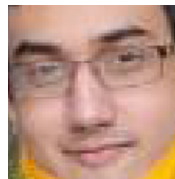


Laporan Tugas 1 IF4073 Interpretasi dan Pengolahan Citra

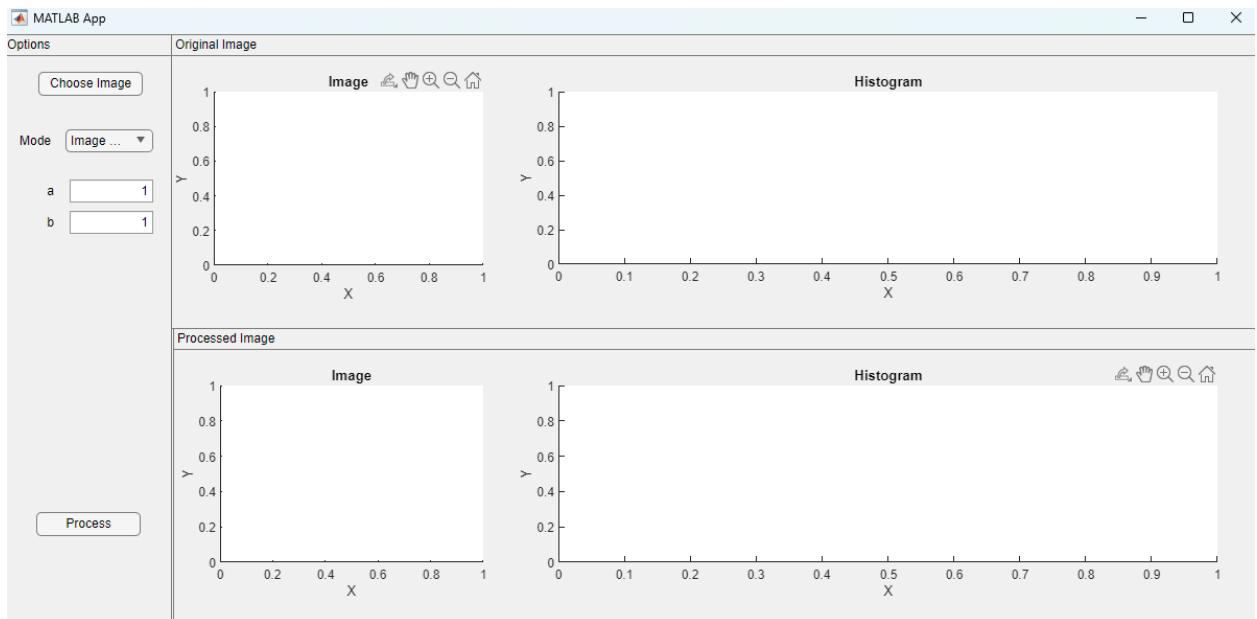
Pembuatan Program Image Enhancement dengan Matlab



Oleh
Safiq Faray - 13519145

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2023

A. Screenshot GUI



B. Kode Program

a. Program 1 : Menghitung dan Menampilkan Histogram Citra

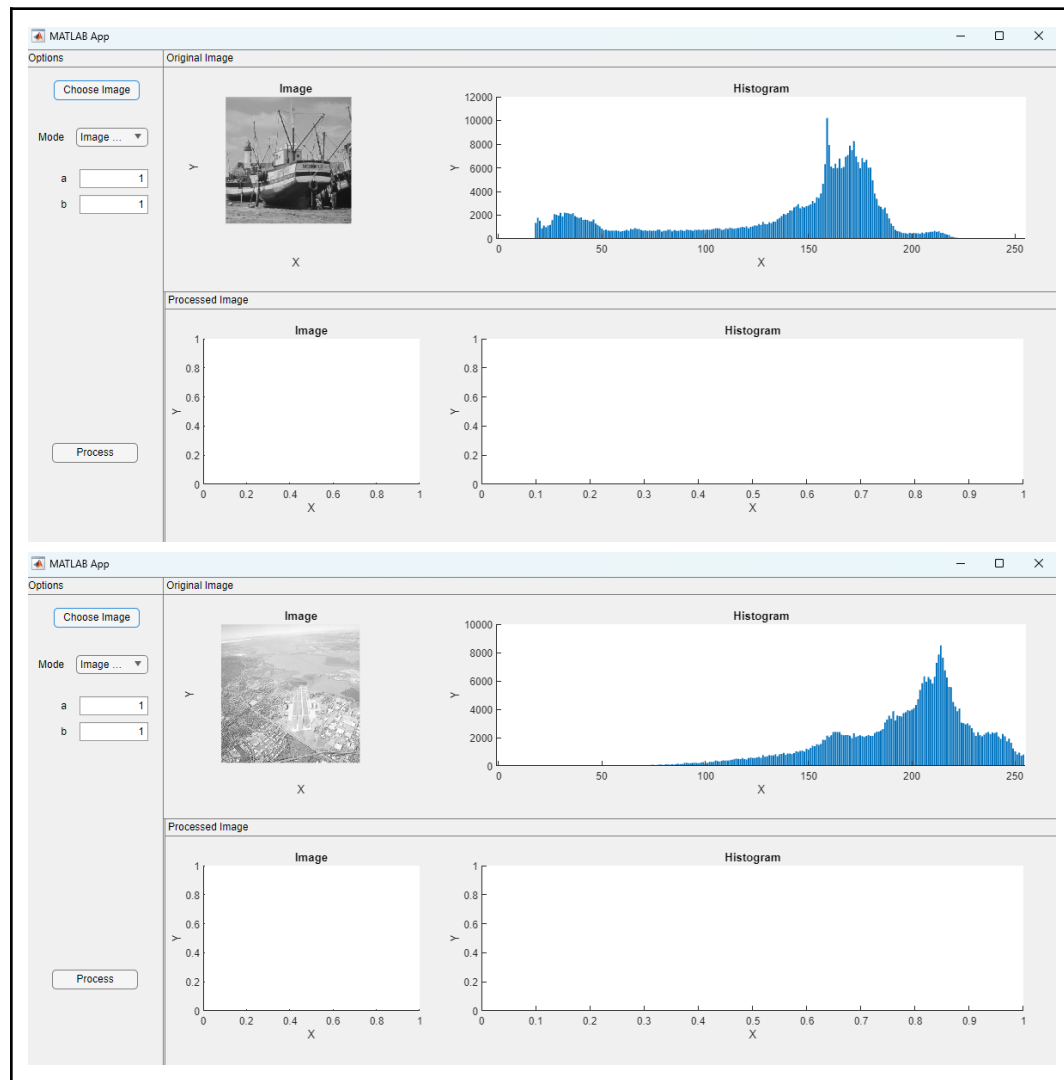
Berikut adalah kode dari program 1.

