Laporan Tugas 3
IF4073 Interpretasi dan Pengolahan Citra

Segmentasi Citra

Disusun oleh:

Tanur Rizaldi Rahardjo / 13519214

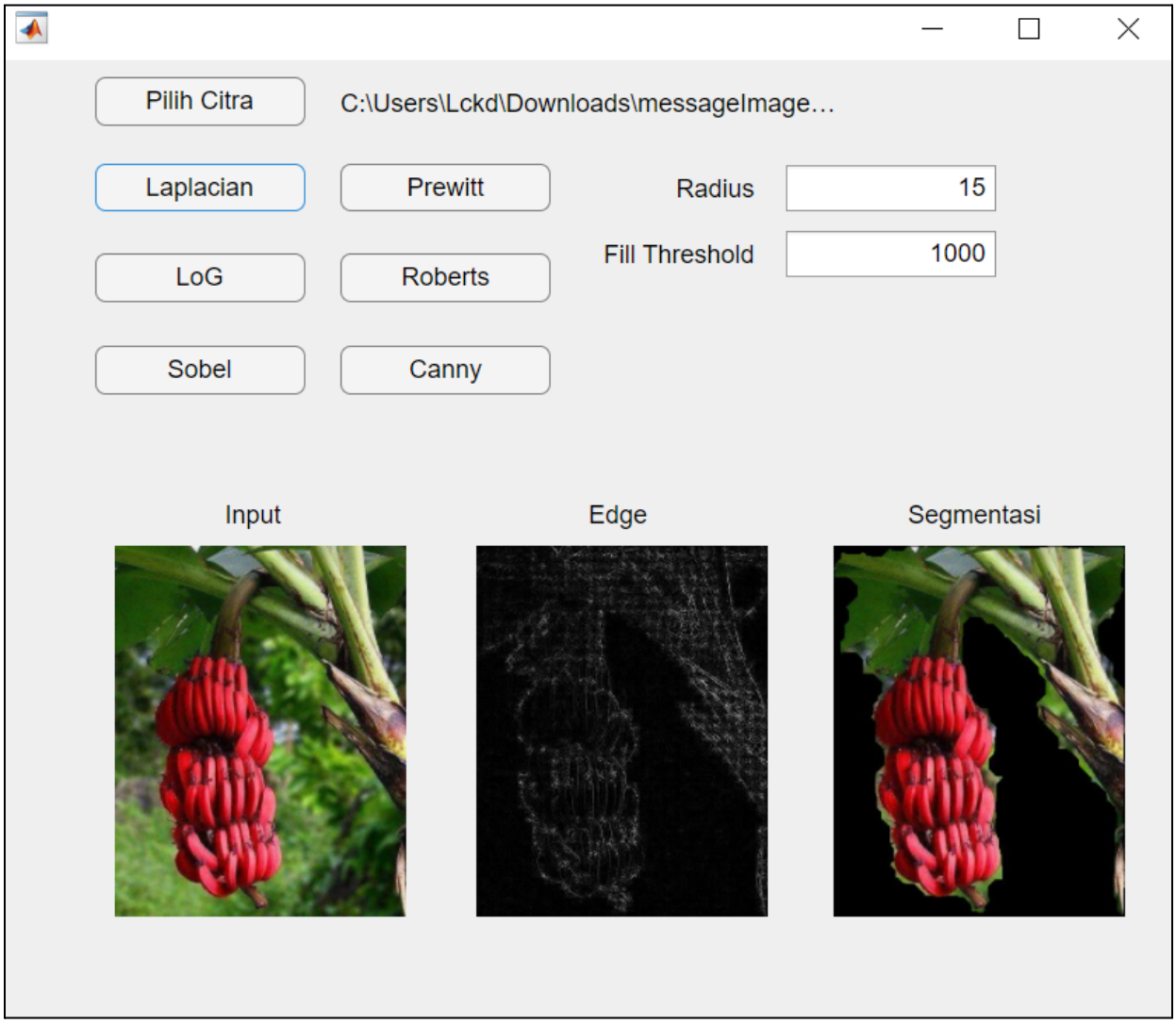
Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

