

# **Tugas 1 IF4073 Interpretasi dan Pengolahan Citra**

**Semester I Tahun 2023/2024**



**Disusun oleh:**

Mahesa Lizardy                    13520116

Bryan Amirul Husna                13520146

**Program Studi Teknik Informatika**

**Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung**

**Jl. Ganesha 10, Bandung 40132**

**2023**

## Daftar Isi

<b>I Pendahuluan</b>	<b>3</b>
1.1. Histogram Citra	3
1.2. Perbaikan Kualitas Citra	3
1.3. Pencerahan Citra (Image Brightening)	3
1.4. Menegatifkan Citra (Image Negative)	4
1.5. Transformasi Log	4
1.6. Transformasi Pangkat	4
1.7. Peregangan Kontras (contrast stretching)	4
1.8. Perataan Histogram (Histogram Equalization)	5
1.9. Spesifikasi Histogram (Histogram Specification)	5
<b>II GUI Program</b>	<b>6</b>
<b>III Pembahasan</b>	<b>12</b>
3.1. Histogram citra Grayscale dan Citra Berwarna	12
3.1.1. Kode Program	12
3.1.2. Testing	13
3.1.3. Analisis	18
3.2. Perbaikan Kualitas Citra	18
3.2.1. Kode Program	18
3.2.2. Testing	21
3.2.3. Testing Negative Image	35
3.2.4. Analisis	37
3.3. Histogram Equalization	38
3.3.1. Kode Program	38
3.3.2. Testing	39
3.3.3. Analisis	46
3.4. Histogram Specification	46
3.4.1. Kode Program	46
3.4.2. Testing	47
3.4.3. Analisis	50
<b>Alamat GitHub Program</b>	<b>51</b>
<b>Referensi</b>	<b>51</b>

# I Pendahuluan

## 1.1. Histogram Citra

Histogram citra (*image histogram*) merupakan informasi yang penting mengenai isi dari citra digital. Histogram citra merupakan grafik yang menggambarkan penyebaran nilai-nilai intensitas pixel dari suatu citra atau bagian tertentu dalam citra. Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (relative) dari intensitas yang terdapat pada citra. Selain itu histogram juga dapat digunakan untuk menunjukkan kecerahan (*brightness*) dan kontras (*contrast*) dari sebuah gambar. Oleh karena itu histogram merupakan alat bantu yang berharga dalam pengolahan citra baik secara kualitatif maupun kuantitatif.

## 1.2. Perbaikan Kualitas Citra

*Image enhancement* atau perbaikan kualitas citra agar tampak lebih baik dilakukan dengan tujuan untuk memperoleh citra yang lebih sesuai digunakan untuk aplikasi lebih lanjut. Misalnya, untuk mengidentifikasi objek dalam citra yang sebelumnya kurang jelas atau tidak terlihat sebelum dilakukan perbaikan citra. *Image Enhancement* juga merupakan salah satu proses awal (*preprocessing*) di dalam bidang *computer vision*. Citra yang perlu dilakukan *image enhancement* adalah citra-citra yang mengandung derau (*noise*). Citra yang terlihat terlalu terang/gelap, citra kurang tajam, citra kabur (*blur*), citra yang terdapat cacat saat akuisisi citra. Maupun citra yang mengalami distorsi geometrik yang disebabkan oleh lensa atau sudut pengambilan gambar. Beberapa contoh proses yang termasuk ke dalam perbaikan citra adalah Pengubahan kecerahan gambar (*image brightening*), Citra negatif (*image negatives*), Peregangan kontras (*contrast stretching*), Pengubahan histogram citra.

## 1.3. Pencerahan Citra (*Image Brightening*)

*Image Brightening* atau Pencerahan Citra merupakan salah satu proses untuk memperbaiki citra dengan cara menambahkan atau mengurangkan sebuah konstanta kepada (atau dari) setiap pixel, atau mengalikan sebuah konstanta ke setiap pixel. *Image Brightening* dapat digambarkan dengan fungsi ( $s = r + b$  dan  $s = ar + b$ , dengan  $a$  dan  $b$  adalah konstanta). Jika  $a$  lebih dari satu atau  $b$  bernilai positif maka kecerahan citra

bertambah sedangkan jika nilai  $a$  kurang dari 1 atau  $b$  negatif kecerahan citra berkurang. Perlu dilakukan operasi *clipping* jika nilai  $ar + b$  berada dibawah nilai intensitas minimum atau di atas nilai intensitas maksimum.

#### 1.4. Menegatifkan Citra (*Image Negative*)

Seperti film negatif yang terdapat pada fotografi, menegatifkan citra merupakan salah satu proses untuk memperbaiki citra dengan cara mengurangi nilai intensitas pixel dari nilai keabuan maksimum ( $L-1$ ). Menegatifkan citra digambarkan dengan fungsi ( $s = (L - 1) - r$ ). Jika citra merupakan citra *grayscale* 8 bit maka ( $s = 255 - r$ ).

#### 1.5. Transformasi Log

Transformasi log merupakan salah satu transformasi dasar keabuan yang digunakan untuk perbaikan kualitas citra. Transformasi log digambarkan dengan fungsi ( $s = c \log(1 + r)$ ). Transformasi Log memiliki beberapa sifat sebagai berikut, Untuk citra yang memiliki rentang yang sempit untuk nilai-nilai keabuan yang rendah (gelap), dipetakan menjadi rentang yang lebih luas pada citra luaran, dan untuk citra yang memiliki rentang yang lebar untuk nilai-nilai keabuan yang tinggi (terang), dipetakan menjadi rentang yang lebih sempit pada citra luaran. Pada transformasi log balikan, yang terjadi adalah kebalikannya.

#### 1.6. Transformasi Pangkat

Transformasi pangkat juga merupakan salah satu transformasi dasar keabuan yang digunakan untuk perbaikan kualitas citra. Transformasi pangkat digambarkan dengan fungsi ( $s = cr^y$ , dengan  $c$  dan  $y$  merupakan konstanta bernilai positif). Jika nilai  $y$  kurang dari 1 maka akan membuat citra menjadi lebih terang, jika nilai  $y$  lebih dari satu akan membuat citra menjadi lebih gelap, jika  $y = 1$  dan  $c = 1$  maka transformasi identitas ( $s = r$ ).

#### 1.7. Peregangan Kontras (*contrast stretching*)

Peregangan kontras merupakan metode sederhana untuk memperbaiki citra yang memiliki kontras rendah. Tujuan dari peregangan kontras adalah untuk meningkatkan

rentang nilai-nilai keabuan untuk citra kontras-rendah (terentang dari nilai  $r_1$  sampai  $r_2$  pada citra dengan nilai keabuan 0 sampai  $L - 1$ ). Nilai-nilai pixel antara  $r_1$  sampai  $r_2$  akan dipetakan menjadi nilai antara  $s_1$  sampai  $s_2$ .

### 1.8. Perataan Histogram (*Histogram Equalization*)

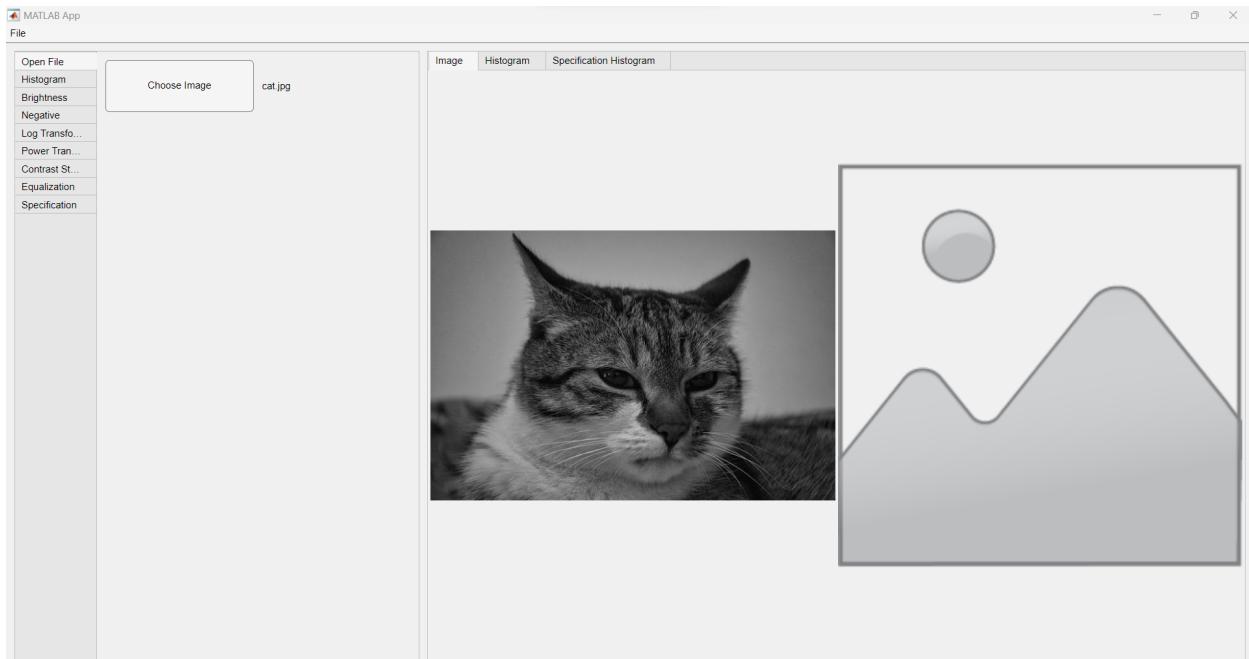
Perataan histogram (*histogram equalization*) merupakan penyebaran nilai-nilai intensitas di dalam citra diubah sedemikian sehingga penyebarannya seragam (*uniform*). Tujuan dari perataan histogram adalah untuk memperoleh penyebaran histogram yang merata, sedemikian sehingga setiap derajat keabuan memiliki jumlah pixel yang relatif sama. Perataan histogram akan memperlebar rentang nilai keabuan, sehingga dapat meningkatkan kekontrasan citra termasuk operasi dalam aras global. Perataan histogram artinya mengubah derajat keabuan  $r$  dengan derajat keabuan yang baru  $s$ , dengan fungsi transformasi  $T$  ( $s = T(r)$ )

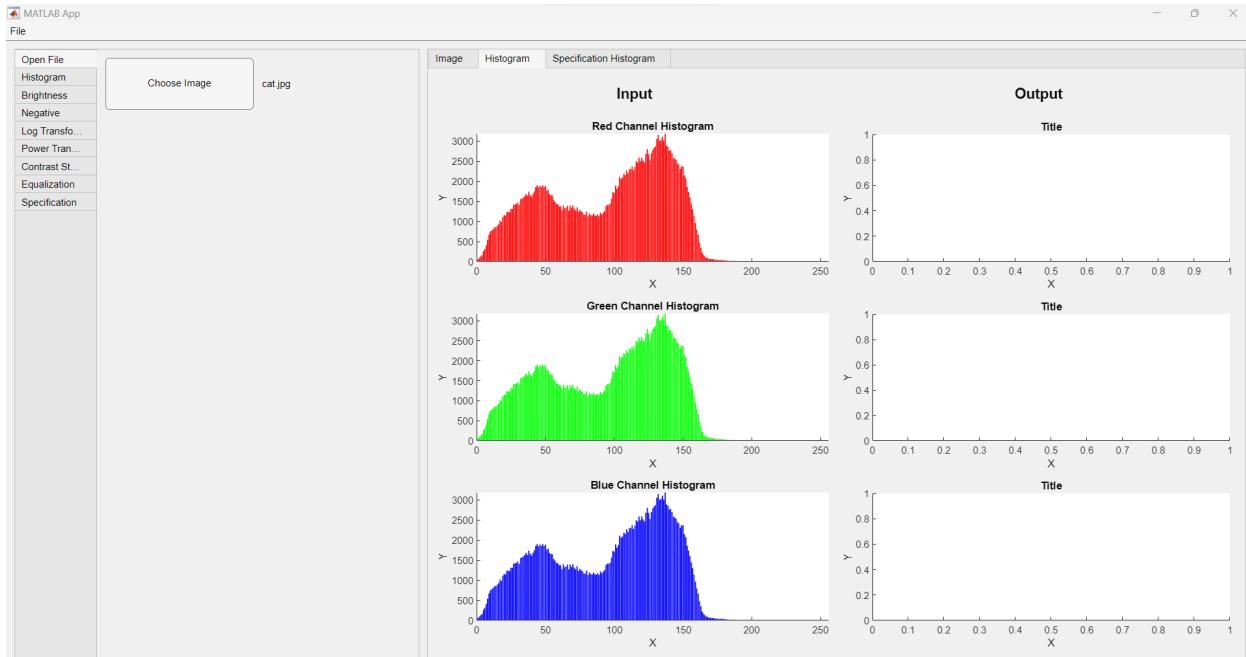
### 1.9. Spesifikasi Histogram (*Histogram Specification*)

Pada Perataan histogram seringkali histogram hasil perataan tidak benar benar tersebar secara merata dikarenakan derajat keabuan terbatas jumlahnya dan Nilai intensitas baru hasil perataan merupakan pembulatan ke derajat keabuan terdekat. Selain itu Jumlah pixel yang digunakan sangat terbatas. Untuk mengatasi masalah tersebut, digunakannya metode spesifikasi histogram. Spesifikasi Histogram memetakan histogram citra semula menjadi histogram yang seragam. Jika histogram tidak seragam, maka cara ini tidak dapat digunakan. Metode spesifikasi histogram (*histogram specification*) atau pencocokan histogram (*histogram matching*) memberikan cara menghasilkan histogram yang ditentukan oleh pengguna.

