

PENGOLAHAN CITRA DIGITAL

EDGE DETECTION

GKV - IF 2020

OUTLINE

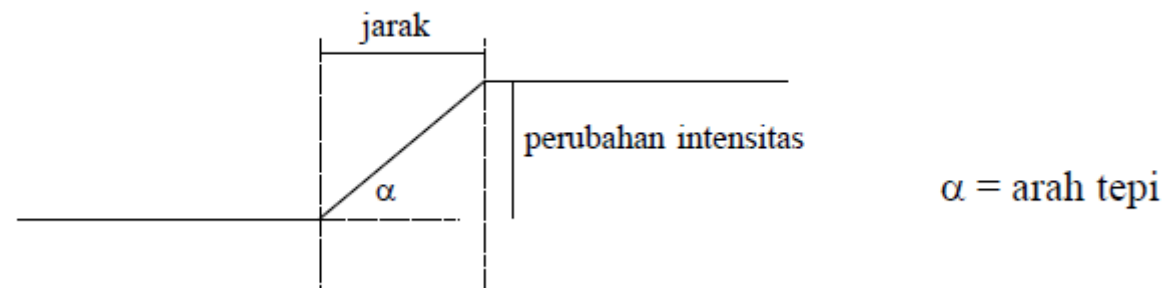
- LATAR BELAKANG
- PENGERTIAN TEPI
- OPERATOR
- ALGORITMA TEPI
- IMPLEMENTASI TEPI

LATAR BELAKANG

- Faktor kunci dalam mengekstraksi ciri adalah kemampuan mendeteksi keberadaan tepi (*edge*) dari objek di dalam citra.
- Setelah tepi objek diketahui, langkah selanjutnya dalam analisis citra adalah segmentasi, yaitu mereduksi citra menjadi objek atau region, misalnya memisahkan objek-objek yang berbeda dengan mengekstraksi batas-batas objek (*boundary*).
- Langkah terakhir dari analisis citra adalah klasifikasi, yaitu memetakan segmen-segmen yang berbeda ke dalam kelas objek yang berbeda pula.

PENGERTIAN TEPI (EDGE)

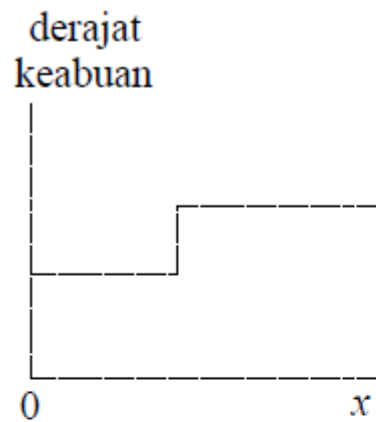
- tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat.
- Perbedaan intensitas inilah yang menampakkan rincian pada gambar.
- Tepi biasanya terdapat pada batas antara dua daerah berbeda pada suatu citra.
- Tepi dapat diorientasikan dengan suatu arah, dan arah ini berbeda-beda pada bergantung pada perubahan intensitas.



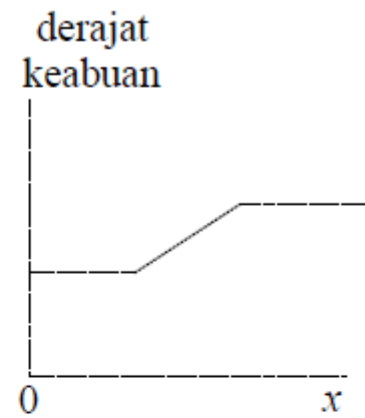
TIPE TEPI - 1

- Tepi curam
Tepi dengan perubahan intensitas yang tajam. Arah tepi berkisar 90° .
- Tepi landai
Disebut juga tepi lebar, yaitu tepi dengan sudut arah yang kecil. Tepi landai dapat dianggap terdiri dari sejumlah tepi-tepi lokal yang lokasinya berdekatan.
- Tepi yang mengandung derau (*noise*)
Umumnya tepi yang terdapat pada aplikasi *computer vision* mengandung derau. Operasi peningkatan kualitas citra (*image enhancement*) dapat dilakukan terlebih dahulu sebelum pendeteksian tepi.