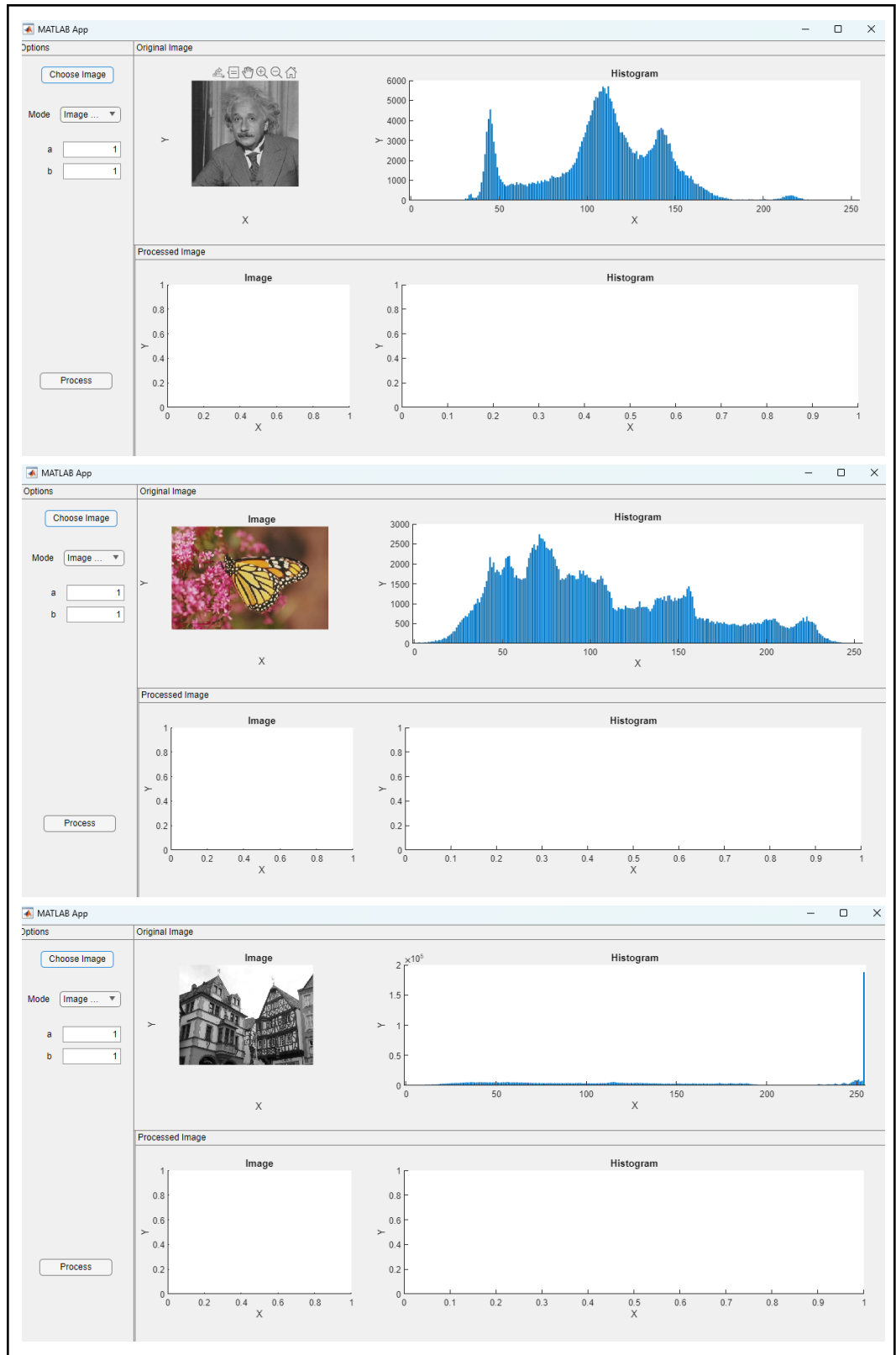
```
classdef CustomHist
methods (Static)
function histImg = getHistImg(img)
    binEdges = 0:255;
    %di spec, tidak ada larangan untuk menggunakan histcounts, yaitu
    %fungsi untuk menghitung frekuensi dan unique values, yang
    %digunakan untuk plot bar
    [freq, uniqueVals] = histcounts(img, binEdges);
    histImg = bar(uniqueVals(1:end-1), freq);
    xlabel("Pixel Value");
    ylabel("Frequency");
    title("Image Histogram");
end
function [freq, uniqueVals] = getHistData(img)
    binEdges = 0:255;
    [freq, uniqueVals] = histcounts(img, binEdges);
end
end
end
```