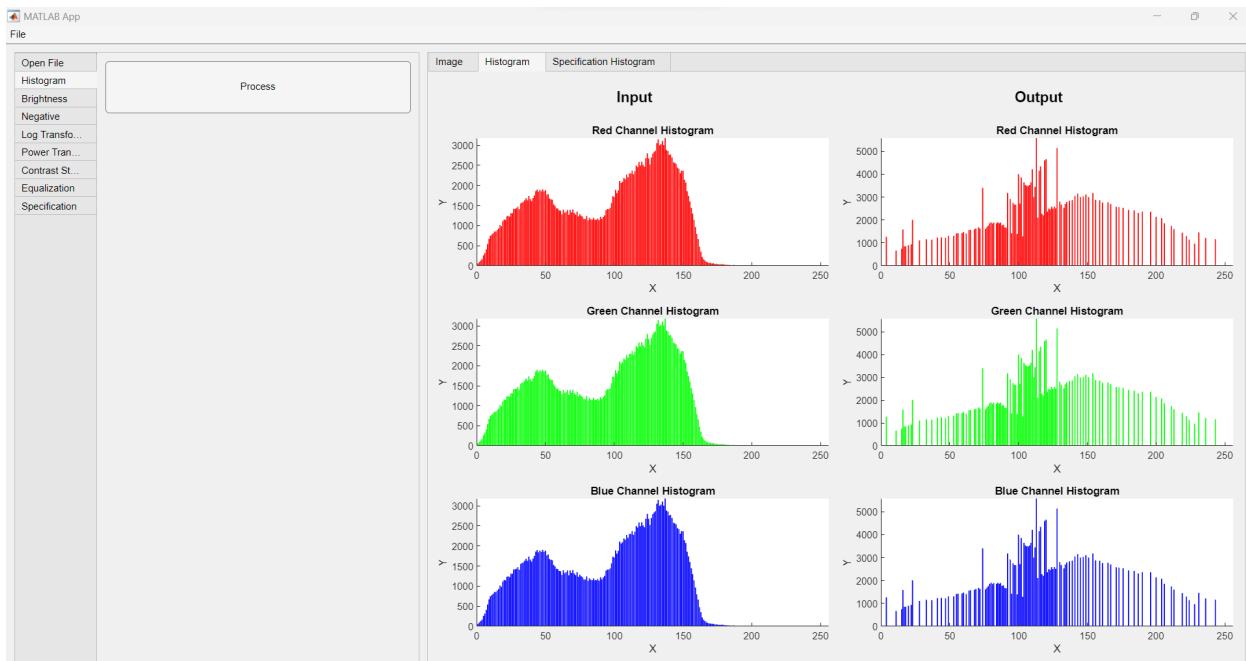
## II GUI Program

Berikut adalah beberapa tangkapan layar GUI program. Sebelum memproses citra, pengguna perlu memasukkan gambar pada menu Open File atau dropdown toolbar “File > Open”. Pada tiap menu transformasi, terdapat tombol “Process” dan “Apply”. Tombol “Process” digunakan untuk melakukan transformasi pada citra masukan, menghasilkan citra dan histogram keluaran yang dapat dilihat pada tab “Image” dan “Histogram”. Tombol “Apply” digunakan untuk mengaplikasikan transformasi sehingga citra luaran menjadi citra masukan (misal pengguna ingin melakukan power transformation setelah equalization). Process harus dilakukan sebelum Apply. Terdapat tab “Specification Histogram” yang khusus digunakan oleh histogram specification, berisi citra dan histogram gambar acuan.

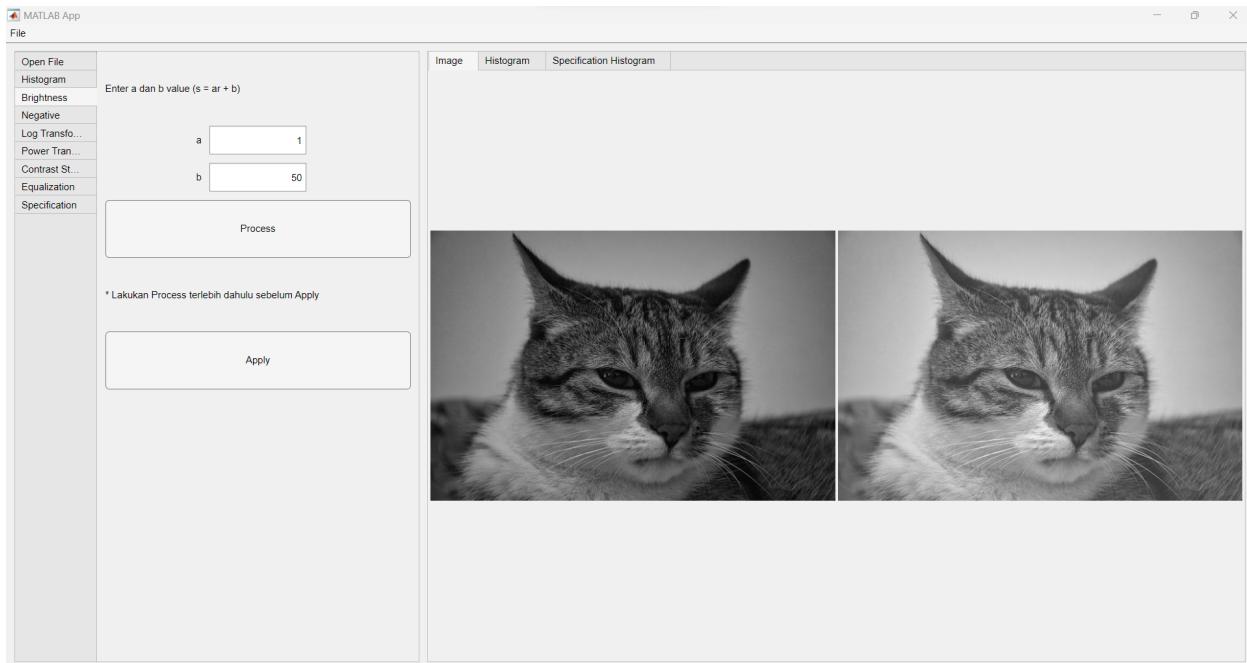




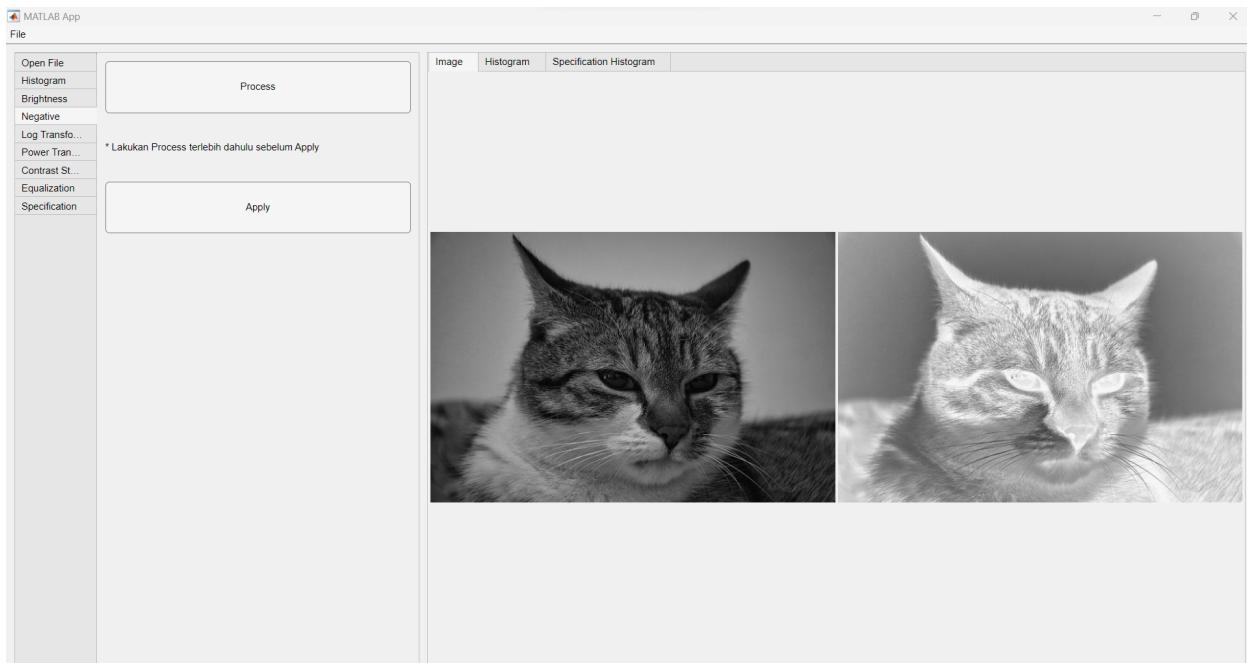
Gambar 2.1. Menu Open File: tab “Image” (atas); tab “Histogram” (bawah)



