

PENGOLAHAN CITRA DIGITAL

DETEKSI TEPI CITRA OBJEK



NAMA : PUTRI ZHACHILIA SUSANTO

KELAS : PTIK C

NIM : 200209502021

PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER

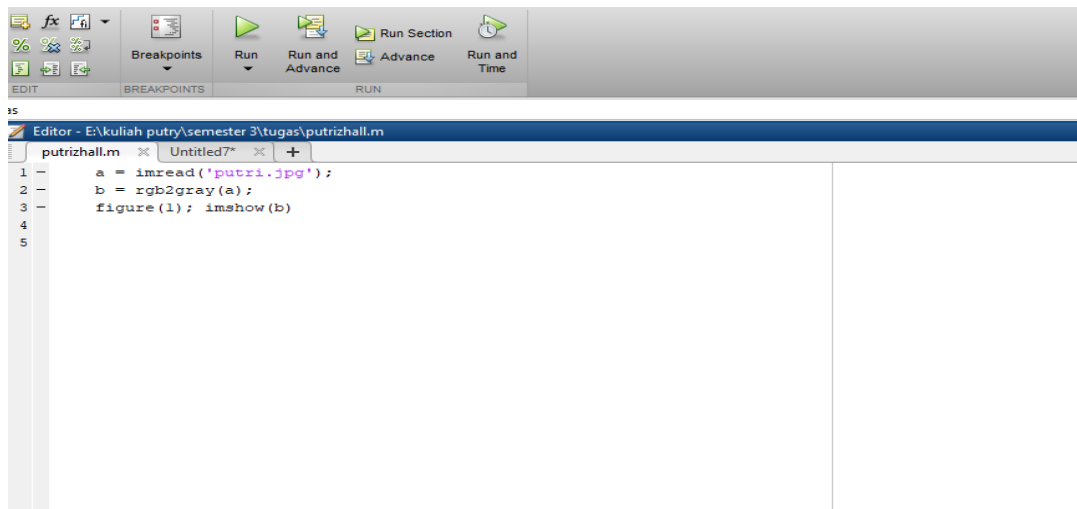
TEKNIK INFORMATIKA DAN KOMPUTER

FAKULTAS TEKNIK

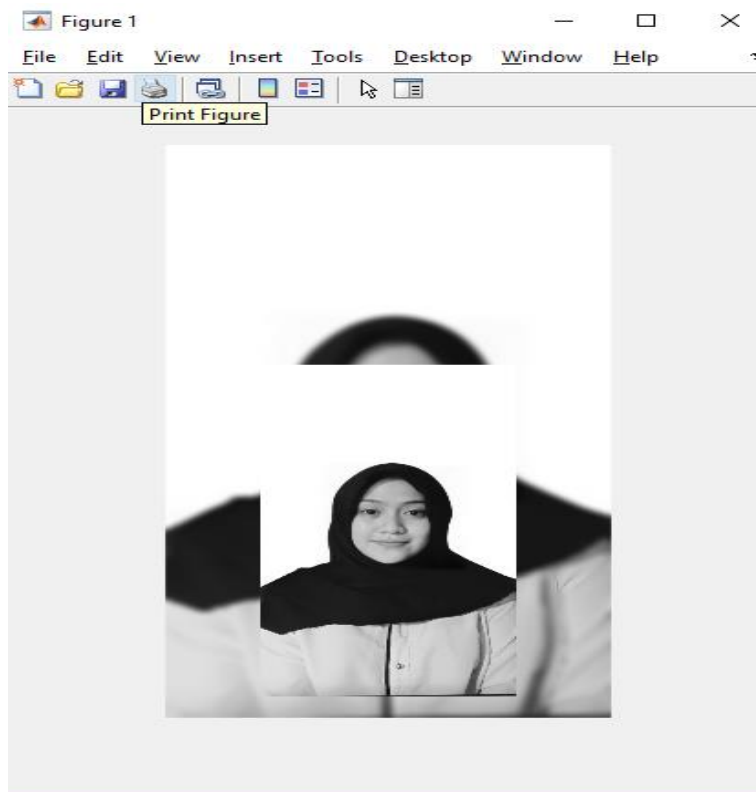
UNIVERSITAS NEGERI MAKASSAR

2020/2021

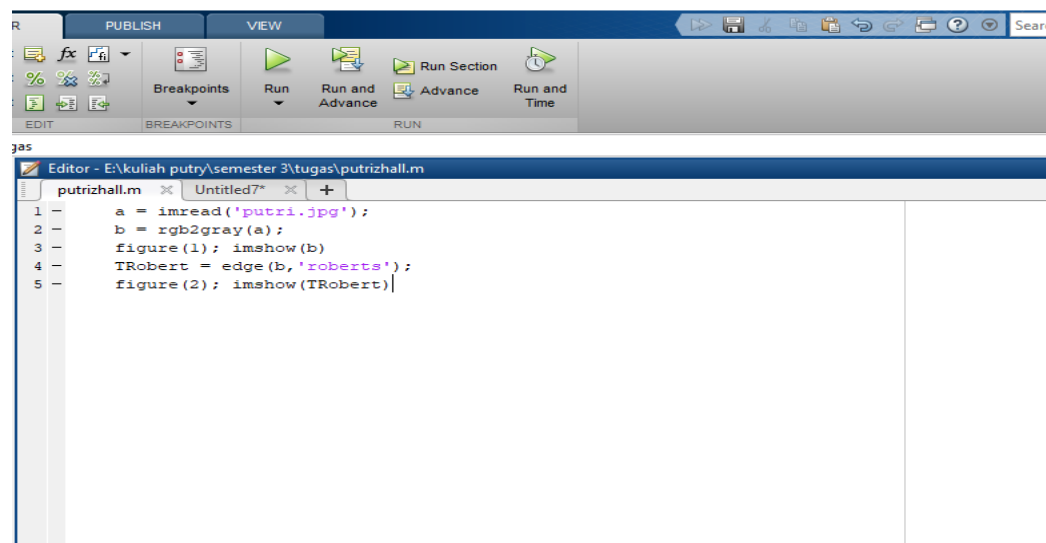
Masukkan **Script Seperti Pada Gambar Kemudian Run**



