

TUGAS VI

MATA KULIAH PENGOLAHAN CITRA DIGITAL

“ DETEKSI TEPI GAMBAR ”



Muh. Alfian Hanafi

200209502008

PTIK B

PRODI PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

FAKULTAS TEKNIK

UNIVERSITAS NEGERI MAKASSAR

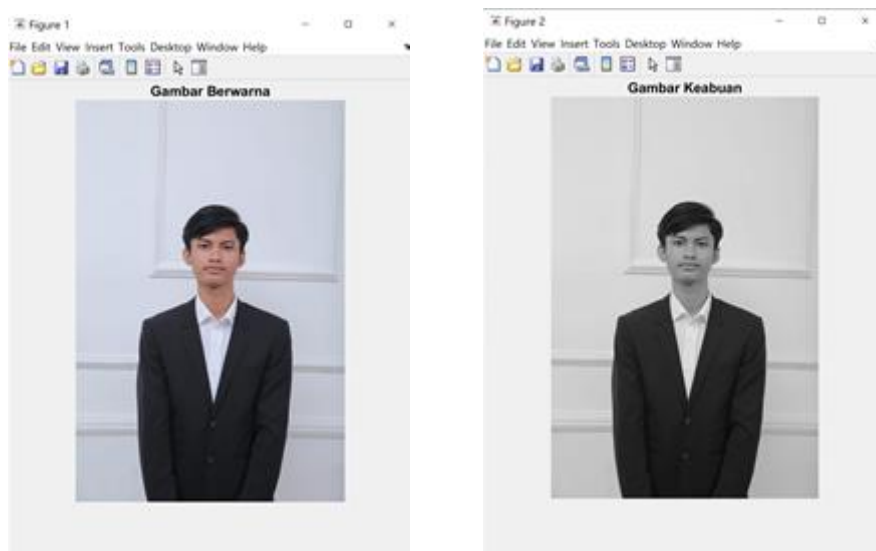
TAHUN 2021

Tepi citra adalah posisi dimana intensitas citra berubah dari tinggi menjadi rendah atau sebaliknya. Tepi dari suatu citra mengandung informasi penting dari citra yang bersangkutan. Deteksi tepi akan menghasilkan tepi objek-objek dalam citra. Informasi terkait Bentuk, ukuran bahkan teksturnya dapat digali dengan deteksi tepi. Deteksi tepi merupakan langkah awal untuk melakukan segmentasi. Terdapat beberapa metode yang dapat digunakan untuk melakukan deteksi tepi citra yaitu Metode Roberts, Metode Sobel, Metode Prewitt, Metode Isotropic, dan Metode Kompas.

- Pertama yang harus dilakukan adalah menampilkan/membaca gambarnya terlebih dahulu. Taruh gambar yang ingin dideteksi tepinya di folder Matlab agar lebih mudah dibaca oleh Matlab tersebut. Caranya dengan menggunakan fungsi `imread()`. Setelah itu ubah gambar menjadi gambar keabuan menggunakan fungsi `rgb2gray()`. Atau dengan mengetikkan script seperti gambar di bawah ini.

```
Deteksi_Tepi_Gambar.m x +
1 gambar = imread('myfoto.jpg');
2 figure(1);
3 imshow(gambar);
4 title('Gambar Berwarna');
5
6 a = rgb2gray(gambar);
7 figure(2);
8 imshow(a);
9 title('Gambar Keabuan');
```

Setelah itu, maka akan muncul dua figure yaitu Gambar Berwarna dan Gambar Keabuan seperti gambar di bawah ini.

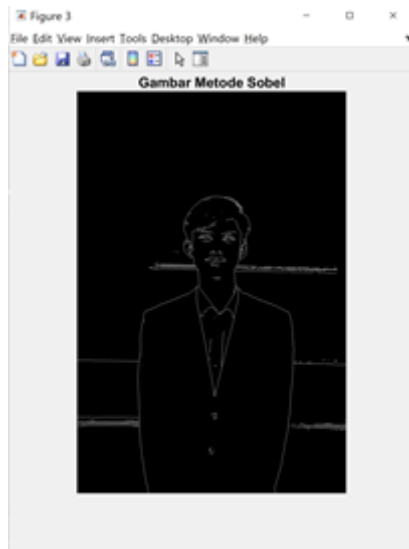


- Pertama yaitu dengan menggunakan **Metode Sobel**. Metode Sobel menghindari perhitungan gradient di titik interpolasi. Berdasarkan besaran gradient laplace,

besaran gradient. Cara yang dilakukan untuk mendeteksi gambar dengan menggunakan metode ini adalah dengan mengetikkan script seperti gambar di bawah ini.

```
%sobel
y = edge(a,'sobel');
figure(3);
imshow(y);
title('Gambar Metode Sobel');
```

