Laporan Tugas 3
Interpretasi dan Pengolahan Citra

Disusun oleh:
13519008 Ronggur Mahendra

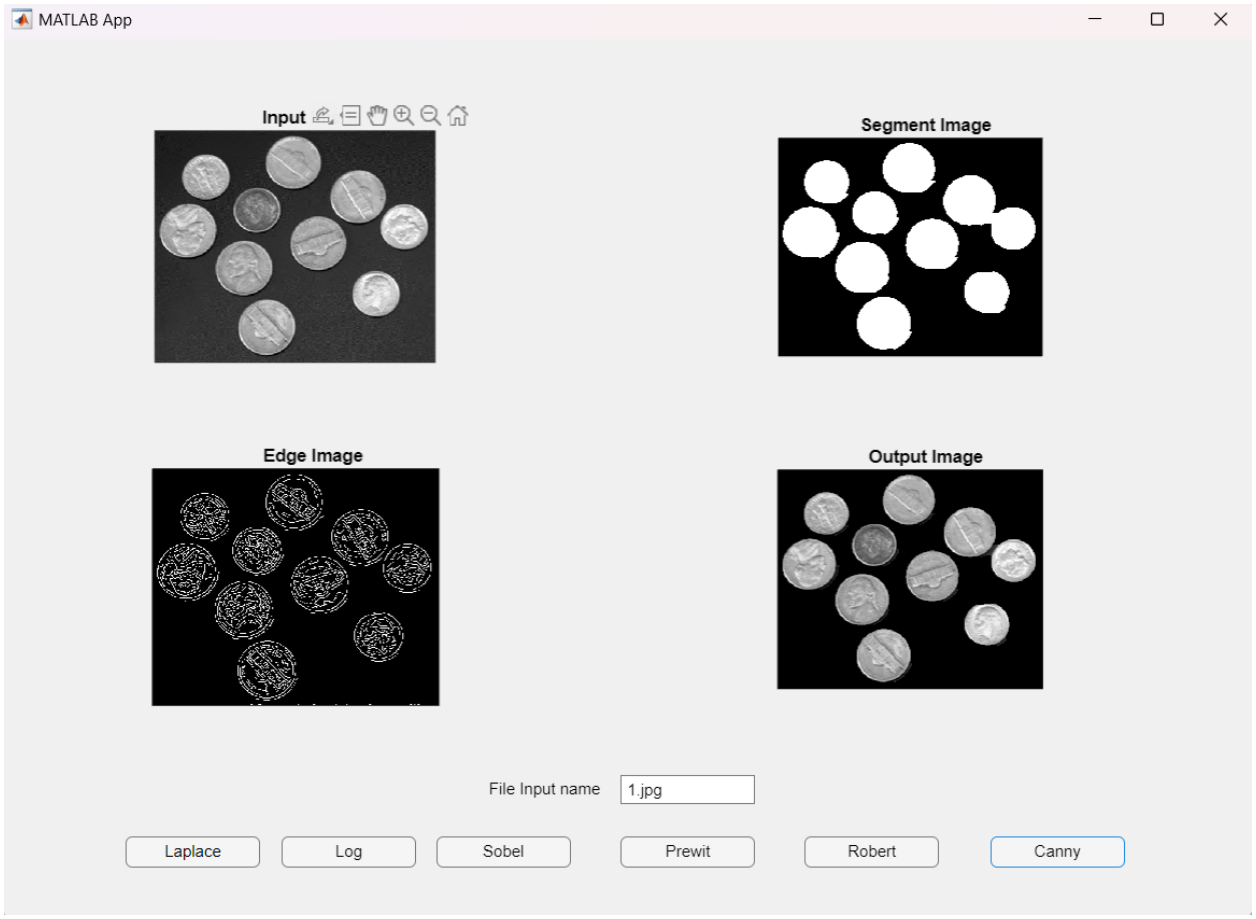
PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2023

1. Screenshot GUI



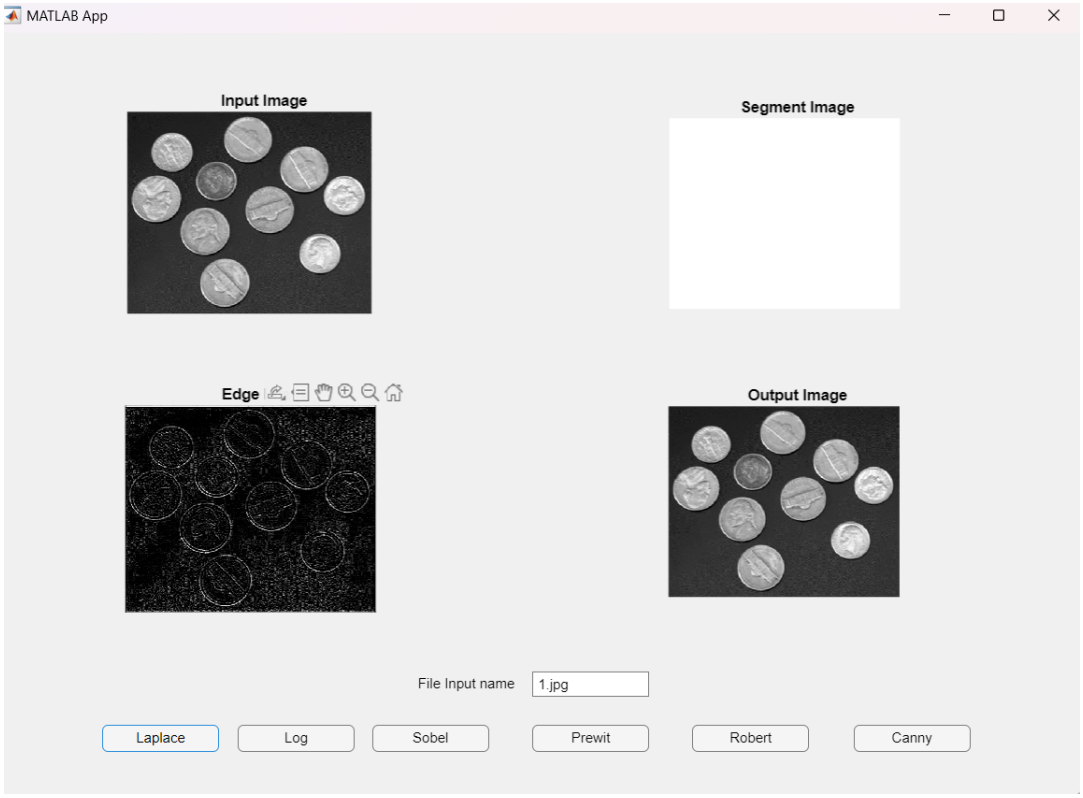
2. Program

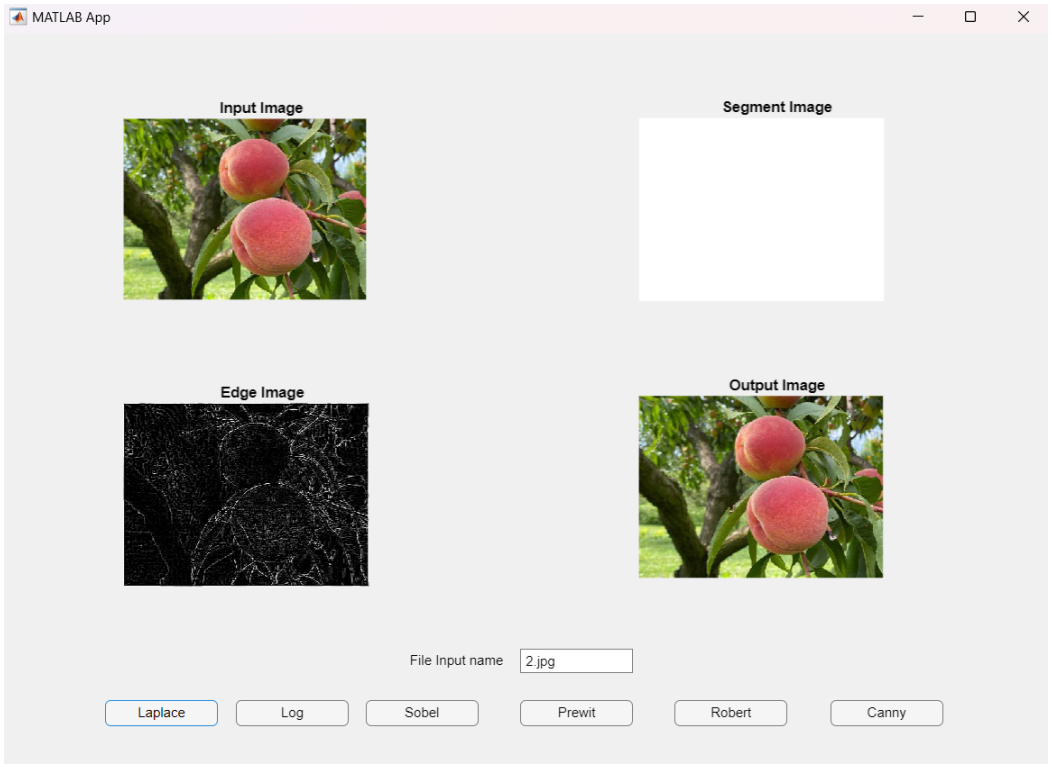
a. Laplace Edge Detection

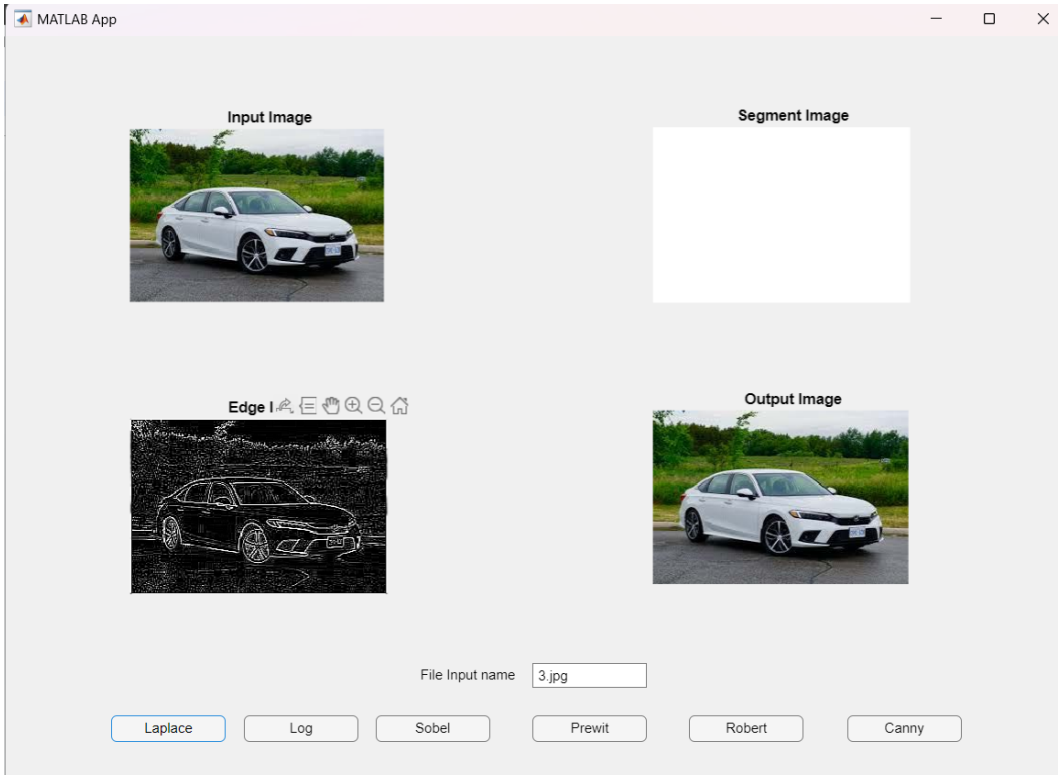
i. Kode Program

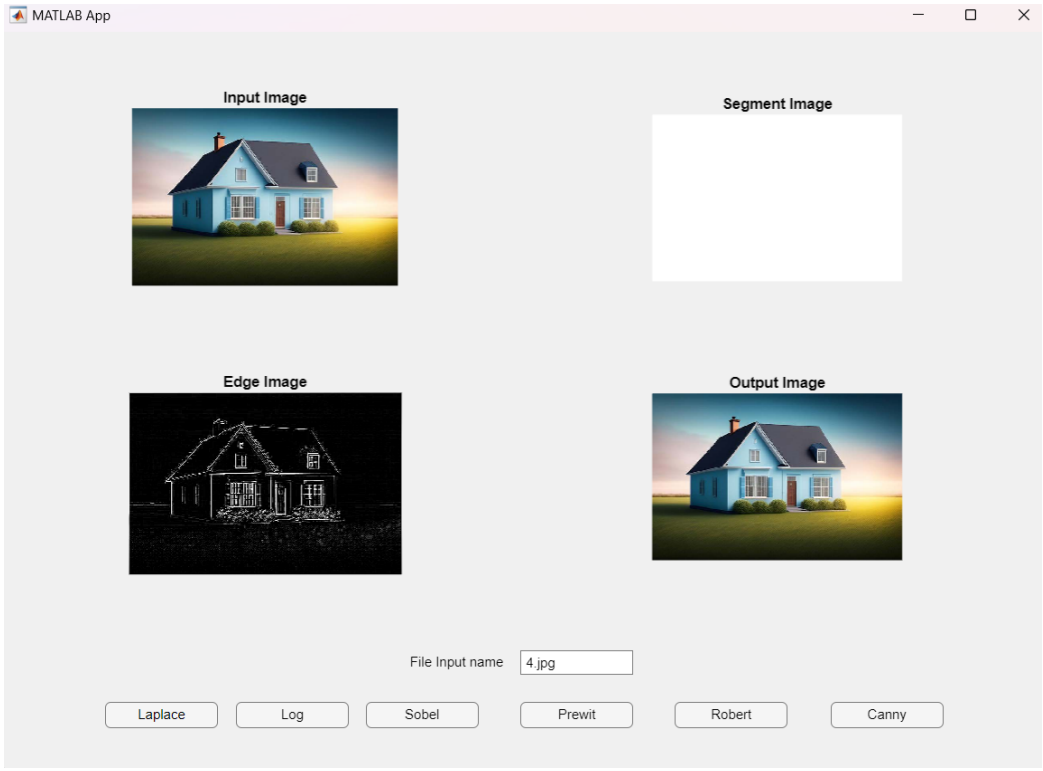
```
% Laplace Edge detenction (sesuai slide)
function result = edgeLaplace(img_input)
    % Menggunakan filter Laplace
    % h = fspecial('log', [5, 5], 2);
    H = [1 1 1; 1 -8 1; 1 1 1];
    % H = [0 1 0; 1 -4 1; 0 1 0];
    result = uint8(convn(double(img_input), double(H)));

end
```

