

LAPORAN TUGAS KECIL 3
IF4073 INTERPRETASI DAN PENGOLAHAN CITRA
Semester I Tahun 2023/2024

EDGE DETECTION DAN IMAGE SEGMENTATION
MENGGUNAKAN MATLAB GUI

Disusun oleh:

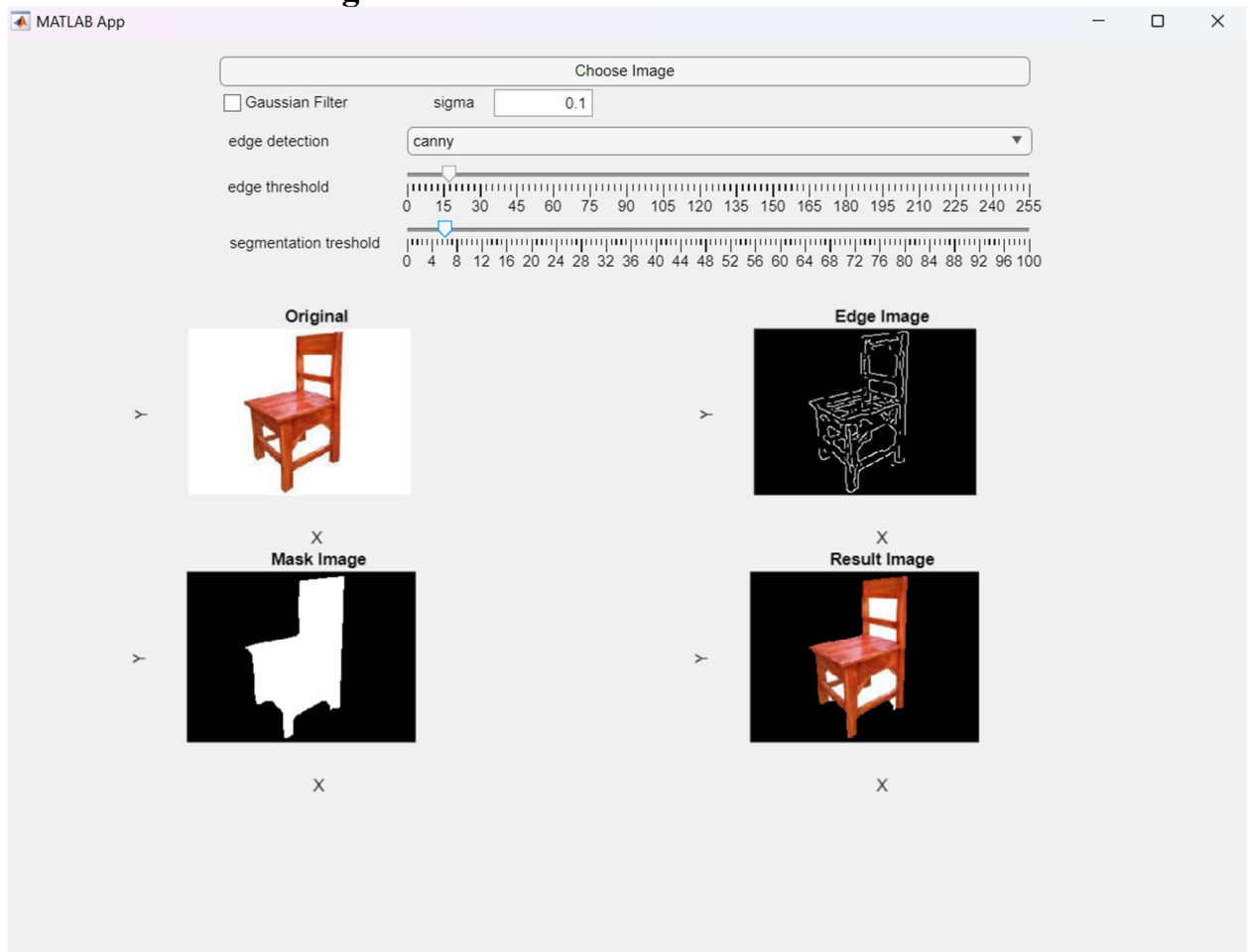
La Ode Rajuh E

(13519170)