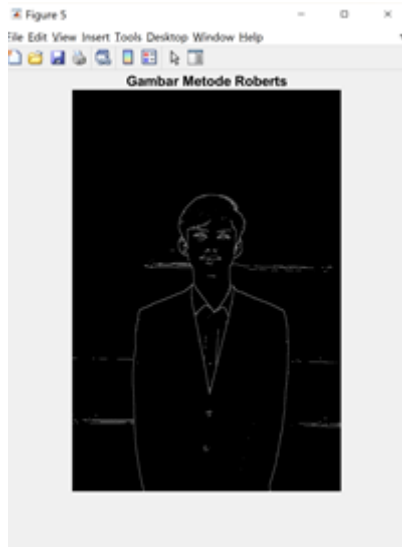
Lalu jika dijalankan,, hasilnya akan seperti gambar di bawah ini.



- Kedua yaitu dengan menggunakan **Metode Roberts**. Metode Roberts menggunakan operator berbasis gradient dan menggunakan kernel ukuran 2 X 2. Mengambil arah diagonal untuk penentuan arah dalam perhitungan nilai gradient, sehingga dapat ditulis dengan persamaan $G=|f(x,y)-f(x+1,y)|+|f(x,y)-f(x,y+1)|$. Cara yang dilakukan untuk mendeteksi gambar dengan menggunakan metode ini adalah dengan mengetikkan script seperti gambar di bawah ini.

```
17 %Roberts
18 - c = edge(a,'Roberts');
19 - figure(4);
20 - imshow(c);
21 - title('Gambar Metode Roberts');
```

Lalu jika dijalankan,, hasilnya akan seperti gambar di bawah ini.



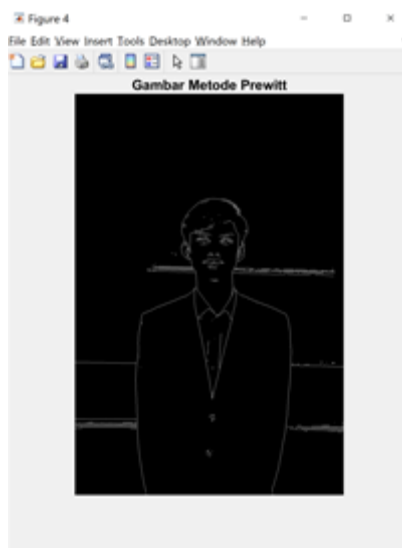
- Ketiga yaitu dengan menggunakan **Metode Prewitt**. Metode Prewitt Jika Konstanta c pada Operator Sobel diubah menjadi 1, maka Operator Sobel akan menjadi operator Prewitt. Perbedaan Operator Prewitt dengan Sobel adalah, Op. Sobel menggunakan pembobotan pada piksel-piksel yang lebih dekat dengan titik pusat kernel, sedangkan Op. Prewitt tidak menekankan pembobotan pada titik tengah. Cara yang dilakukan untuk mendeteksi gambar dengan menggunakan metode ini adalah dengan mengetikkan script seperti gambar di bawah ini.

```

23      %Prewitt
24      d = edge(a, 'Prewitt');
25      figure(5);
26      imshow(d);
27      title('Gambar Metode Prewitt');

```

Lalu jika dijalankan,, hasilnya akan seperti gambar di bawah ini.