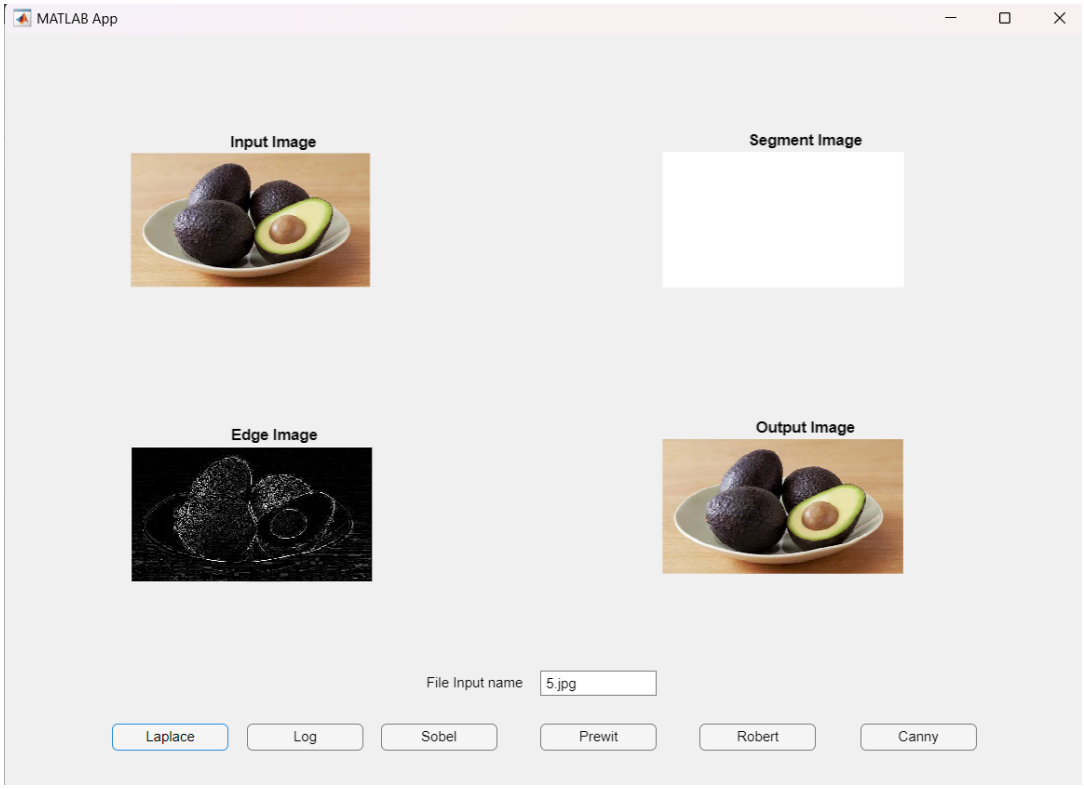
ii. Hasil Eksekusi

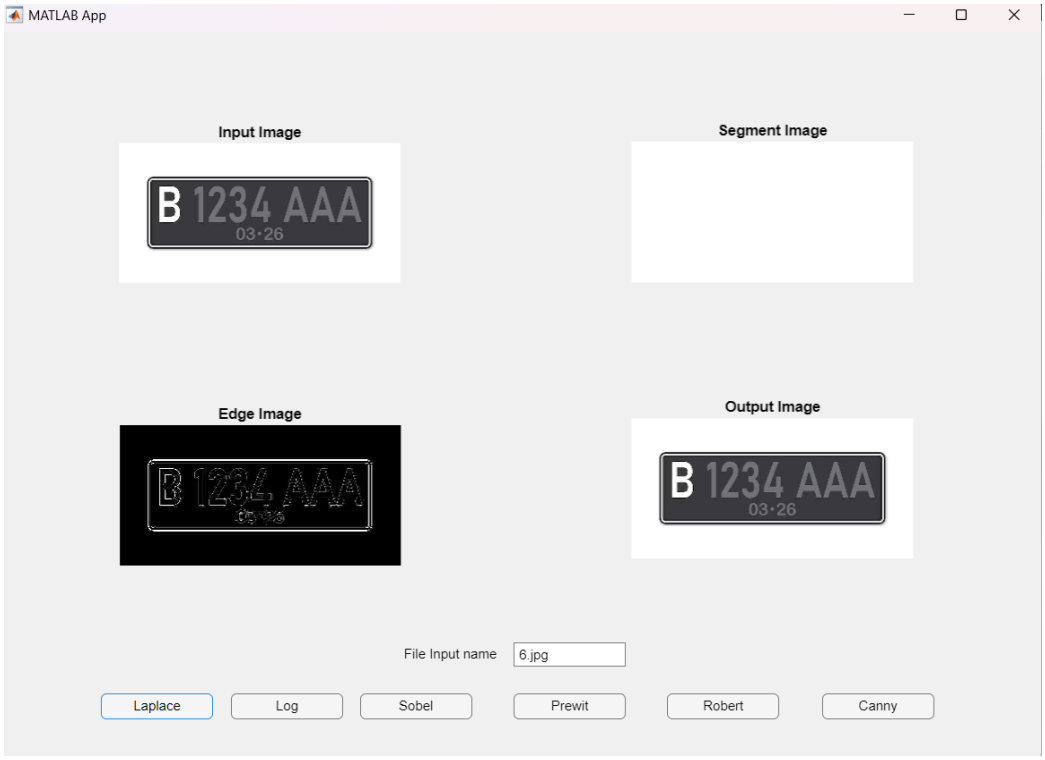


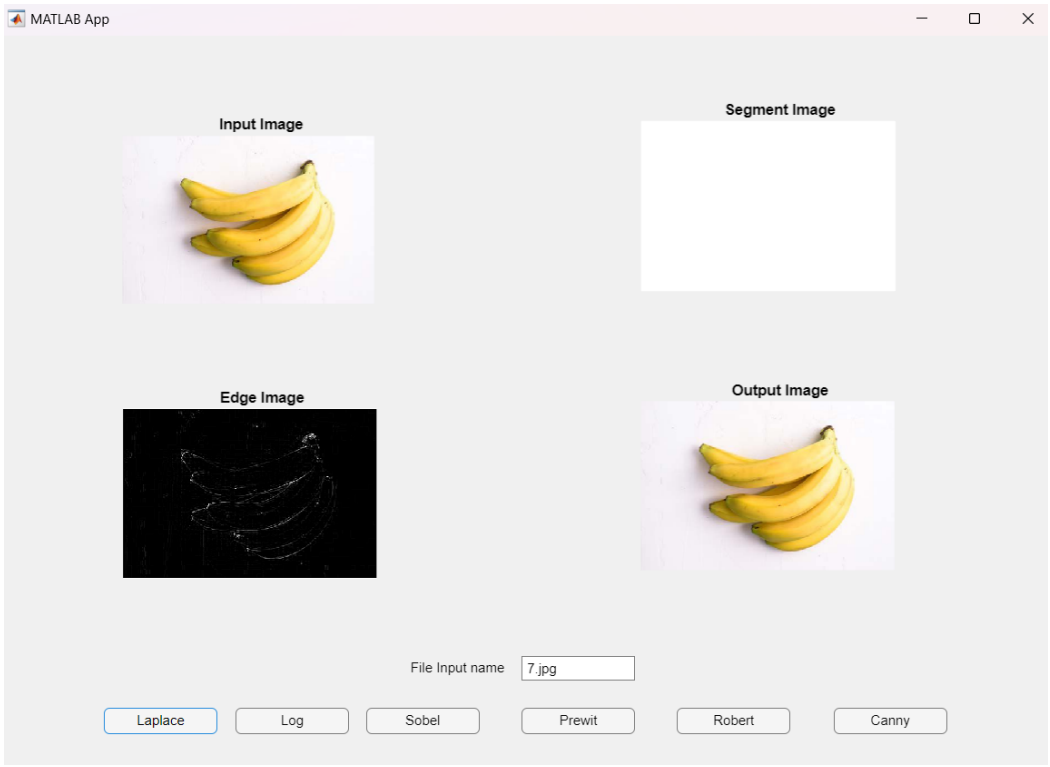

MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
2.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge I
Output Image
File Input name
3.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
4.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
5.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny





iii. Analisis cara kerja

Laplace edge detection gagal men-segmentasi semua image sample.

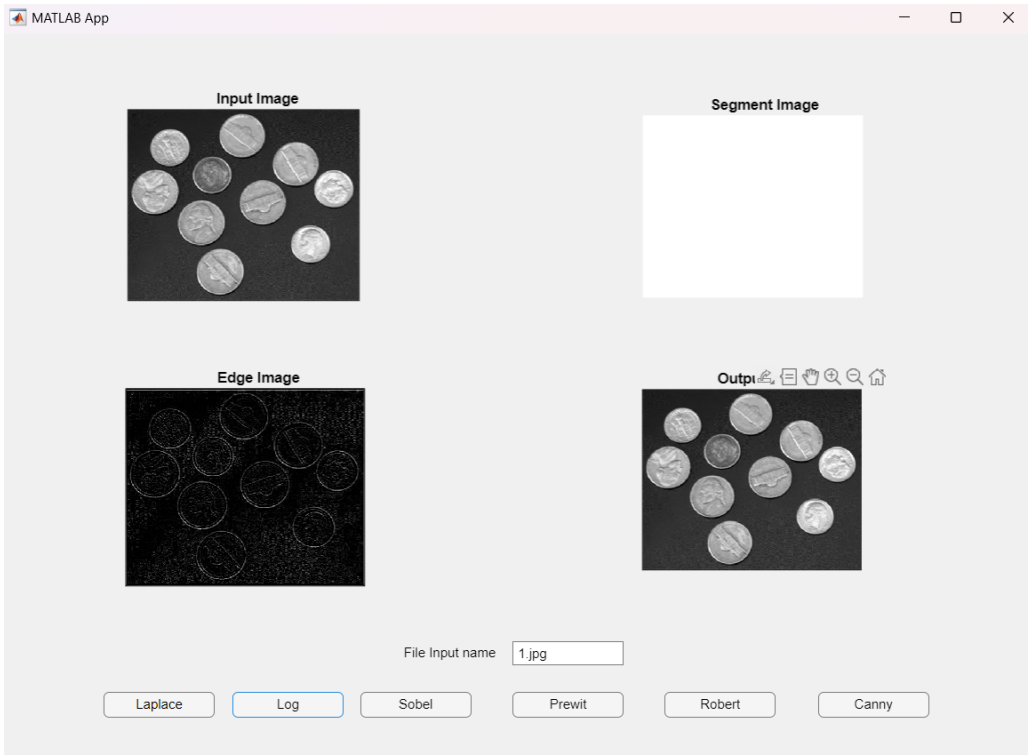
Cara kerja Laplace Edge Detection adalah dengan mengkonvolusi citra asli dengan operator Laplacian untuk mengidentifikasi perubahan cepat dalam intensitas piksel, yang menyoroti tepi dan detail citra.

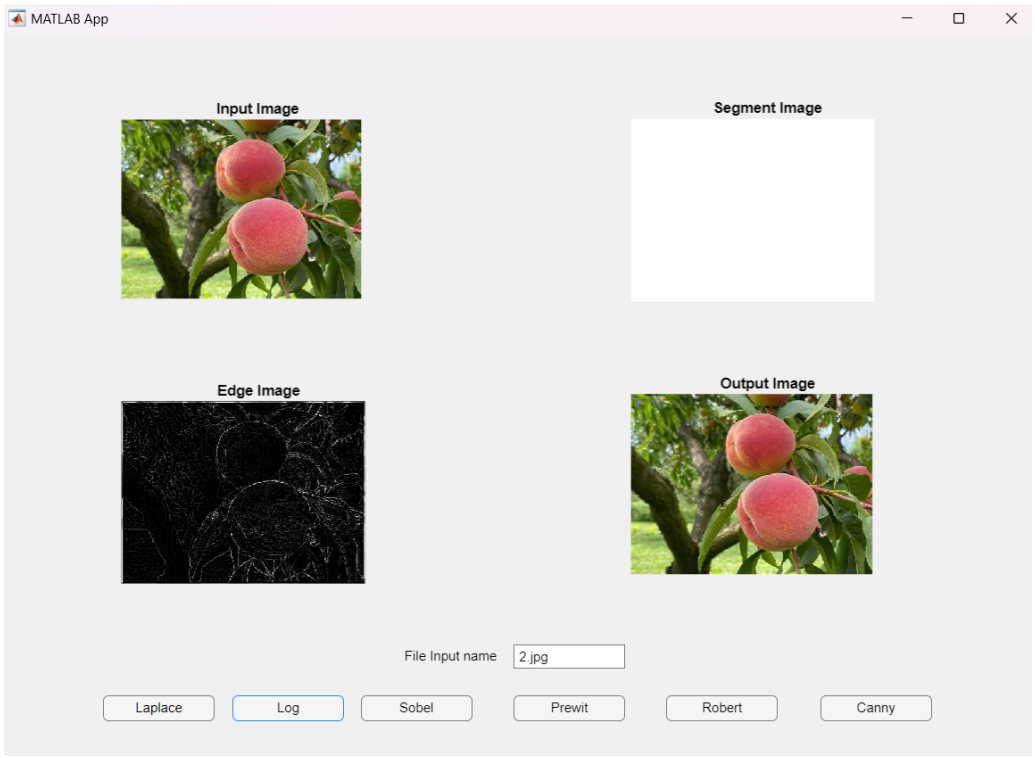
b. LOG(laplace of gaussian) Edge Detection