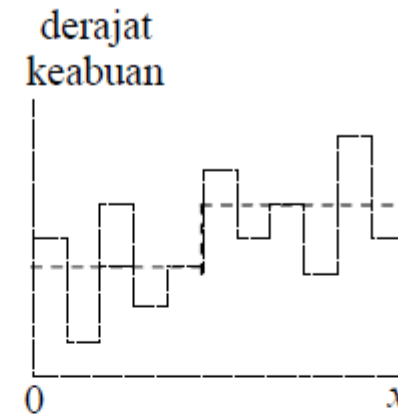
TIPE TEPI - 2



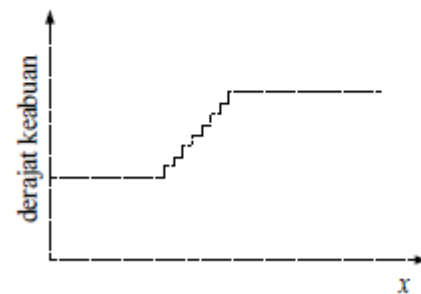
(a) Tepi curam



(b) tepi landai



(c) tepi curam dengan derau



(d) *break down* tepi landai

4	4	4	8	8	8	8	8
4	4	4	8	8	8	8	8
4	4	4	8	8	8	8	8
4	4	4	8	8	8	8	8
4	4	4	8	8	8	8	8

(e) citra dengan tepi curam

4	4	5	6	7	8	8	8
4	4	5	6	7	8	8	8
4	4	5	6	7	8	8	8
4	4	5	6	7	8	8	8
4	4	5	6	7	8	8	8

(f) citra dengan tepi landai

TUJUAN

- Tujuan operasi deteksi tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Karena tepi termasuk ke dalam komponen berfrekuensi tinggi, maka pendeteksian tepi dapat dilakukan dengan filter lolos-tinggi.

TEKNIK DETEKSI TEPI

- Terdapat beberapa teknik yang digunakan untuk mendeteksi tepi, antara lain:
 1. Operator gradien pertama (*differential gradient*)
 2. Operator turunan kedua (*Laplacian*)
 3. Operator kompas (*compass operator*)

Operator gradien pertama

- Perubahan intensitas yang besar dalam jarak yang singkat dipandang sebagai fungsi yang memiliki kemiringan yang besar.
- Kemiringan fungsi biasanya dilakukan dengan menghitung turunan pertama (*gradient*).
- Karena citra $f(x,y)$ adalah fungsi 2 dimensi dalam bentuk diskrit, maka turunan pertamanya adalah secara parsial, masing-masing dalam arah- x dan dalam arah- y .

Operator gradien pertama-2

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} G_x \\ G_y \end{bmatrix}$$

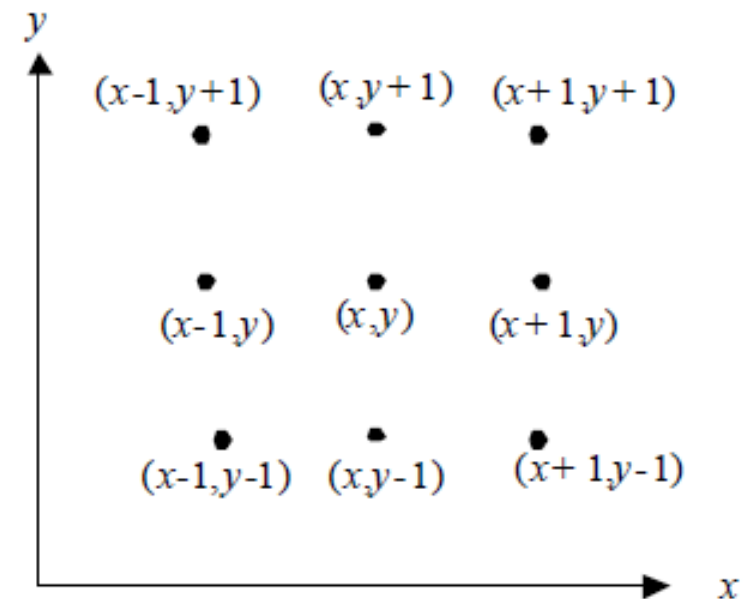
$$G_x = \frac{\partial f(x, y)}{\partial x} = \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

$$G_y = \frac{\partial f(x, y)}{\partial y} = \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}$$

$$\Delta x = \Delta y = 1$$

$$G_x = \frac{\partial f(x, y)}{\partial x} = f(x + 1, y) - f(x, y)$$

$$G_y = \frac{\partial f(x, y)}{\partial y} = f(x, y + 1) - f(x, y)$$



$$G_1(x) = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad \text{dan} \quad G_1(y) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Operator gradien pertama-3

- Operator gradien selisih-terpusat (*center-difference*):

$$D_x(x, y) = \frac{\partial f(x, y)}{\partial x} = \frac{f(x+1, y) - f(x-1, y)}{2}$$

$$D_y(x, y) = \frac{\partial f(x, y)}{\partial y} = \frac{f(x, y+1) - f(x, y-1)}{2}$$

$$D_2(x) = [-1 \quad 0 \quad 1] \quad \text{dan} \quad D_2(y) = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

