

PRAKTIKUM 4

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.

A. 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*.

Pada gambar 1. merupakan contoh jendela ketetanggaan 3x3. Nilai pixel pada posisi X dipengaruhi oleh 8 . nilai pixel tetangganya → berbeda pada point processing (tidak dipengaruhi oleh pixel tetangganya).

1	2	3
8	X	4
7	6	5

Gambar 4.1. Contoh jendela ketetanggaan

Beikut pada gambar 2, contoh sebuah mask berukuran 3x3. Filter ini akan diterapkan / dikonvolusikan pada setiap jendela ketetanggaan 3x3 pada citra (gambar 3).

W_{1c}	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

Gambar 4.2. Mask berukuran 3x3

G_{11}	G_{12}	G_{13}	G_{14}	G_{15}
G_{21}	G_{22}	G_{23}	G_{24}	G_{25}
G_{31}	G_{32}	G_{33}	G_{34}	G_{35}
G_{41}	G_{42}	G_{43}	G_{44}	G_{45}
G_{51}	G_{52}	G_{53}	G_{54}	G_{55}

Gambar 4.3. Contoh piksel citra

Dengan menggunakan rumus berikut, akan diperoleh nilai G_{22} yang baru hasil konvolusi.

$$G'_{22} = w_1 G_{11} + w_2 G_{12} + w_3 G_{13} + w_4 G_{21} + w_5 G_{22} + w_6 G_{23} + w_7 G_{31} + w_8 G_{32} + w_9 G_{33}$$

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 Filtering (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.

$$H(x, y) \geq 0 \quad \text{dan} \quad \sum_{xy} H(x, y) = 1$$

Salah satu kernel LPF, kernel rata-rata : $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} * 1/9$

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 positif, nol dan negatif, dan jumlah dari semua nilainya sama dengan nol

$$\sum_{xy} H(x, y) = 0$$

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

B. Latihan

1. Membuat program untuk melakukan filter spasial pada sebuah citra dengan menggunakan *smoothingtype* medianBlur .

Median.cpp

```
#include <iostream>
#include <cv.h>
#include <highgui.h>

using namespace std;
using namespace cv;

int main()
{

    Mat image;
```

```

image = imread("Lenna.png",0);
Mat dst(image.rows,image.cols,CV_8UC1,Scalar(0,0,0,0));
medianBlur(image,dst,3);

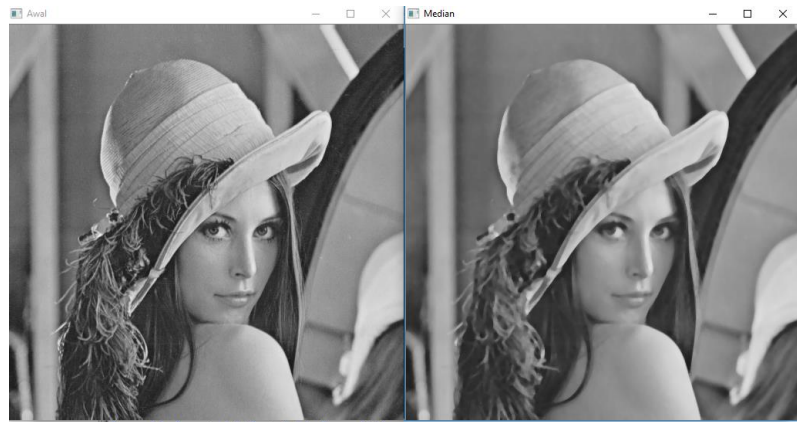
imshow("Awal",image);
imshow("Median",dst);

cvWaitKey(0);

return 0;
}

```

Output :



2. 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.cpp

```

Mat image,kernel;
int ind = 0;
int kernel_size;
Point anchor;
double delta;
int ddepth;
image = imread("Lenna.png",0);

```

```

Mat dst(image.rows,image.cols,CV_8UC1,Scalar(0,0,0,0));
kernel = (Mat_<int>(3,3) << 0,-1,0,
                           -1,4,-1,
                           0,-1,0);

    /// Initialize arguments for the filter
    anchor = Point( -1, -1 );
    delta = 0;
    //medianBlur(image,dst,5);
    filter2D(image,dst,image.depth(),kernel,anchor,delta,BORDER_DEFAULT);
    imshow("Awal",image);
    imshow("Hasil Filter",dst);

    cvWaitKey(0);

    return 0;

```

Output :