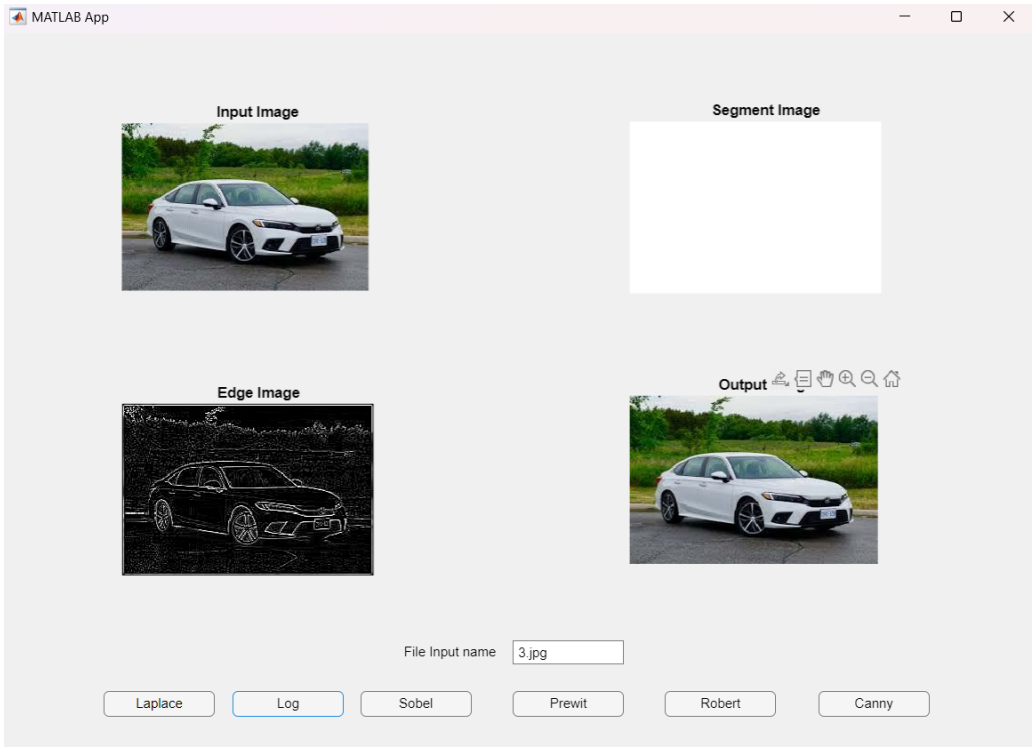
i. Kode Program

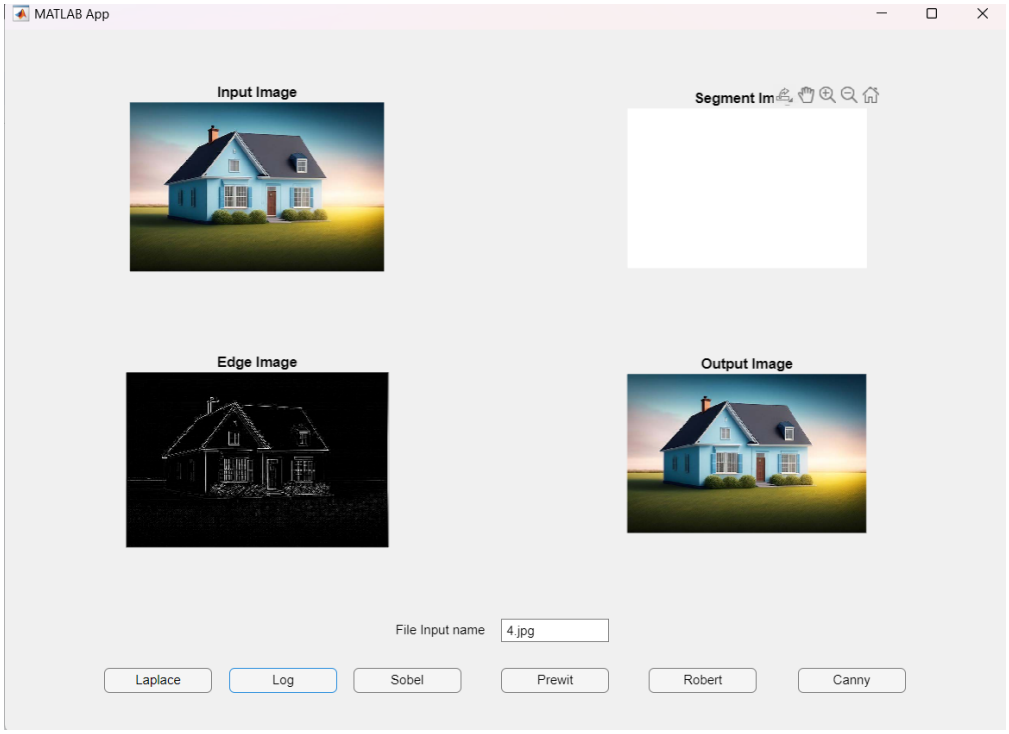
```
% Log Laplace of Gaussian
function result = edgeLog(img_input)
        % Menggunakan filter LoG (Laplace of Gaussian) dengan kernel default
        % H = [0 1 0; 1 -4 1; 0 1 0];
        h = fspecial('log');
        result = uint8(convn(double(img_input), double(h)));
end
```

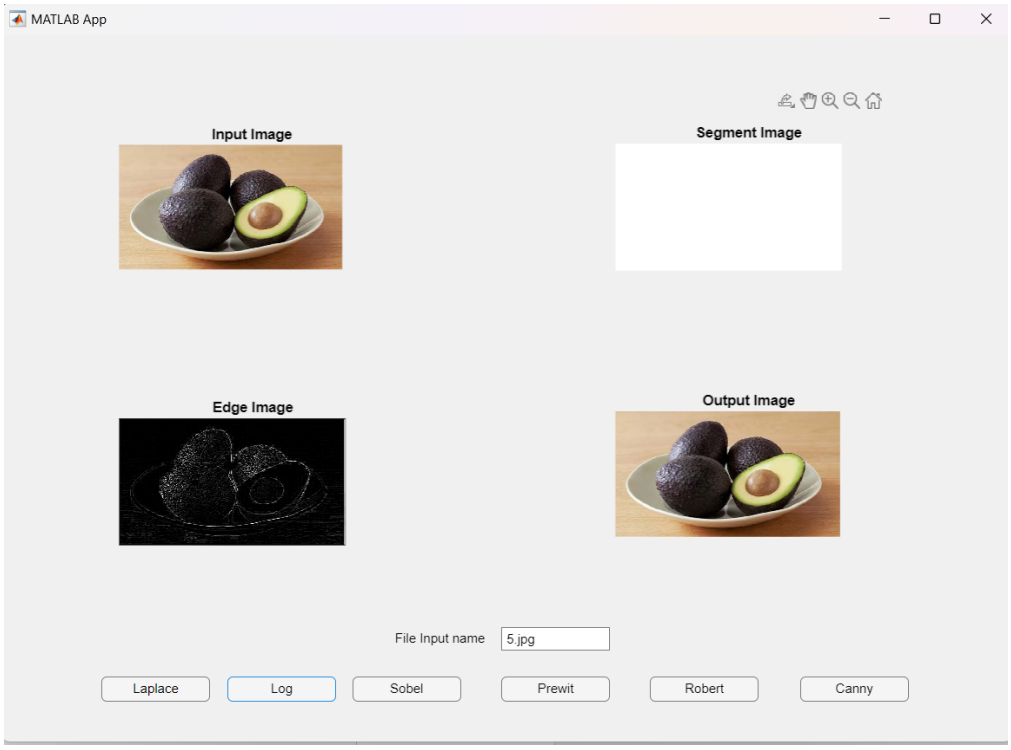
ii. Hasil Eksekusi

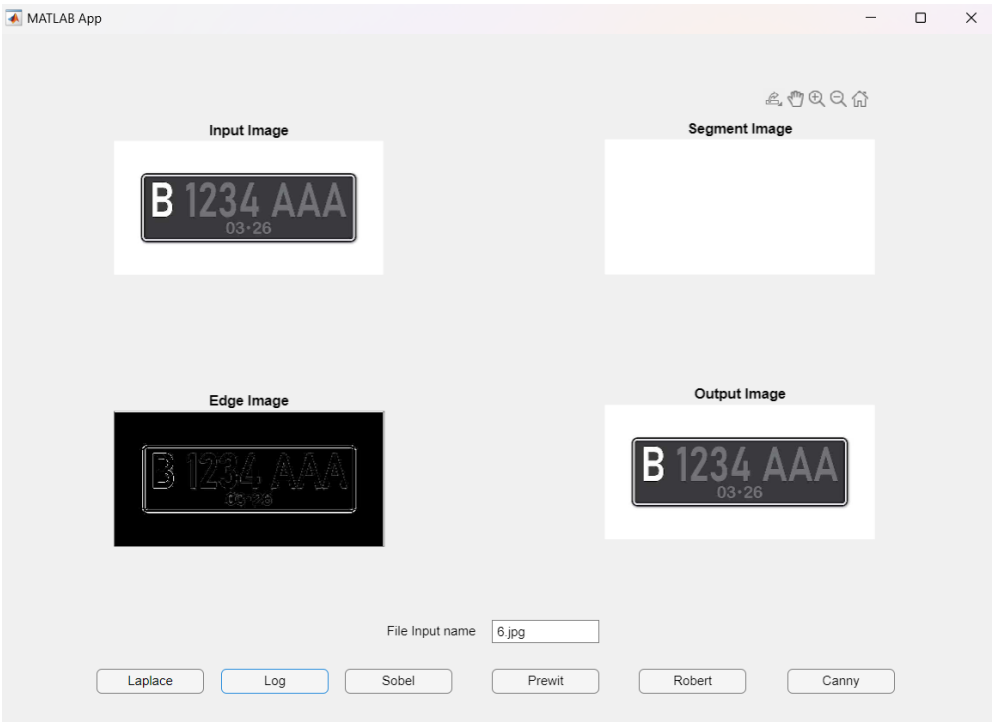


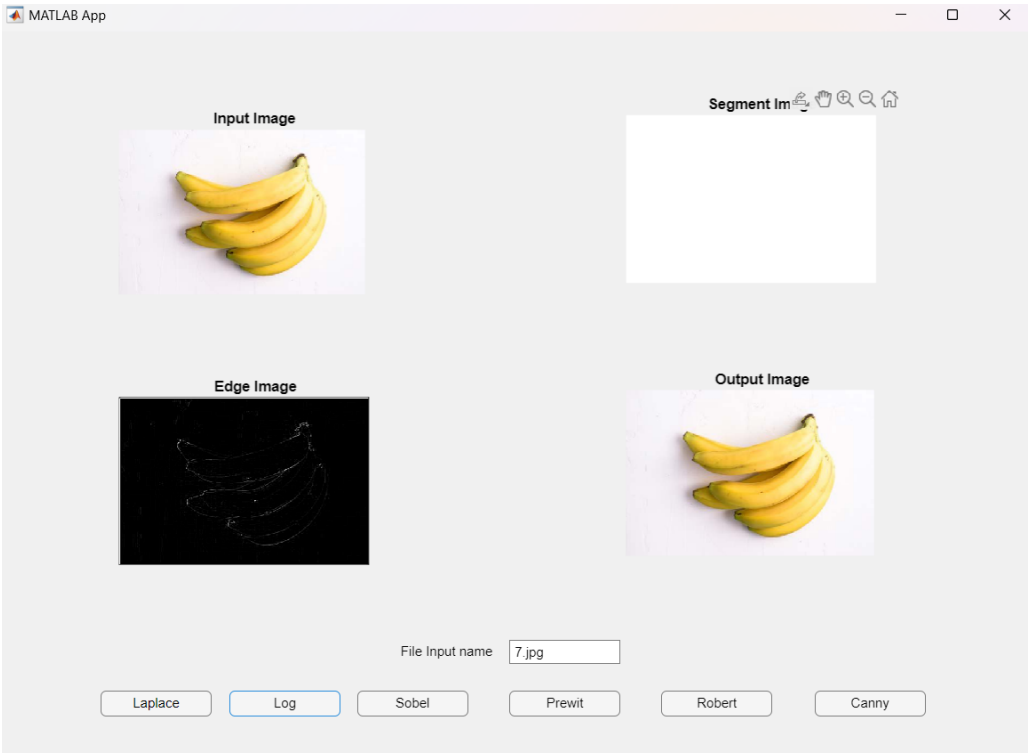

MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
2.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output
File Input name
3.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Im
Edge Image
Output Image
File Input name
4.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
5.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny





iii. Analisis cara kerja

LOG(laplace of gaussian) edge detection gagal men-segmentasi semua image sample.

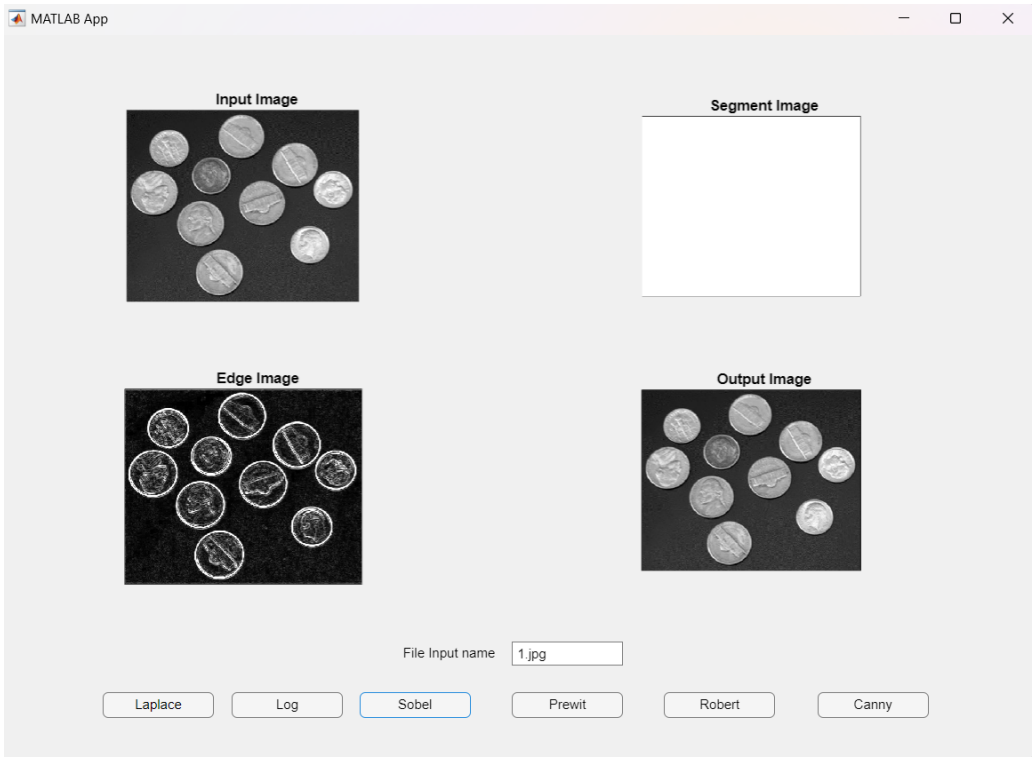
Cara kerja LOG (Laplace of Gaussian) Edge Detection adalah dengan mengkonvolusi citra asli dengan filter Laplacian of Gaussian untuk menyoroti perubahan gradien citra, yang memungkinkan pendeteksian tepi dan fitur tajam pada berbagai skala spasial.