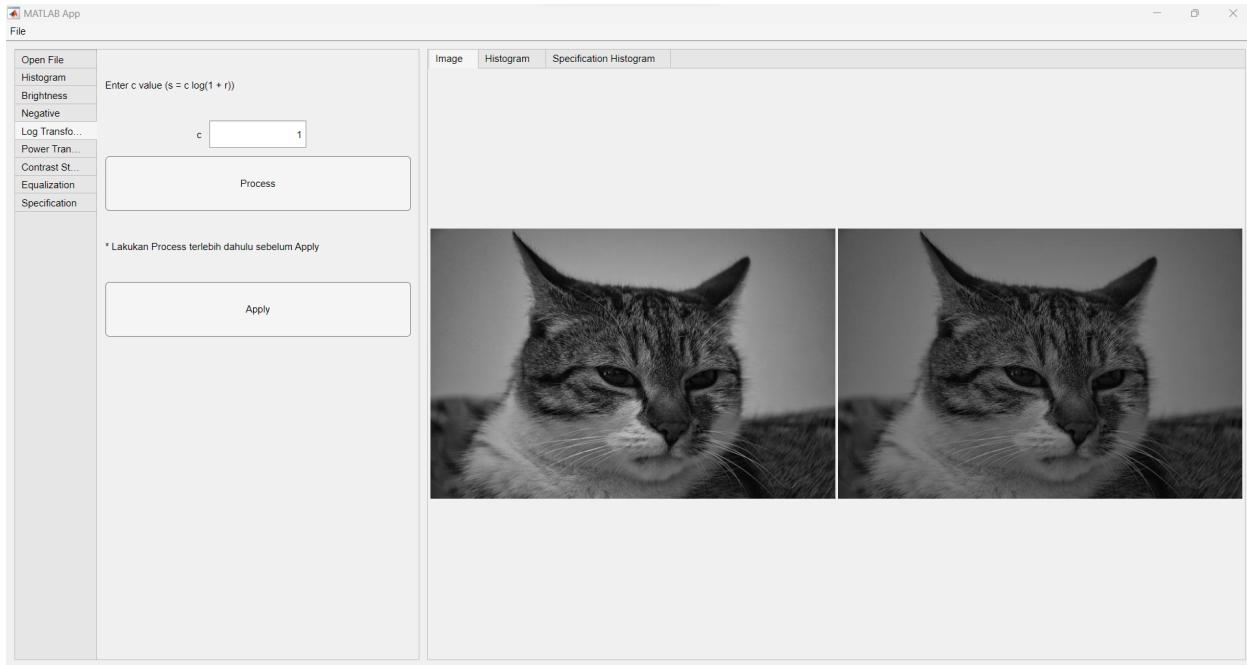
Gambar 2.2. Menu Histogram



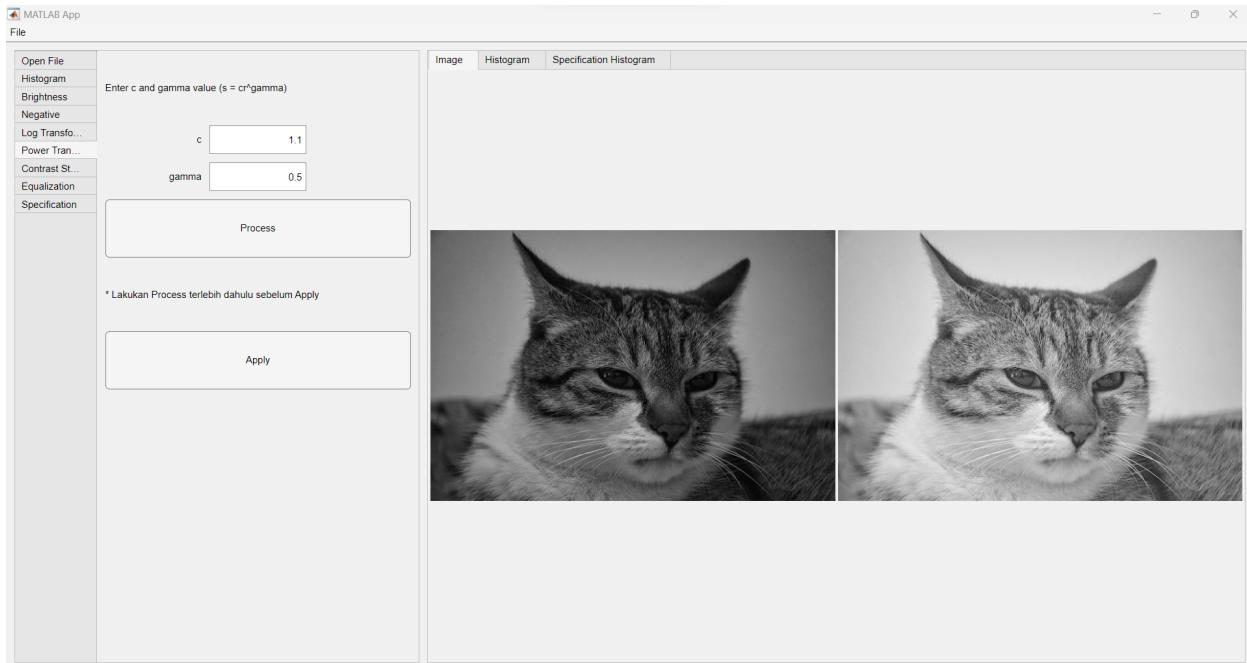
Gambar 2.3. Menu Brightness



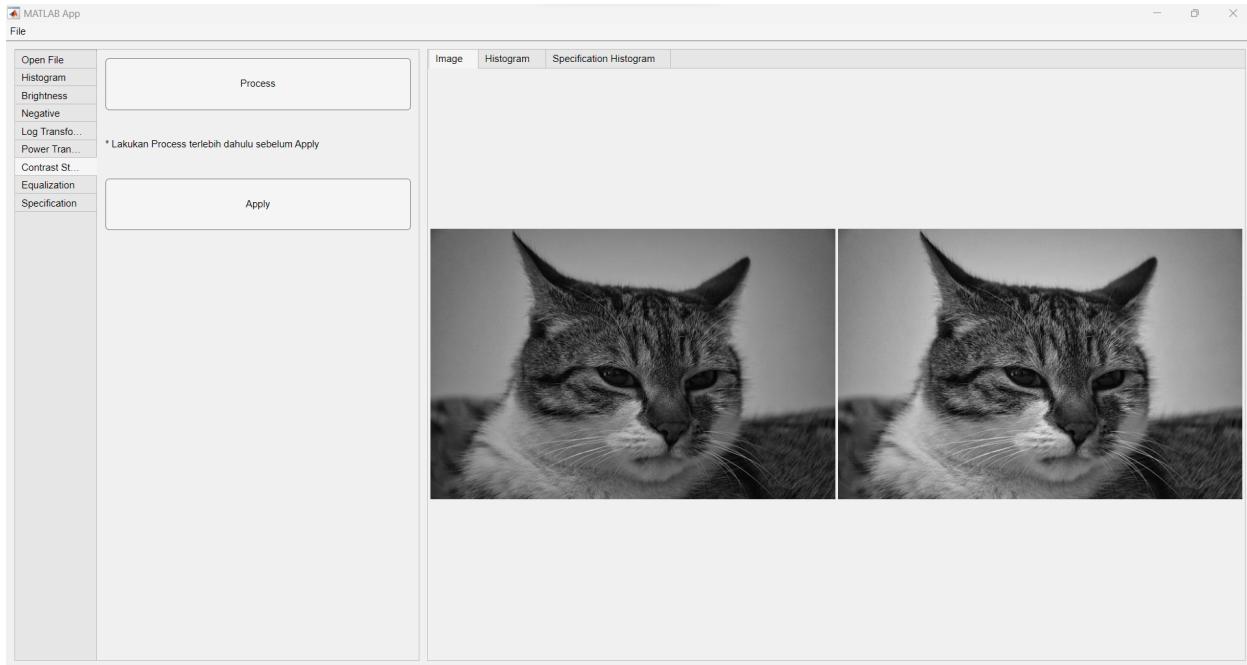
Gambar 2.4. Menu Negative



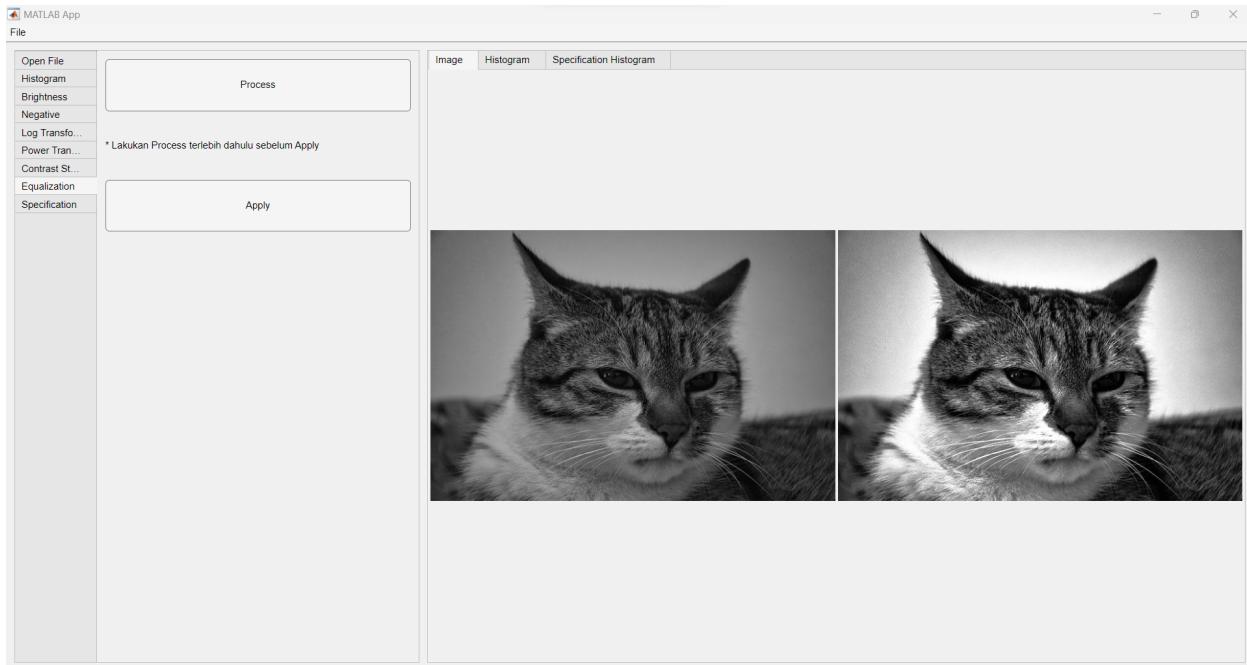
Gambar 2.5. Menu Log Transformation



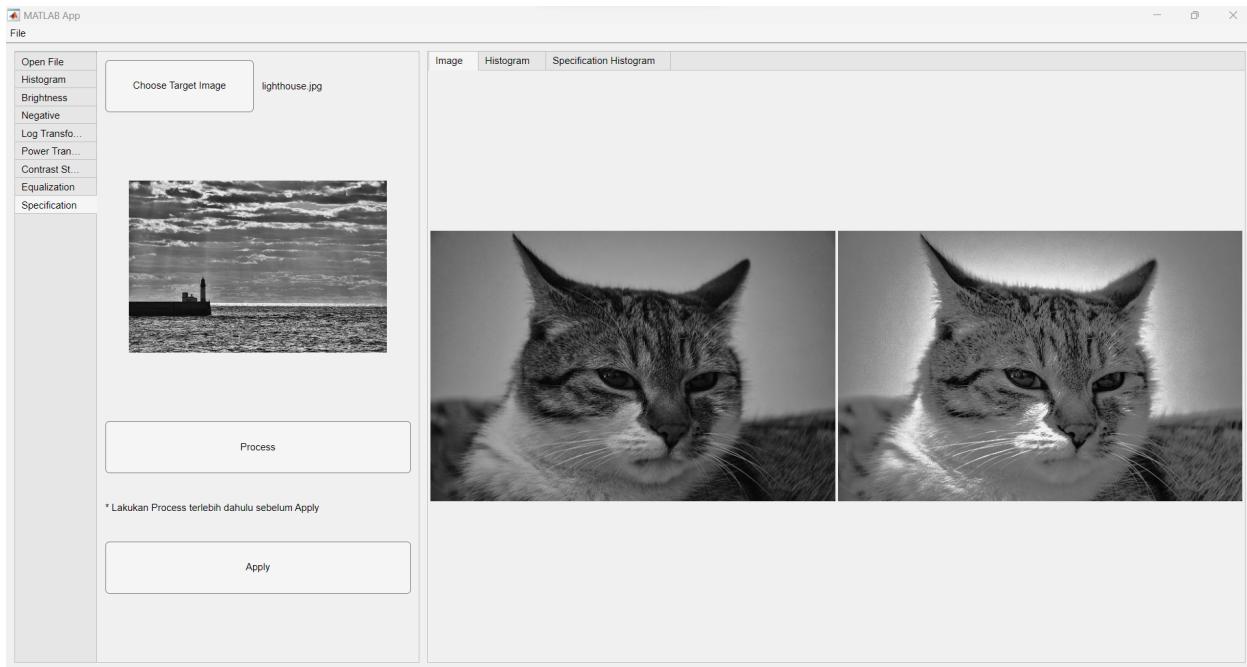
Gambar 2.6. Menu Power Transformation



Gambar 2.7. Menu Contrast Stretching



Gambar 2.8. Menu Equalization



Gambar 2.9. Menu Specification

### III Pembahasan

#### 3.1. Histogram citra Grayscale dan Citra Berwarna

Pada program ini akan dihitung dan ditampilkan histogram dari citra grayscale maupun citra berwarna dengan 256 derajat keabuan.

##### 3.1.1. Kode Program

Berikut merupakan potongan kode program untuk menampilkan histogram citra rgb

```
...
redChannel = app.Inputimagedata(:, :, 1);
greenChannel = app.Inputimagedata(:, :, 2);
blueChannel = app.Inputimagedata(:, :, 3);

% create histogram matrix
redHistogram = zeros(256, 1);
greenHistogram = zeros(256, 1);
blueHistogram = zeros(256, 1);

% Calculate histograms for each channel
[height, width] = size(redChannel);

% calculate r,g, and b value
for i = 1:height
    for j = 1:width
        redValue = redChannel(i, j) + 1; % Add 1 to avoid indexing issues
        greenValue = greenChannel(i, j) + 1;
        blueValue = blueChannel(i, j) + 1;

        redHistogram(redValue) = redHistogram(redValue) + 1;
        greenHistogram(greenValue) = greenHistogram(greenValue) + 1;
        blueHistogram(blueValue) = blueHistogram(blueValue) + 1;
    end
End

% Plot histograms
bar(app.UIAxes, 0:255, redHistogram, 'r'); % Red channel histogram
bar(app.UIAxes2, 0:255, greenHistogram, 'g'); % Green channel histogram
bar(app.UIAxes3, 0:255, blueHistogram, 'b'); % Blue channel histogram
...
```