2022

1. *Screenshot* GUI Program



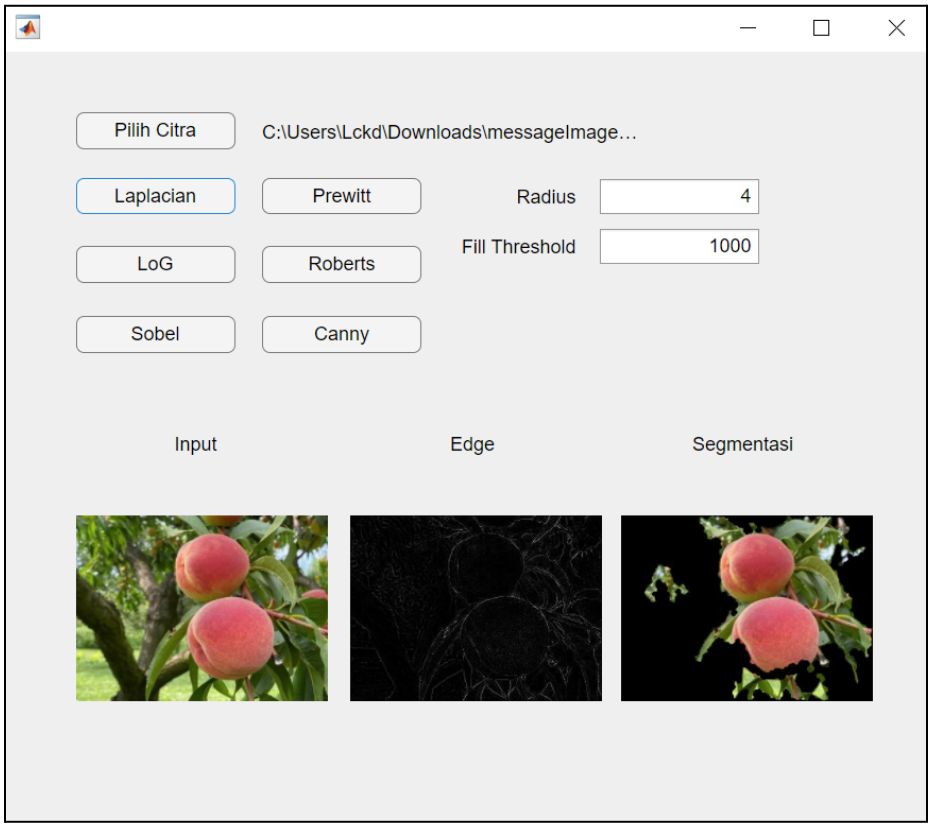
2. Rincian Setiap Program

2.1. Laplacian

Kode Program

```
function res = laplacianEdge(~, image)
    mask = [0 1 0;
            1 -4 1;
            0 1 0];
    res = uint8(conv2(double(rgb2gray(image)), double(mask), 'same'));
end
```