c. Sobel Edge Detection

i. Kode Program

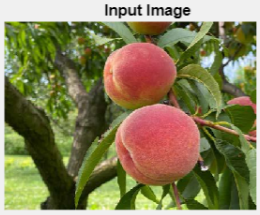
```
% sobel Edge detenction (sesuai slide)
function result = edgeSobel(img_input)
    % Menggunakan operator Sobel untuk mendeteksi tepi
    % sobelEdge = edge(grayImage, 'sobel');
    Sx = [-1 0 1; -2 0 2; -1 0 1];
    Sy = [1 2 1; 0 0 0; -1 -2 -1];
    Jx = convn(double(img_input), double(Sx), 'same');
    Jy = convn(double(img_input), double(Sy), 'same');
    result = uint8(sqrt(Jx.^2 + Jy.^2));
end
```

ii. Hasil Eksekusi



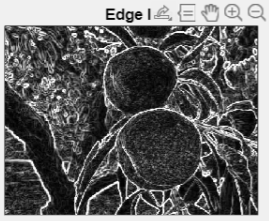
MATLAB App

Input Image

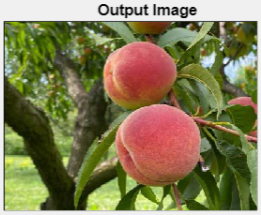


Segment Image

Edge I



Output Image



File Input name 2.jpg

Laplace | Log | Sobel | Prewit | Robert | Canny

MATLAB App

Input Image

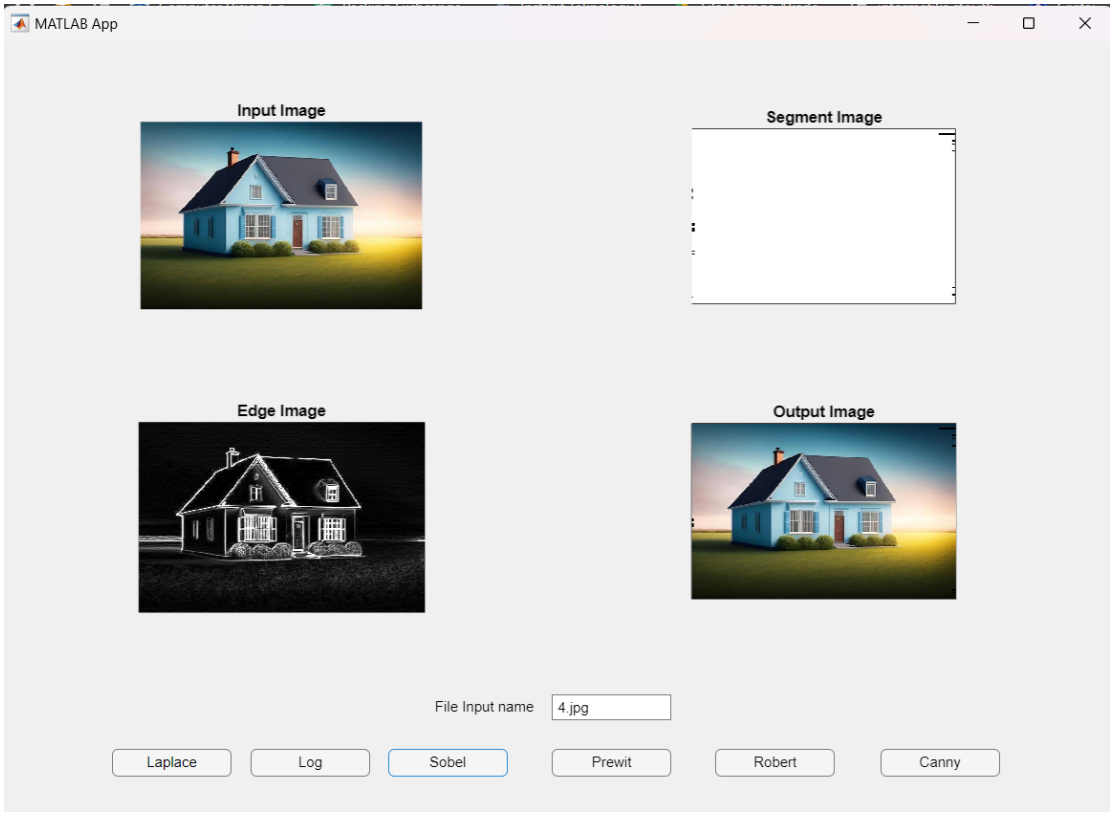
Segment Image

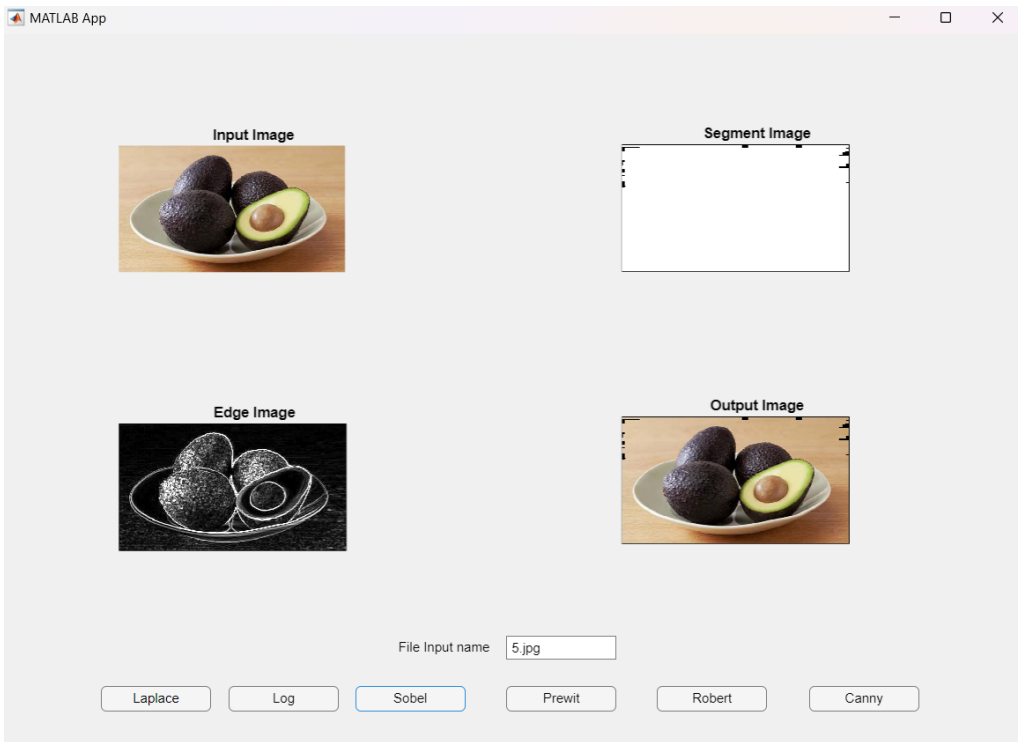
Edge I

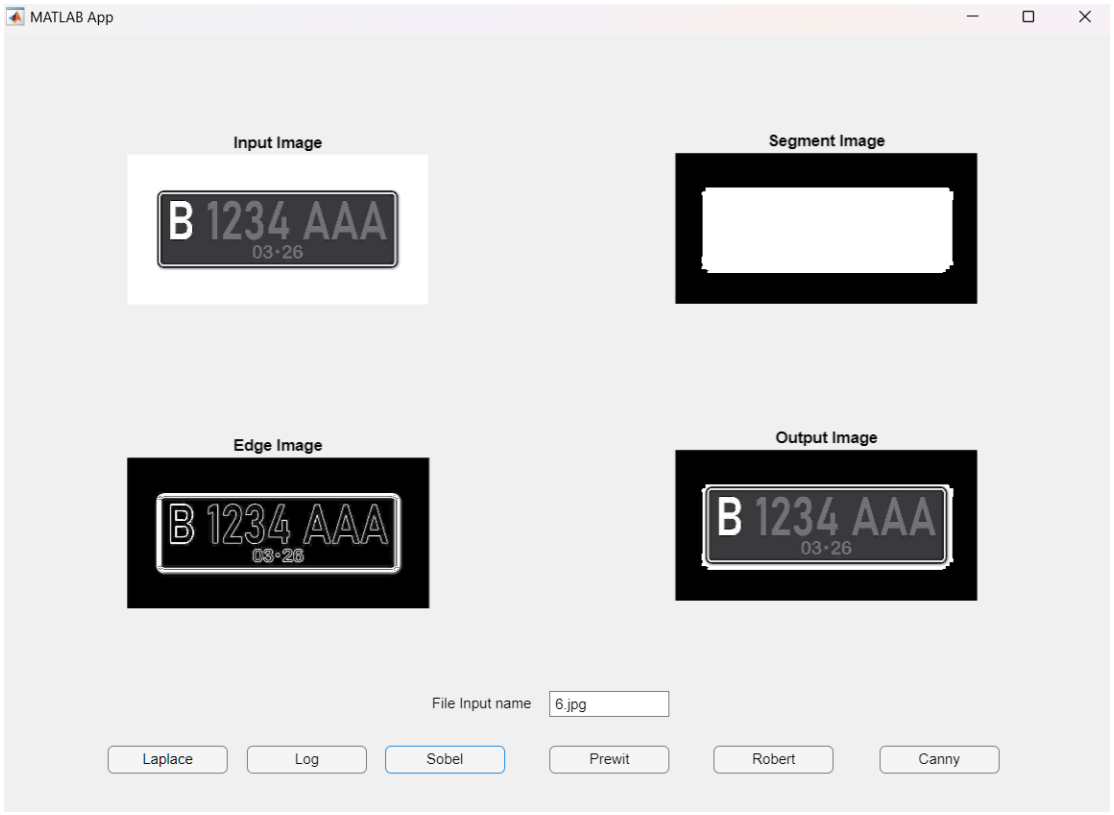
Output Image

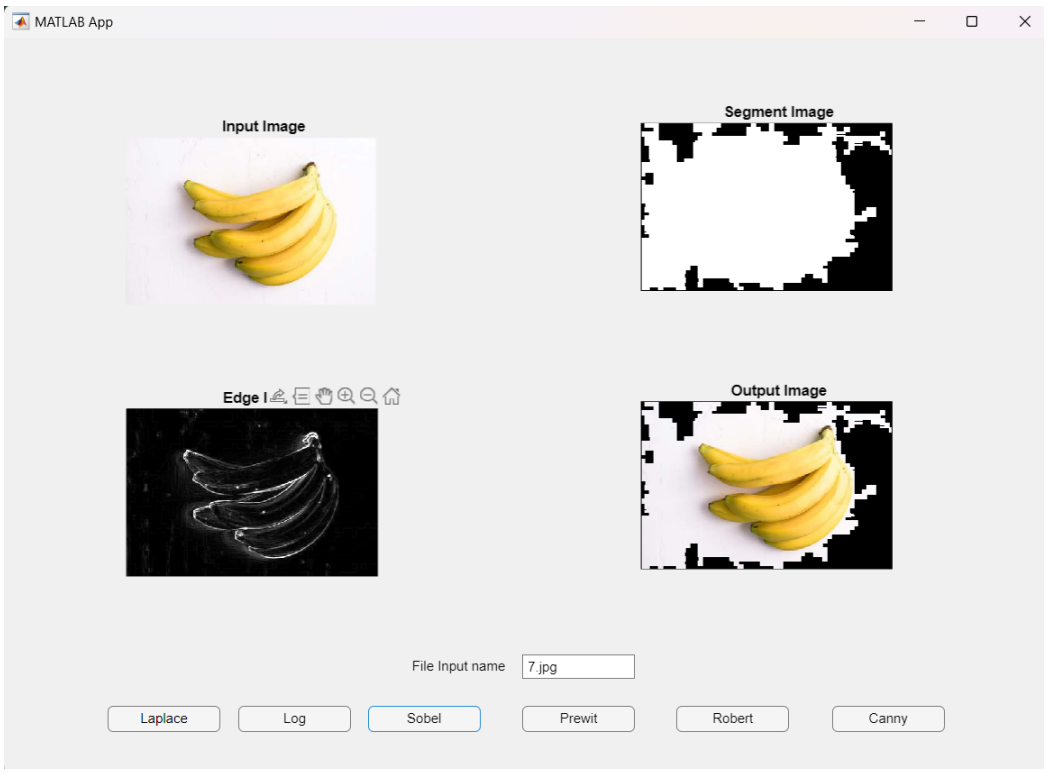
File Input name 3.jpg

Laplace | Log | Sobel | Prewit | Robert | Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
4.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
5.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny





d. Analisis cara kerja

Sobel edge detection berhasil men-segmentasi image mudah (6.jpg) namun kesulitan untuk mensegmentasi image yang lebih sulit.

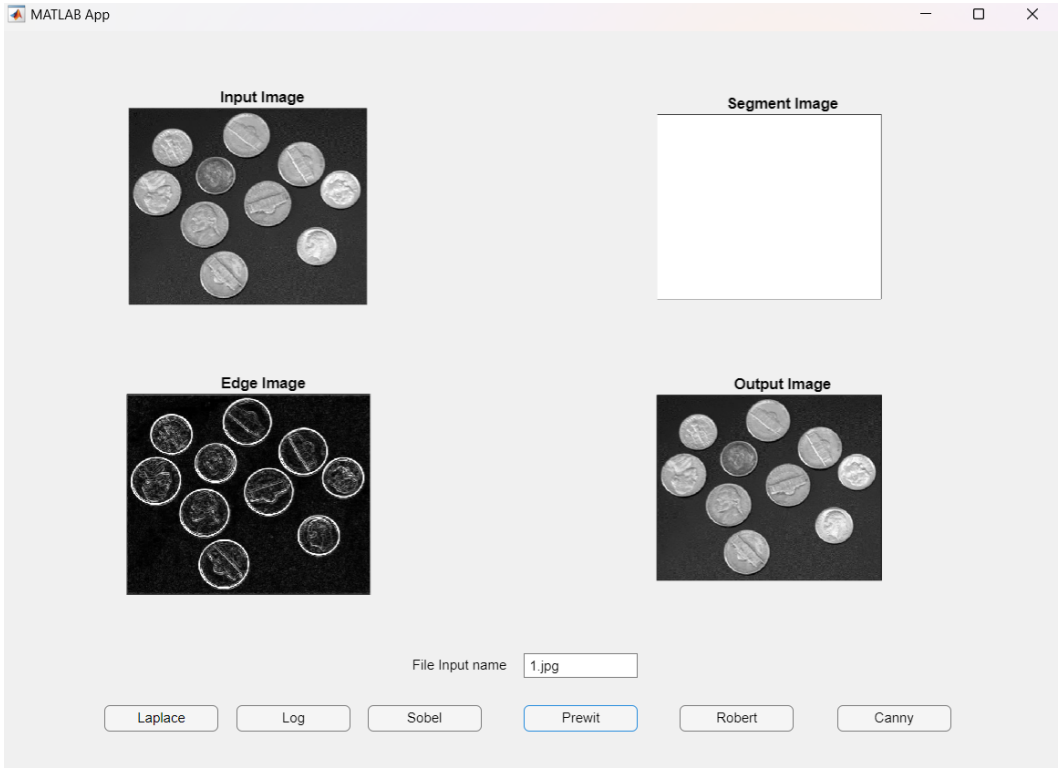
Cara kerja Sobel Edge Detection adalah dengan menggunakan sepasang operator konvolusi yang lebih sensitif terhadap perubahan gradien dalam arah horizontal dan vertikal untuk mendeteksi tepi dalam gambar.