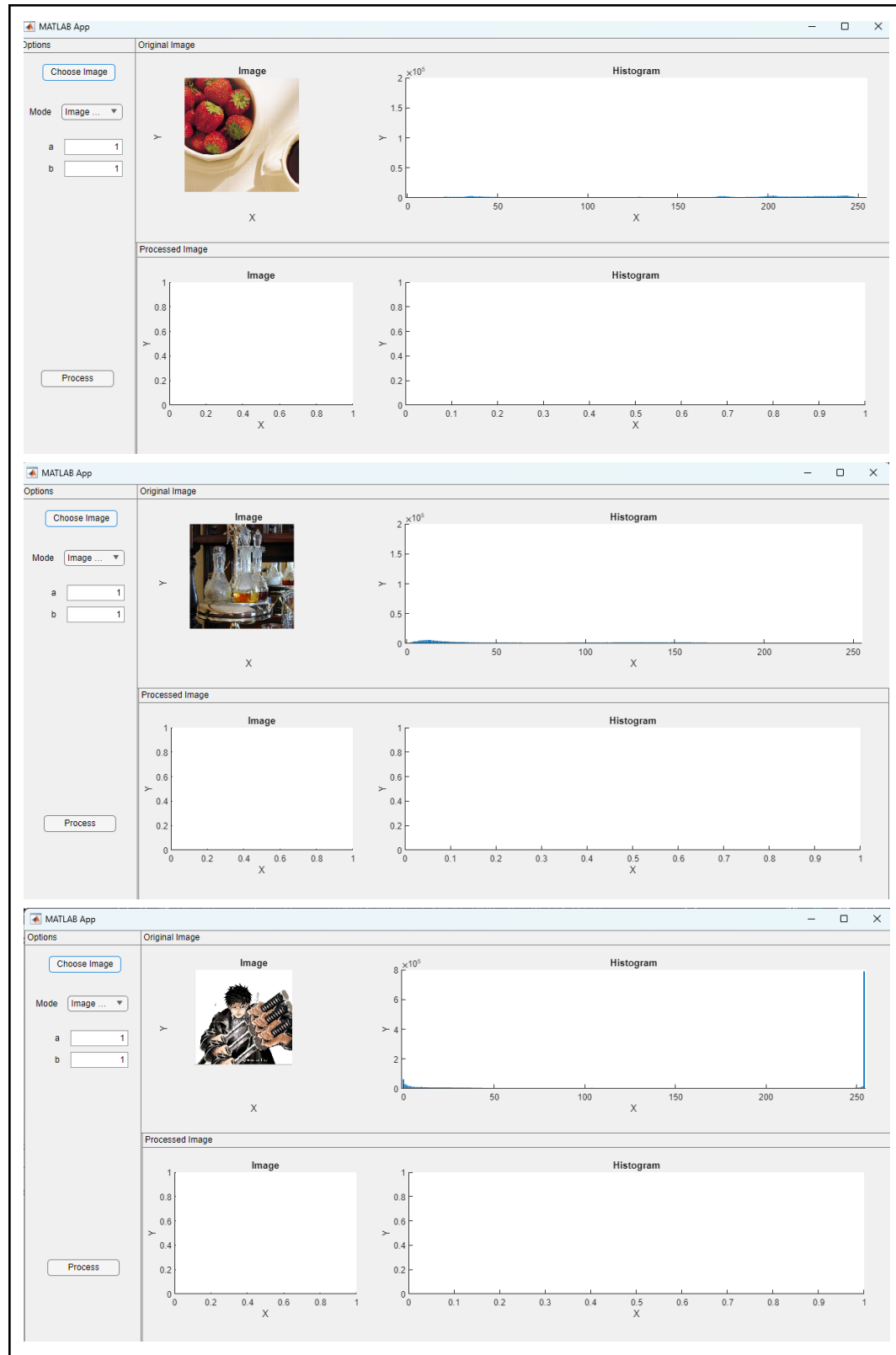
Kode dikumpulkan ke dalam class static bernama “CustomHist”. Untuk perhitungan histogram digunakan fungsi “histcounts”, karena tidak ada larangan

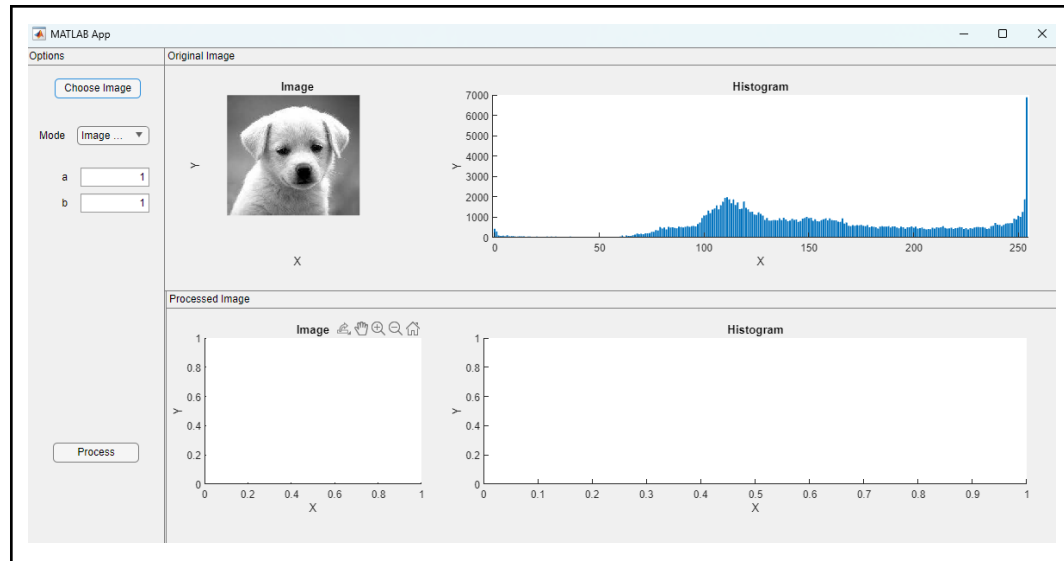
untuk menggunakan fungsi tersebut. Alasan menggunakan fungsi tersebut adalah karena efisiensi waktu. Jika menggunakan fungsi buatan sendiri dengan kompleksitas $O(3 \cdot n^2)$, akan ada *noticeable* delay pada saat generate plot histogram.