Sedangkan berikut untuk citra grayscale

```
...
% set to show image for greyscale
rgbimage = cat(3, app.Inputimagedata(:, :, 1), app.Inputimagedata(:, :,
```

```

1),app.Inputimagedata(:, :, 1));
% Display the image in the uiimage component
app.InputImage.ImageSource = rgbimage;

% grey channel
greyChannel = app.Inputimagedata(:, :, 1);

% grey histogram
greyHistogram = zeros(256, 1);

% Calculate histograms for each channel
[height, width] = size(greyChannel);

% calculate grey value
for i = 1:height
    for j = 1:width
        greyValue = greyChannel(i, j) + 1; % Add 1 to avoid indexing issues
        greyHistogram(greyValue) = greyHistogram(greyValue) + 1;
    end
end
% plot histograms
bar(app.UIAxes,0:255, greyHistogram, 'k'); % Grey channel histogram
...

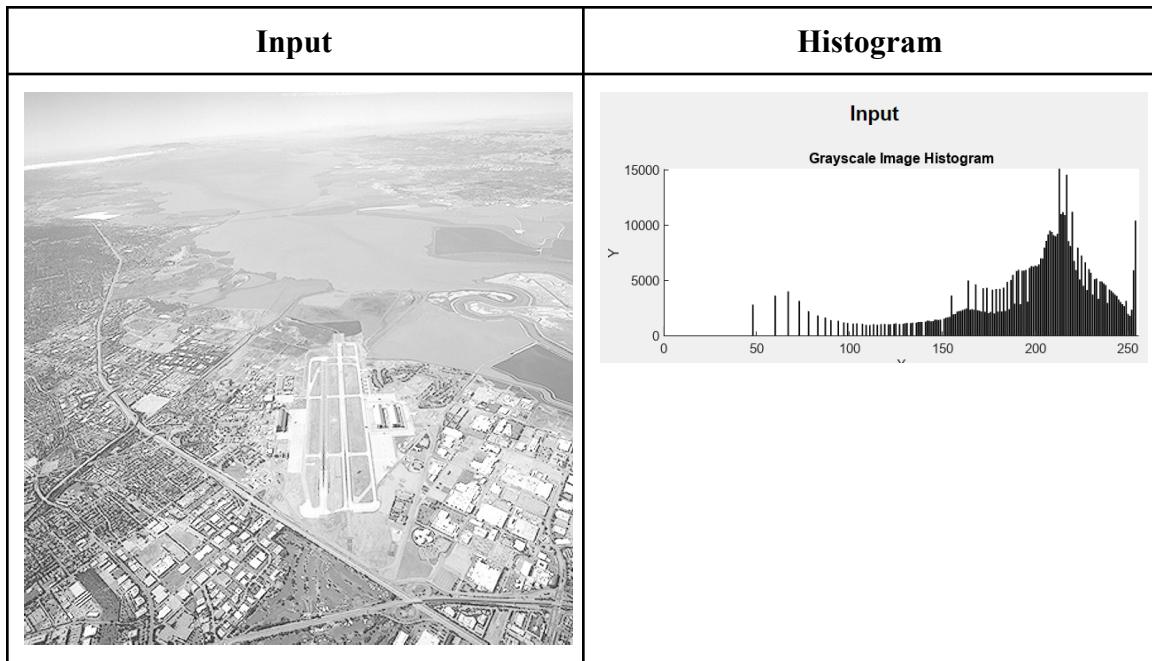
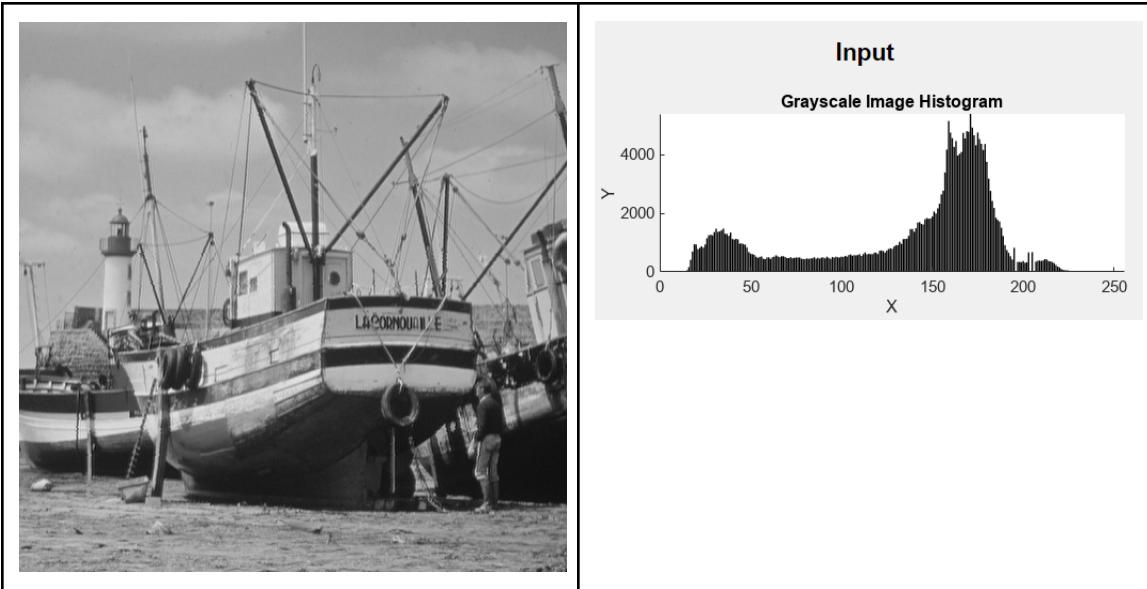
```

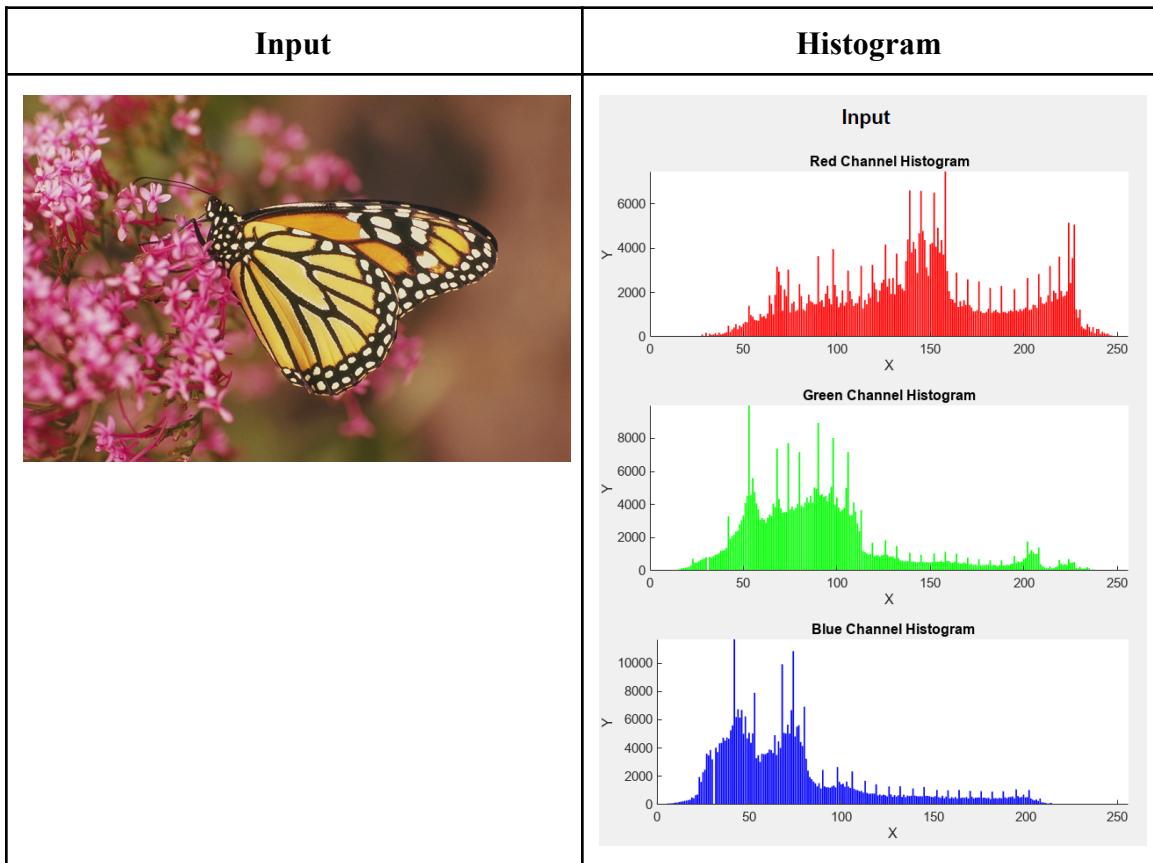
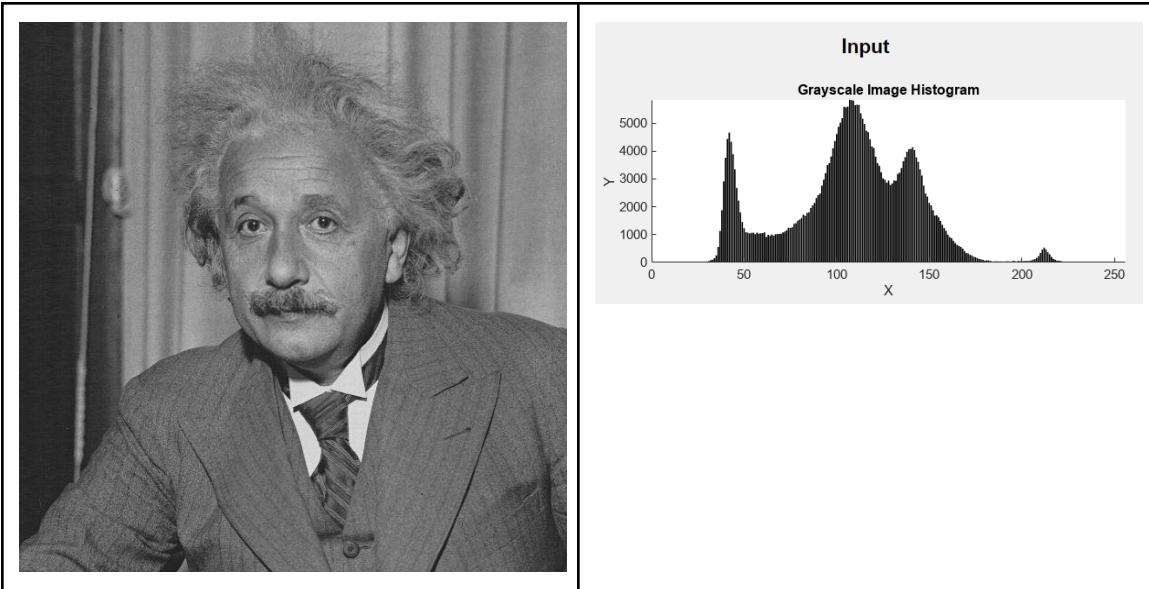
Pada program tersebut pertama tama akan dilakukan pemisahan untuk nilai red, green, dan blue untuk image yang rgb sedangkan untuk grayscale hanya mengambil nilai grey nya saja. Setelah dipisah, akan dilakukan inisialisasi matriks untuk menyimpan histogram untuk setiap saluran warna. Dilakukan loop untuk menghitung frekuensi kemunculan intensitas warna dan menyimpannya ke dalam matriks yang sudah diinisialisasi sebelumnya. Histogram tersebut nantinya akan ditampilkan pada elemen UIAxes yang terdapat pada GUI.

