

## PENGOLAHAN CITRA DIGITAL

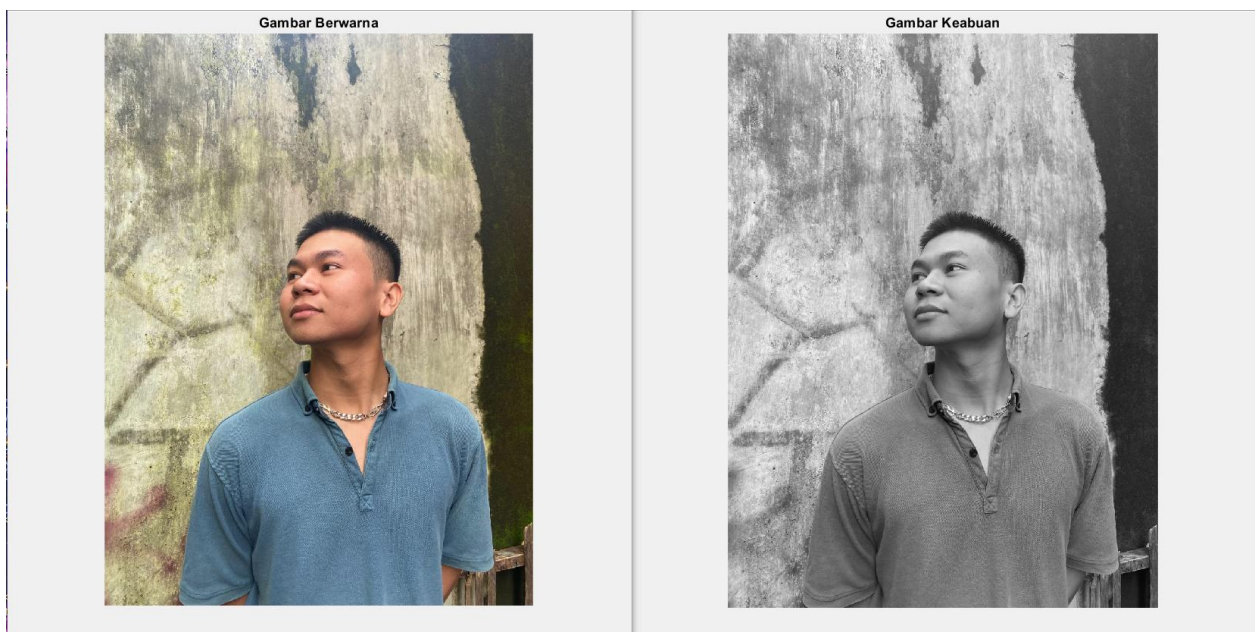
Nama : TRI ADE IRFAN

Kelas : PTIK B 2020

NIM :200209502014

Hal pertama yang harus dilakukan sebelum melakukan transformasi intensitas citra pada matlab yaitu membaca file gambar. File gambar ditaruh pada path tempat file matlab disimpan agar dapat diakses dengan mudah. Caranya adalah dengan menggunakan fungsi `imread()`. Setelah itu ubah gambar menjadi gambar keabuan menggunakan fungsi `rgb2gray()`.

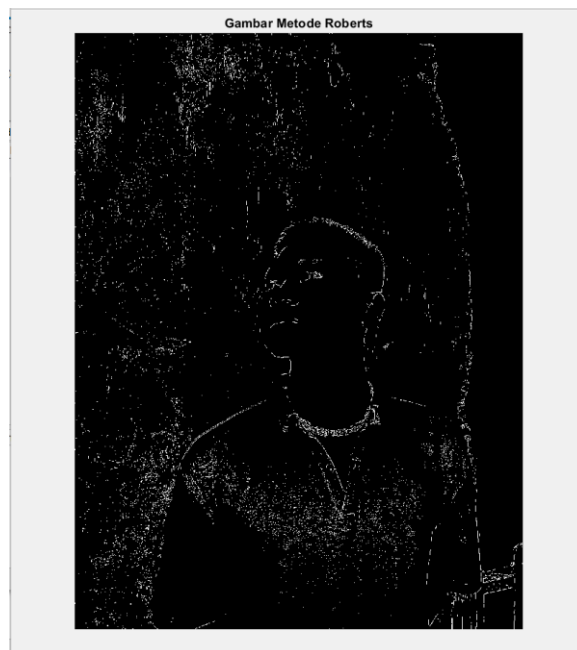
```
Editor - C:\Users\ASUS\Downloads\New folder\tugas.m
tugas.m x +
1 - gambar = imread('777.jpeg');
2 - figure(1);
3 - imshow(gambar);
4 - title('Gambar Berwarna');
5
6 - a = rgb2gray(gambar);
7 - figure(2);
8 - imshow(a);
9 - title('Gambar Keabuan');
10
```