Contoh penggunaan adalah sebagai berikut.









b. Program 2 : Perbaikan Kualitas Citra

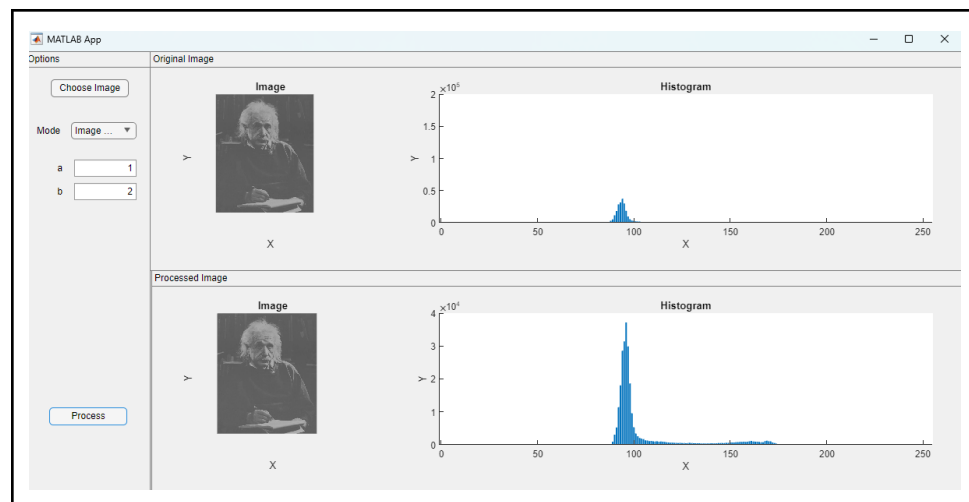
i. Image Brightening

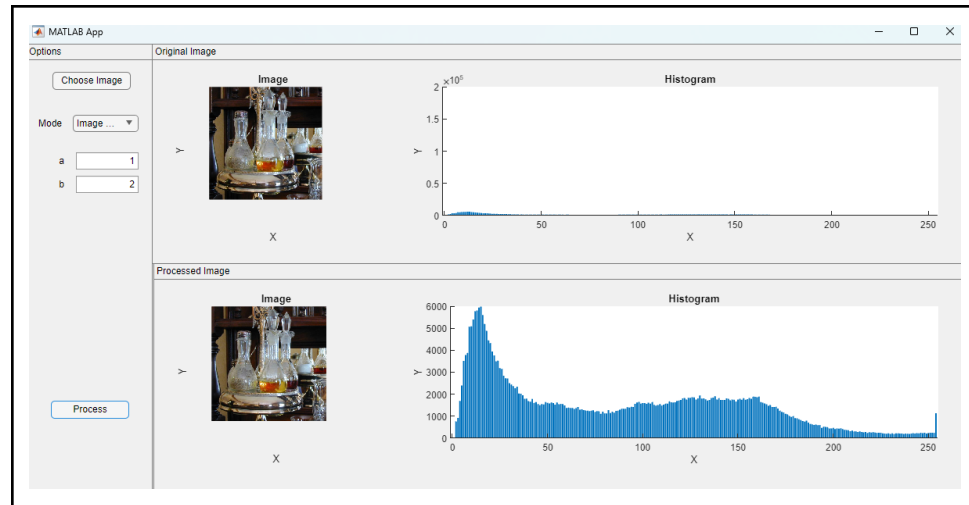
Berikut adalah kodenya (berada pada kelas “ImageManipulation” dan merupakan method static)

```
function imgBright = imageBrighten(img, a, b)
    imgBright = a * img + b;
end
```

Menggunakan rumus $s = ar + b$ yang dimana a adalah faktor matriks inputan pengguna, r adalah matrik gambar, dan b adalah konstanta inputan pengguna.

Contoh penggunaan adalah sebagai berikut.



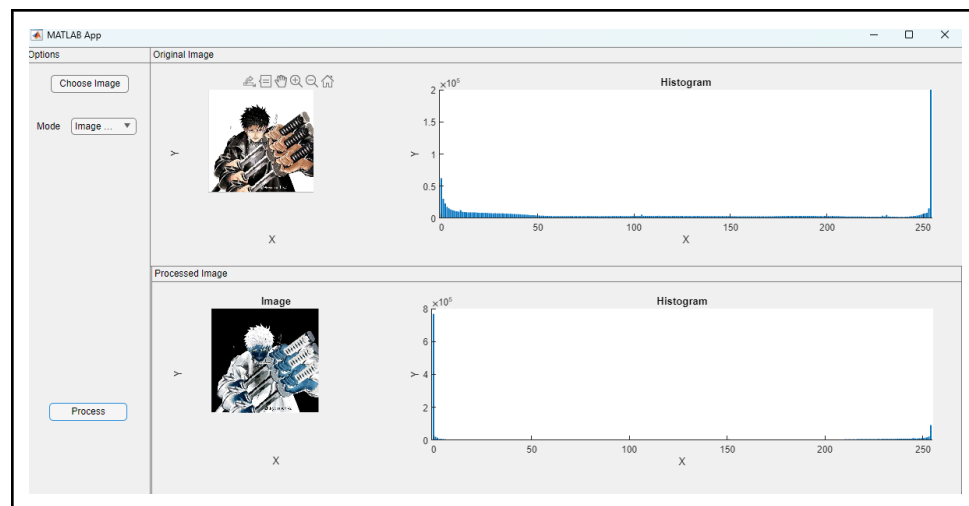


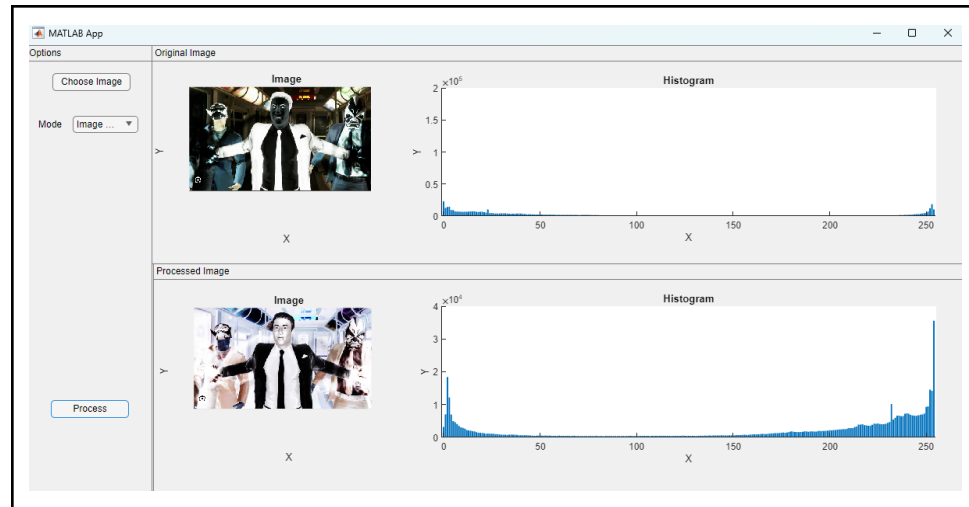
ii. Citra Negatif

Berikut adalah kodenya (berada pada kelas “ImageManipulation” dan merupakan method static)

```
function imgNeg = imageNegative(img)
    maxVal = 255;
    imgNeg = maxVal - img;
end
```

Dengan rumus $\text{max grey length} - \text{img grey level}$. Berikut adalah contoh penggunaannya.