Contoh Hasil Eksekusi

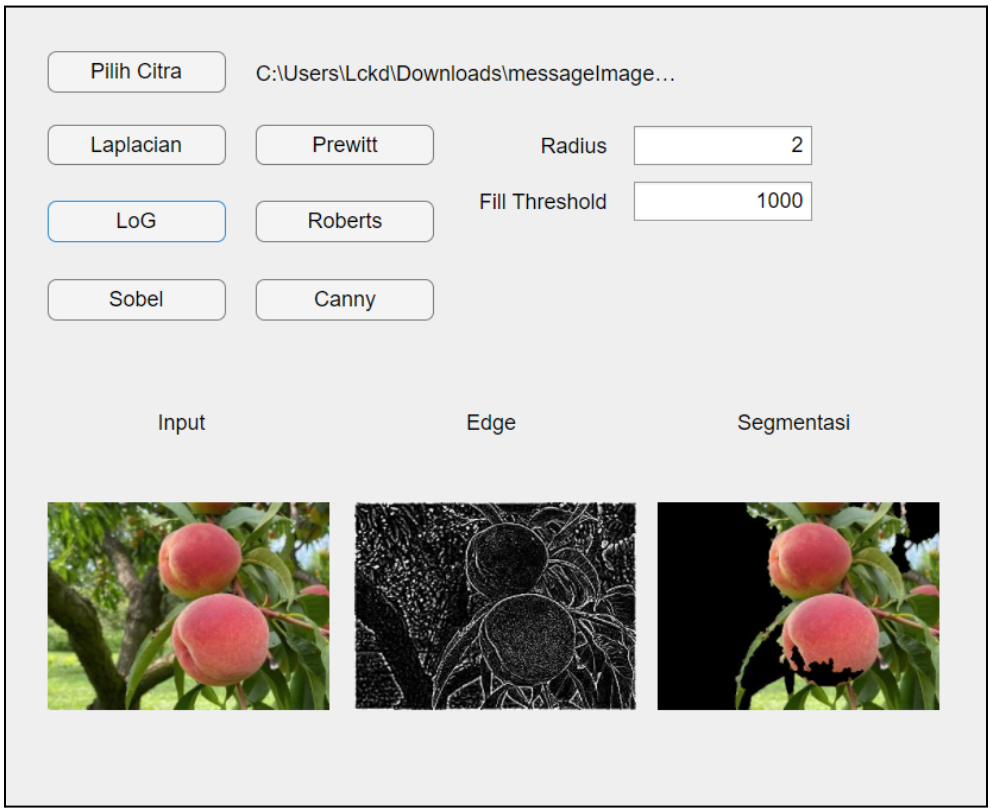


2.2. LoG

Kode Program

```
function res = logEdge(~, image)
    mask = [0 0 -1 0 0;
            0 -1 -2 -1 0;
            -1 -2 16 -2 -1;
            0 -1 -2 -1 0;
            0 0 -1 0 0];
    res = uint8(conv2(double(rgb2gray(image)), double(mask), 'same'));
end
```

Contoh Hasil Eksekusi

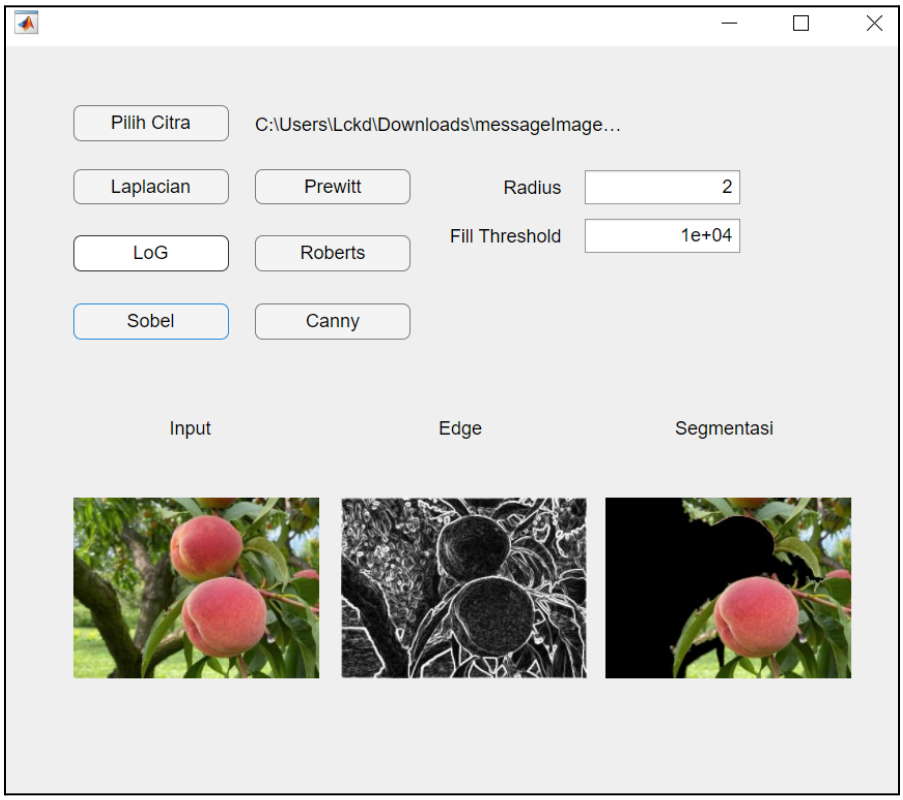


2.3. Sobel

Kode Program

```
function res = sobelEdge(~, image)
    maskX = [-1 0 1;
             -2 0 2;
             -1 0 1];
    maskY = [1 2 1;
             0 0 0;
             -1 -2 -1];
    sobelJX = conv2(double(rgb2gray(image)), double(maskX), 'same');
    sobelJY = conv2(double(rgb2gray(image)), double(maskY), 'same');
    res = uint8(sqrt(sobelJX.^2 + sobelJY.^2));
end
```