## MENDETEKSI TEPI GAMBAR DENGAN 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  $G=|f(x,y)-f(x+1,y)|+|f(x,y)-f(x,y+1)|$ .

```
%robert
x = edge(a,'roberts');
figure(5);
imshow(x);
title('Gambar Metode Roberts');
```



Hasil dari Metode Roberts

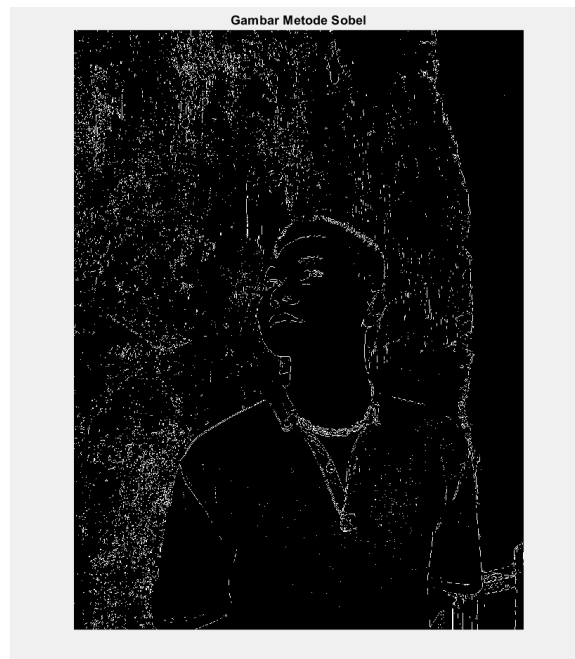
## MENDETEKSI TEPI GAMBAR DENGAN METODE SOBEL

Metode Sobel menghindari perhitungan gradient di titik interpolasi.

Berdasarkan besaran gradient laplace, besaran gradient dapat ditulis

dengan  $M \approx \sqrt{S_x^2 + S_y^2}$

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

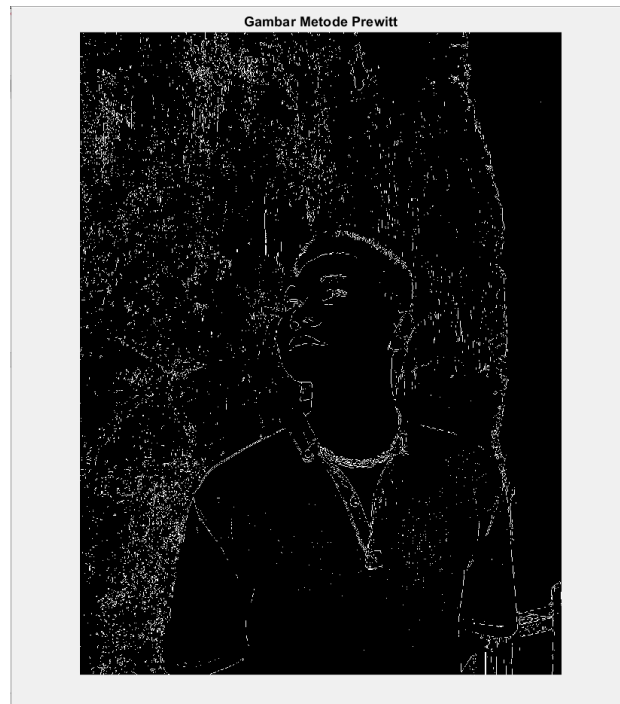


Hasil dari Metode Sobel

## MENDETEKSI GAMBAR DENGAN 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.

```
%prewitt
q = edge(a, 'prewitt');
figure(4);
imshow(q);
title('Gambar Metode Prewitt');
```



