; 4 &amp; -2 \\ 1 &amp; -2 &amp; 1 \end{bmatrix}</math> <math display="block">\Sigma = 0</math> </div> <div style="text-align: center;"> <math display="block">(vi) \begin{bmatrix} 0 &amp; 1 &amp; 0 \\ 1 &amp; -4 &amp; 1 \\ 0 &amp; 1 &amp; 0 \end{bmatrix}</math> <math display="block">\Sigma = 0</math> </div> </div> <p style="text-align: center;">Gambar 23. Kernel low pass filter</p> |
| <b>Estimasi waktu</b> | 25 menit  |

## Prerequisite

1. Asumsi telah membuat fungsi konvolusi
2. Citra noise sebanyak minimal 5 buah citra
3. Tambahkan menu Sharpening image di GUI

## Tugas

1. Buat prosedur untuk menjalankan program penajaman citra
2. Analisis hasil citra dengan mencoba kernel-kernel citra pada Gambar.23
3. Analisis citra jika diberikan kernel filter Laplace berikut:  
$$(1.0 / 16) * \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$
4. Analisis piksel citra berdasarkan aplikasi yang dibuat berdasarkan citra sebelum dikonvolusi dan yang setelah dikonvolusi!

## Judul

## PRAKTEK D5- MEDIAN FILTER

## Deskripsi

Median filter berbeda dengan yang sudah dikerjakan pada praktek D1-D4 yang merupakan filterisasi secara linear, 3 modul praktek berikutnya merupakan proses filterisasi non linear.

Sebagai contoh, tinjau matriks citra dengan kelompok *pixel* (berbentuk kotak diarsir). Pixel yang sedang diproses adalah yang mempunyai intensitas 35. Langkah yang perlu dilakukan adalah mengurutkan *pixel-pixel* tersebut:

|    |    |    |    |    |
|----|----|----|----|----|
| 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 |

9 10 10 10 10 10 11 12 35

Median dari kelompok tersebut adalah 10

Hasil setelah 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  $3 \times 3$  *pixel*

Gambar 24. Proses median filter (sumber: (Putra, 2010))

|                       |  |
|-----------------------|--|
| <b>Estimasi waktu</b> | 15 menit   |
| <b>Prerequisite</b>   | <ol style="list-style-type: none"> <li>1. Asumsi telah membuat fungsi konvolusi</li> <li>2. Citra noise sebanyak minimal 5 buah citra</li> </ol>   |
| <b>Alur Proses</b>    | <ol style="list-style-type: none"> <li>1. Konversi citra ke grayscale dapat menggunakan fungsi <code>cv2.COLOR_BGR2GRAY</code></li> <li>2. <code>img_out= copy image</code></li> <li>3. <code>h=ukuran tinggi citra</code></li> <li>4. <code>w=ukuran baris citra</code></li> <li>5. <code>for i=3 to h-3</code> <ul style="list-style-type: none"> <li><code>for j=3 to w-3</code> <ul style="list-style-type: none"> <li><code>neighbors=[]</code></li> <li><code>for k=-3 to 4</code> <ul style="list-style-type: none"> <li><code>for l=-3 to 4</code> <ul style="list-style-type: none"> <li><code>a=image(i+k,j+l)</code></li> <li>menambahkan a ke neighbors</li> </ul> </li> </ul> </li> <li><code>end for</code></li> </ul> </li> <li><code>end for</code></li> </ul> </li> <li>mengurutkan neighbors</li> <li><code>median=posisi neighbors ke-24</code></li> <li>posisikan nilai piksel sebelumnya dengan nilai median</li> <li><code>end for</code></li> <li><code>end for</code></li> <li>tampilkan citra hasil median</li> </ol> |
| <b>TUGAS</b>          | <ol style="list-style-type: none"> <li>1. Implementasikan pseudo code diatas dengan membuat prosedur 'Median Filter'</li> <li>2. Lakukan pengujian dan analisis hasil pengujian terhadap minimal 5 buah citra noise yang berbeda!</li> <li>3. Analisis piksel citra berdasarkan aplikasi yang dibuat berdasarkan citra sebelum dan yang setelah difilter Median!</li> </ol>  |

## Judul PRAKTEK D6- MAX FILTERING

**Deskripsi** Maximum filter adalah proses menggantikan nilai piksel dengan nilai piksel maksimum yang dipengaruhi piksel area tetangga.

|                       |   |
|-----------------------|---|
| <b>Estimasi waktu</b> | 20 menit  |
| <b>Prerequisite</b>   | <ol style="list-style-type: none"> <li>1. Citra noise sebanyak minimal 5 buah citra</li> <li>2. Tambahkan menu 'Max Filter' di GUI</li> </ol>   |
| <b>Alur Proses</b>    | <ol style="list-style-type: none"> <li>1. Konversi citra ke grayscale dapat menggunakan fungsi <code>cv2.COLOR_BGR2GRAY</code></li> <li>2. <code>img_out= copy image</code></li> <li>3. <code>h=ukuran tinggi citra</code></li> <li>4. <code>w=ukuran baris citra</code></li> <li>5. <code>for i=3 to h-3</code> <ul style="list-style-type: none"> <li><code>for j=3 to w-3</code> <ul style="list-style-type: none"> <li><code>neighbors=[]</code></li> <li><code>for k=-3 to 4</code> <ul style="list-style-type: none"> <li><code>for l=-3 to 4</code> <ul style="list-style-type: none"> <li><code>baca piksel pada (i + k, j + l)</code></li> <li><code>if a &lt; min:</code> <ul style="list-style-type: none"> <li><code>min=a</code></li> </ul> </li> </ul> </li> <li><code>end for</code></li> </ul> </li> <li><code>end for</code></li> <li><code>img_out. itemset((i, j), b)</code></li> </ul> </li> <li><code>end for</code></li> </ul> </li> <li><code>tampilkan citra hasil max filter</code></li> <li><code>end</code></li> </ol> |
| <b>Tugas</b>          | <ol style="list-style-type: none"> <li>1. Buat prosedur untuk menjalankan program Maximum Filtering</li> <li>2. Buat prosedur untuk menampilkan citra hasil minimum filtering</li> <li>3. Analisis citra berdasarkan piksel sebelum dan setelah dilakukan proses maximum filtering</li> <li>4. Analisis citra berdasarkan piksel sebelum dan setelah dilakukan proses minimum filtering</li> </ol>  |