### 3.1.2. Testing

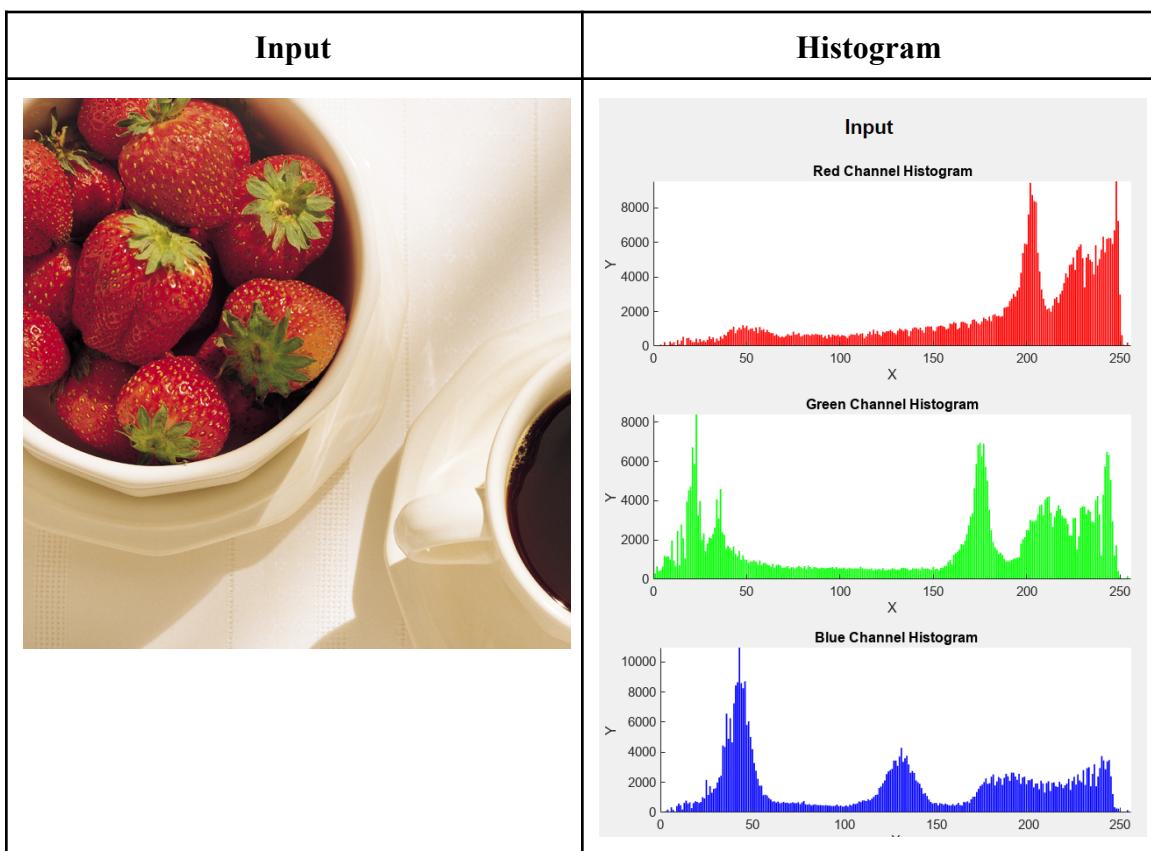
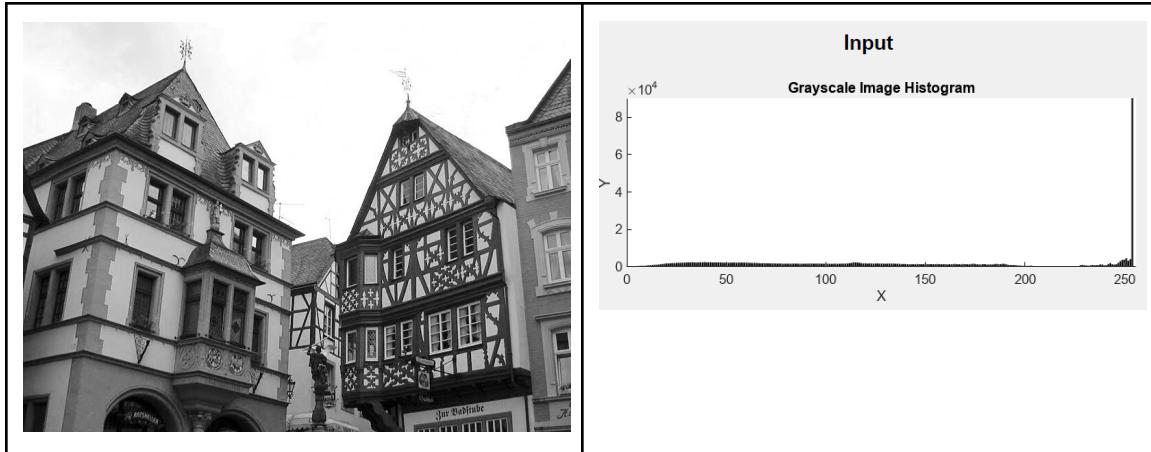
Berikut testing program Histogram Citra menggunakan beberapa image berikut ini (note: beberapa gambar diambil berdasarkan source di online karena tidak menemukan gambar tersebut pada link yang telah disediakan)

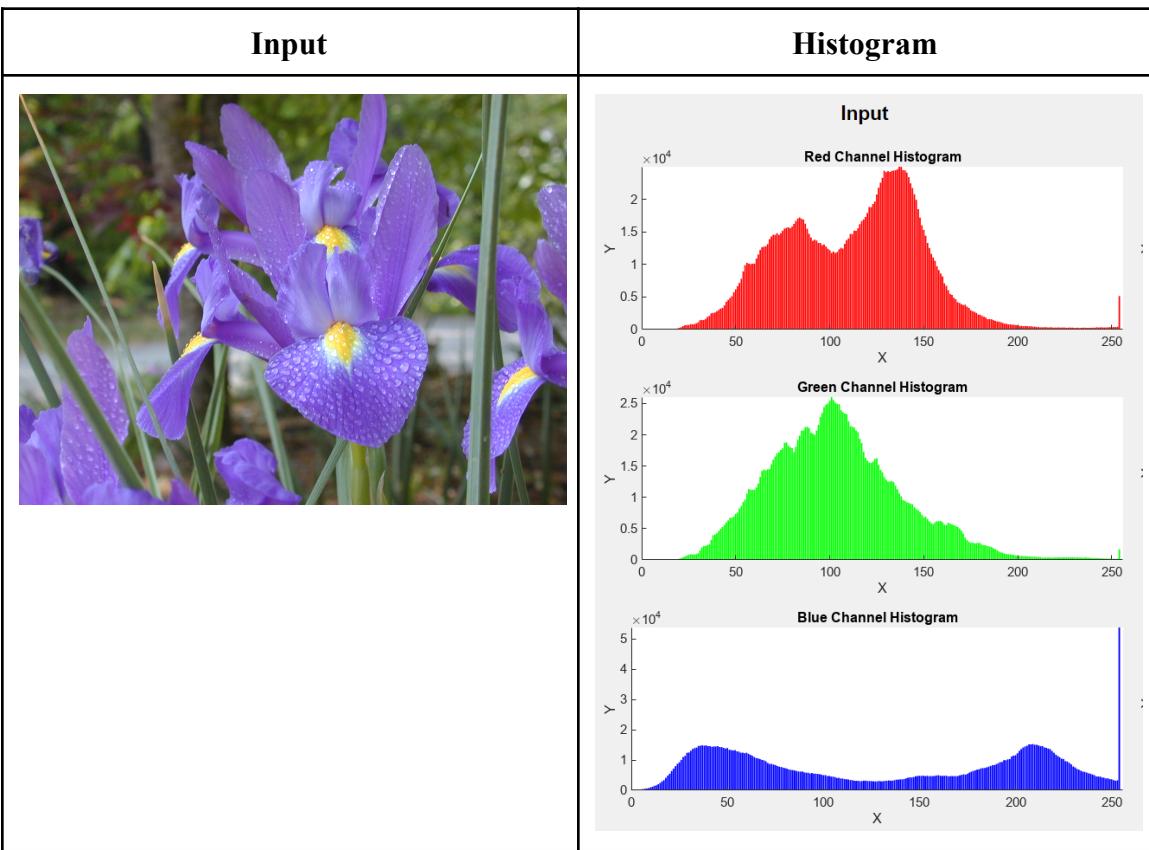
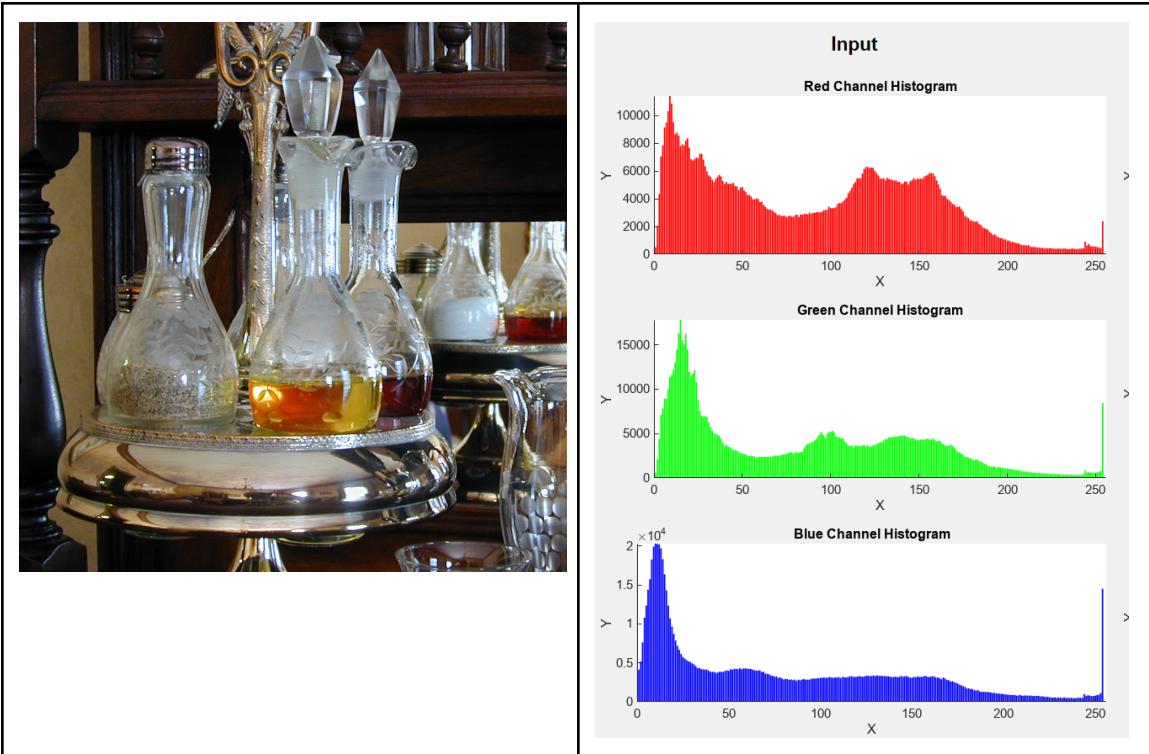
Input	Histogram
-------	-----------

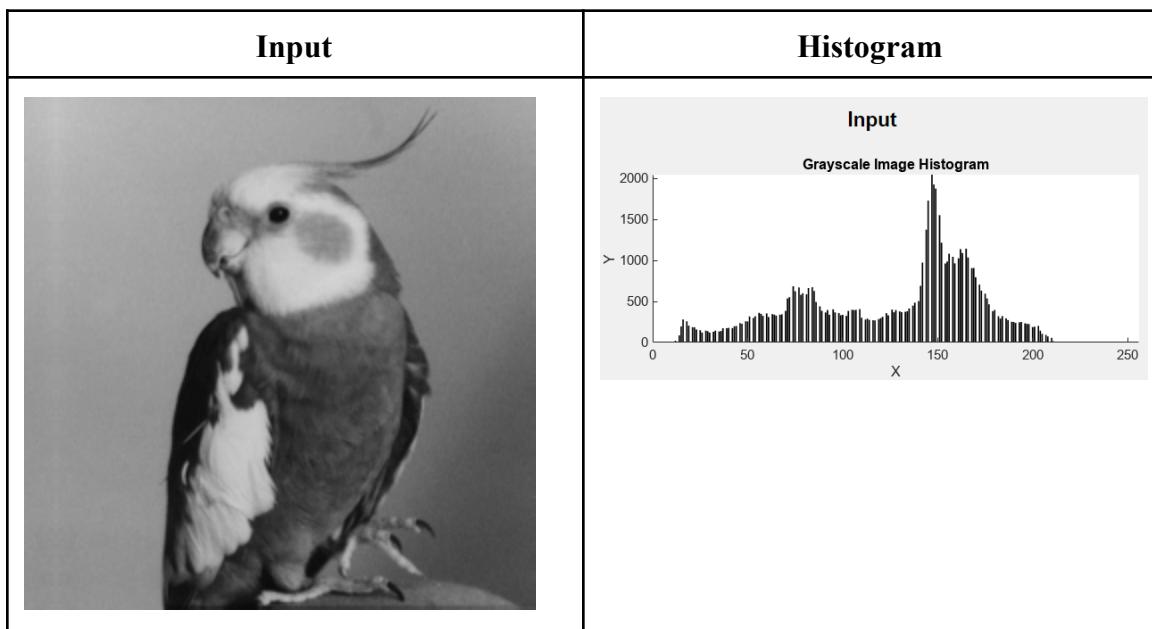




<b>Input</b>	<b>Histogram</b>
--------------	------------------







### 3.1.3. Analisis

Berdasarkan testing yang telah dilakukan dengan menggunakan 5 gambar *grayscale*, dan 3 gambar RGB, program mampu menampilkan histogram dari masing-masing gambar dengan baik.