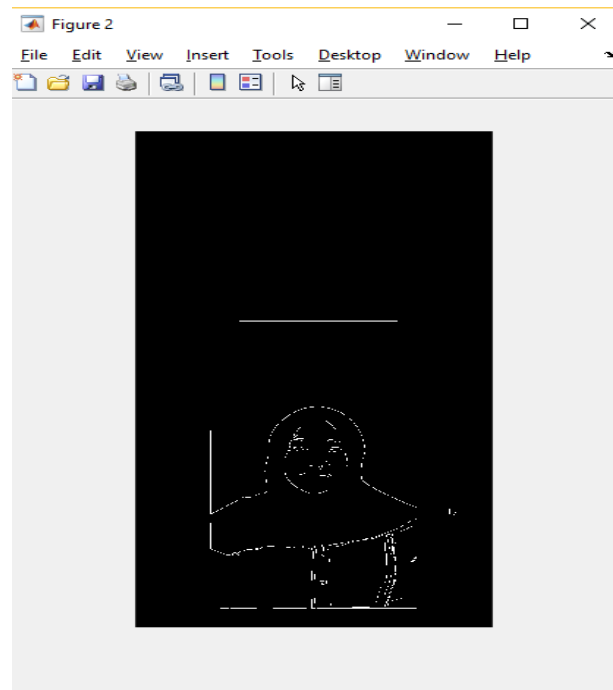
Inilah **hasilnya setelah di run**



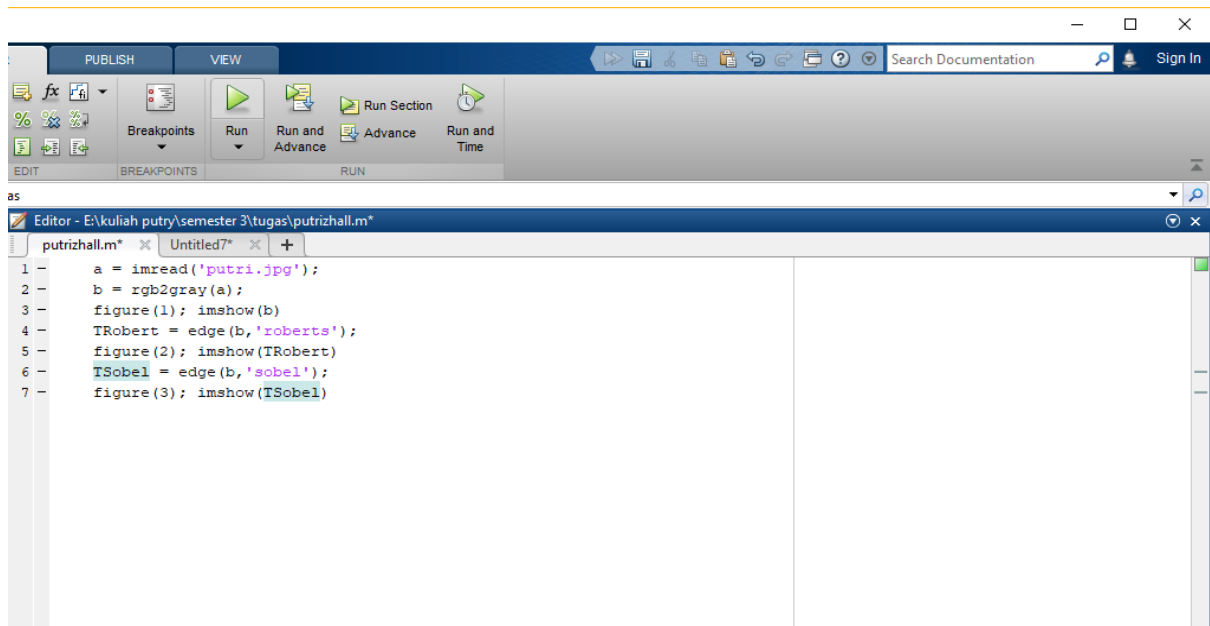
Kemudian **masukkan sintaks yang ada pada no4 dan 5 perhatikan gambar kemudian run**