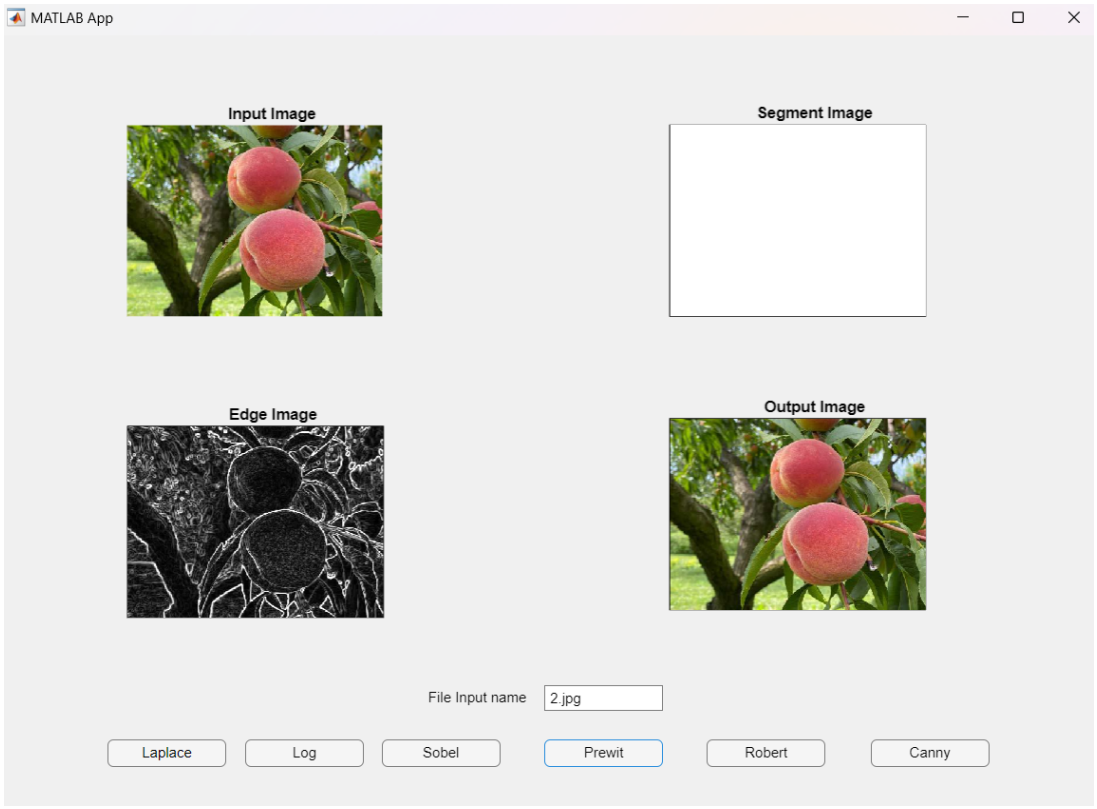
e. Prewitt Edge Detection

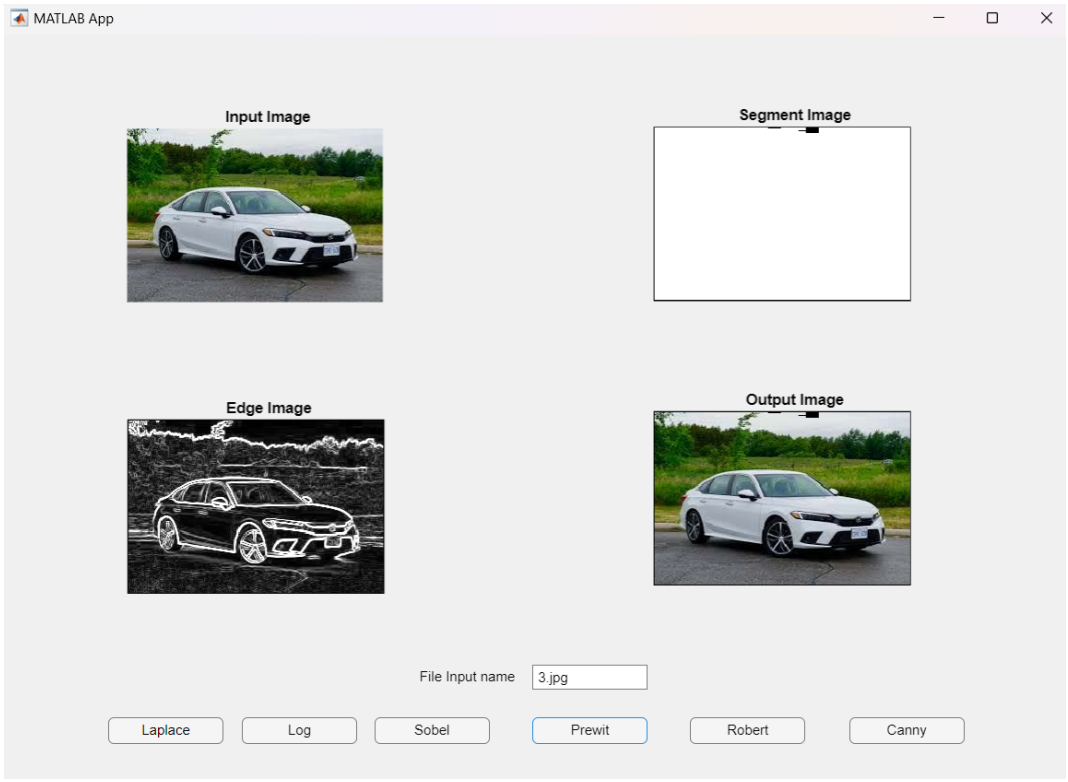
i. Kode Program

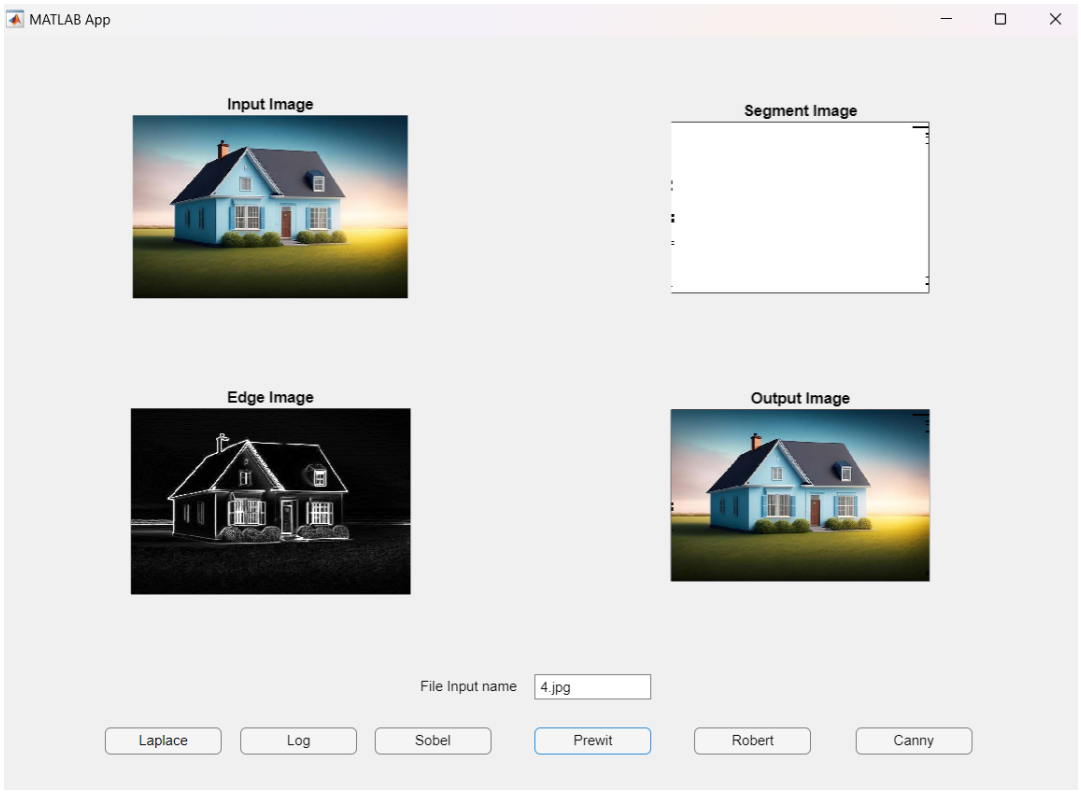
```
% Prewitt Edge Detection
function result = edgePrewit(img_input)
    % Menggunakan operator Prewitt untuk mendeteksi tepi
    % prewittEdge = edge(grayImage, 'prewitt');
    Px = [-1 0 1; -1 0 1; -1 0 1];
    Py = [-1 -1 -1; 0 0 0; 1 1 1];
    Jx = convn(double(img_input), double(Px), 'same');
    Jy = convn(double(img_input), double(Py), 'same');
    result = uint8(sqrt(Jx.^2 + Jy.^2));
end
```