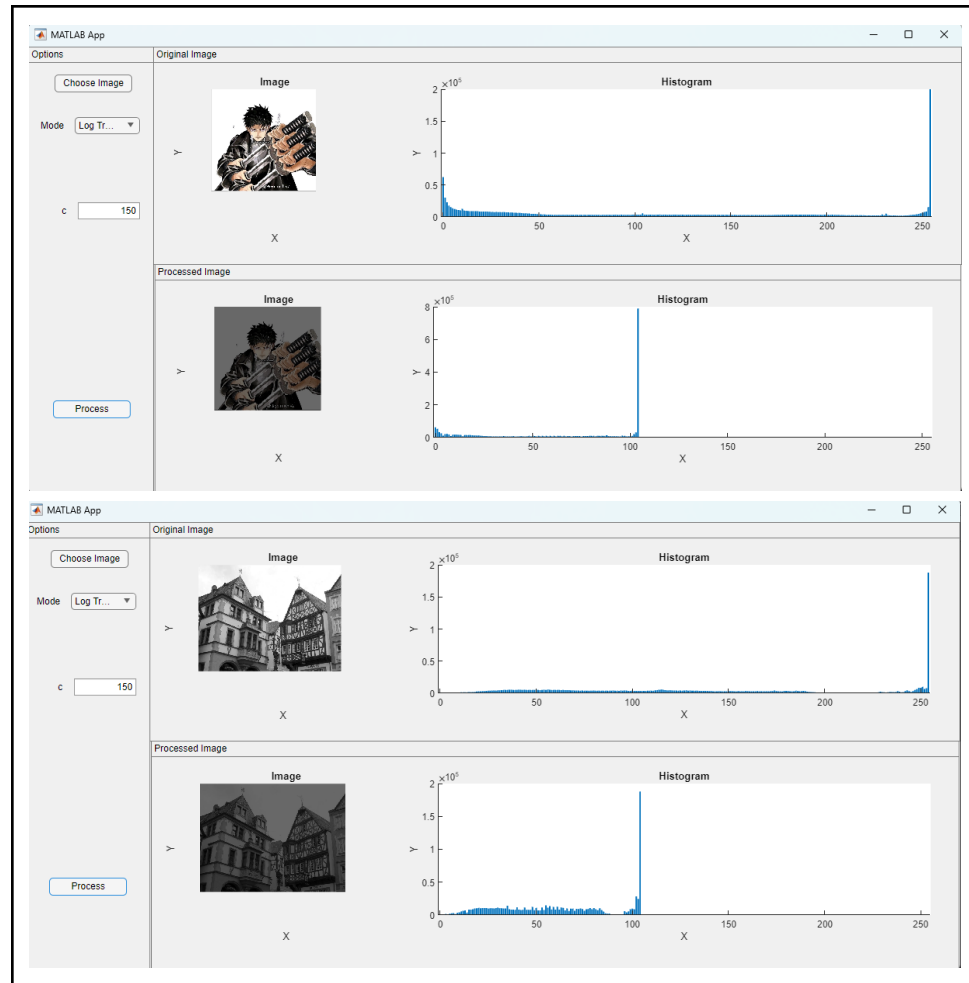


iii. Transformasi Log

Berikut adalah kodenya (berada pada kelas “ImageManipulation” dan merupakan method static)

```
function logTransform = logTransformation(img, c)
    img = im2double(img);
    logTransform = uint8(c * log(img + 1));
end
```

Algoritma yang diterapkan adalah dengan mengubah matriks gambar menjadi bentuk double, lalu diterapkan rumus $c \log(1 + r)$ yang kemudian akan diubah kembali menjadi integer. Transformasi log sendiri merupakan salah satu cara untuk menegatifkan citra. Berikut adalah contoh penggunaannya.