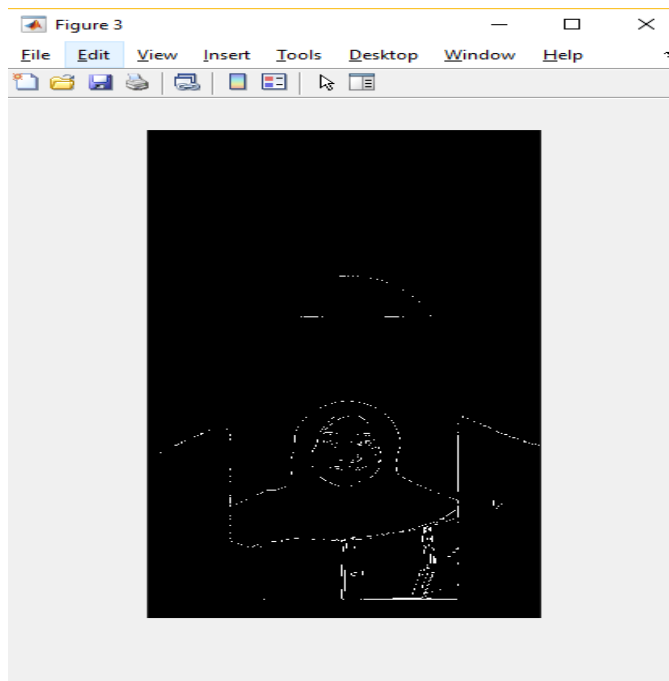
Inilah **Hasilnya**



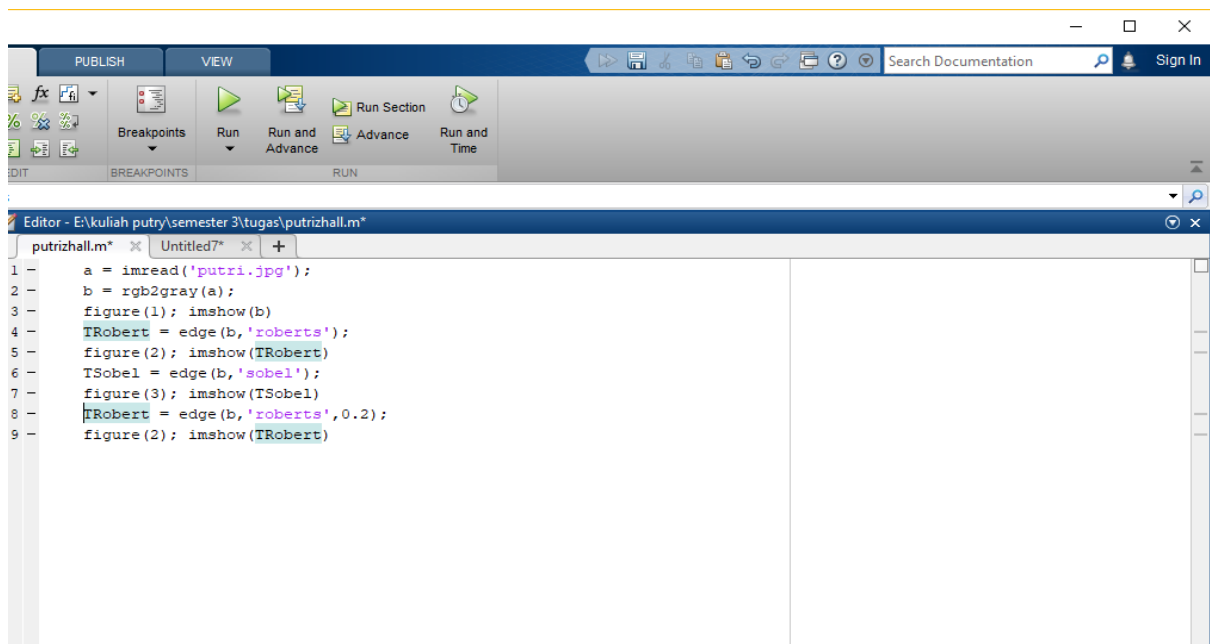
Kemudian **masukkan sintaks no6 dan 7 kemudian run**



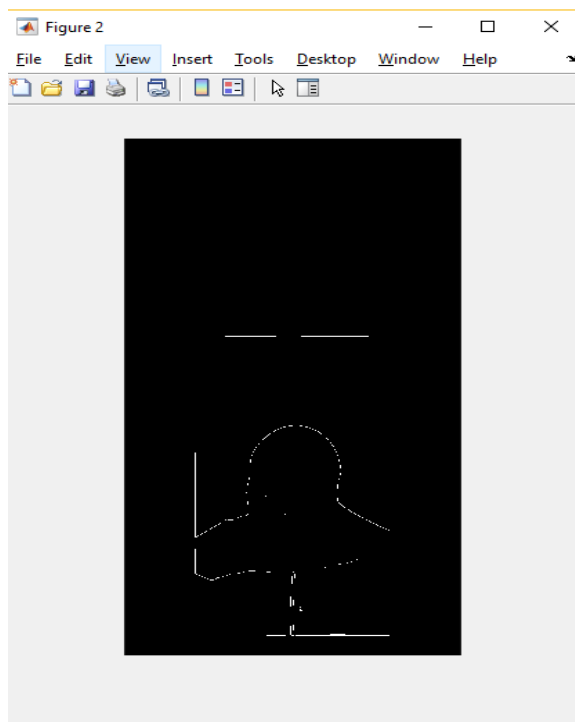
Inilah **Hasilnya**



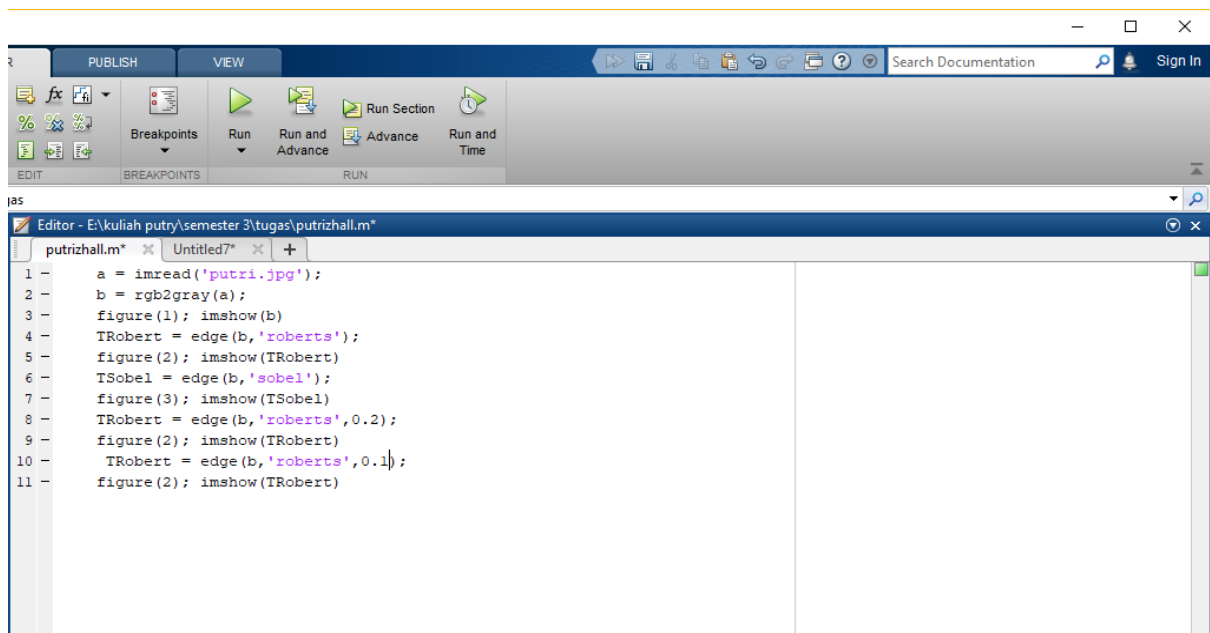
Kemudian **masukkan sintaks no8 dan 9** kemudian run