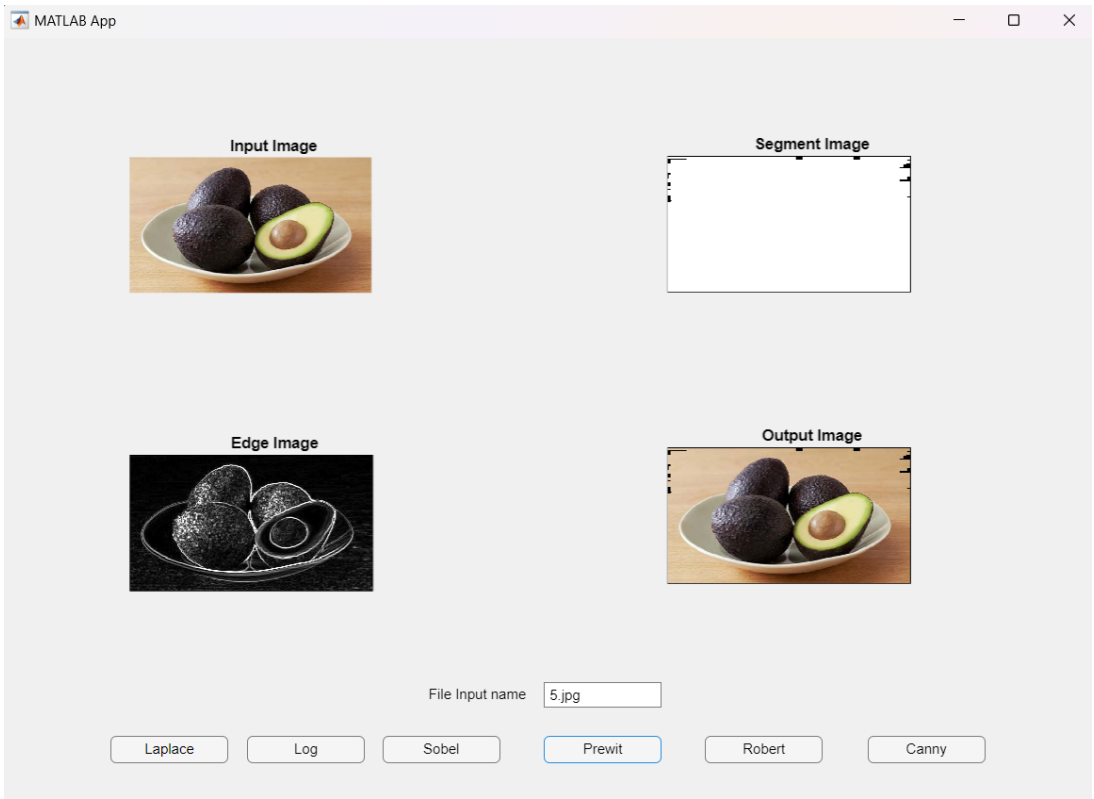
ii. Hasil Eksekusi

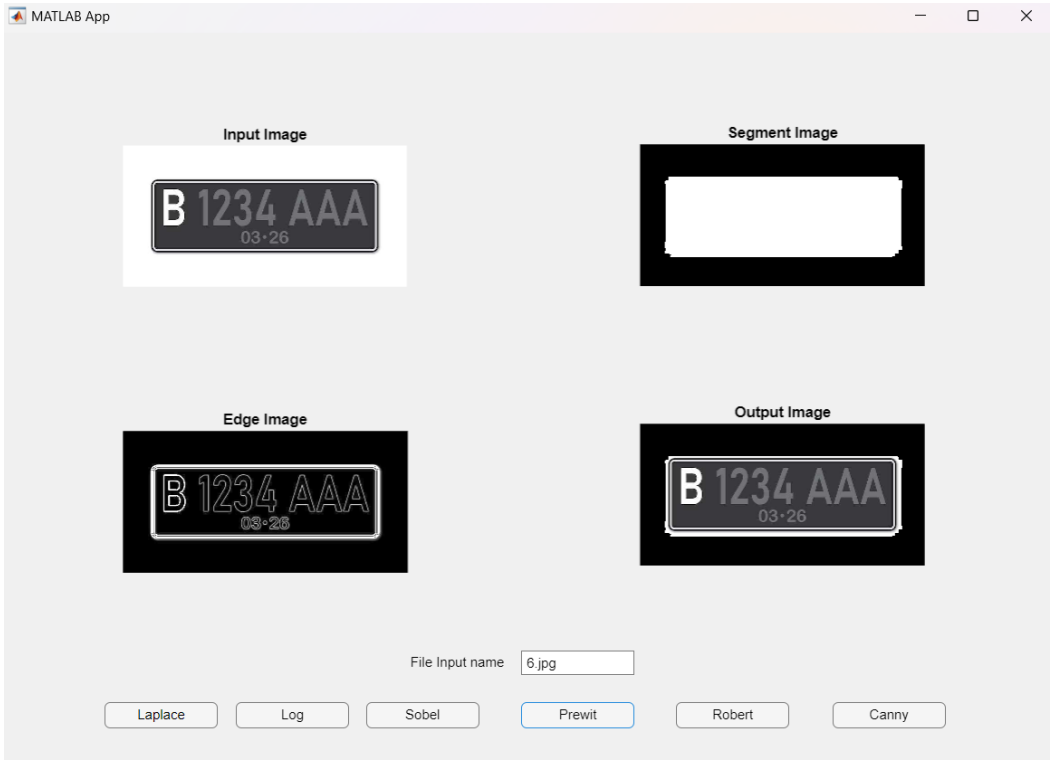


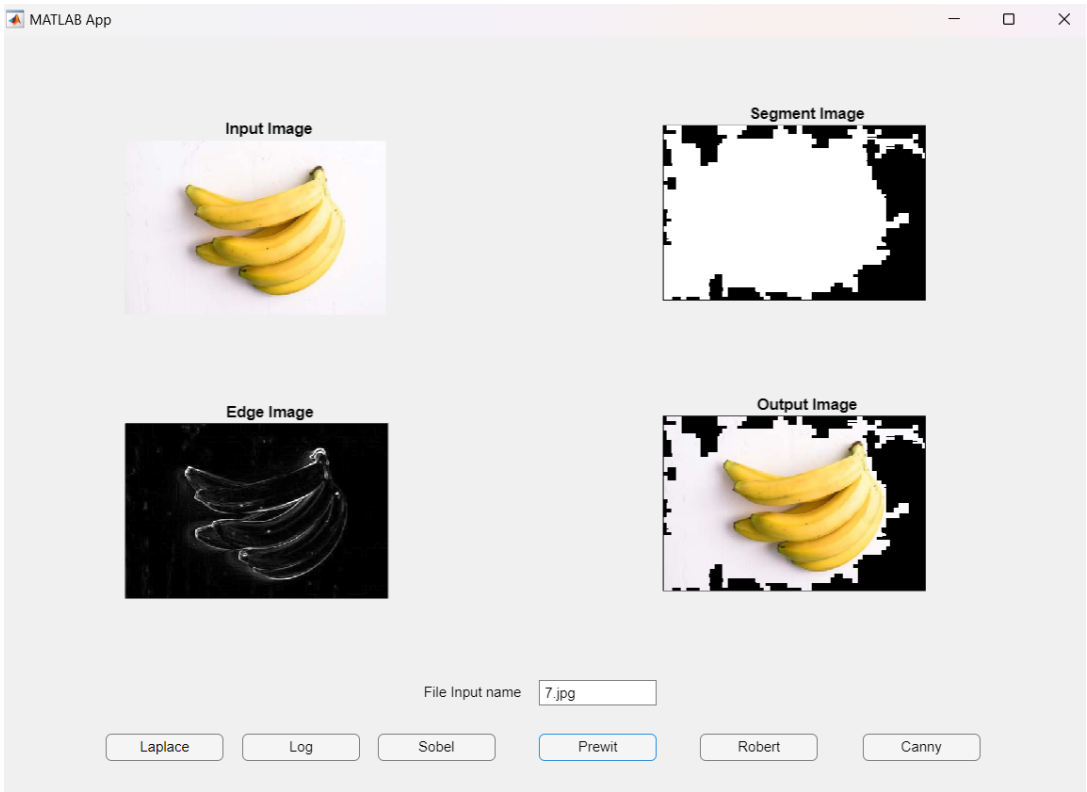

MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
2.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
3.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
4.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
5.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
6.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
7.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny

iii. Analisis cara kerja

Prewitt edge detection berhasil men-segmentasi image mudah (6.jpg) namun kesulitan untuk mensegmentasi image yang lebih sulit.

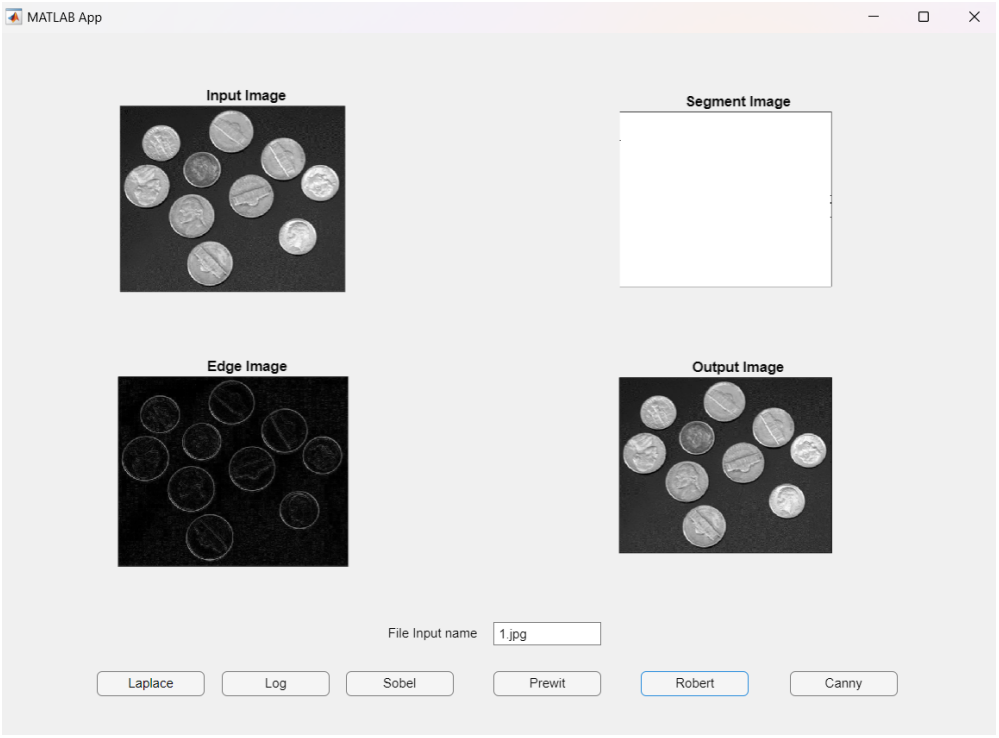
Cara kerja Prewitt Edge Detection adalah dengan menggunakan sepasang operator konvolusi untuk menghitung gradien citra dalam arah horizontal dan vertikal, kemudian menggabungkan kedua hasil untuk mendeteksi tepi dalam berbagai arah.

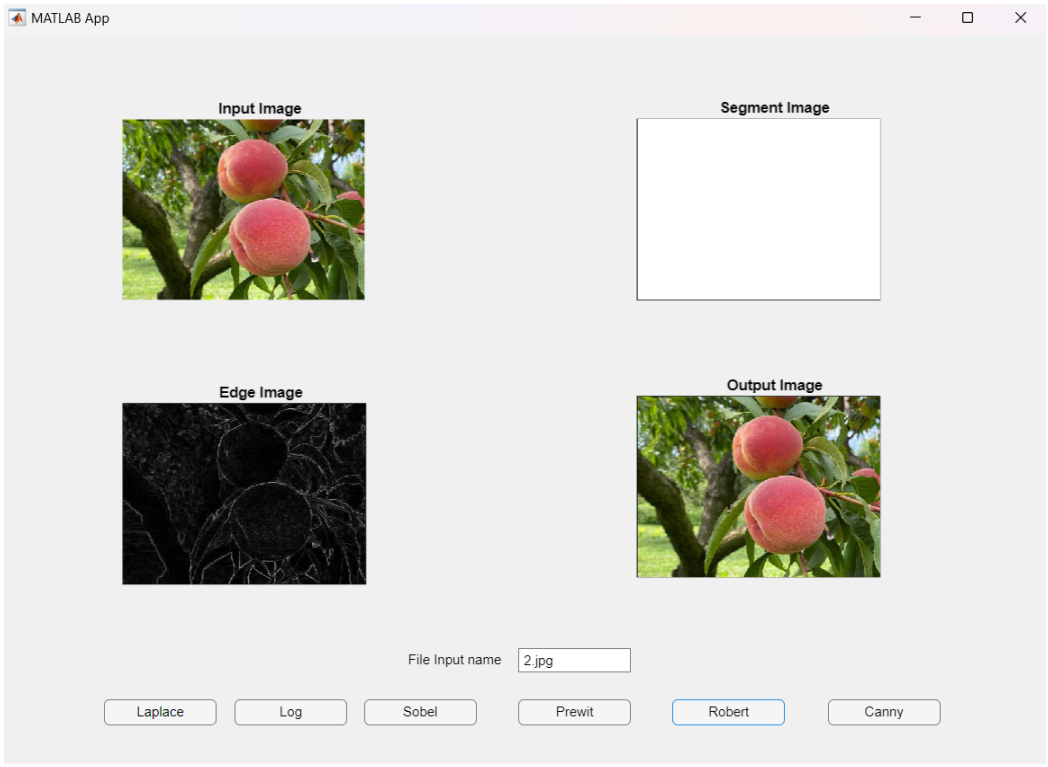
f. Robert Edge Detection

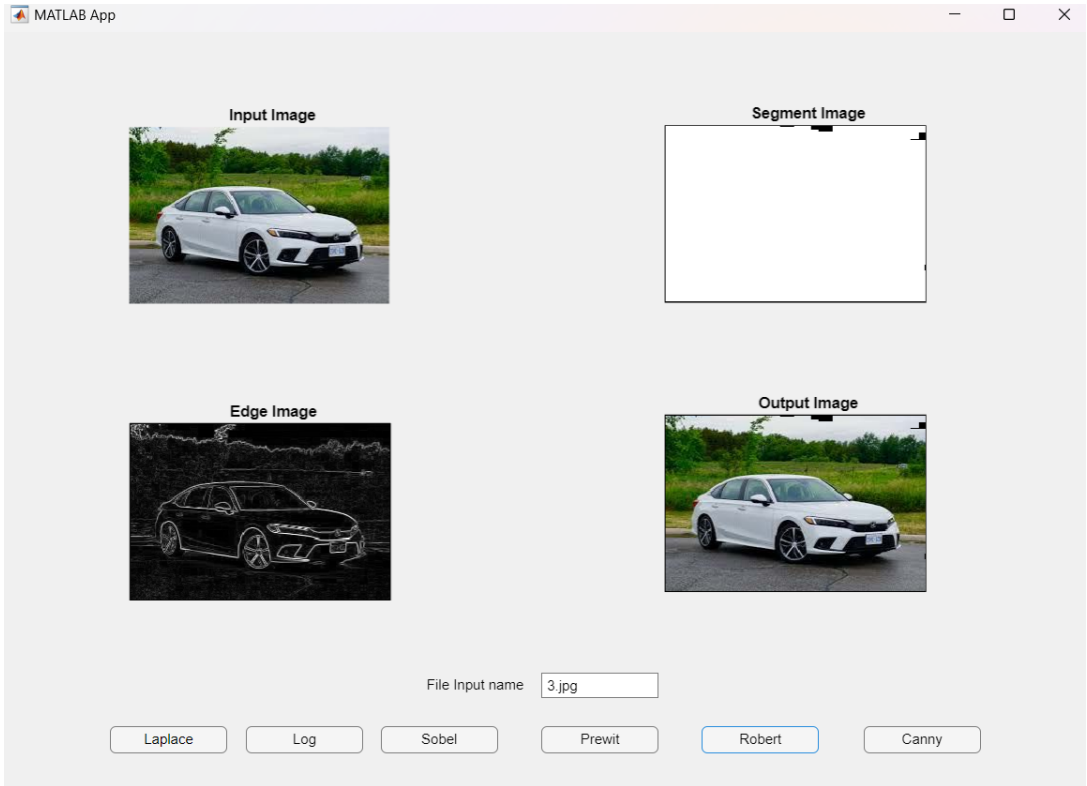
i. Kode Program

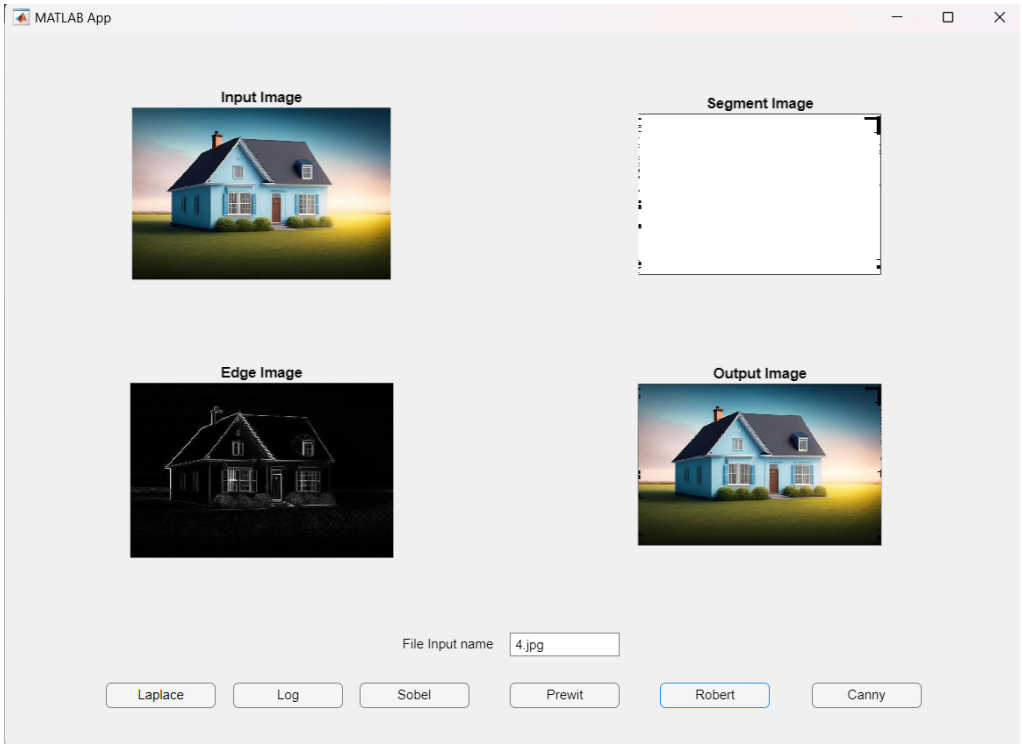
```
% Roberts edge detenction
function result = edgeRoberts(img_input)
    % Menggunakan operator Roberts untuk mendeteksi tepi
    % robertsEdge = edge(grayImage, 'roberts');
    Rx = [1 0; 0 -1];
    Ry = [0 1; -1 0];
    Jx = convn(double(img_input), double(Rx), 'same');
    Jy = convn(double(img_input), double(Ry), 'same');
    result = uint8(sqrt(Jx.^2 + Jy.^2));
end