Operator gradien pertama-4

- Operator Sobel

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x, y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

$$s_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$s_y = (a_0 + ca_1 + a_{22}) - (a_6 + ca_5 + a_4)$$

$$c = 2$$

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Operator gradien pertama-5

- Operator Prewitt

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x, y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

$$s_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

$$s_y = (a_0 + ca_1 + a_{22}) - (a_6 + ca_5 + a_4)$$

$$c = 1$$

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

dan

$$P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

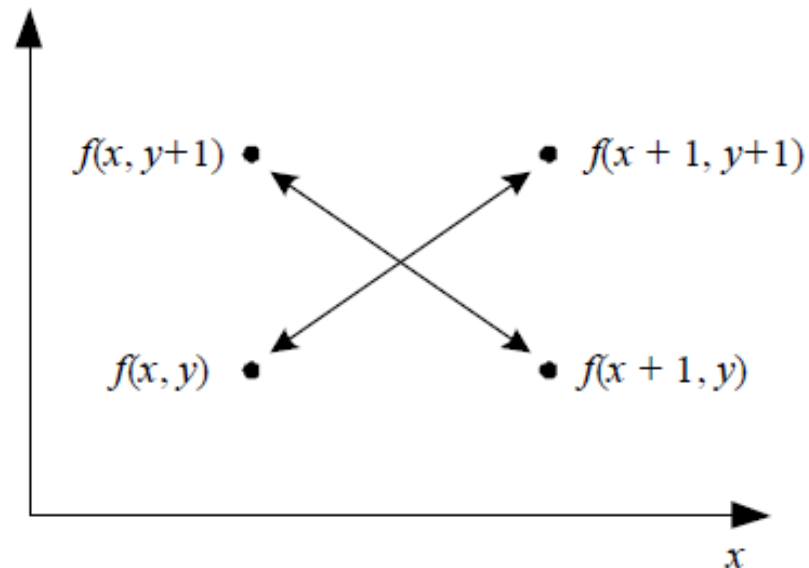
Operator gradien pertama-5

- Operator Roberts
 - Sering disebut juga operator silang. Gradien Roberts dalam arah- x dan arah- y dihitung dengan rumus:

$$R_+(x, y) = f(x+1, y+1) - f(x, y)$$

$$R_-(x, y) = f(x, y+1) - f(x+1, y)$$

$$R_+ = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ dan } R_- = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$



- Magnitude Tepi $G[f(x, y)] = |R_+| + |R_-|$

Operator Turunan Ke-dua

- Operator turunan kedua disebut juga operator Laplace. Operator Laplace mendeteksi lokasi tepi lebih akurat khususnya pada tepi yang curam.
- Pada tepi yang curam, turunan keduanya mempunyai persilangan nol (*zero-crossing*), yaitu titik di mana terdapat pergantian tanda nilai turunan kedua sedangkan pada tepi yang landai tidak terdapat persilangan nol.
- Persilangan nol merupakan lokasi tepi yang akurat.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Operator Turunan Ke-dua

$$\begin{aligned}\nabla^2 f &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ &= \frac{f(x+\Delta x, y) - 2f(x, y) + f(x-\Delta x, y)}{(\Delta x)^2} + \frac{f(x, y+\Delta y) - 2f(x, y) + f(x, y-\Delta y)}{(\Delta y)^2}\end{aligned}$$

$$\Delta x = \Delta y = 1$$

$$\begin{aligned}\nabla^2 f(x, y) &= f(x+1, y) - 2f(x, y) + f(x-1, y) + f(x, y+1) - 2f(x, y) + f(x, y-1) \\ &= f(x, y-1) + f(x-1, y) - 4f(x, y) + f(x+1, y) + f(x, y+1)\end{aligned}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

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

Operator Kompas

- Operator kompas (*compass operator*) digunakan untuk mendeteksi semua tepi dari berbagai arah di dalam citra.
- Operator kompas yang dipakai untuk pendeteksian tepi menampilkan tepi dari 8 macam arah mata angin: Utara, Timur Laut, Timur, Tenggara, Selatan, Barat Daya, dan Barat Laut.
- Pendeteksian tepi dilakukan dengan mengkonvolusikan citra dengan berbagai *mask kompas*, lalu dicari nilai kekuatan tepi (*magnitude*) yang terbesar dan arahnya.