INSTITUT TEKNOLOGI BANDUNG

1. Screenshots GUI Program



2. Program

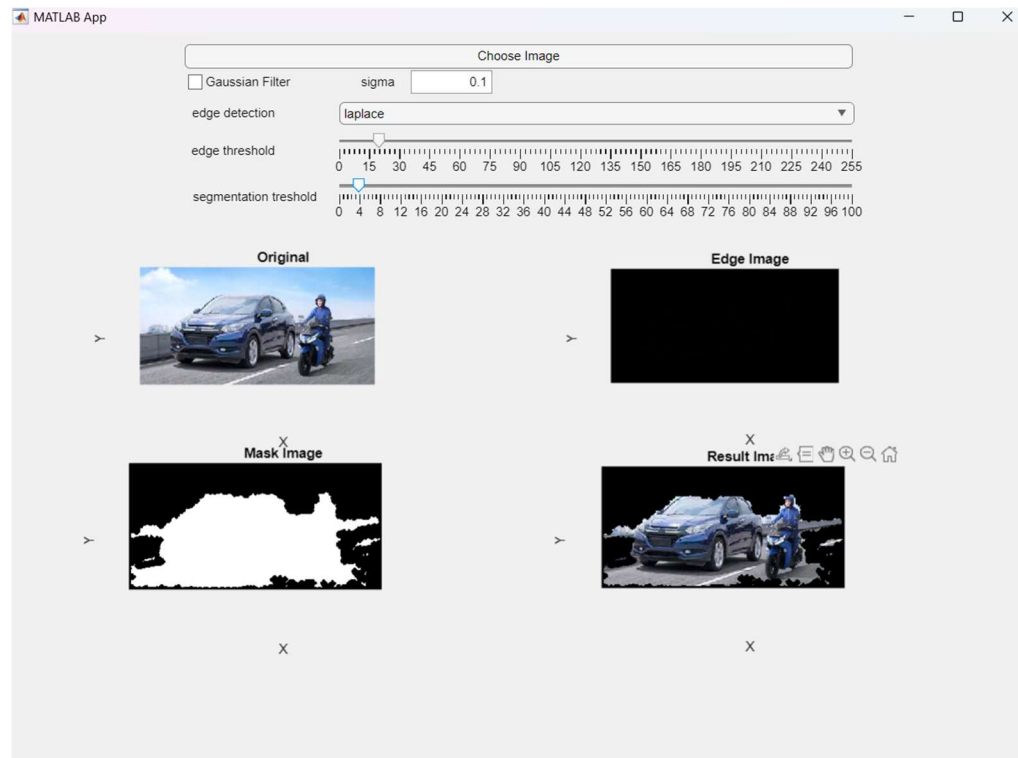
2.1 Laplace

a. Kode

```
function edgeImage = laplace(app, image, threshold)
    H = [0 1 0; 1 -4 1; 0 1 0];
    edgeImage = uint8(convn(double(image), double(H), 'same') > threshold);
end
```

Hanya dilakukan konvolusi terhadap salah satu matrix laplace. Disini ditetapkan yang dipakai adalah yang pusatnya -4, tidak dapat diubah oleh pengguna.

b. Screenshots hasil



Terdapat error dalam menampilkan edge imagenya, dan karena waktu deadline sudah dekat, belum sempat diperbaiki, namun tepinya masih bisa dipakai untuk membuat masknya.