## 3.2. Perbaikan Kualitas Citra

### 3.2.1. Kode Program

Berikut adalah fungsi untuk melakukan *image brightening*. Fungsi menghitung citra keluaran dengan rumus  $s = a * r + b$ . Nilai keluaran mungkin lebih dari 255 atau kurang dari 0 sehingga nilai-nilai yang melebihi 255 dipetakan ke 255 dan yang kurang dari 0 dipetakan ke 0.

```

function result = imageBrightening(a, b, r)
    % Doing calculation
    s = a * r + b;

    % Replace values > 255 to 255 and values < 0 to 0
    s(s > 255) = 255;
    s(s < 0) = 0;

    % Set output image
    result = s;
end

```

Berikut adalah fungsi untuk melakukan *negative* dan *inverse negative transformation*. Fungsi menghitung citra keluaran dengan rumus  $255 - \text{Inputimagedata}$ . Nilai pasti berada dalam rentang 0-255 sehingga tidak diperlukan pemrosesan setelah kalkulasi.

```
function result = negativeImage(Inputimagedata)
    result = 255 - Inputimagedata;
end
```

Berikut adalah fungsi untuk melakukan *log transformation*. Nilai piksel diubah terlebih dahulu ke *double*, hasilnya memiliki rentang 0.0-1.0. Setelah itu, citra keluaran dihitung sebagai  $s = c * \log(1 + r)$ . Nilai hasil mungkin di luar rentang 0.0-1.0 sehingga yang melebihi 1 diubah ke 1 dan yang kurang dari 0 diubah ke 0. Hasilnya kemudian diubah ke *uint8* yang memiliki rentang 0-255 dan dikembalikan.

```
function result = logTransformation(c, Inputimagedata)
    r = im2double(Inputimagedata); % convert image data to double

    % Doing calculation
    s = c * log(1 + r);

    % Replace values > 1 to 1 and values < 0 to 0
    s(s > 1) = 1;
    s(s < 0) = 0;

    % Set output image (and map resulting data to uint8)
    result = im2uint8(s);
end
```

Berikut adalah fungsi untuk melakukan *power transformation*. Nilai piksel diubah terlebih dahulu ke *double*, hasilnya memiliki rentang 0.0-1.0. Setelah itu, citra keluaran dihitung sebagai  $s = c * (r .^ \gamma)$ . Nilai hasil mungkin di luar rentang 0.0-1.0 sehingga yang melebihi 1 diubah ke 1 dan yang kurang dari 0 diubah ke 0. Hasilnya kemudian diubah ke *uint8* yang memiliki rentang 0-255 dan dikembalikan.

```
function result = powerTransformation(c, gamma, Inputimagedata)
    r = im2double(Inputimagedata); % convert image data to double

    % Doing calculation
    s = c * (r .^ gamma);

    % Replace values > 1 to 1 and values < 0 to 0
    s(s > 1) = 1.0;
```

```

s(s < 0) = 0.0;

% Set output image (and map resulting data to uint8)
result = uint8(s * 255);
end

```

Berikut adalah fungsi untuk melakukan *contrast stretching*. Pertama, dicari nilai piksel terendah (rmin) dan nilai tertinggi (rmax). Selanjutnya, dilakukan *stretching* dengan rumus  $(\text{Inputimagedata} - \text{rmin}) \cdot (255 / (\text{rmax} - \text{rmin}))$ . Nilai pasti dalam rentang 0-255 sehingga dapat langsung dikembalikan tanpa pemrosesan tambahan.

```

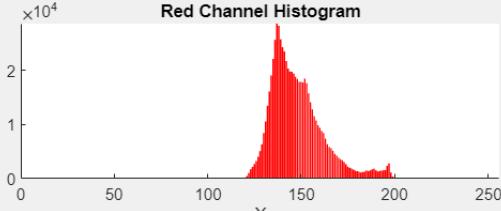
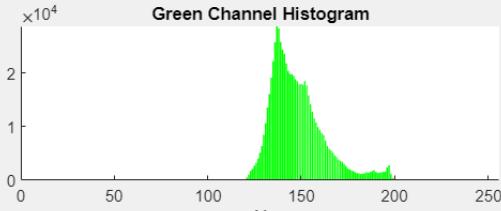
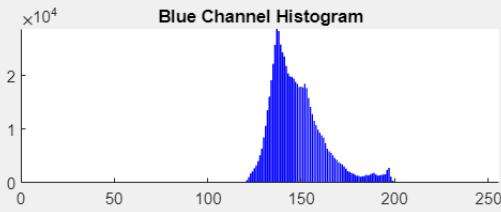
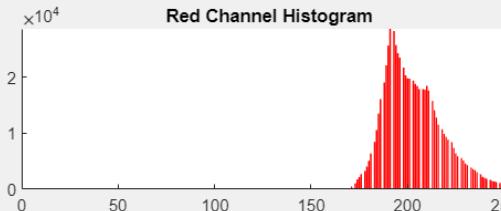
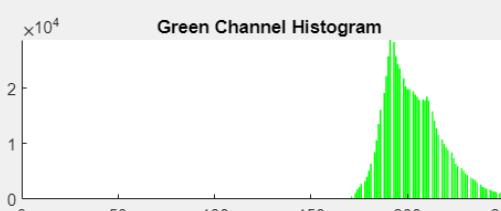
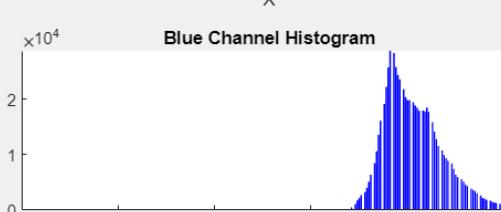
function result = contrastStretching(numChannels, Inputimagedata)
    % Check number of channels
    if numChannels == 1
        % Find minimum and maximum value
        rmin = min(min(Inputimagedata));
        rmax = max(max(Inputimagedata));

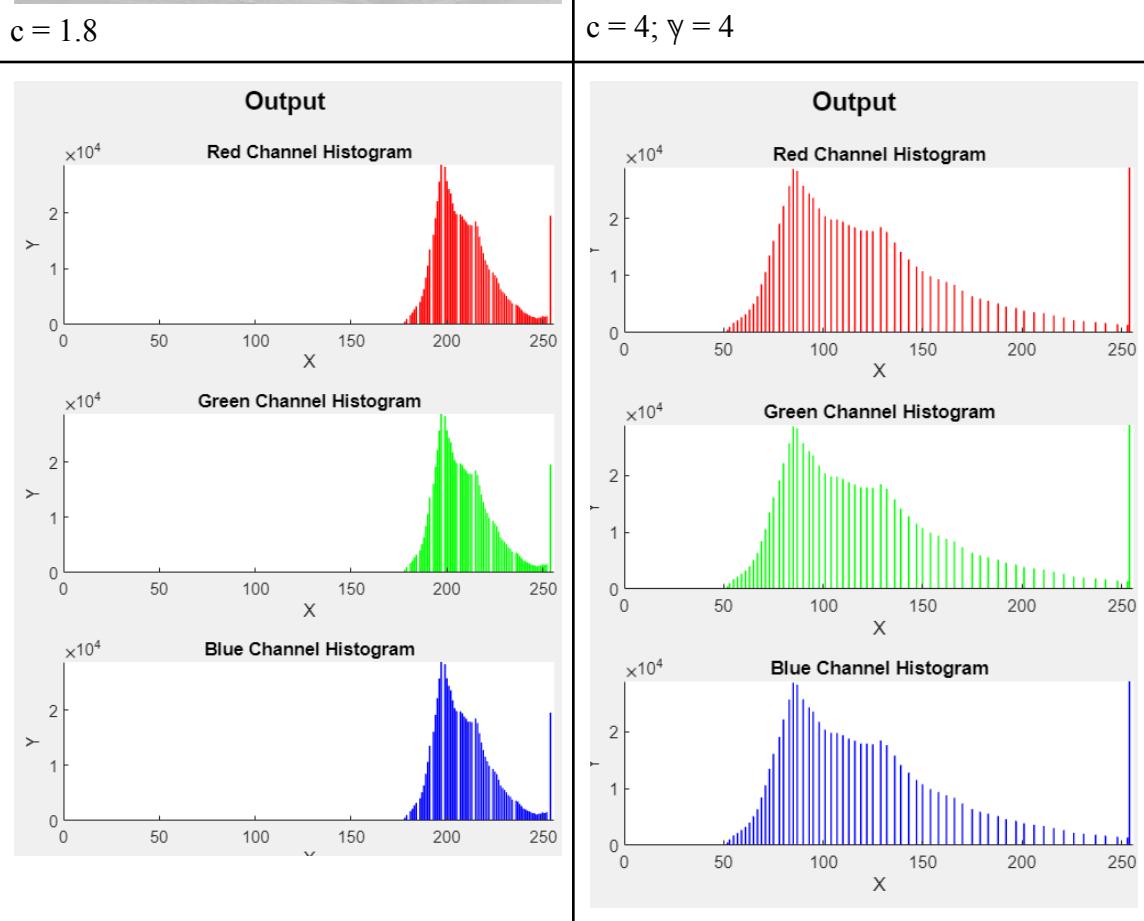
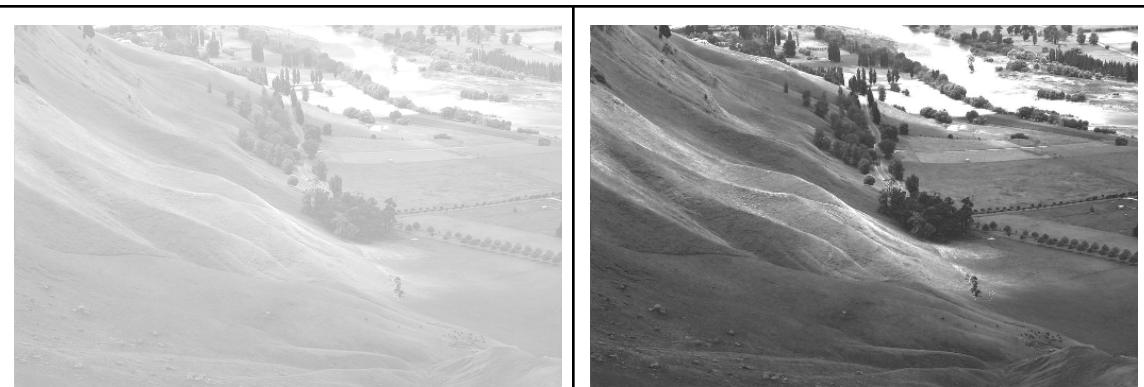
        % Do calculation and set result
        result = (Inputimagedata - rmin) .* (255/ (rmax - rmin));
    elseif numChannels == 3
        % Find minimum and maximum value for each channel
        rmin = [0, 0, 0];
        rmax = [0, 0, 0];
        for i = 1:numChannels
            rmin(i) = min(min(Inputimagedata(:,:,i)));
            rmax(i) = max(max(Inputimagedata(:,:,i)));
        end
        rmin = max(rmin);
        rmax = min(rmax);

        % Do calculation and set result
        result = (Inputimagedata - rmin) .* (255/ (rmax - rmin));
    else
        msgbox('Unsupported image format', 'Error', 'error');
    end
end

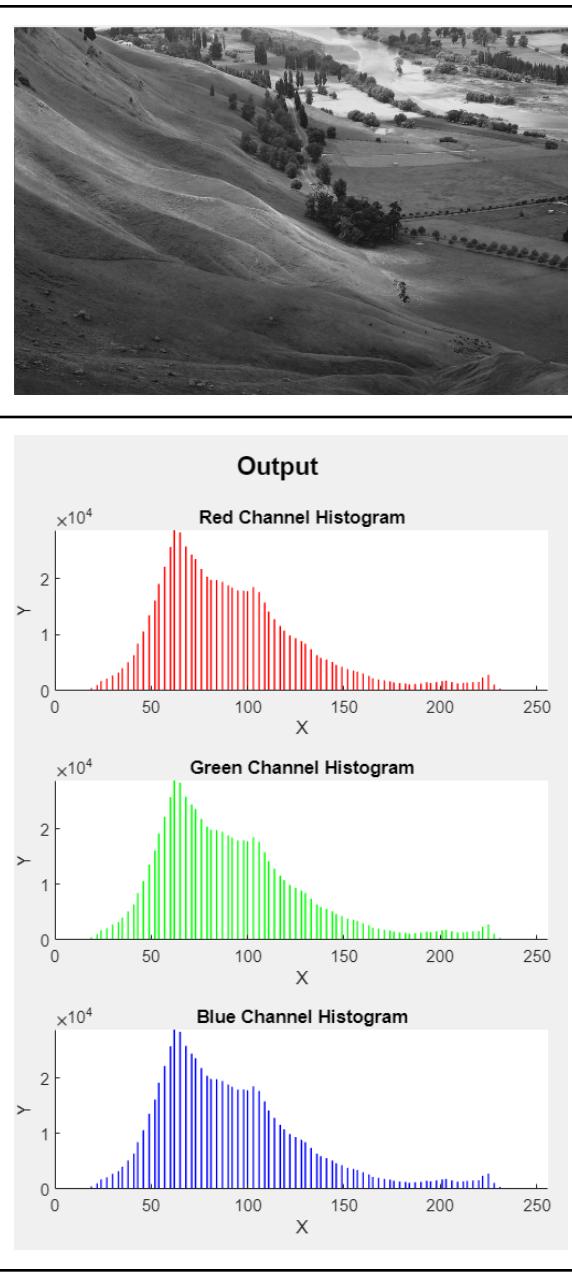
```