Contoh Hasil Eksekusi

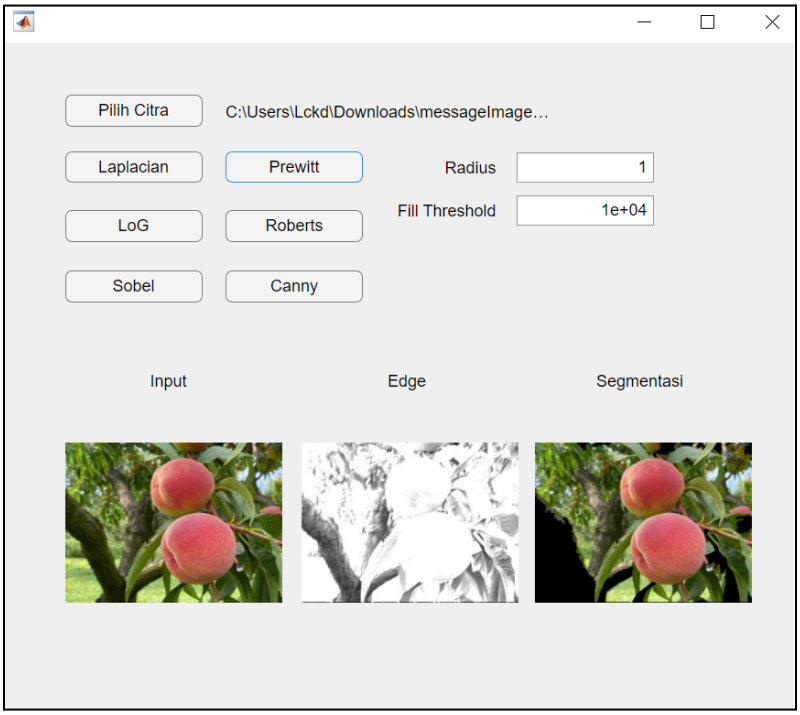


2.4. Prewitt

Kode Program

```
function res = prewittEdge(~, image)
    maskX = [-1 0 1;
             -1 0 1;
             -1 0 1];
    maskY = [-1 -1 -1;
             -1 0 1;
             -1 0 1];
    prewittX = conv2(double(rgb2gray(image)), double(maskX), 'same');
    prewittY = conv2(double(rgb2gray(image)), double(maskY), 'same');
    res = uint8(sqrt(prewittX.^2 + prewittY.^2));
end
```

Contoh Hasil Eksekusi



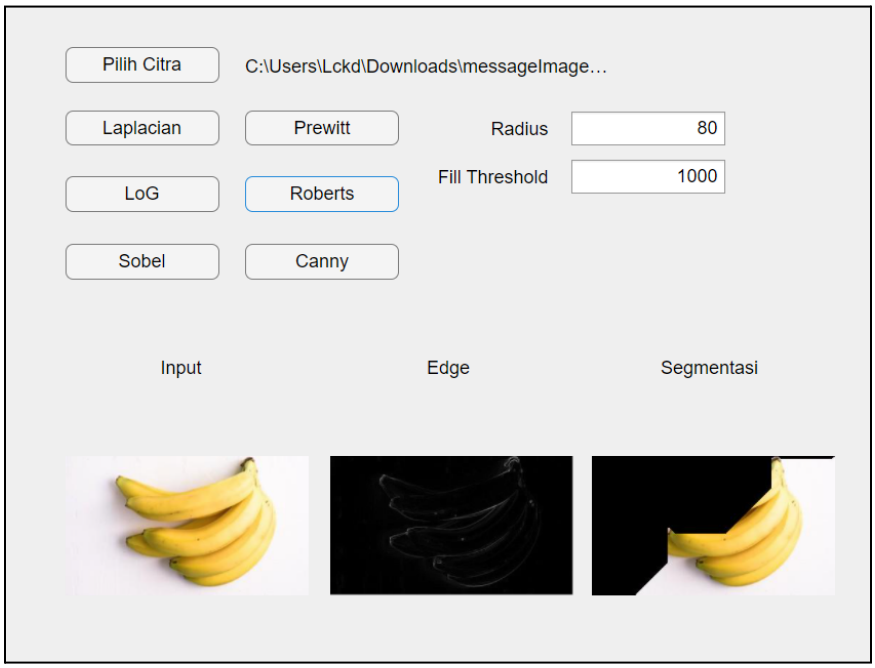
Catatan : Untuk suatu alasan, setelah implementasi segmentasi, operator Prewitt mengalami masalah yang tidak diketahui

