**PRAKTIKUM 5**

***IMAGE ENHANCEMENT* DENGAN *MASK PROCESSING***

**Materi**

* Mask Processing
* Konvolusi
* Filter Spasial

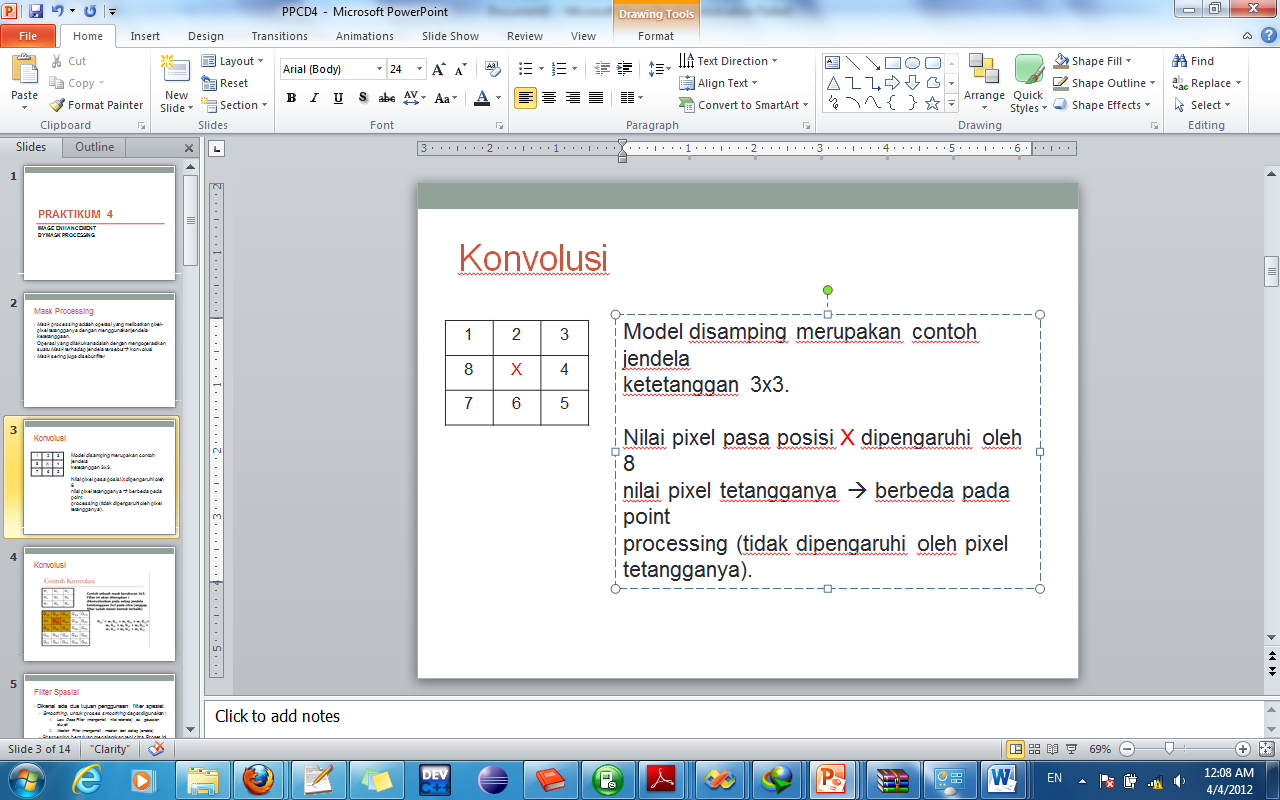
**Tujuan Praktikum**

* Mahasiswa dapat melakukan *enhancement* pada citra dengan *mask processing* menggunakan konvolusi matriks dan filter spasial.