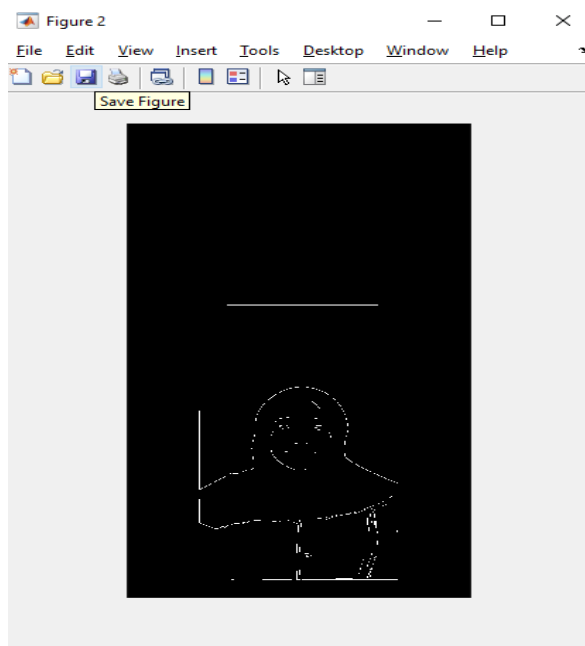
Inilah **Hasilnya**



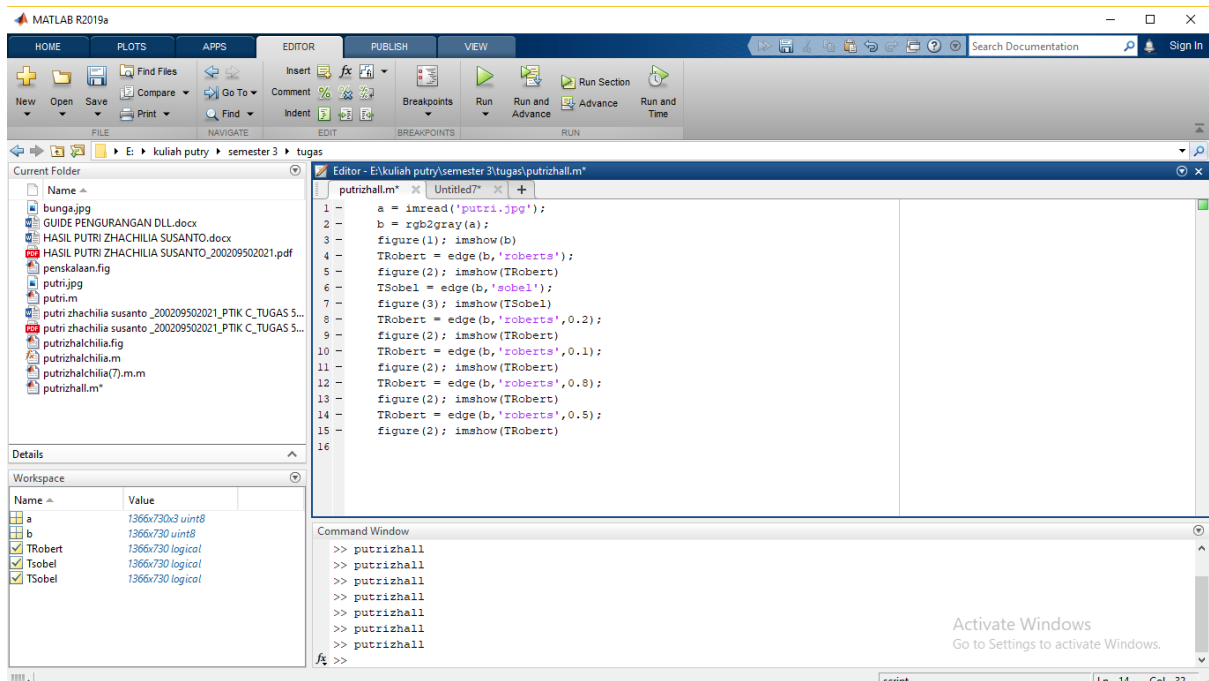
Kemudian **masukkan sintaks no10 dan 11 kemudian run**