### 3.2.2. Testing

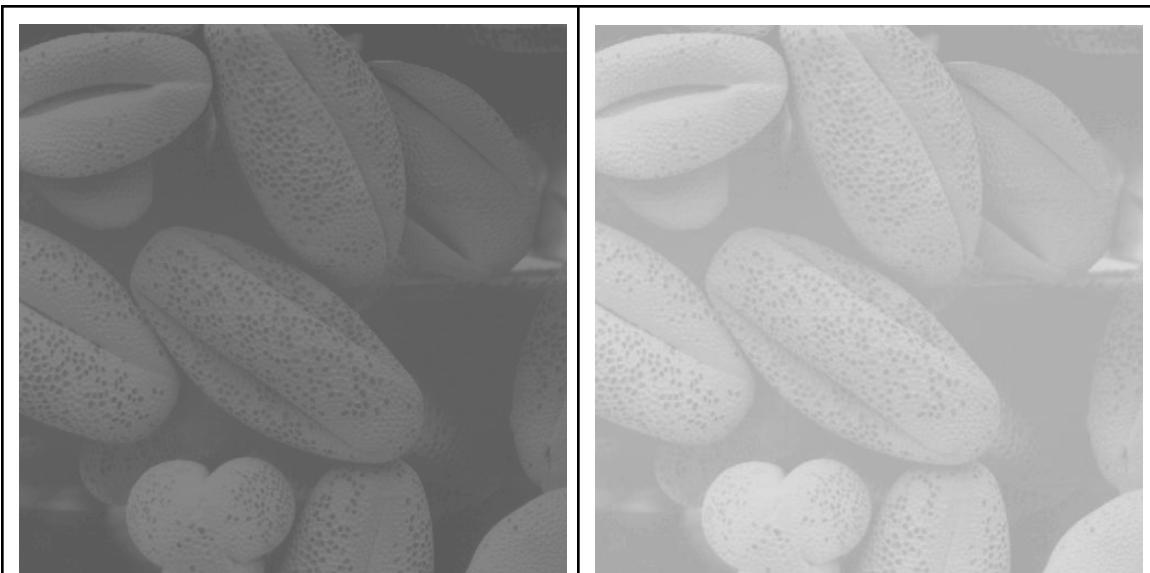
Input	Output (Image brightening)
	
	$a = 1.25; b = 20$
<p style="text-align: center;"><b>Input</b></p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Red Channel Histogram</p>  </div> <div style="text-align: center;"> <p>Green Channel Histogram</p>  </div> <div style="text-align: center;"> <p>Blue Channel Histogram</p>  </div> </div>	<p style="text-align: center;"><b>Output</b></p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Red Channel Histogram</p>  </div> <div style="text-align: center;"> <p>Green Channel Histogram</p>  </div> <div style="text-align: center;"> <p>Blue Channel Histogram</p>  </div> </div>
<b>Output (Transformasi Log)</b>	<b>Output (Transformasi Pangkat)</b>



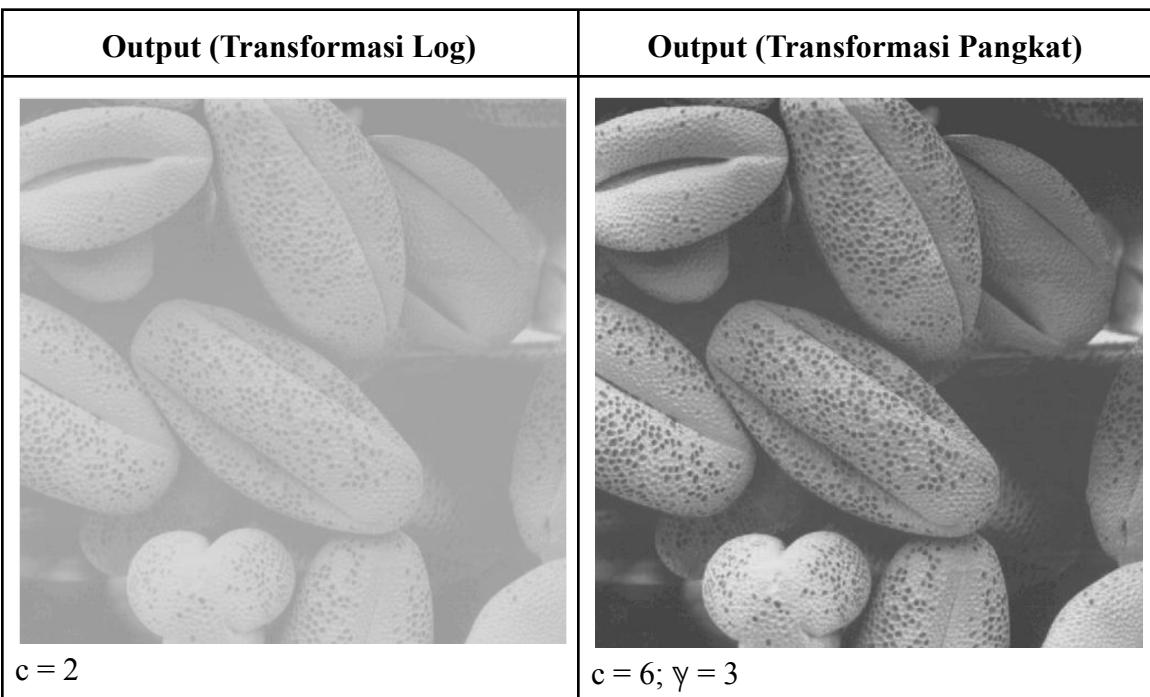
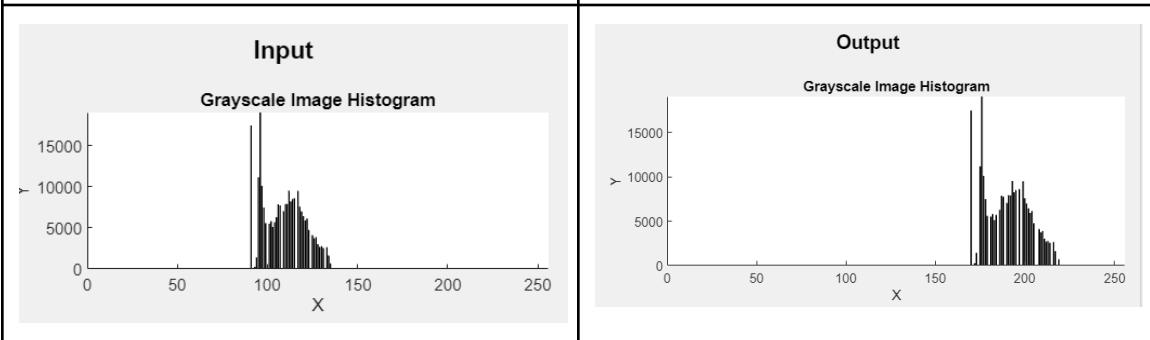
**Output (Peregangan Kontras)**

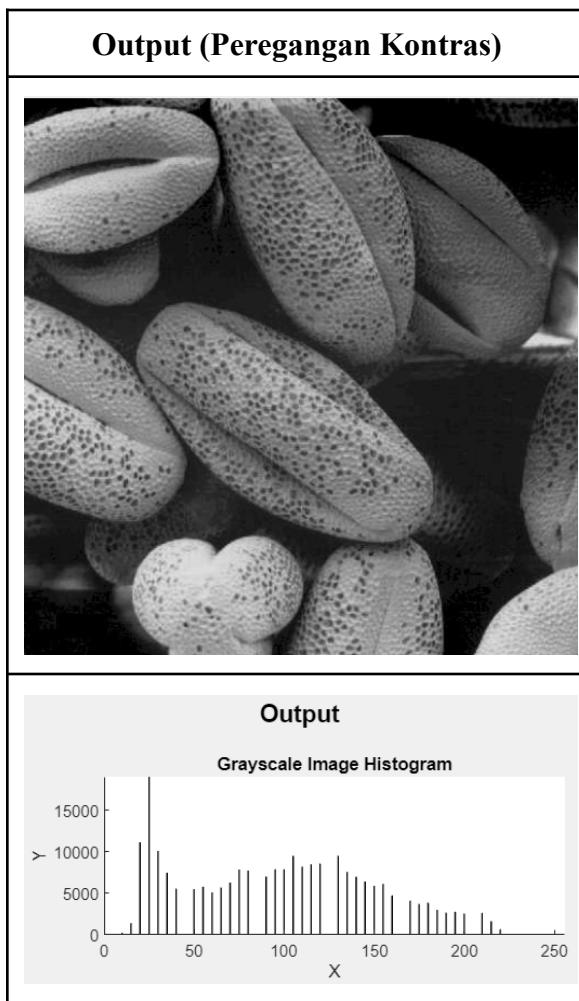
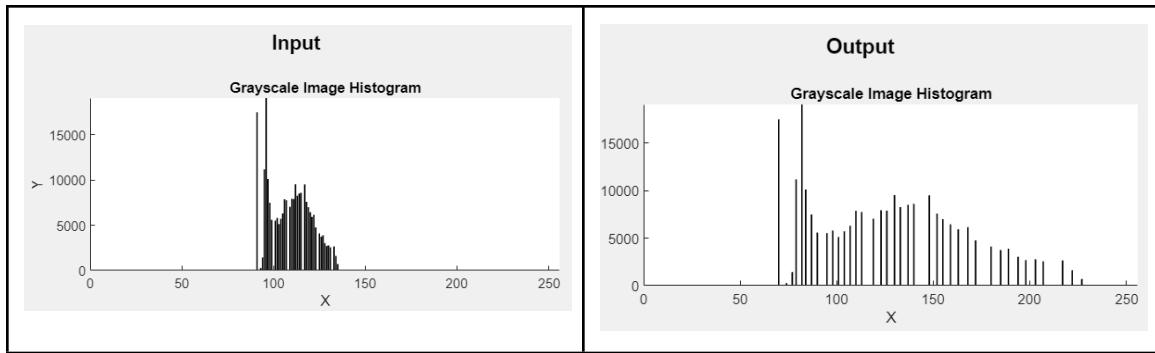


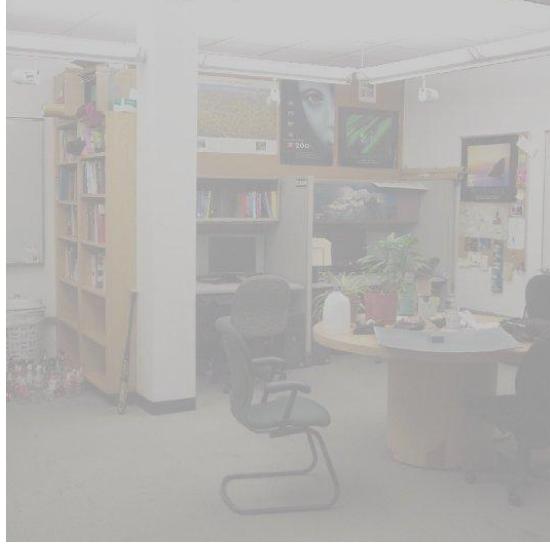
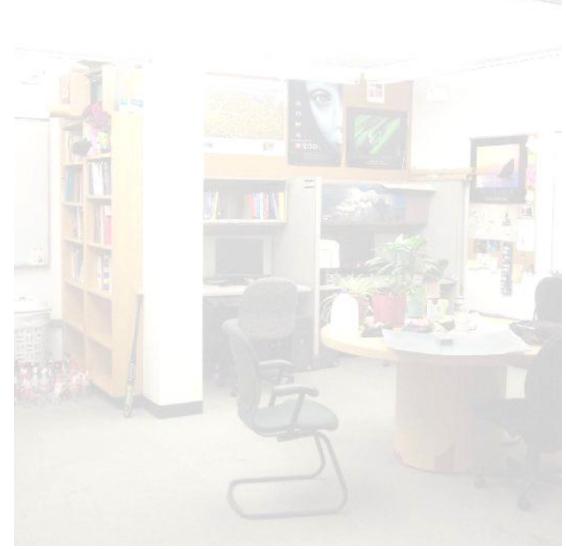
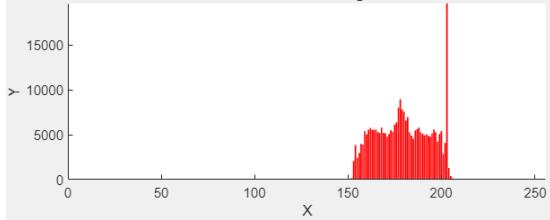
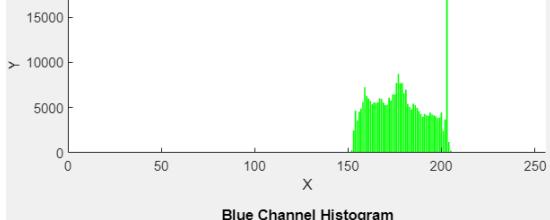
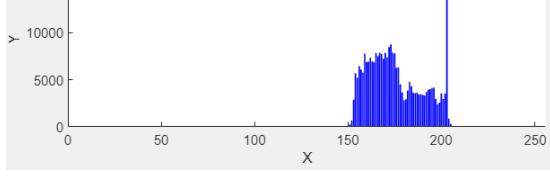
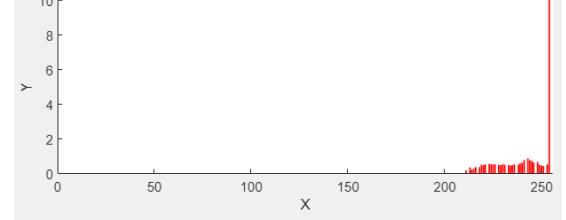
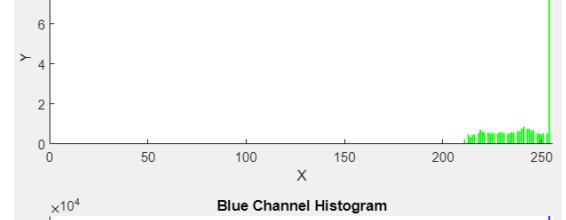
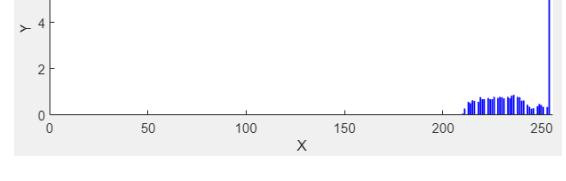
<b>Input</b>	<b>Output (Image brightening)</b>
--------------	-----------------------------------