1. **Penyajian**

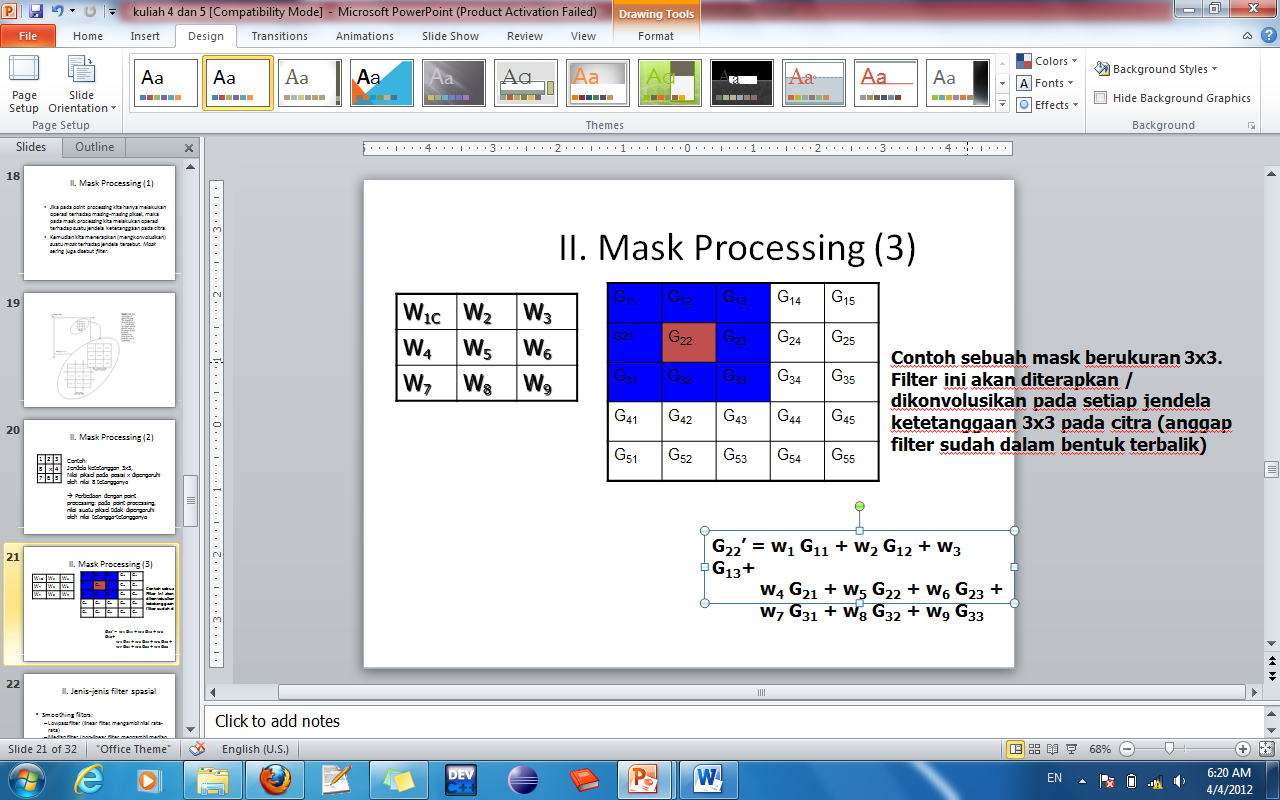
*Mask processing* adalah operasi yang melibatkan pixel-pixel tetangganya dengan menggunakan jendela ketetanggaan. Operasi yang dilakukan adalah dengan mengoperasikan suatu *Mask* terhadap jendela tersebut 🡪 konvolusi. *Mask* sering juga disebut *filter*.

Gambar 5.1 merupakan contoh jendela ketetanggan 3x3. Nilai pixel pada posisi X dipengaruhi oleh 8 nilai pixel tetangganya 🡪 berbeda pada *point processing* (tidak dipengaruhi oleh pixel tetangganya).