2.5. Roberts

Kode Program

```
function res = robertsEdge(~, image)
    maskX = [1 0;
             0 -1];
    maskY = [0 1;
             -1 0];
    robertsX = conv2(double(rgb2gray(image)), double(maskX), 'same');
    robertsY = conv2(double(rgb2gray(image)), double(maskY), 'same');
    res = uint8(sqrt(robertsX.^2 + robertsY.^2));
end
```

Contoh Hasil Eksekusi

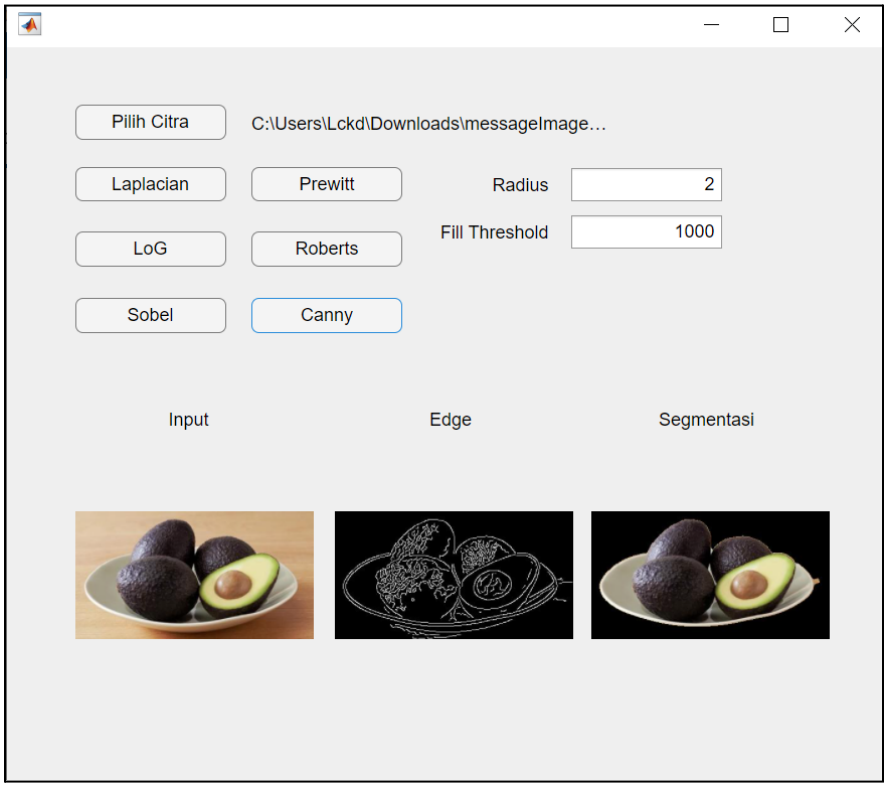


2.6. Canny

Kode Program

```
function res = cannyEdge(~, image)
    res = edge(im2gray(image), 'canny');
end
```

Contoh Hasil Eksekusi



2.7. Segmentasi

Kode Program

```
function res = segmentation(~, image, imageEdge, rad, thres)
    binarizedEdge = imbinarize(imageEdge, graythresh(imageEdge));
    closedEdge = imclose(binarizedEdge, strel('disk', rad));
    filledEdge = imfill(closedEdge, 'holes');

    filtered = imopen(filledEdge, strel(ones(3, 3)));
    mask = uint8(bwareaopen(filtered, thres));

    r = image(:, :, 1) .* mask;
    g = image(:, :, 2) .* mask;
    b = image(:, :, 3) .* mask;
    res = cat(3, r, g, b);
end
```

Analisa

Hasil edge yang didapat dari operator-operator sebelumnya akan dijadikan image binary dengan otsu method yang dijelaskan dikelas. Matlab menyediakan fungsi graythresh yang mengimplementasikan otsu method untuk mendapatkan gray threshold. Setelah edge menjadi binary, edge akan ditutup dengan imclose dan dilanjutkan menutup lubang agar menjadi mask dengan imfill. Setelah itu akan diproses tambahan dengan bwareaopen untuk menutup lubang yang berukuran dibawah threshold user input. Hasil akhir akan ditampilkan ke UI.

3. Alamat Github Program

https://github.com/Lock1/Image-3-IF4073