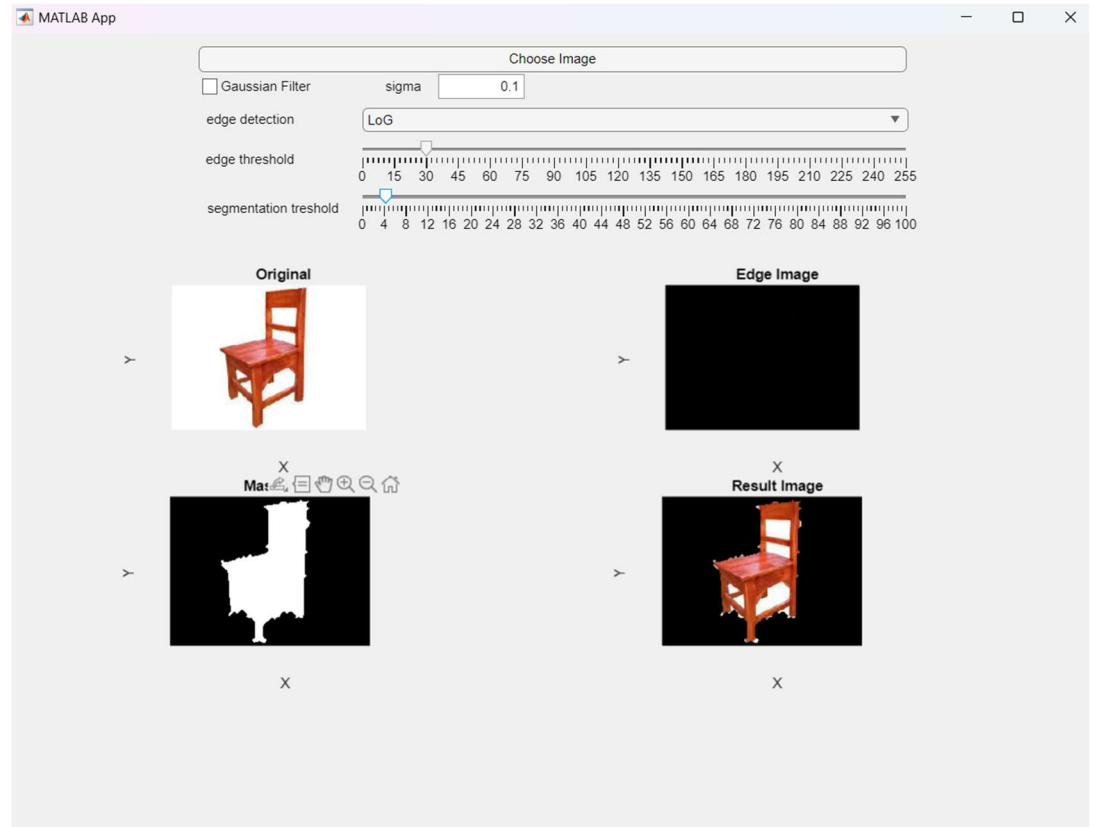
2.2 Laplace of Gaussian

a. Kode dan analisis

```
function edgeImage = LoG(app, image, threshold)|
    H = fspecial("log",5,0.5);
    edgeImage = uint8(convn(double(image), double(H), 'same') > threshold);
end
```

Hanya dilakukan convolusi terhadap matrix Laplacian of Gaussian yang diambil menggunakan fungsi `fspecial` pada matlab. Kemudian dilihat apakah lebih besar dari treshold menjadi 1 dan jika kurang menjadi 0.

b. Screenshots hasil



Sama seperti sebelumnya, untuk edgenya tidak terlihat, harusnya karena masih dalam 0 dan 1 pada derajat keabuan 256 sehingga terlihat hitam semua. Setidaknya masih berfungsi dengan cukup baik dalam membuat masknya.