```

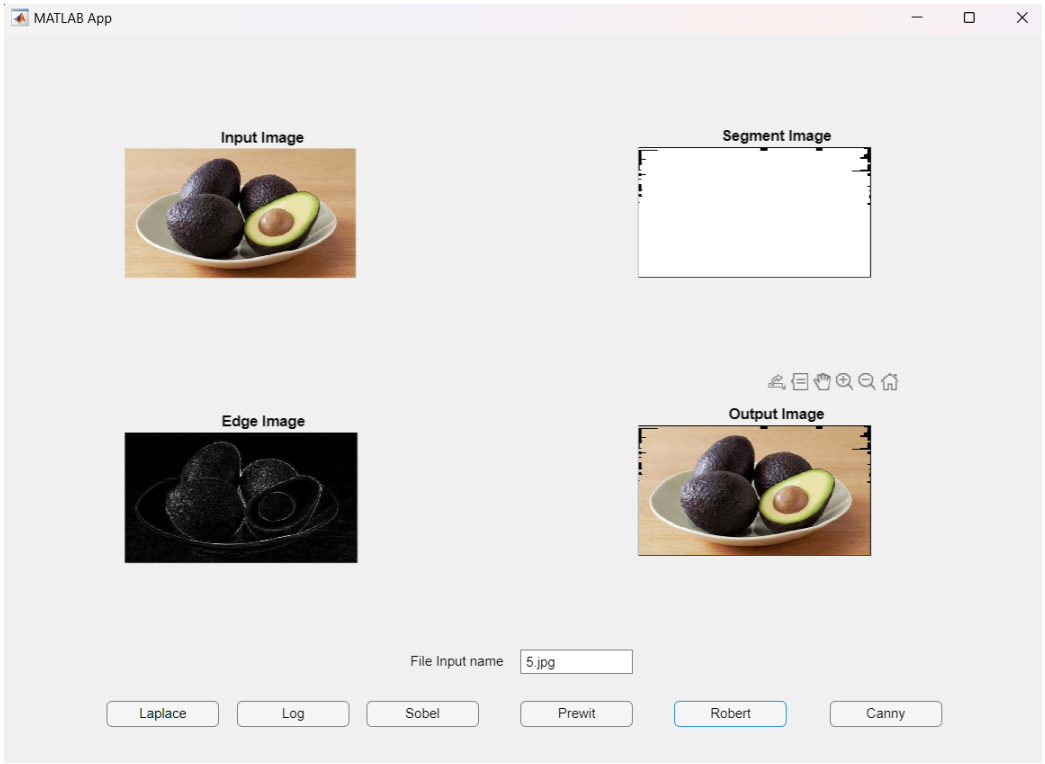
ii. Hasil Eksekusi

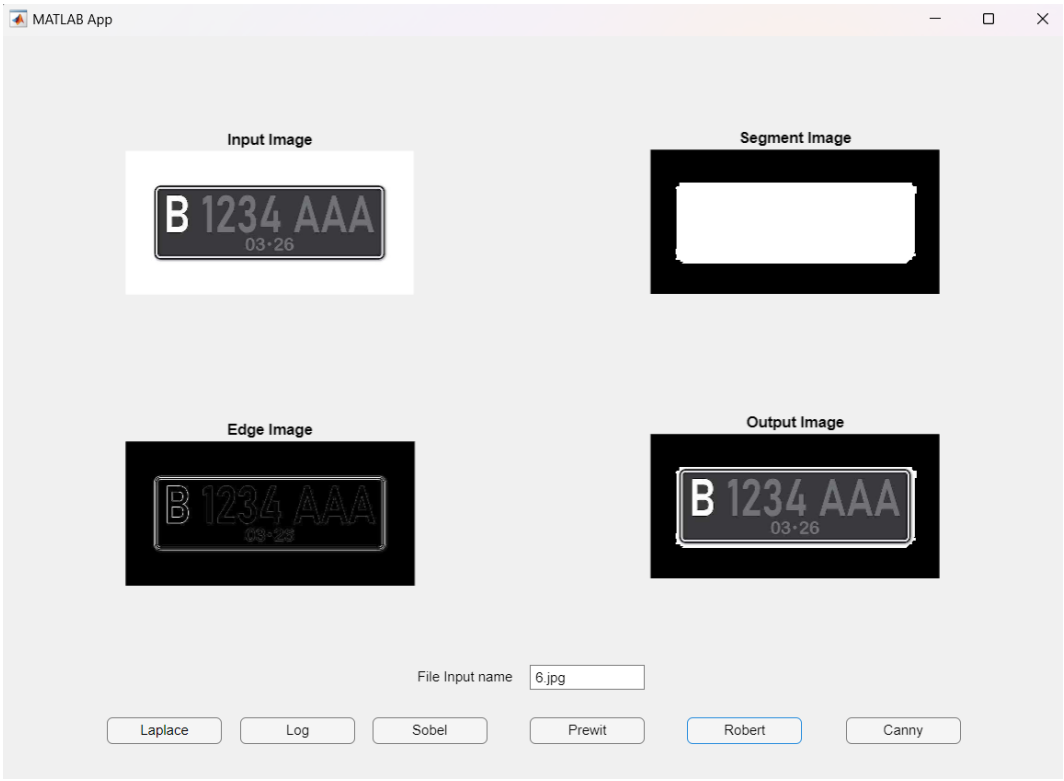


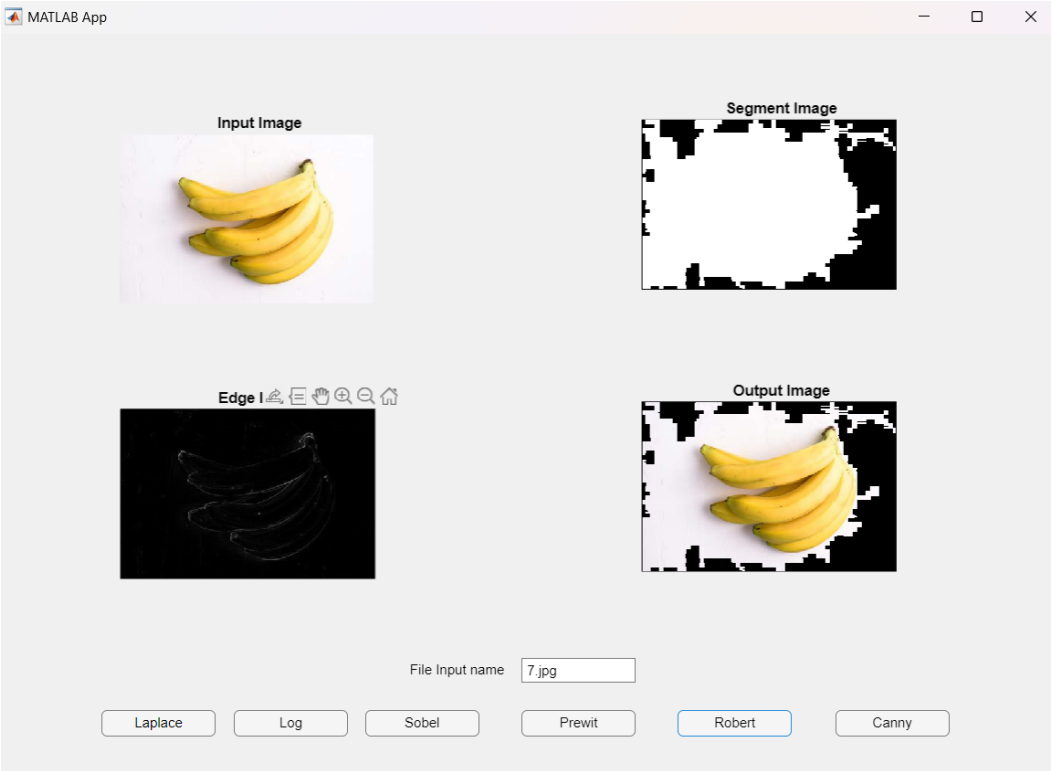

MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
2.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
3.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
4.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
5.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny





iii. Analisis cara kerja

Robert edge detection berhasil men-segmentasi image mudah (6.jpg) namun kesulitan untuk mensegmentasi image yang lebih sulit.

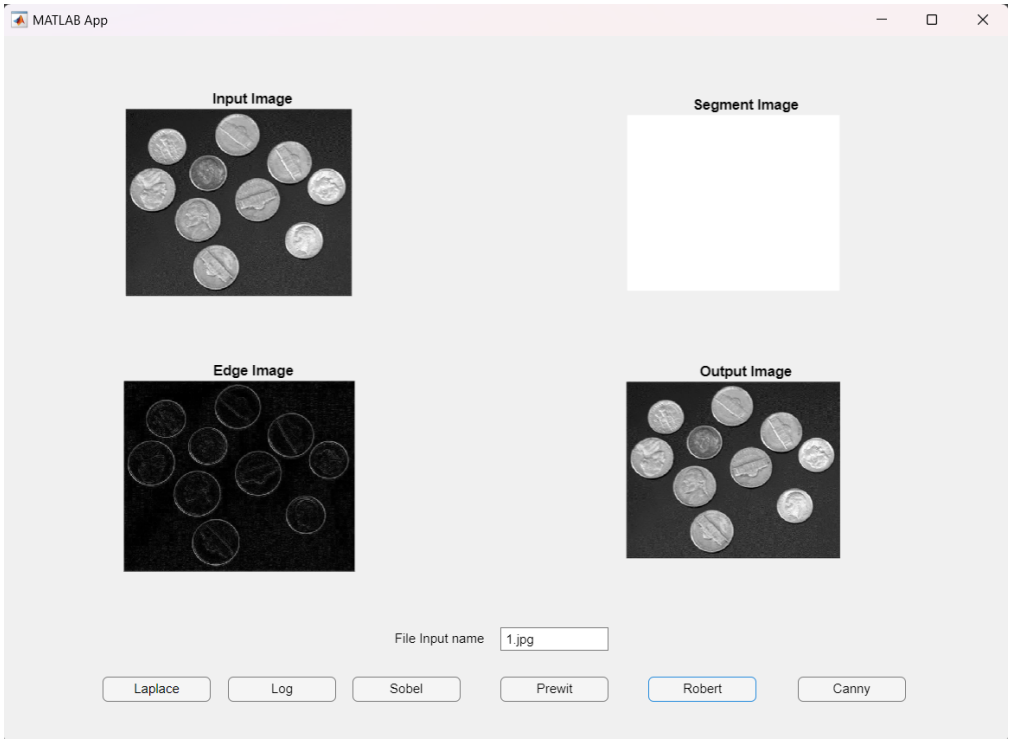
Cara kerja Robert Edge Detection adalah dengan menggunakan sepasang operator konvolusi kecil untuk menghitung perbedaan gradien antara piksel tetangga dalam arah horizontal dan vertikal dalam gambar.