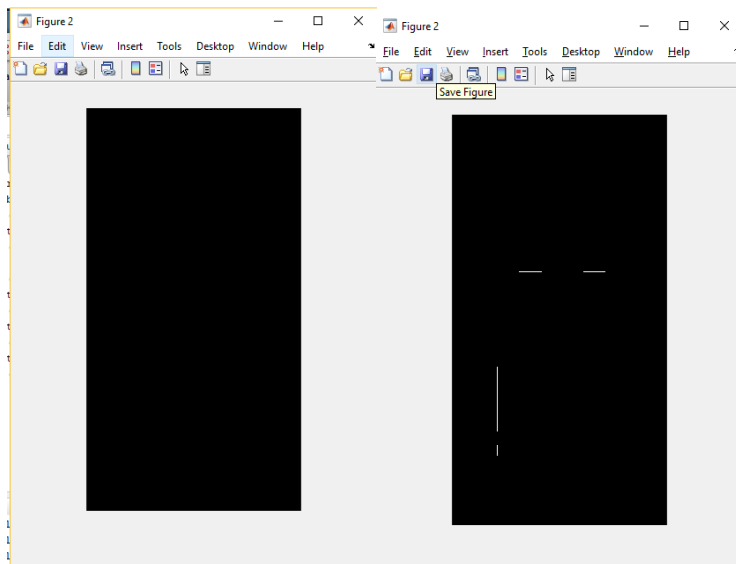
Inilah **Hasilnya**



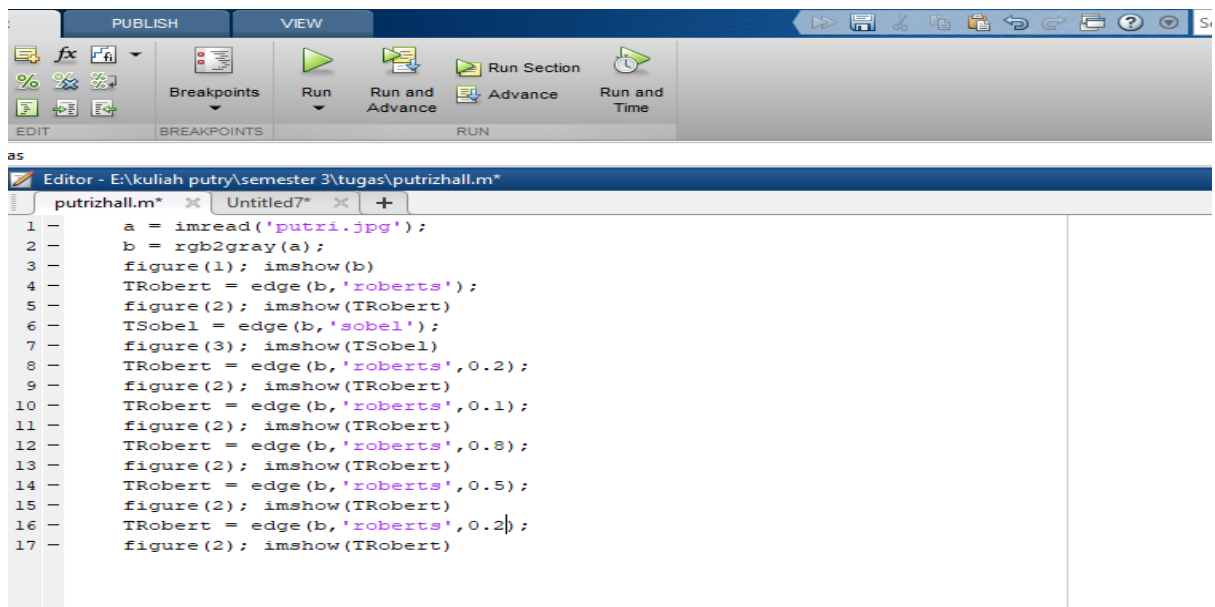
Kemudian **masukkan sintaks no12,13,14 dan 15** kemudian run



Iniilah **Hasilnya**



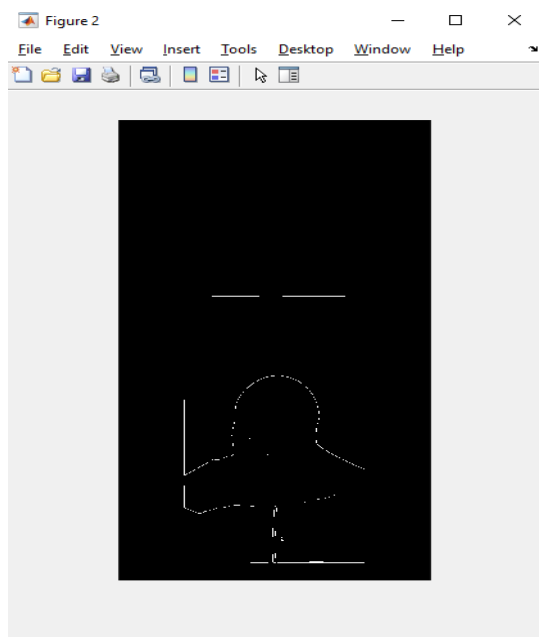
Kemudian **masukkan sintaks no16 dan 17** kemudian run