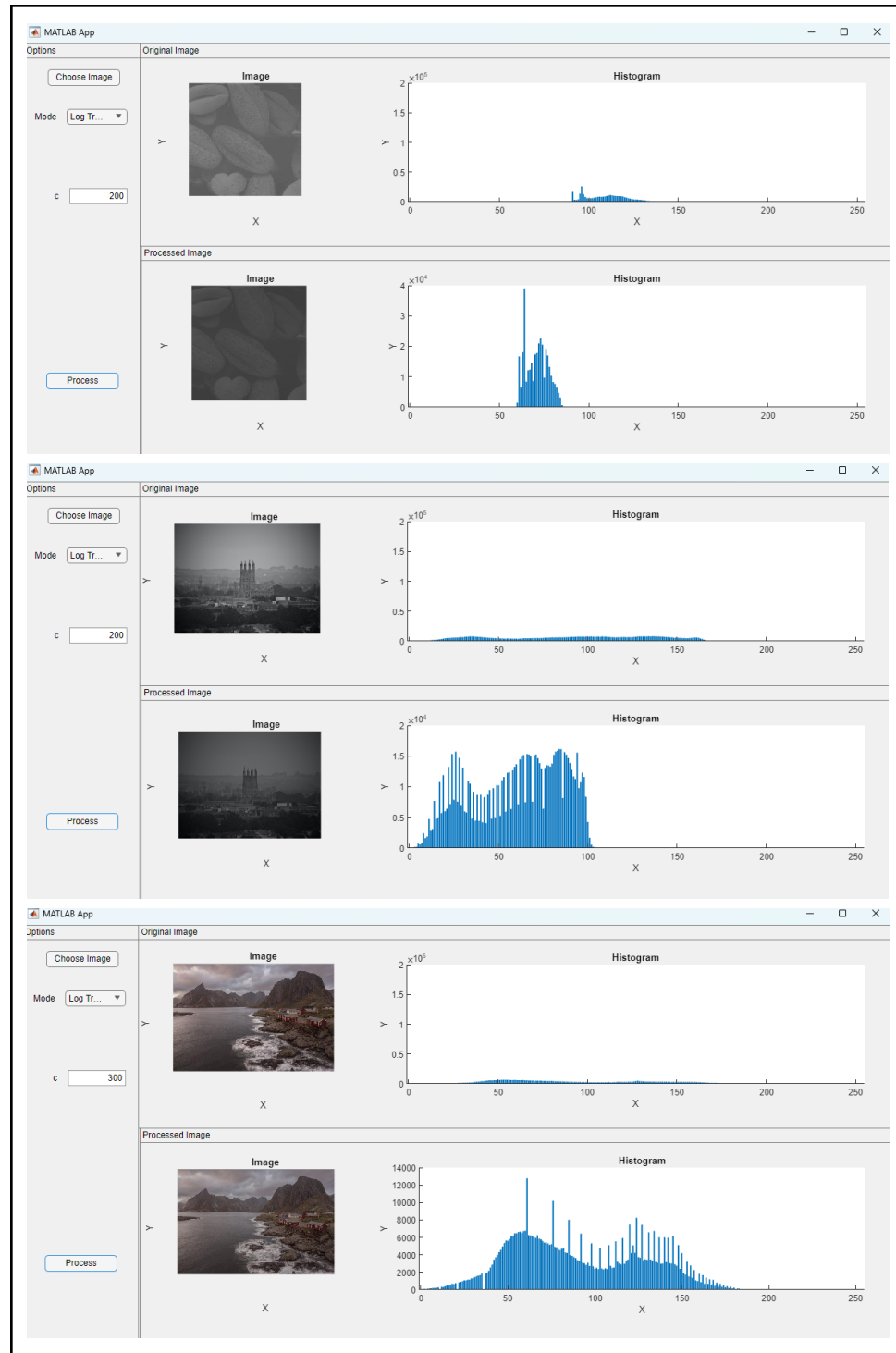


iv. Transformasi Pangkat

Berikut adalah kodenya (berada pada kelas “ImageManipulation” dan merupakan method static)

```
function powerTransform = powerTransformation(img, c, gamma)
    img = im2double(img);
    powerTransform = uint8(c * (img^gamma));
end
```

Algoritma yang diterapkan adalah dengan mengubah matriks gambar menjadi bentuk double, lalu diterapkan rumus cr^γ yang kemudian akan diubah kembali menjadi integer. Transformasi pangkat sendiri merupakan salah satu cara untuk menegatifkan citra. Berikut adalah contoh penggunaannya.



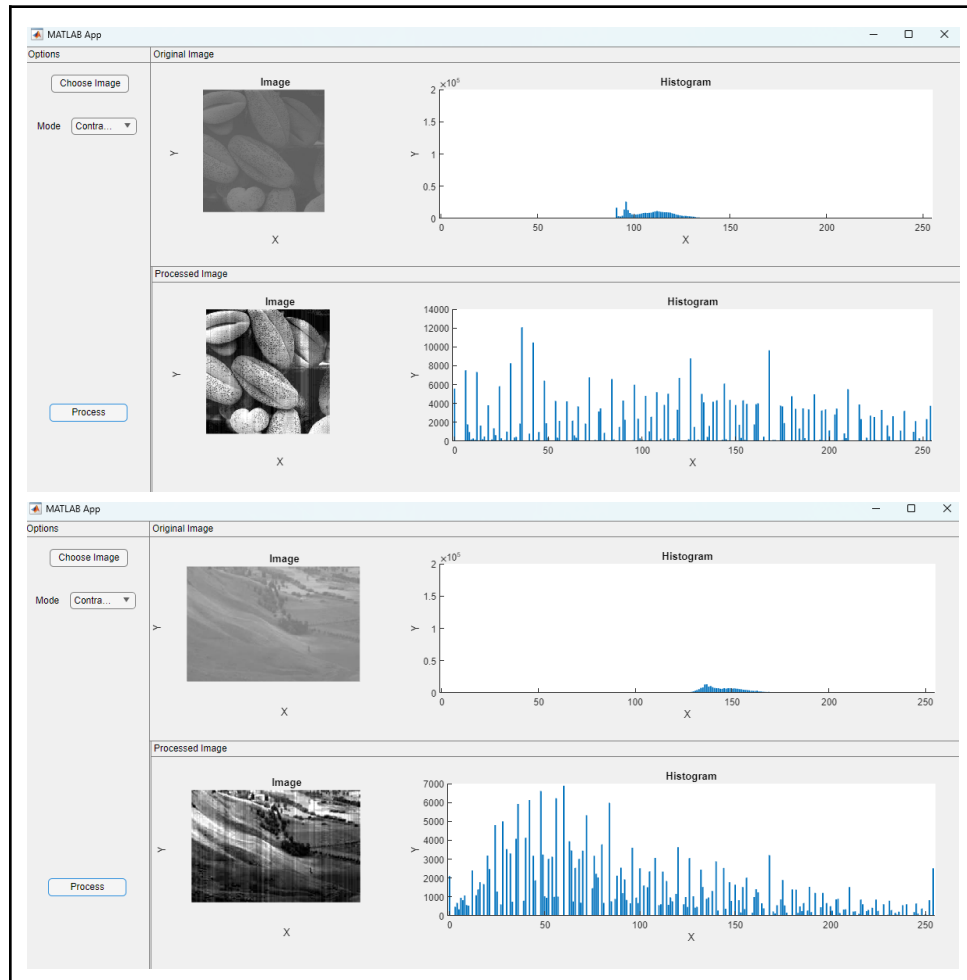
- v. **Peregangan Kontras**
 Berikut adalah kodenya (berada pada kelas “ImageManipulation” dan merupakan method static)