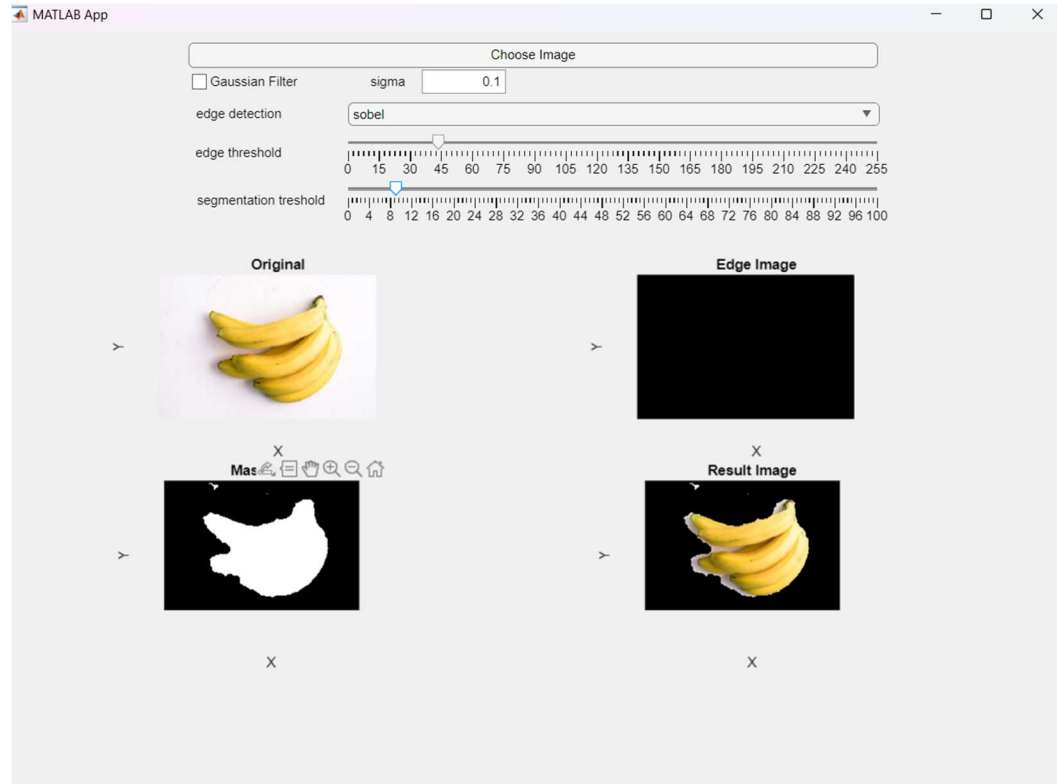
2.3 Sobel

a. Kode dan analisis

```
function edgeImage = sobel(app, image, threshold)
    Sx = [-1 0 1; -2 0 2; -1 0 1];
    Sy = [1 2 1; 0 0 0; -1 -2 -1];
    Jx = convn(double(image), double(Sx), 'same');
    Jy = convn(double(image), double(Sy), 'same');
    edgeImage = uint8(abs(Jx) + abs(Jy) > threshold);
end
```

Disini untuk menghitung kemiringan terhadap x dan terhadap y ditetapkan menggunakan matrix seperti pada kode diatas. Setelah dihitung kemiringan x dan y nya, akan dijumlahkan sebelum dibandingkan dengan ambang besaran tepi yang diterima.

b. Screenshots hasil



Masalahnya masih sama dengan yang sebelum sebelumnya, namun tepi masih dapat digunakan dengan baik untuk menemukan mask objeknya.