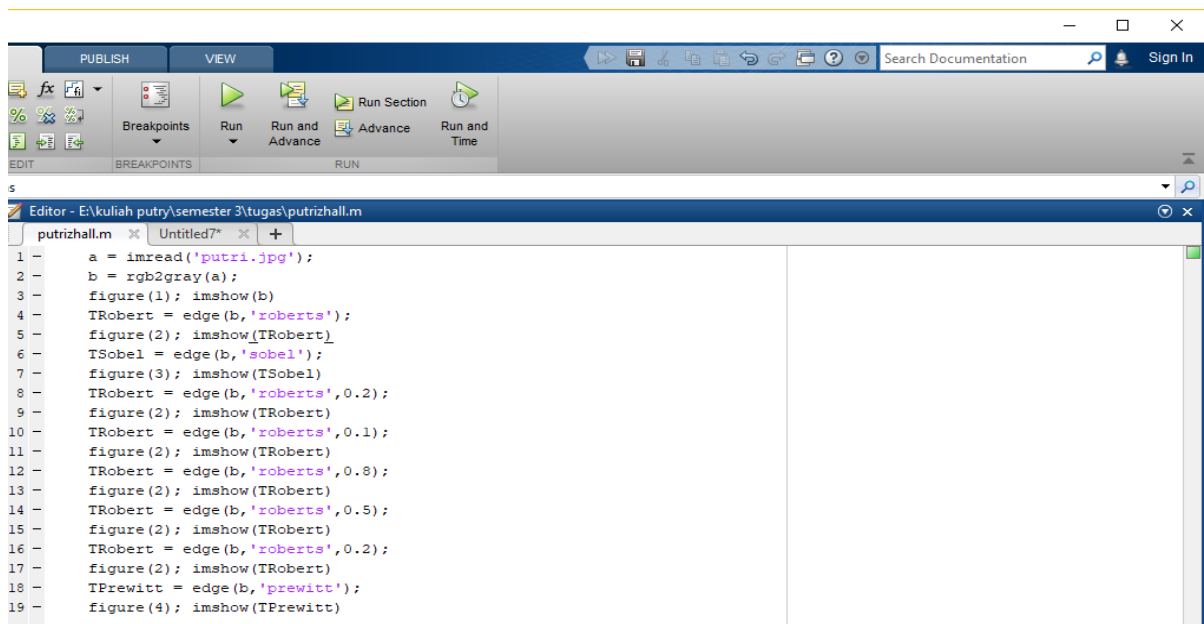
The image shows a MATLAB editor window with the following code:

```
1 - a = imread('putri.jpg');  
2 - b = rgb2gray(a);  
3 - figure(1); imshow(b)  
4 - TRobert = edge(b, 'roberts');  
5 - figure(2); imshow(TRobert)  
6 - TSobel = edge(b, 'sobel');  
7 - figure(3); imshow(TSobel)  
8 - TRobert = edge(b, 'roberts', 0.2);  
9 - figure(2); imshow(TRobert)  
10 - TRobert = edge(b, 'roberts', 0.1);  
11 - figure(2); imshow(TRobert)  
12 - TRobert = edge(b, 'roberts', 0.8);  
13 - figure(2); imshow(TRobert)  
14 - TRobert = edge(b, 'roberts', 0.5);  
15 - figure(2); imshow(TRobert)  
16 - TRobert = edge(b, 'roberts', 0.2);  
17 - figure(2); imshow(TRobert)
```

Inilah **Hasilnya**



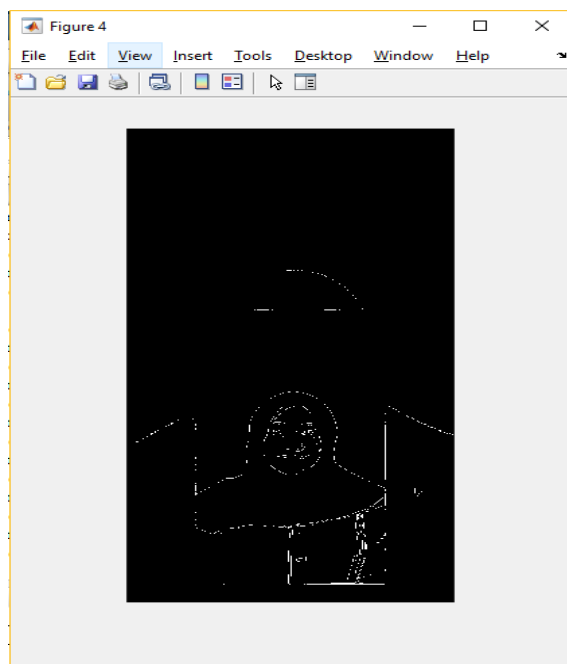
Kemudian **masukkan sintaks no18 dan 19 kemudian run**



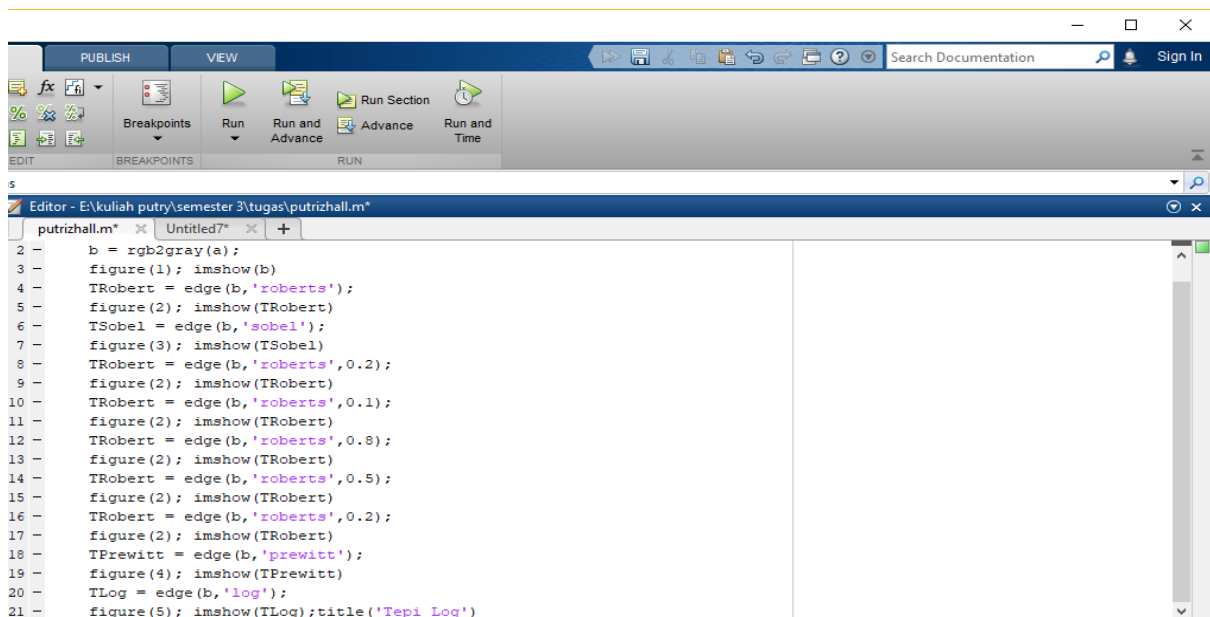
The image shows the MATLAB Editor window with a script named 'putrizhall.m'. The script performs edge detection on an image 'putri.jpg' using various methods: Roberts, Sobel, and Prewitt. It displays the original image and the edge detection results for each method in separate figures. The script is as follows:

```
1 a = imread('putri.jpg');  
2 b = rgb2gray(a);  
3 figure(1); imshow(b)  
4 TRobert = edge(b,'roberts');  
5 figure(2); imshow(TRobert)  
6 TSobel = edge(b,'sobel');  
7 figure(3); imshow(TSobel)  
8 TRobert = edge(b,'roberts',0.2);  
9 figure(2); imshow(TRobert)  
10 TRobert = edge(b,'roberts',0.1);  
11 figure(2); imshow(TRobert)  
12 TRobert = edge(b,'roberts',0.8);  
13 figure(2); imshow(TRobert)  
14 TRobert = edge(b,'roberts',0.5);  
15 figure(2); imshow(TRobert)  
16 TRobert = edge(b,'roberts',0.2);  
17 figure(2); imshow(TRobert)  
18 TPrewitt = edge(b,'prewitt');  
19 figure(4); imshow(TPrewitt)
```