Operator Kompas

- Operator Kompas berdasarkan arah mata angin

Utara	Timur Laut	Timur	Tenggara
$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
Selatan	Barat Daya	Barat	Barat Laut
$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$

- Magnitute Tepi

$$G[f(x, y)] = \max_i \{G_i[f(x, y)] \mid i = 1, 2, \dots, p\}$$

ALGORITMA DETEKSI TEPI

1. Inputkan citra (x) , kernel operator (k_x, k_y) , parameter ambang (T)
2. Tentukan ukuran citra (M, N)
3. Konvolusi citra dengan kernel $k_x (D_x)$
4. Konvolusi citra dengan kernel $k_y (D_y)$
5. Hitung kekuatan tepi citra $K[f(x,y)] = |D_x| + |D_y|$
6. Lakukan Proses Thresholding terhadap $K[f(x,y)]$

IMPLEMENTASI - Prewitt

```
Editor - C:\Users\user\Documents\MATLAB\PCD\tepi.m
fitness.m x translasi.m x flipp.m x histogram.m x konvolusi.m x medfilter.m x kontras_1.m x tepi.m x
1 function y=tepi(x,kx,ky,T)
2 -   Dx=konvolusi(x,kx);
3 -   Dy=konvolusi(x,ky);
4 -   KT=abs(Dx)+abs(Dy);
5 -   y=thresholding(KT,T);
6

Command Window
>> x=imread('lena.bmp');
>> Px=[-1 0 1;-1 0 1;-1 0 1]

Px =

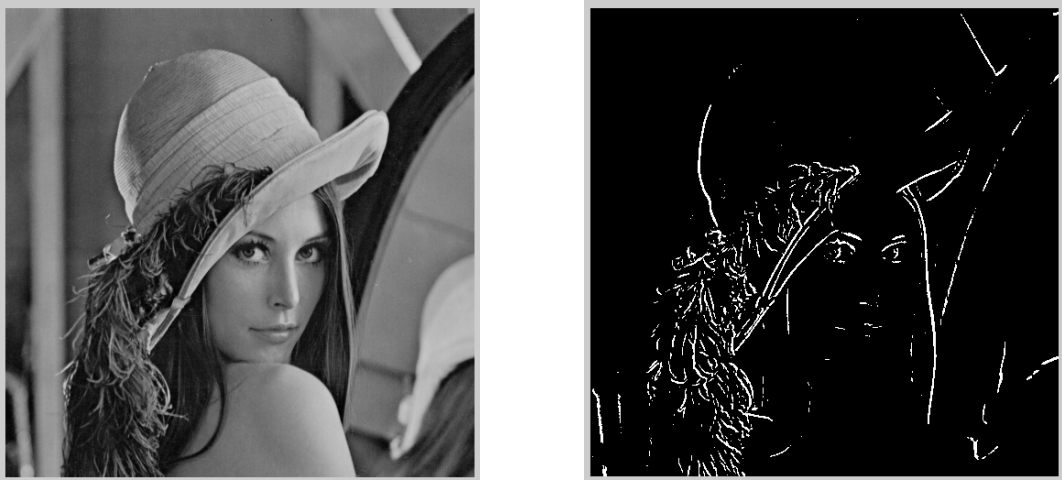
    -1     0     1
    -1     0     1
    -1     0     1

>> Py=[1 1 1;0 0 0;-1 -1 -1]

Py =

     1     1     1
     0     0     0
    -1    -1    -1

>> y=tepi(x,Px,Py,128);
>> figure(1)
>> imshow(x);
>> figure(2)
>> imshow(y);
fx >>
```



IMPLEMENTASI - Sobel

Command Window

```
>> x=imread('lena.bmp');  
>> Sx=[-1 0 1;-2 0 2;-1 0 1]
```

Sx =

```
-1    0    1  
-2    0    2  
-1    0    1
```

```
>> Sy=[1 2 1;0 0 0;-1 -2 -1]
```

Sy =

```
1    2    1  
0    0    0  
-1   -2   -1
```

```
>> y=tepi(x,Sx,Sy,128);
```

```
>> figure(1)
```

```
>> imshow(x);
```

```
>> figure(2)
```

```
>> imshow(y);
```

fx >>



Prewitt



Sobel

IMPLEMENTASI-Kompas


- Algoritma :
 1. Inputkan citra (x), kernel operator (Tr, Tg, S, BD, B, BL, U, TL), parameter ambang (T)
 2. Tentukan ukuran citra (M,N)
 3. Konvolusi citra dengan kernel Tr (K1), Tg(K2), S(K3), BD(K4), B(K5), BL(K6), U(K7), TL(K8).
 4. Hitung kekuatan tepi citra $K[f(x,y)] = \max(K1, K2, \dots, K8)$
 5. Lakukan Proses Thresholding terhadap $K[f(x,y)]$

IMPLEMENTASI-Kompas


deteksi

Deteksi Tepi (Edge Detection)

Citra Asli



Hasil Deteksi Tepi



Operator

☐ Sobel

☐ Prewit

☒ Kompas

Ambang

T

Informasi Citra

File

Format

Ambil Citra

Proses Deteksi Tepi

Keluar