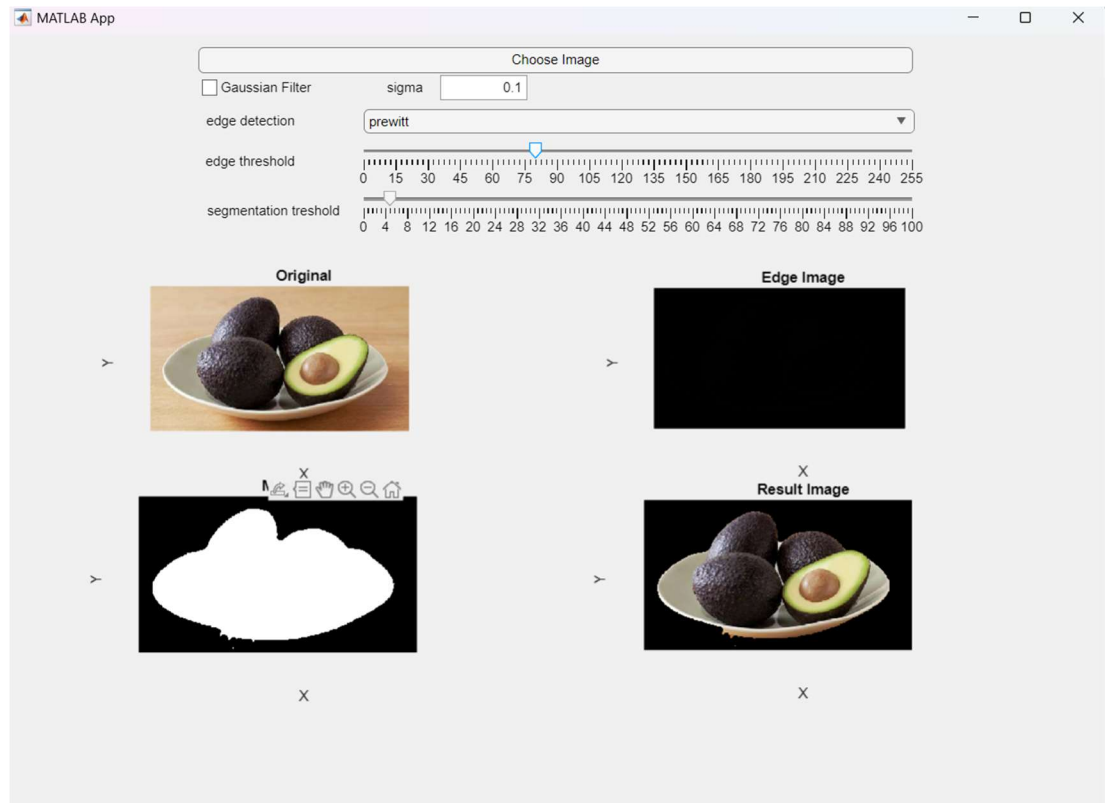
2.4 Prewitt

a. Kode dan analisis

```
function edgeImage = prewitt(app, image, threshold)
    Px = [-1 0 1; -1 0 1; -1 0 1];
    Py = [-1 -1 -1; 0 0 0; 1 1 1];
    Jx = convn(double(image), double(Px), 'same');
    Jy = convn(double(image), double(Py), 'same');
    edgeImage = uint8(abs(Jx) + abs(Jy) > threshold);
end
```

Untuk cara perhitungannya sama seperti sobel, hanya saja matrix yang digunakan berbeda, menggunakan matrix prewitt untuk menghitung kemiringannya.

b. Screenshots hasil



Segmentasi yang didapat cukup bersih, namun ketika parameter thresholnya diubah, akan ada bagian alpukat dan piringnya hilang. Seharusnya dikarenakan beberapa bagian dari tepi piring lebih kuat dibanding tepi alpukan dan juga sebaliknya.