Hasil dari Metode Prewitt

## **MENDETEKSI TEPI GAMBAR DENGAN 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

```
%laplacian  
m = edge(a,'log');  
figure(7);  
imshow(m);  
title('Gambar Metode Laplacian');
```



Hasil dari Metode Laplacian

## **MENDETEKSI GAMBAR DENGAN METODE CANNY**

Metode Canny adalah algoritma deteksi tepi yang banyak digunakan dalam berbagai penelitian karena dinilai sebagai algoritma deteksi tepi yang paling optimal. Langkah awal pada algoritma Canny adalah mengimplementasikan tapis Gaussian pada citra untuk menghilangkan derau. Kemudian dilanjutkan dengan melakukan deteksi tepi pada citra dengan salah satu algoritma deteksi tepi yang ada, misalnya Sobel atau Prewitt.

```
%canny
b = edge(a, 'canny');
figure(6);
imshow(b);
title('Gambar Metode Canny');
```

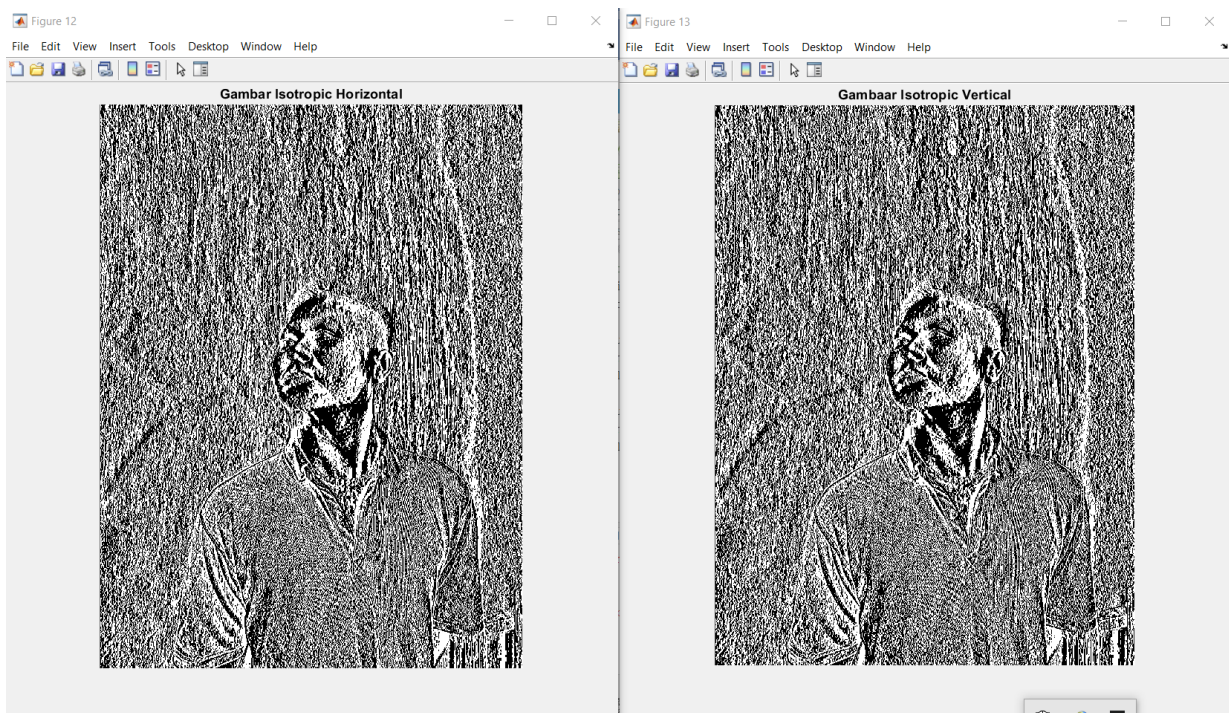




```
%isotropic
horizontal = [-1 0 1; -sqrt(2) 0 sqrt(2); -1 0 1];
vertical = [-1 -sqrt(2) -1; 0 0 0; 1 sqrt(2) 1];

figure(12);
imshow(conv2(a,horizontal,'same'));
title('Gambar Isotropic Horizontal');

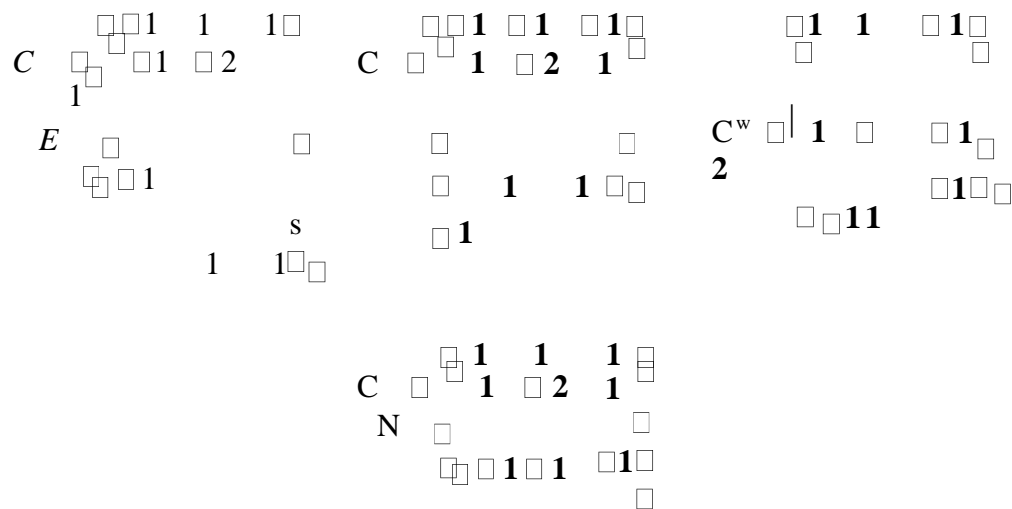
figure(13);
imshow(conv2(a,vertical,'same'));
title('Gambar Isotropic Vertical');
```