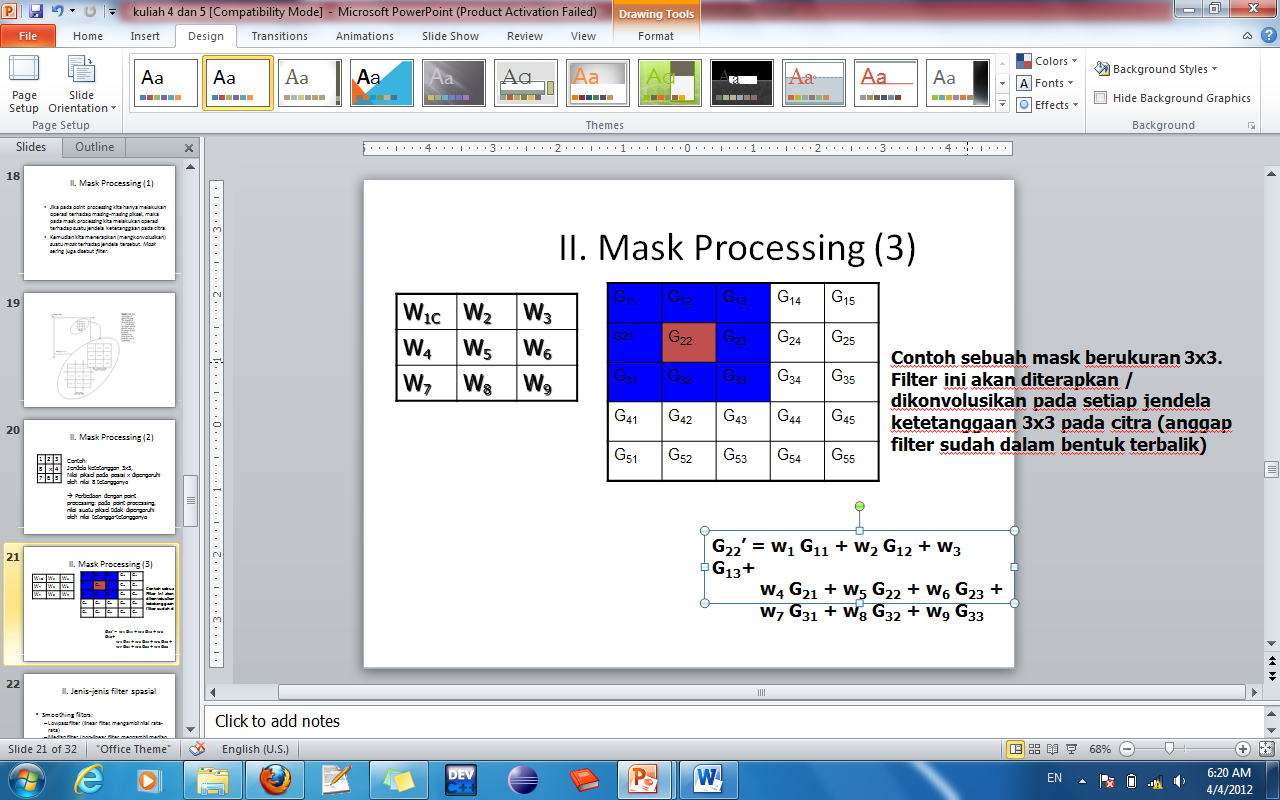


Gambar 5.1 Contoh jendela ketetanggan

Berikut pada Gambar 5.2, contoh sebuah *mask* berukuran 3x3. Filter ini akan diterapkan / dikonvolusikan pada setiap jendela ketetanggaan 3x3 pada citra (Gambar 5.3).



Gambar 5.2 *Mask* berukuran 3x3



Gambar 5.3 Contoh piksel citra

Dengan menggunakan rumus berikut, diperoleh nilai G22 yang baru hasil konvolusi.

G22’ = w1 G11 + w2 G12 + w3 G13+ w4 G21 + w5 G22 + w6 G23 + w7 G31 + w8 G32 + w9 G33

**Filter Spasial**

Dikenal ada dua tujuan penggunaan filter spasial:

* + *Smoothing*, untuk proses *smoothing* dapat digunakan :
    1. *Low Pass Filter* (mengambil nilai rata-rata) ex : gaussian blur, dll
    2. *Median Filter* (mengambil median dari setiap jendela)
  + *Sharpening*, bertujuan menajamkan tepi citra. Proses ini dapat menggunakan :
    1. Pendeteksian tepi (Roberts, Prewitt, Sobel) (UAS)
    2. *High Pass Filter* (mengambil nilai tinggi)