2.5 Roberts

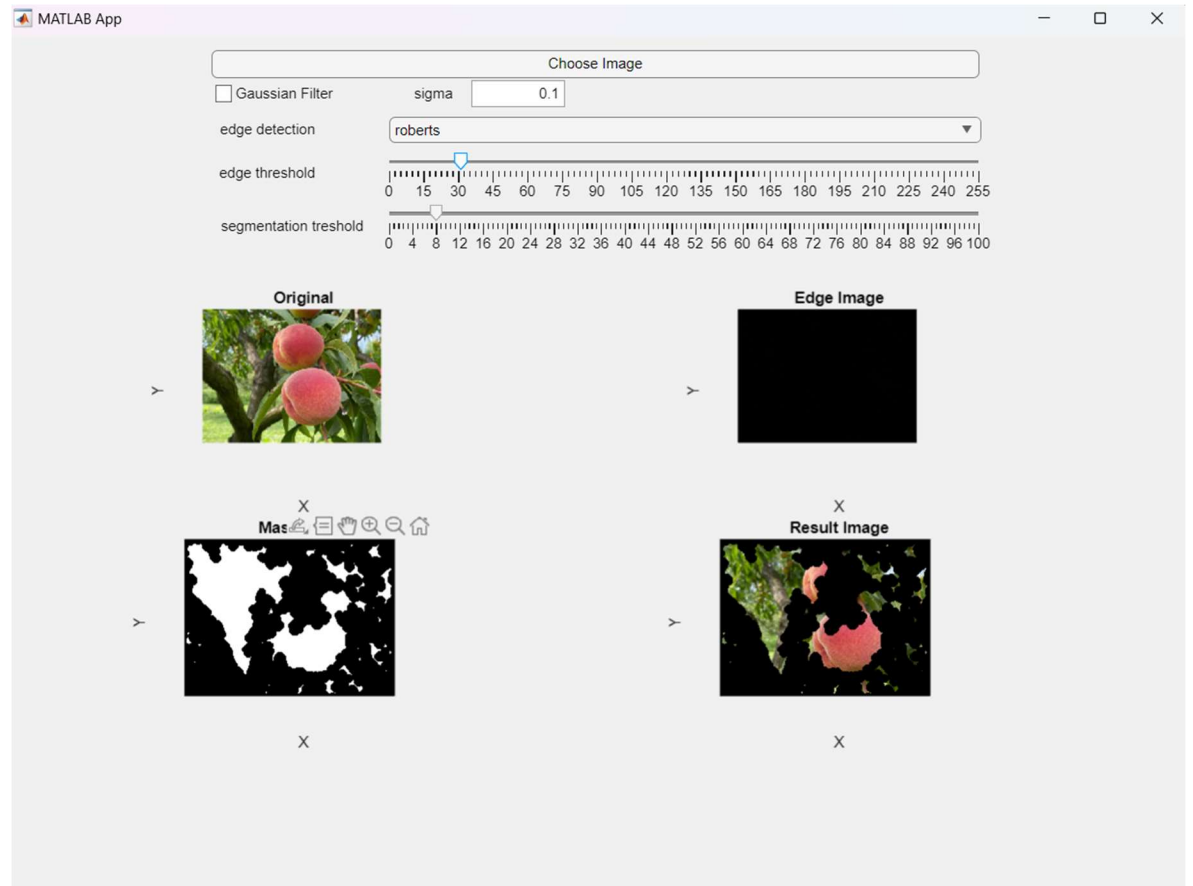
a. Kode dan Analisis

```
function edgeImage = roberts(app, image, threshold)
    Rx = [1 0; 0 -1];
    Ry = [0 1; -1 0];
    Jx = convn(double(image), double(Rx), 'same');
    Jy = convn(double(image), double(Ry), 'same');
    edgeImage = uint8(abs(Jx) + abs(Jy) > threshold);

end
```

Sama seperti sobel dan prewitt, yang berubah hanya matrix untuk menghitung kemiringannya.

b. Screenshots Hasil



Hasil yang didapatkan kurang memuaskan (bahkan menggunakan tipe lainnya). untuk gambar ini cukup sulit, karena tepi yang dihasilkan lebih kuat pada bagian daun daripada bagian peachnya. Mungkin bisa digunakan elemen warna, sebagai input pengguna, apakah objeknya dominan merah, hijau, atau biru agar tepinya bisa lebih kuat pada bagian buahnya.