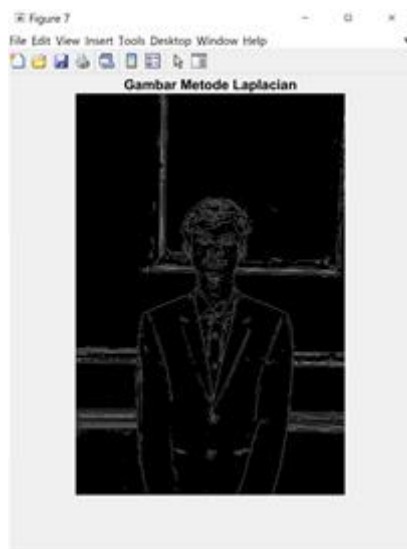
- Keempat yaitu dengan menggunakan **Metode Laplacian**. Titik-titik tepi dilacak dengan cara menemukan titik perpotongan dengan sumbu x oleh turunan kedua - > sehingga sering di sebut sebagai zero crossing operator. Sangat sensitif terhadap noise yang terletak pada titik-titik tepi. -> dapat diatasi dengan Laplacian of Gaussian yang merupakan kombinasi dari operator laplacian dengan operator gaussian. Cara yang dilakukan untuk mendeteksi gambar dengan menggunakan metode ini adalah dengan mengetikkan script seperti gambar di bawah ini.

```

29 %Laplacian
30 - e = edge(a,'log');
31 - figure(6);
32 - imshow(e);
33 - title('Gambar Metode Laplacian');

```

Lalu jika dijalankan,, hasilnya akan seperti gambar di bawah ini.



- Kelima yaitu dengan menggunakan **Metode Compass**. Pada metode compass masing-masing kernel di konvolusi dan diambil nilai terbesar dan dijadikan sebagai nilai baru dari suatu titik. Dan pada metode ini juga menggunakan pola empat mata angin yaitu utara, selatan, timur, dan barat. Konvolusi keempat kernel dengan citra yang akan dilakukan proses deteksi garis kemudian cari nilai maksimum dari keempat proses konvolusi.

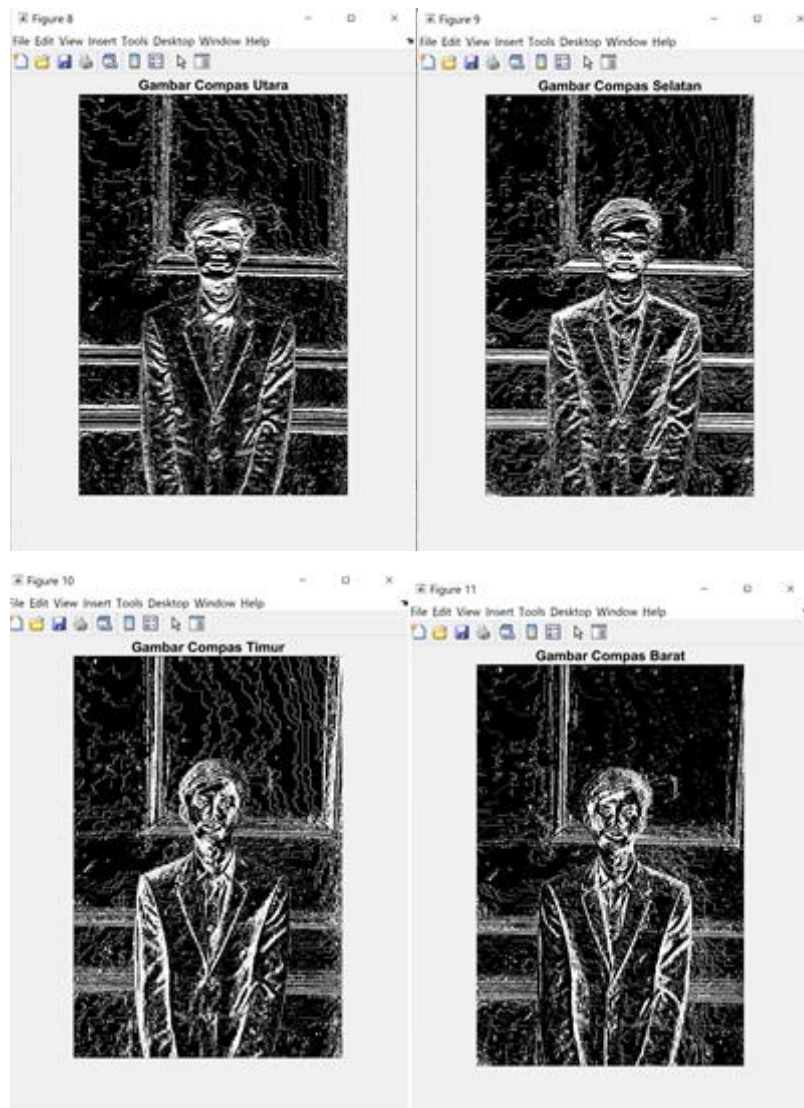
$$C_E = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \quad C_S = \begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad C_W = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$$