Inilah **Hasilnya**



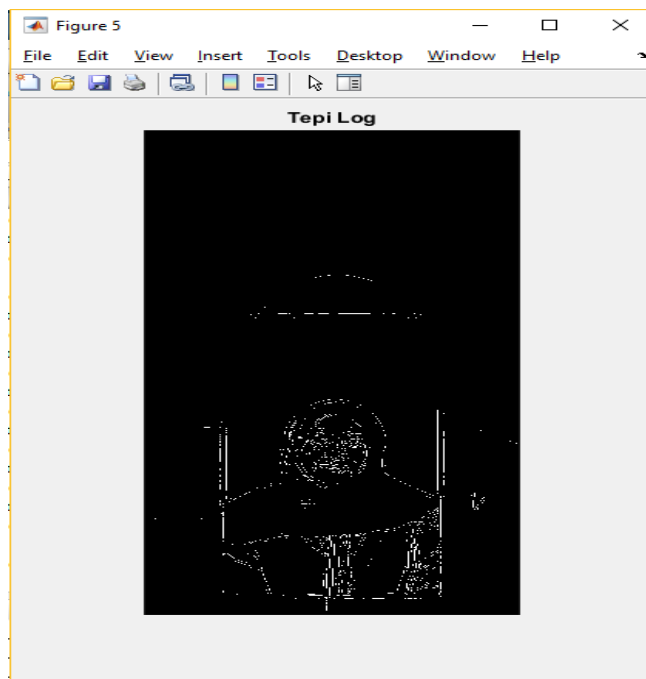
Kemudian **masukkan sintaks no20 dan 21 kemudian run**



The image shows the MATLAB Editor window with a script named 'putrizhall.m'. The script performs edge detection on an image 'a' using various edge detection functions. The code is as follows:

```
2 b = rgb2gray(a);
3 figure(1); imshow(b)
4 TRobert = edge(b, 'roberts');
5 figure(2); imshow(TRobert)
6 TSobel = edge(b, 'sobel');
7 figure(3); imshow(TSobel)
8 TRobert = edge(b, 'roberts', 0.2);
9 figure(2); imshow(TRobert)
10 TRobert = edge(b, 'roberts', 0.1);
11 figure(2); imshow(TRobert)
12 TRobert = edge(b, 'roberts', 0.8);
13 figure(2); imshow(TRobert)
14 TRobert = edge(b, 'roberts', 0.5);
15 figure(2); imshow(TRobert)
16 TRobert = edge(b, 'roberts', 0.2);
17 figure(2); imshow(TRobert)
18 TPrewitt = edge(b, 'prewitt');
19 figure(4); imshow(TPrewitt)
20 TLog = edge(b, 'log');
21 figure(5); imshow(TLog); title('Tepi Log')
```