Hasil dari Metode Isotropic

## MENDETEKSI GAMBAR DENGAN METODE COMPASS

Pada metode compass masing-masing kernel di konvolusi dan diambil nilai terbesar dan dijadikan sebagai nilai baru dari suatu titik. Metode compass 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.



```
%compas
utara = [1 1 1; 1 -2 1; -1 -1 -1];
selatan = [-1 -1 -1; 1 -2 1; 1 1 1];
timur = [-1 1 1; -1 -2 1; -1 1 1];
barat = [1 1 -1; 1 -2 -1; 1 1 -1];

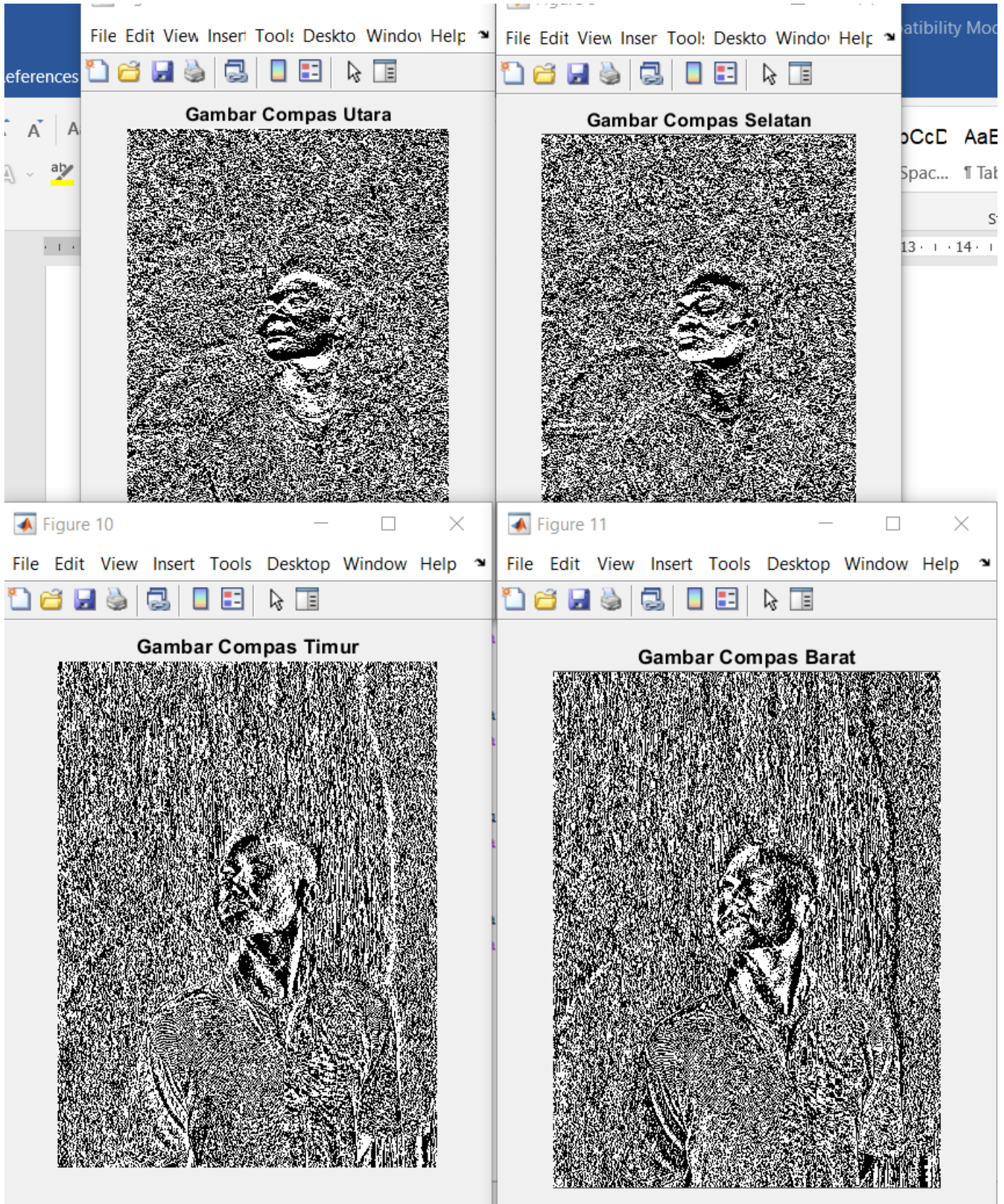
figure(8);
imshow(conv2(a,utara,'same'));
title('Gambar Compas Utara');

figure(9);
imshow(conv2(a,selatan,'same'));
title('Gambar Compas Selatan');

figure(10);
imshow(conv2(a,timur,'same'));
title('Gambar Compas Timur');

figure(11);
imshow(conv2(a,barat,'same'));
title('Gambar Compas Barat');
```





Hasil dari Metode Compass

## **KESIMPULAN**

Dari semua metode yang sudah dilihat hasilnya, **Metode Canny** merupakan metode dengan hasil yang lebih optimal dari metode lainnya. Tepi gambar yang dideteksi lebih banyak dan lebih jelas, inilah mengapa metode ini dinilai sebagai algoritma deteksi tepi yang paling optimal. Selain itu, **Metode Laplacian** juga menghasilkan hasil yang optimal walaupun hasil deteksi tepi object tidak sejelas menggunakan metode Canny.