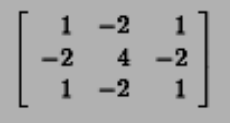
**Low Pass Filter**

* Pixel yang mempunyai *noise* pada umumnya memiliki frekuensi yang tinggi.
* *Low Pass Filter* (LPF) mengambil komponen citra yang berfrekuensi rendah dan menahan komponen berfrekuensi tinggi.
* LPF digunakan untuk melakukan proses efek blur dan reduksi *noise*.
* Ciri-ciri kernel dari LPF adalah semua nilainya positif dan jumlah dari semua nilainya sama dengan satu.

 dan 

**High Pass Filter**

* *High Pass Filter* (HPF) akan memperkuat komponen citra yang berfrekuensi tinggi dan menahan komponen berfrekuensi rendah.
* Hasilnya, *edge* atau tepi citra terlihat lebih tajam.
* HPF digunakan untuk melakukan proses deteksi tepi.
* Ciri-ciri kernel dari HPF adalah nilai-nilainya terdiri dari positif, nol, dan negatif, serta jumlah dari semua nilainya sama dengan nol.





**Noise**

* Noise adalah degradasi pada sinyal citra yang disebabkan karena gangguan eksternal.
* Noise ada empat macam yaitu :
  + *Salt and pepper noise*
  + *Gaussian noise*
  + *Speckle noise*
  + *Periodic noise*

1. **Latihan**
2. Membuat program untuk melakukan filter spasial pada sebuah citra dengan menggunakan fungsi *medianBlur* pada library OpenCV.

**Median.py**

import cv2  
import numpy as np  
  
image = cv2.imread("LennaOriginal.png", 0)  
  
row, col = image.shape  
canvas = np.zeros((row, col, 1), np.uint8)

# fungsi medianBlur memiliki 3 parameter

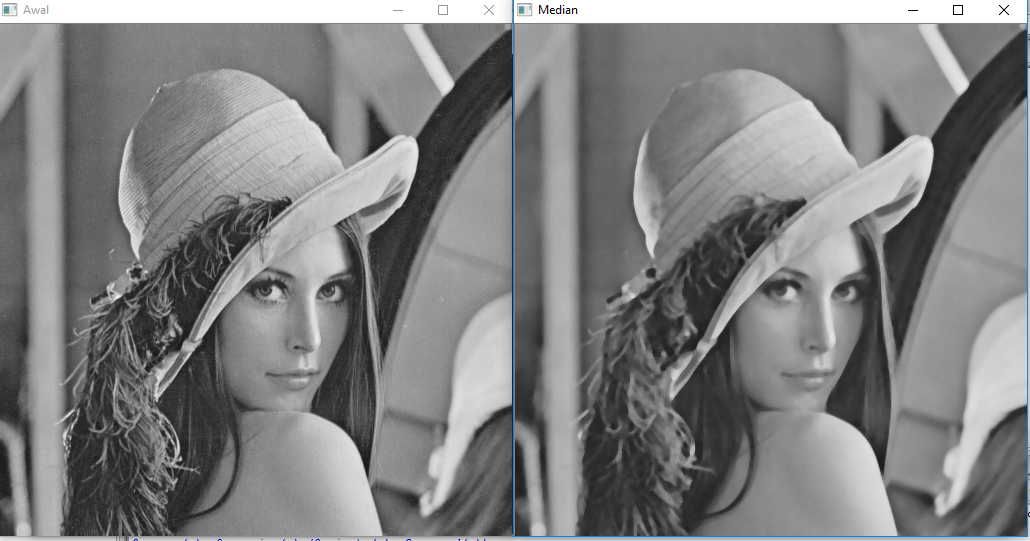
# param1 => citra original

# param2 => ukuran kernel, harus bilangan ganjil

# param3 => canvas tujuan untuk menyimpan hasil *blurring*  
cv2.medianBlur(image, 3, canvas);  
  
cv2.imshow("Awal", image)  
cv2.imshow("Median", canvas)  
  
cv2.waitKey(0)

cv2.destroyAllWindows()