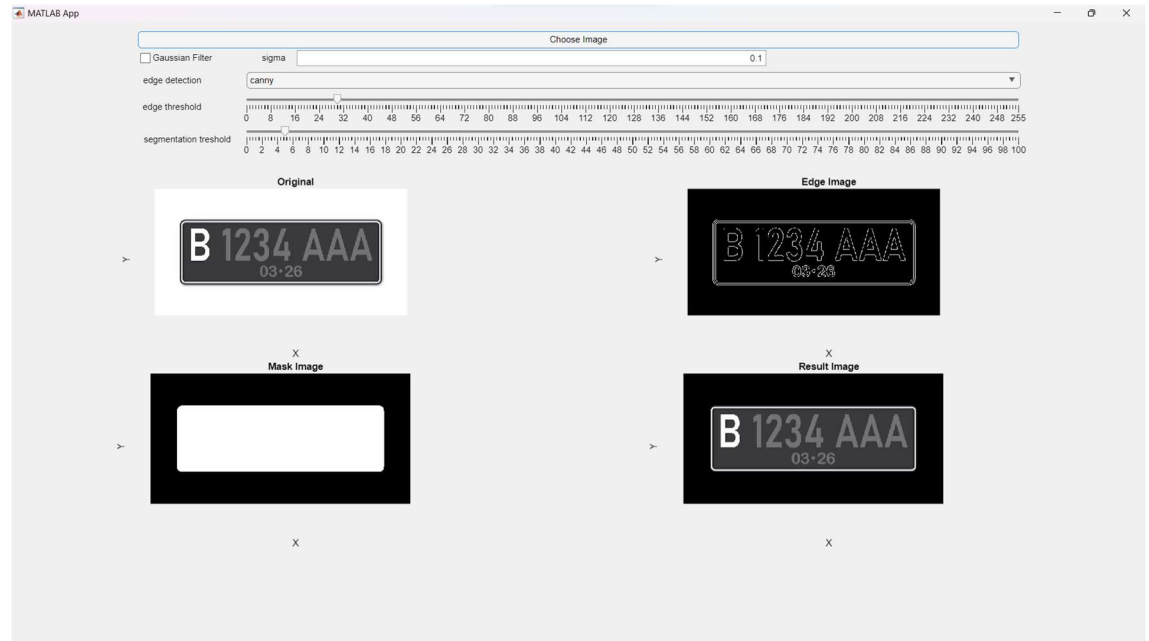
2.6 Canny

a. Kode dan Analisis

```
function edgeImage = canny(app, image, threshold)
    edgeImage = edge(image, "Canny", threshold/255);
end
```

Menggunakan fungsi bawaan dari matlab.

b. Screenshots Hasil



Untuk gambar ini cukup bersih karena tepinya sangat jelas.

2.7. Segmentation

a. Mask

```
function maskImage = generateMask(app, edgeImage, threshold)
% bersihkan pixel tepi
tempImage = imclearborder(edgeImage);

% hubungkan tepi
tempImage = imclose(tempImage, strel('disk', round(threshold)));

% Isi daerah yang tepinya terhubung
tempImage = imfill(tempImage, 8, 'holes');

% erosi pada tepi yang bukan daerah utama
tempImage = imerode(tempImage, strel('disk', 2));

maskImage = uint8(tempImage);
end
```

b. Main Program


```

function main(app)
    cla(app.UIAxes);
    cla(app.UIAxes2);
    cla(app.UIAxes3);
    cla(app.UIAxes4);
    imshow(app.OriginalImage, 'Parent', app.UIAxes);
    %ubah ke rgb untuk selanjutnya diproses tepinya
    grayImage = rgb2gray(app.OriginalImage);
    %dilakukan blurring untuk menghilangkan noise
    blurImage = imgaussfilt(grayImage, app.sigmaEditField.Value);
    %ambil edge
    edgeImage = generateEdgeImage(app, blurImage, app.edgedetectionDropDown.Value, app.edgethresholdSlider.Value);
    imshow(edgeImage, "Parent", app.UIAxes2);
    %buat mask
    maskImage = generateMask(app, edgeImage, app.segmentationthresholdSlider.Value);
    imshow(maskImage.*255, "Parent", app.UIAxes3);
    %yang sesuai dengan mask ditampilkan
    resultImage = app.OriginalImage .* maskImage;
    imshow(resultImage, "Parent", app.UIAxes4);

end

```

Penjelasan sudah ada di komentar pada screenshot kode.

c. Hasil dan analisis

Hasil sudah ditampilkan di screnshots sebelumnya.

Gambar segmentasi objek yang dihasilkan masih sangat bergantung pada kekuatan tepi objek yang ingin dipisahkan. Jika tepinya lemah dan terdapat tepi lain yang lebih kuat, maka mau tidak mau tepi tersebut akan ditampilkan. Apalagi jika tepi kuat itu bersambung semua sehingga membentuk suatu area, algoritma pembuatan mask yang saya buat belum dapat menghandle itu semua. Mungkin masukannya adalah bisa ditambah bantuan seperti masukan warna dari pengguna, dimana disebutkan warna dominan dari objek yang ingin dipisahkan, sehingga dapat dimanipulasi saat membuat grayscale image dengan menekankan warna dominan lebih kuat.

3. Alamat Github Program

[makhlukcupu/Tugas-3-Citra-13519170: edge detection and object segmentation \(github.com\)](https://github.com/makhlukcupu/Tugas-3-Citra-13519170)

4. Saran untuk penulis

- Pelajari dulu dasar pengolahan array dalam matlab sebelum mulai mengerjakan tugasnya, agar kodenya bisa jauh dipersingkat.
- Tidak menunda pengerjaan