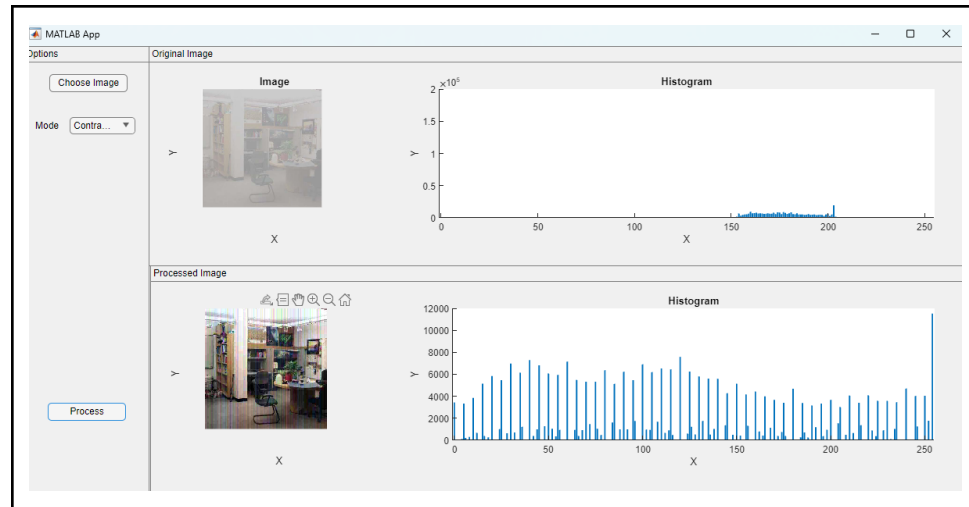
```

function contrastStretch = contrastStretching(img)
    [~, ~, channels] = size(img);
    stretchedChannels = zeros(size(img));
    for ch=1:channels
        processedImg = img(:,:,ch);
        rmin = min(processedImg);
        rmax = max(processedImg);
        stretchedChannels(:,:,ch) = (processedImg -
rmin).*(255./(rmax-rmin));
    end
    contrastStretch = uint8(stretchedChannels);
end

```

Cara kerjanya adalah dengan memroses masing-masing channel gambar, dan diterapkan rumus $(\text{processedImg} - \text{rmin}).*(255./(\text{rmax}-\text{rmin}))$. Rmin dan rmax merupakan nilai pixel minimum dan maksimum pada sebuah gambar. Berikut adalah contoh penggunaannya.





c. Program 3 : Histogram Equalization

Berikut adalah kodenya (berada pada kelas “ImageManipulation” dan merupakan method static)

```
function histEq = getHistEqual(img)
    [height, width, ~] = size(img);
    hist = CustomHist;
    [freq, ~] = hist.getHistData(img);

    histEq = zeros(height, 1);
    NPixels = height * width;
    NGray = 255;
    for i=1:NGray
        sum = 0;
        for j=1:i
            sum = sum + (freq(j) / NPixels);
        end
        histEq(i) = floor(NGray * sum);
    end
End

function histEqImg = histogramEqual(img)
    [height, width, channel] = size(img);

    histEqImg = img;
    for ch=1:channel
        imageChannel = img(:, :, ch);

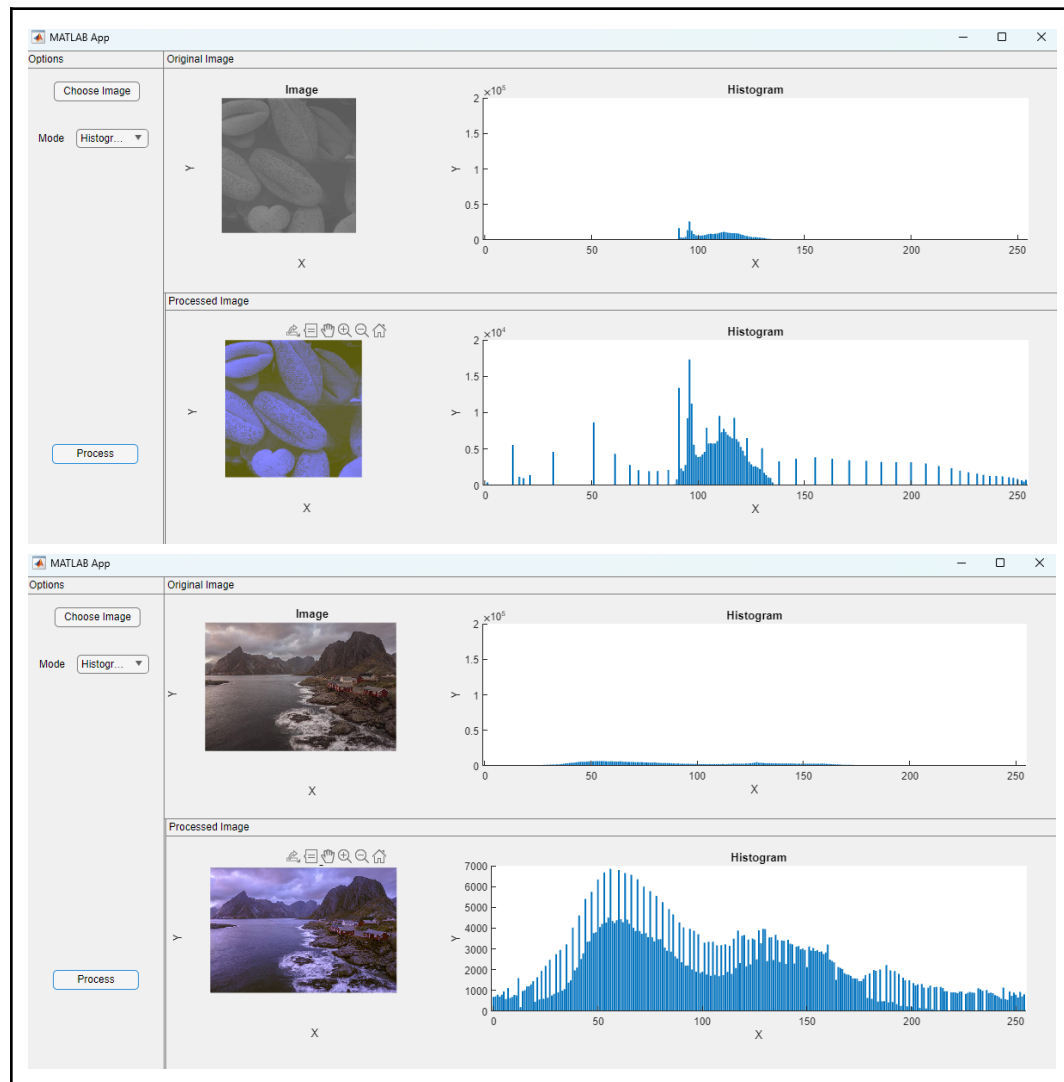
        histEq = ImageManipulation.getHistEqual(imageChannel);
        for i=1:height
```