$$C_N = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

Cara yang dilakukan untuk mendeteksi gambar dengan menggunakan metode ini adalah dengan mengetikkan script seperti gambar di bawah ini.

```
35 %Compass
36 - Utara = [1 1 1; 1 -2 1; -1 -1 -1];
37 - Selatan = [-1 -1 -1; 1 -2 1; 1 1 1];
38 - Timur = [-1 1 1; -1 -2 1; -1 1 1];
39 - Barat = [1 1 -1; 1 -2 -1; 1 1 -1];
40
41 - figure(7);
42 - imshow(conv2(a,Utara,'same'));
43 - title('Gambar Compass Utara');
44
45 - figure(8);
46 - imshow(conv2(a,Selatan,'same'));
47 - title('Gambar Compass Selatan');
48
49 - figure(9);
50 - imshow(conv2(a,Timur,'same'));
51 - title('Gambar Compass Timur');
52
53 - figure(10);
54 - imshow(conv2(a,Barat,'same'));
55 - title('Gambar Compass Barat');
```

Maka hasilnya jika di jalankan akan seperti gambar di bawah ini.



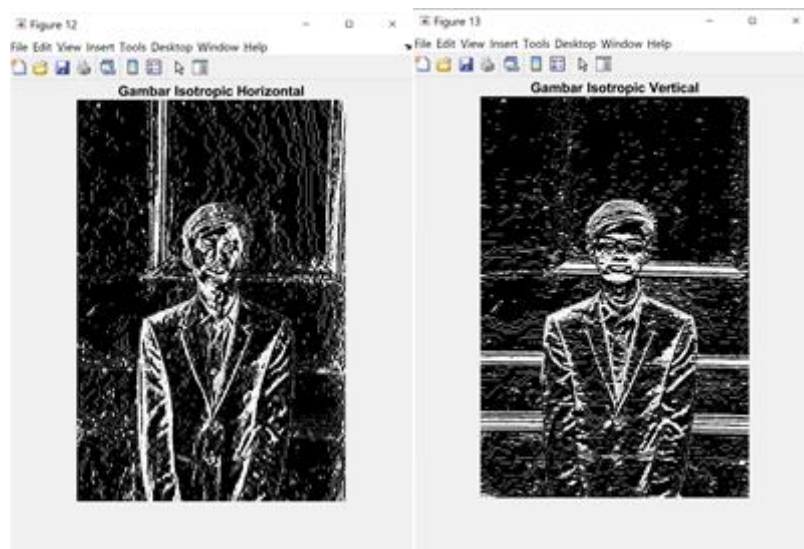
- Dan yang terakhir yaitu dengan menggunakan **Metode Isotropic**. Metode Isotropic menggunakan kernel 3 X 3. Kemudian operator ini berfungsi untuk mendeteksi tepi yang curam. Mendeteksi gambar dengan metode isotropic terbagi menjadi dua yaitu Horizontal dan Vertikal. Operator ini berfungsi untuk mendeteksi tepi yang curam.

$$I_y = \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix} \quad I_x = \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

Cara yang dilakukan untuk mendeteksi gambar dengan menggunakan metode ini adalah dengan mengetikkan script seperti gambar di bawah ini.

```
57      %Isotropic
58 -    Horizontal = [-1 0 1; -sqrt(2) 0 sqrt(2); -1 0 1];
59 -    Vertikal = [-1 -sqrt(2) -1; 0 0 0; 1 sqrt(2) 1];
60
61 -    figure(11);
62 -    imshow(conv2(a,Horizontal,'same'));
63 -    title('Gambar Isotropic Horizontal');
64
65 -    figure(12);
66 -    imshow(conv2(a,Vertikal,'same'));
67 -    title('Gambar Isotropic Vertikal');
```

Lalu jika dijalankan, maka hasilnya akan seperti gambar berikut ini.



KESIMPULAN :

Dari beberapa metode yang sudah digunakan untuk mendeteksi tepi, Metode Laplacian merupakan metode yang memiliki kinerja paling baik untuk mendeteksi tepi, dibandingkan dengan metode lainnya. Tepi gambar yang dideteksi lebih banyak dan lebih jelas atau lebih detail garisnya, inilah mengapa metode tersebut dinilai sebagai algoritma deteksi tepi yang paling optimal untuk digunakan.