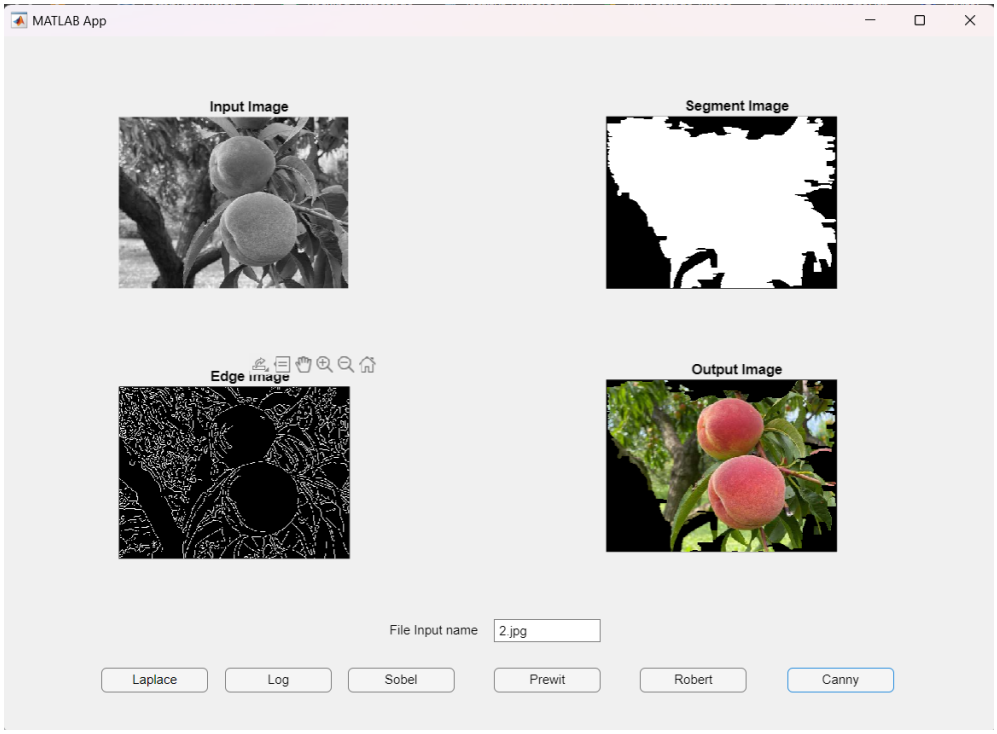
g. Canny Edge Detection

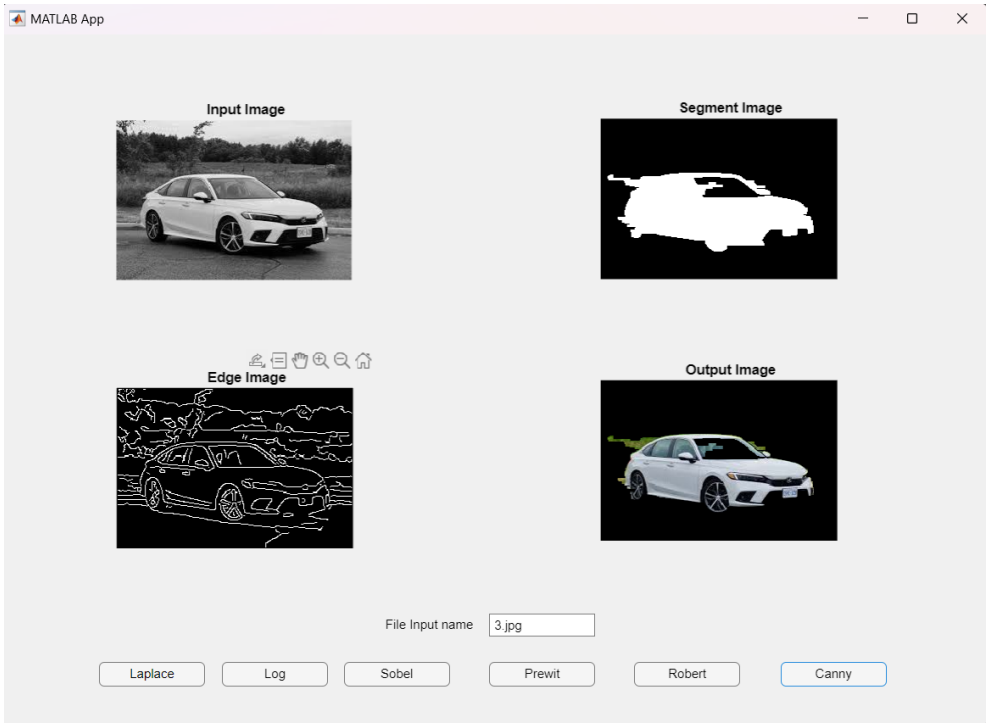
i. Kode Program

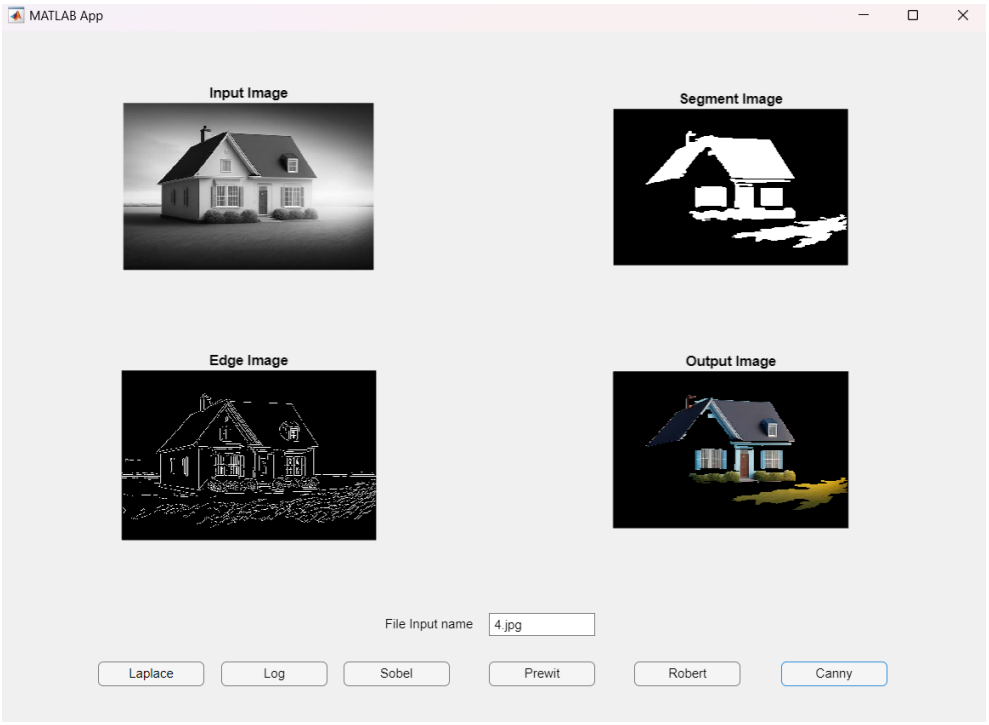
```
% using build in canny edge detection then cast it into uint
function result = edgeCanny(img_input)
    % Menggunakan deteksi tepi Canny bawaan MATLAB        You, 1
    result = uint8(edge(img_input, 'canny'));
end

```

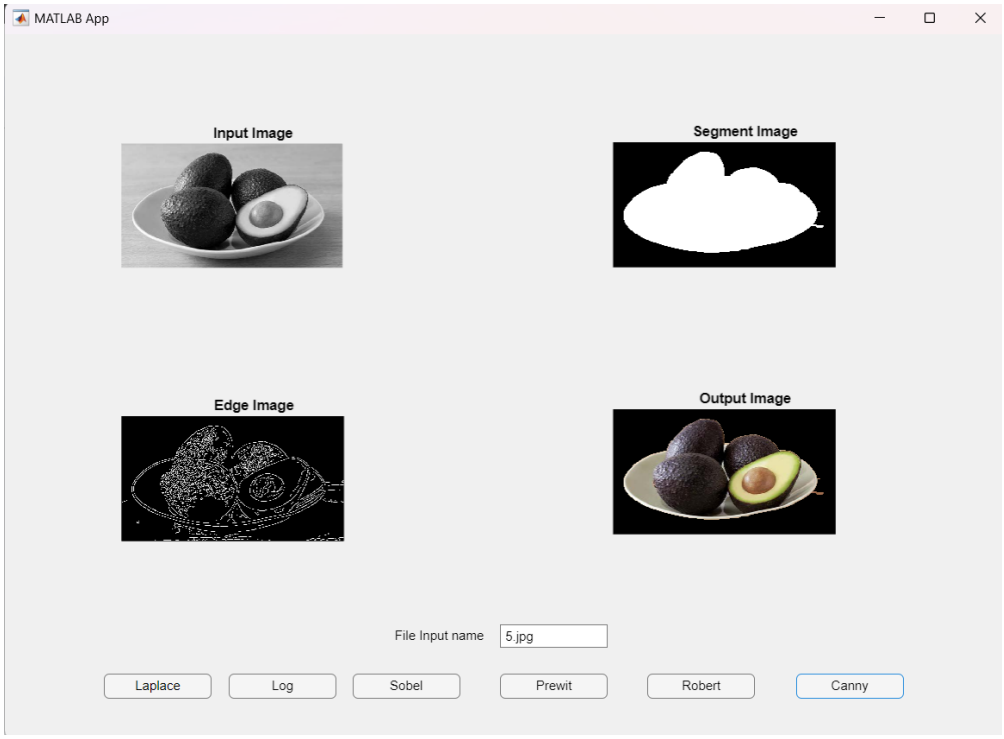
ii. Hasil Eksekusi

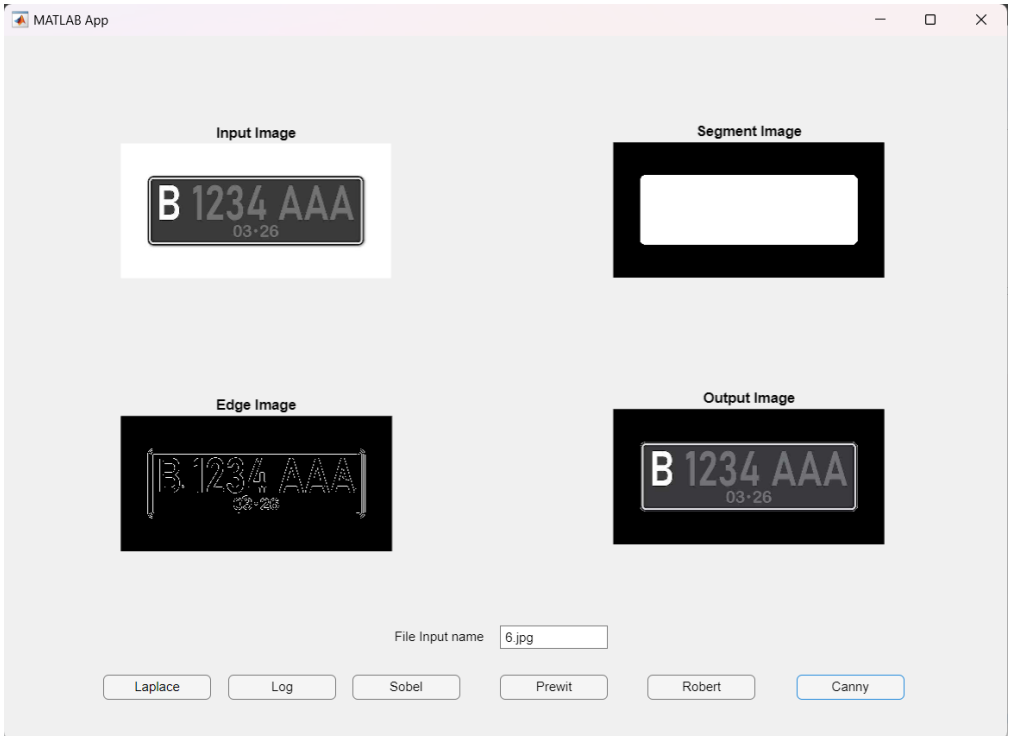


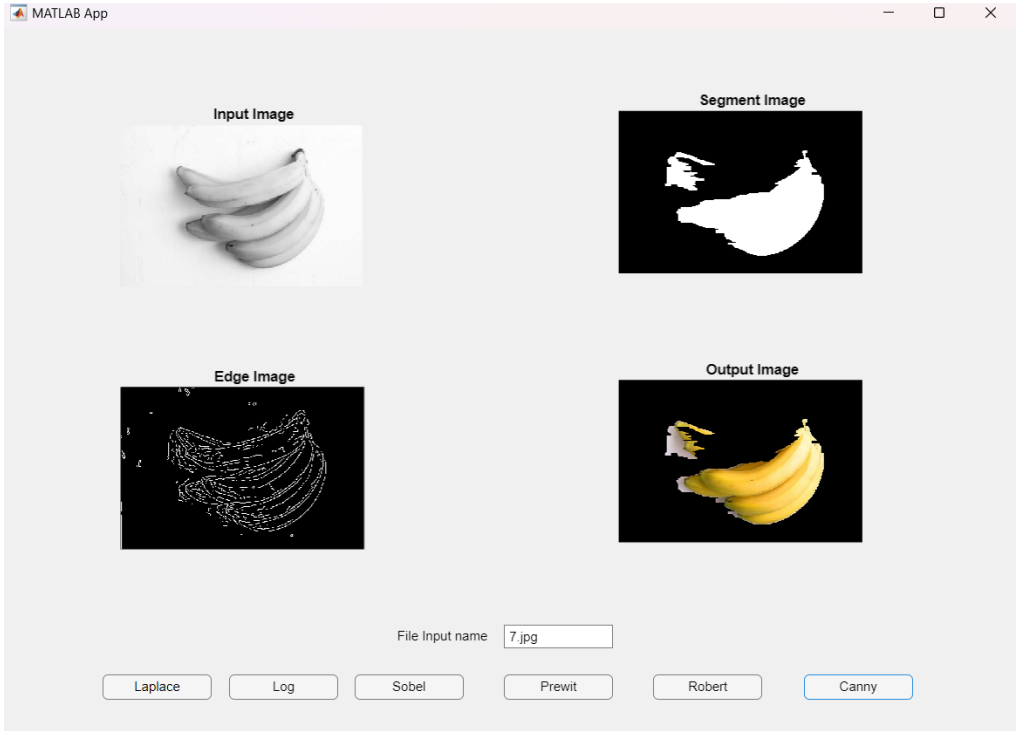

MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
2.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
3.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
4.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny


MATLAB App
Input Image
Segment Image
Edge Image
Output Image
File Input name
5.jpg
Laplace
Log
Sobel
Prewit
Robert
Canny





iii. Analisis cara kerja

Dari semua algoritma edge detection yang lain canny mendapatkan performa segmentation terbaik.

Cara kerja canny edge detection adalah dengan mengurangi noise, menghitung gradien citra, menerapkan penyaringan non-maksimum, dan mengaplikasikan ambang biner untuk mengidentifikasi dan menonjolkan tepi pada gambar.

h. Segmentation

i. Kode Program

```
You, 1 second ago | 1 author (You)
filename = app.FileInputnameEditField.Value
% Buka file
inputImage = imread(filename);
% konversi ke grayscale apabila rgb
if ndims(inputImage) == 2
    inputImageGrey = inputImage
elseif ndims(inputImage) == 3
    inputImageGrey = rgb2gray(inputImage)
end
imshow(inputImageGrey, 'Parent', app.UIAxes);
% lakukan edge detection
resultImageLaplace = edgeLaplace(inputImageGrey);
resultImageLog = edgeLog(inputImageGrey);
resultImageSobel = edgeSobel(inputImageGrey);
resultImagePrewit = edgePrewit(inputImageGrey);
resultImageRoberts = edgeRoberts(inputImageGrey);
resultImageCanny = edgeCanny(inputImageGrey);

% cast ke uint8
edgeImage = uint8(resultImageCanny)
imshow(edgeImage, 'Parent', app.UIAxes_2);
labeledImage = bwlabel(edgeImage);

% tutup disconnected edges
closedImage = imclose(labeledImage,strel('line',10,0));

% Fill edges
filledImage = imfill(closedImage, 'holes');

% filter apabila ukuran kecil
openedImage = imopen(filledImage, strel(ones(3,3)));
maskImage = bwareaopen(openedImage,1500);

imshow(maskImage, 'Parent', app.UIAxes_3);
segmentedImage = inputImage
% cut image dalam segment       You, now • Uncommitted changes
if ndims(inputImage) == 2
    segmentedImage(~maskImage) = 0;
elseif ndims(inputImage) == 3
    colorMask = cat(3, maskImage, maskImage, maskImage);
    segmentedImage(~colorMask) = 0;
end

```

ii. Hasil Eksekusi

Hasil eksekusi dapat dilihat pada section - section sebelumnya

iii. Analisis cara kerja

Hasil eksekusi segmentasi memiliki performa yang cukup baik apabila menggunakan edge detection canny atau gambar sederhana seperti 6.jpg

Cara kerja dari segmentasi adalah pertama menggunakan algoritma edge detection tertentu kemudian edge image di cast ke uint8 lalu dari edge tersebut agar didapatkan area menggunakan fungsi bawaan imclose ditutup, lalu di fill agar didapatkan are. Setelah didapatkan segmen - regmen area image awal kemudian di cut apabila mask == 0.

3. Alamat Github

https://github.com/ronggurmahendra/Tugas3-Citra-2023.git