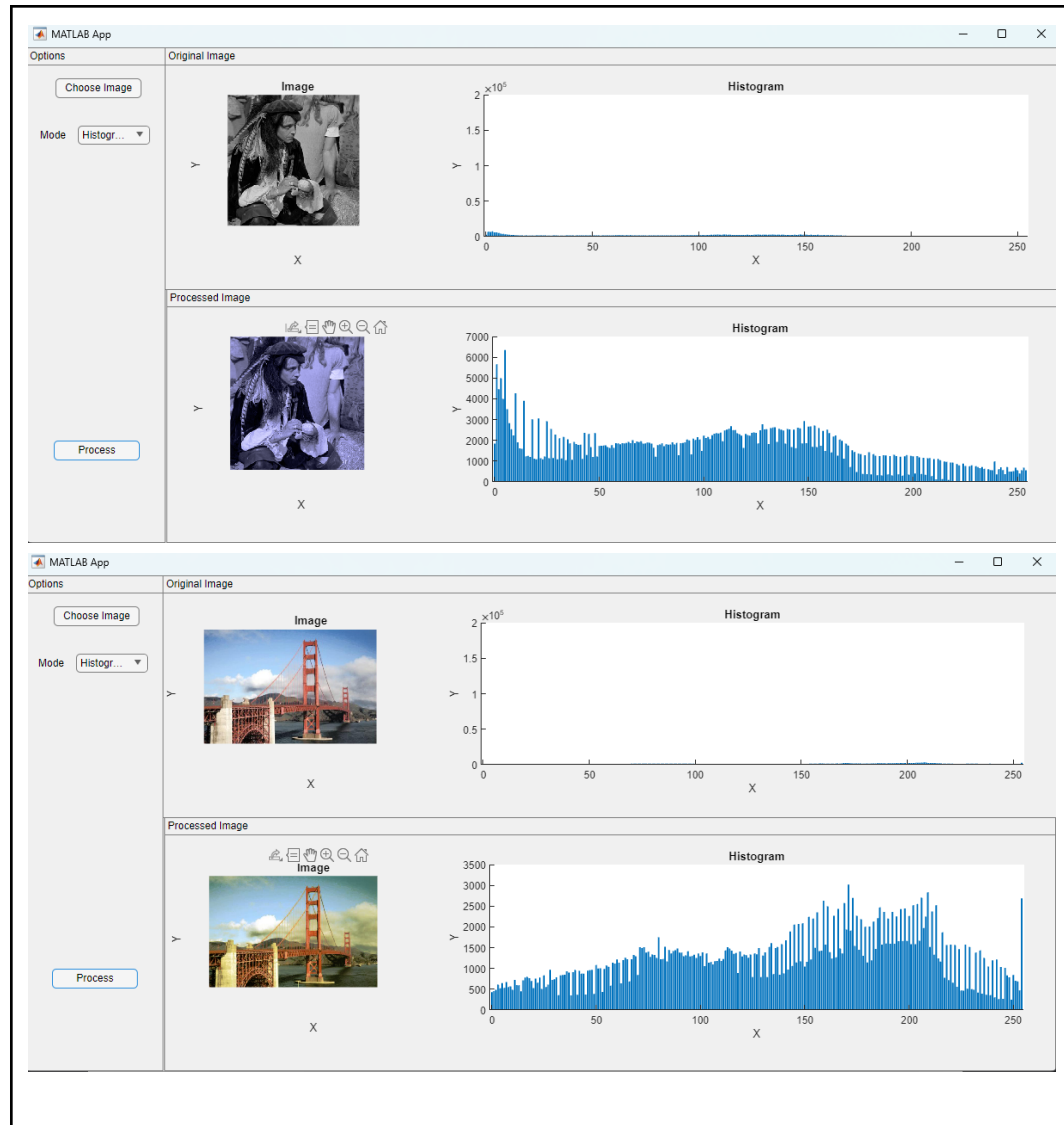
```

for j=1:width
    histEqImg(i,j, channel) = histEq(imageChannel(i,j)+1);
end
end
end
histEqImg = uint8(histEqImg);
end

```

Algoritma yang disusun akan menghitung frekuensi sebuah pixel dibagi banyaknya pixel, lalu akan menjumlah kumulatif dan menyesuaikan value pixel gambar original sehingga histogram dapat menjadi lebih seimbang. Berikut adalah contoh penggunaannya.





d. Program 4 : Histogram Specification

Berikut adalah kodenya (berada pada kelas "ImageManipulation" dan merupakan method static)

```
function histSpec = histogramSpecification(img, ref)
    [height, width, channel] = size(img);
    [~, ~, channelRef] = size(ref);
    hist = CustomHist;
    histSpec = img;
    NGray = 255;
    for ch=1:channel
        imageOriginalChannel = img(:, :, ch);
        if (channel <= channelRef)
```

```

        imageRefChannel = ref(:, :, ch);
    else
        imageRefChannel = ref(:, :, 1);
    end

    histEqOriginal =
    ImageManipulation.getHistEqual(imageOriginalChannel);
    histEqRef = ImageManipulation.getHistEqual(imageRefChannel);
    invHist = zeros(height);
    for i=1:height
        minval = abs(histEqOriginal(i) - histEqRef(1));
        minj = 0;
        for j=1:NGray
            if (abs(histEqOriginal(1) - histEqRef(j)) < minval)
                minval = abs(histEqOriginal(i) - histEqRef(j));
                minj = j;
            end
            invHist(i) = minj;
        end
    end
    for i=1:height
        for j=1:width
            histSpec(i,j, channel) = invHist(imageOriginalChannel(i,j)+1);
        end
    end
    histSpec = uint8(histSpec);
end

```

Algoritma yang disusun adalah dengan melakukan histogram equalization kepada gambar input dan gambar referensi terlebih dahulu. Kemudian, dicari inverse histogramnya yang nantinya akan menjadi gambar akhir. Berikut adalah contoh penggunaannya.



Alamat Github : <https://github.com/Haruray/tugas1-citra>