**Output :**

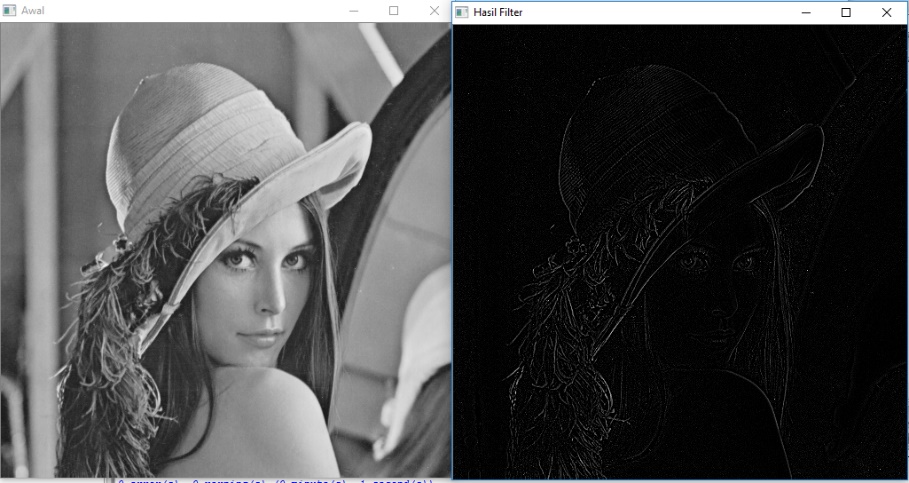


1. Membuat program untuk melakukan filter spasial dengan menerapkan matriks konvolusi pada sebuah citra. Matriks konvolusi yang digunakan :

|  |  |  |
| --- | --- | --- |
| 0 | -1 | 0 |
| -1 | 4 | -1 |
| 0 | -1 | 0 |

**Filter2D.py**

import cv2  
import numpy as np  
  
def kernel():  
 filter = np.ones((3, 3), np.float32)  
 for i in range(0, 3):  
 for j in range(0, 3):  
 if i==1 and j==1:  
 filter[i, j] \*= 4  
 elif (i+j)%2==0:  
 filter[i, j] \*= 0  
 else:  
 filter[i, j] \*= -1  
 return filter  
  
def konvolusi(img, kernel):  
 row, col = img.shape  
 mrow, mcol = kernel.shape  
 h = int(mrow/2)  
  
 canvas = np.zeros((row, col), np.uint8)  
 for i in range(0, row):  
 for j in range(0, col):  
 if i==0 or j==0 or i==row-1 or j==col-1:  
 canvas.itemset((i, j), 0)  
 else:  
 imgsum = 0  
 for k in range(-h, mrow-h):  
 for l in range(-h, mcol-h):  
 res = img[i+k, j+l] \* kernel[h+k, h+l]  
 imgsum += res  
  
 if imgsum > 255:  
 canvas.itemset((i,j), 255)  
 elif imgsum < 0:  
 canvas.itemset((i,j), 0)  
 else:  
 canvas.itemset((i,j), imgsum)  
 return canvas  
  
image = cv2.imread("LennaOriginal.png", 0)  
mask = kernel()  
  
result = konvolusi(image, mask);  
  
cv2.imshow("Awal", image)  
cv2.imshow("Hasil Filter", result)  
  
cv2.waitKey(0)  
cv2.destroyAllWindows()

**Output :**

Nama :

NRP :

Nama Dosen :

Nama Asisten :

1. **LEMBAR KERJA PRAKTIKUM**

Lakukan *enhancement* pada citra LennaInput.jpg agar menghasilkan output yang diharapkan LennaOutput.jpg. Kedua citra dapat diunduh pada website New LMS ([LKP 3 [P1 - Selasa]](https://newlms.ipb.ac.id/mod/assign/view.php?id=22205)). Gunakan langkah berikut:

1. Buatlah fungsi untuk melakukan median filter!
2. Buatlah fungsi untuk melakukan low pass filter!
3. Tentukan ukuran kernel (dan nilai kernel untuk LPF) serta banyaknya proses filtering yang dibutuhkan. Berikan alasan terhadap pilihan tersebut!
4. Terapkan kedua fungsi tersebut pada citra LennaInput.jpg sehingga dapat menghasilkan citra sedekat mungkin dengan LennaOutput.jpg
5. Filter manakah yang lebih baik? Berikan alasan mengapa memilih filter tersebut!

**\*) Seluruh operasi dilakukan secara manual tanpa library**, kecuali operasi dasar seperti imwrite(), imshow(), imread()