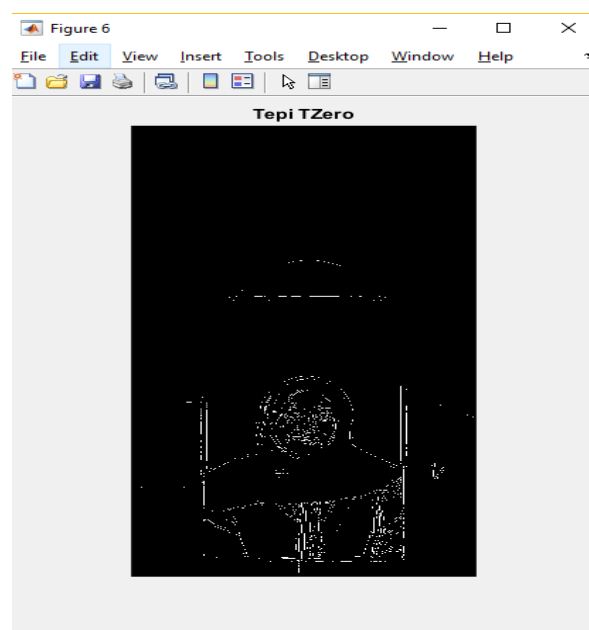
Ini **hasil TEPI LOG**



Kemudian **masukkan sintaks no22 dan 23 kemudian run**

```
Editor - E:\kuliah putry\semester 3\tugas\putrizhall.m*
putrizhall.m* x Untitled7* x +
4 - TRobert = edge(b, 'roberts');
5 - figure(2); imshow(TRobert)
6 - TSobel = edge(b, 'sobel');
7 - figure(3); imshow(TSobel)
8 - TRobert = edge(b, 'roberts', 0.2);
9 - figure(2); imshow(TRobert)
10 - TRobert = edge(b, 'roberts', 0.1);
11 - figure(2); imshow(TRobert)
12 - TRobert = edge(b, 'roberts', 0.8);
13 - figure(2); imshow(TRobert)
14 - TRobert = edge(b, 'roberts', 0.5);
15 - figure(2); imshow(TRobert)
16 - TRobert = edge(b, 'roberts', 0.2);
17 - figure(2); imshow(TRobert)
18 - TPrewitt = edge(b, 'prewitt');
19 - figure(4); imshow(TPrewitt)
20 - TLog = edge(b, 'log');
21 - figure(5); imshow(TLog); title('Tepi Log')
22 - TZero = edge(b, 'zerocross');
23 - figure(6); imshow(TZero); title('Tepi TZero')
```

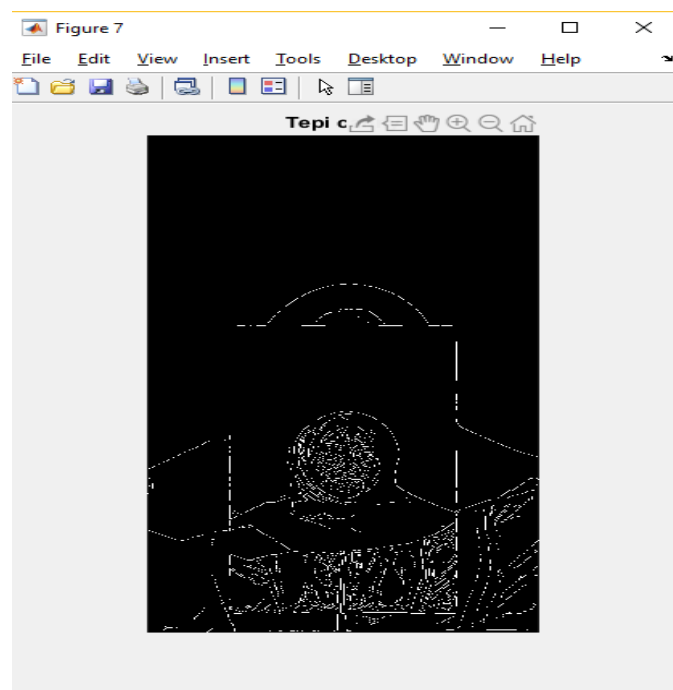
Hasil **Tepi TZero**



Kemudian **masukkan sintaks no24 dan 25 kemudian run**

```
Editor - E:\kuliah putry\semester 3\tugas\putrizhall.m
putrizhall.m  Untitled7*  +
6 - TSobel = edge(b,'sobel');
7 - figure(3); imshow(TSobel)
8 - TRobert = edge(b,'roberts',0.2);
9 - figure(2); imshow(TRobert)
10 - TRobert = edge(b,'roberts',0.1);
11 - figure(2); imshow(TRobert)
12 - TRobert = edge(b,'roberts',0.8);
13 - figure(2); imshow(TRobert)
14 - TRobert = edge(b,'roberts',0.5);
15 - figure(2); imshow(TRobert)
16 - TRobert = edge(b,'roberts',0.2);
17 - figure(2); imshow(TRobert)
18 - TPrewitt = edge(b,'prewitt');
19 - figure(4); imshow(TPrewitt)
20 - TLog = edge(b,'log');
21 - figure(5); imshow(TLog);title('Tepi Log')
22 - TZero = edge(b,'zerocross');
23 - figure(6); imshow(TZero);title('Tepi TZero')
24 - Tcanny = edge(b,'canny');
25 - figure(7); imshow(Tcanny);title('Tepi canny')
```

Hasil **Tepi Canny**



SOAL

Lakukan deteksi tepi dari foto anda. Bandingkan metode deteksi tepi dan berikan penjelasan metode yang mana yang lebih baik.

JAWABAN

Untuk saya pribadi lebih memilih Tepi Canny Mengapa karna dengan menggunakan metode canny terlihat garis tepi pada gambar baik pada bagian dalam maupun tepi gambar terlihat tebal, garis vertical maupun horizontal pada bagian depan foto sangat jelas jika dibandingkan dengan dua metode di atas dan Canny Peningkatan kualitas bertujuan menghasilkan Gambar dengan kualitas yang lebih baik dibandingkan dengan Deteksi Tepi Lainnya dan Operator *Canny* menggunakan *Gaussian Derivative Kernel* untuk menyaring *noise* dari citra awal untuk mendapatkan hasil deteksi tepi yang halus. Pada metode LoG garis tepiyang dihasilkan akan keliatan tidak begitu jelas, dimana pada *image* dengan derajatkeabuan (*gray level*), nilai intensitas antara batas tepi keliatan hampir sama dengan background dari citra yang dicari tepiannya tersebut. Dan Berdasarkan hasil analisis diatas maka disimpulkan bahwa deteksi tepi paling baik dihasilkan dari pengguna metode canny. Deteksi tepi dengan menggunakan metode canny adalah deteksi tepi terbaik dikarenakan morfologi garis yang dihasilkan oleh deteksi tepi ini lebih baik terlihat pada garis tepi gambar, baik pada bagian dalam maupun tepi gambar terlihat tebal, garis vertical maupun horizontal pada bagian depan obyek sangat jelas jika dibandingkan dengan dua metode di atas