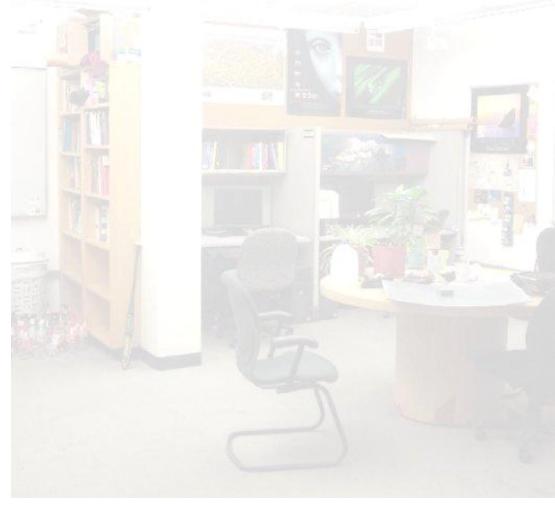
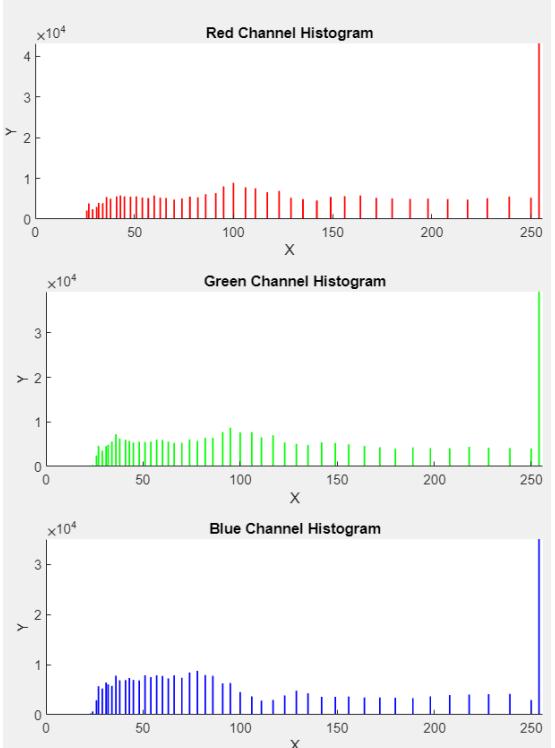
$$a = 1.1; b = 70$$



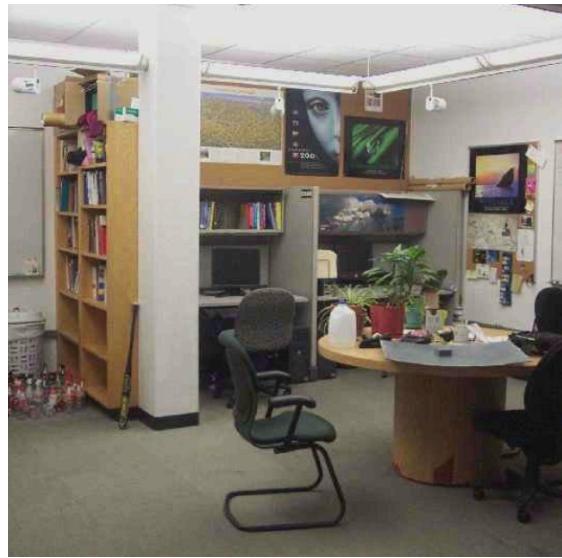


Input	Output (Image brightening)
	
<p data-bbox="300 895 850 925"><b>Input</b></p> <p data-bbox="300 937 850 967">Red Channel Histogram</p>  <p data-bbox="300 1212 850 1241">Green Channel Histogram</p>  <p data-bbox="300 1486 850 1516">Blue Channel Histogram</p> 	<p data-bbox="868 895 1434 925"><b>Output</b></p> <p data-bbox="868 937 1434 967">Red Channel Histogram</p>  <p data-bbox="868 1212 1434 1241">Green Channel Histogram</p>  <p data-bbox="868 1486 1434 1516">Blue Channel Histogram</p> 

$$a = 1.25; b = 20$$

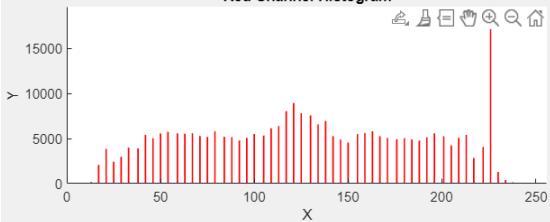
Output (Transformasi Log)	Output (Transformasi Pangkat)																																																																																				
 <p data-bbox="295 846 388 878"><math>c = 1.8</math></p>	 <p data-bbox="879 846 1046 878"><math>c = 10; \gamma = 9</math></p>																																																																																				
<p data-bbox="295 910 850 941"><b>Output</b></p>  <table border="1"> <caption>Red Channel Histogram Data (approximate values)</caption> <thead> <tr> <th>Pixel Value (X)</th> <th>Frequency (Y)</th> </tr> </thead> <tbody> <tr><td>0-50</td><td>~100</td></tr> <tr><td>50-100</td><td>~100</td></tr> <tr><td>100-150</td><td>~100</td></tr> <tr><td>150-200</td><td>~100</td></tr> <tr><td>200-250</td><td>~100</td></tr> <tr><td>250</td><td>~80,000</td></tr> </tbody> </table> <table border="1"> <caption>Green Channel Histogram Data (approximate values)</caption> <thead> <tr> <th>Pixel Value (X)</th> <th>Frequency (Y)</th> </tr> </thead> <tbody> <tr><td>0-50</td><td>~100</td></tr> <tr><td>50-100</td><td>~100</td></tr> <tr><td>100-150</td><td>~100</td></tr> <tr><td>150-200</td><td>~100</td></tr> <tr><td>200-250</td><td>~100</td></tr> <tr><td>250</td><td>~80,000</td></tr> </tbody> </table> <table border="1"> <caption>Blue Channel Histogram Data (approximate values)</caption> <thead> <tr> <th>Pixel Value (X)</th> <th>Frequency (Y)</th> </tr> </thead> <tbody> <tr><td>0-50</td><td>~100</td></tr> <tr><td>50-100</td><td>~100</td></tr> <tr><td>100-150</td><td>~100</td></tr> <tr><td>150-200</td><td>~100</td></tr> <tr><td>200-250</td><td>~100</td></tr> <tr><td>250</td><td>~80,000</td></tr> </tbody> </table>	Pixel Value (X)	Frequency (Y)	0-50	~100	50-100	~100	100-150	~100	150-200	~100	200-250	~100	250	~80,000	Pixel Value (X)	Frequency (Y)	0-50	~100	50-100	~100	100-150	~100	150-200	~100	200-250	~100	250	~80,000	Pixel Value (X)	Frequency (Y)	0-50	~100	50-100	~100	100-150	~100	150-200	~100	200-250	~100	250	~80,000	<p data-bbox="879 910 1434 941"><b>Output</b></p>  <table border="1"> <caption>Red Channel Histogram Data (approximate values)</caption> <thead> <tr> <th>Pixel Value (X)</th> <th>Frequency (Y)</th> </tr> </thead> <tbody> <tr><td>0-50</td><td>~100</td></tr> <tr><td>50-100</td><td>~100</td></tr> <tr><td>100-150</td><td>~100</td></tr> <tr><td>150-200</td><td>~100</td></tr> <tr><td>200-250</td><td>~100</td></tr> <tr><td>250</td><td>~40,000</td></tr> </tbody> </table> <table border="1"> <caption>Green Channel Histogram Data (approximate values)</caption> <thead> <tr> <th>Pixel Value (X)</th> <th>Frequency (Y)</th> </tr> </thead> <tbody> <tr><td>0-50</td><td>~100</td></tr> <tr><td>50-100</td><td>~100</td></tr> <tr><td>100-150</td><td>~100</td></tr> <tr><td>150-200</td><td>~100</td></tr> <tr><td>200-250</td><td>~100</td></tr> <tr><td>250</td><td>~40,000</td></tr> </tbody> </table> <table border="1"> <caption>Blue Channel Histogram Data (approximate values)</caption> <thead> <tr> <th>Pixel Value (X)</th> <th>Frequency (Y)</th> </tr> </thead> <tbody> <tr><td>0-50</td><td>~100</td></tr> <tr><td>50-100</td><td>~100</td></tr> <tr><td>100-150</td><td>~100</td></tr> <tr><td>150-200</td><td>~100</td></tr> <tr><td>200-250</td><td>~100</td></tr> <tr><td>250</td><td>~40,000</td></tr> </tbody> </table>	Pixel Value (X)	Frequency (Y)	0-50	~100	50-100	~100	100-150	~100	150-200	~100	200-250	~100	250	~40,000	Pixel Value (X)	Frequency (Y)	0-50	~100	50-100	~100	100-150	~100	150-200	~100	200-250	~100	250	~40,000	Pixel Value (X)	Frequency (Y)	0-50	~100	50-100	~100	100-150	~100	150-200	~100	200-250	~100	250	~40,000
Pixel Value (X)	Frequency (Y)																																																																																				
0-50	~100																																																																																				
50-100	~100																																																																																				
100-150	~100																																																																																				
150-200	~100																																																																																				
200-250	~100																																																																																				
250	~80,000																																																																																				
Pixel Value (X)	Frequency (Y)																																																																																				
0-50	~100																																																																																				
50-100	~100																																																																																				
100-150	~100																																																																																				
150-200	~100																																																																																				
200-250	~100																																																																																				
250	~80,000																																																																																				
Pixel Value (X)	Frequency (Y)																																																																																				
0-50	~100																																																																																				
50-100	~100																																																																																				
100-150	~100																																																																																				
150-200	~100																																																																																				
200-250	~100																																																																																				
250	~80,000																																																																																				
Pixel Value (X)	Frequency (Y)																																																																																				
0-50	~100																																																																																				
50-100	~100																																																																																				
100-150	~100																																																																																				
150-200	~100																																																																																				
200-250	~100																																																																																				
250	~40,000																																																																																				
Pixel Value (X)	Frequency (Y)																																																																																				
0-50	~100																																																																																				
50-100	~100																																																																																				
100-150	~100																																																																																				
150-200	~100																																																																																				
200-250	~100																																																																																				
250	~40,000																																																																																				
Pixel Value (X)	Frequency (Y)																																																																																				
0-50	~100																																																																																				
50-100	~100																																																																																				
100-150	~100																																																																																				
150-200	~100																																																																																				
200-250	~100																																																																																				
250	~40,000																																																																																				

## Output (Peregangan Kontras)

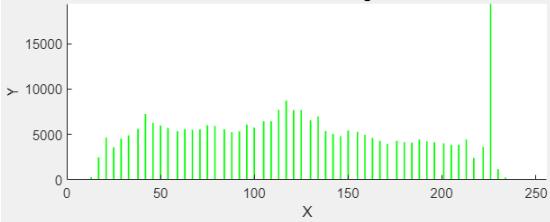


Output

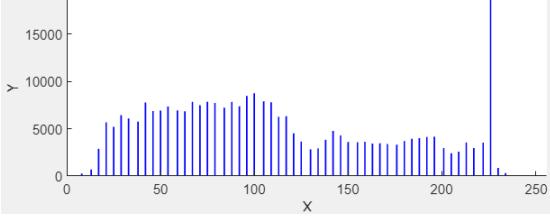
Red Channel Histogram

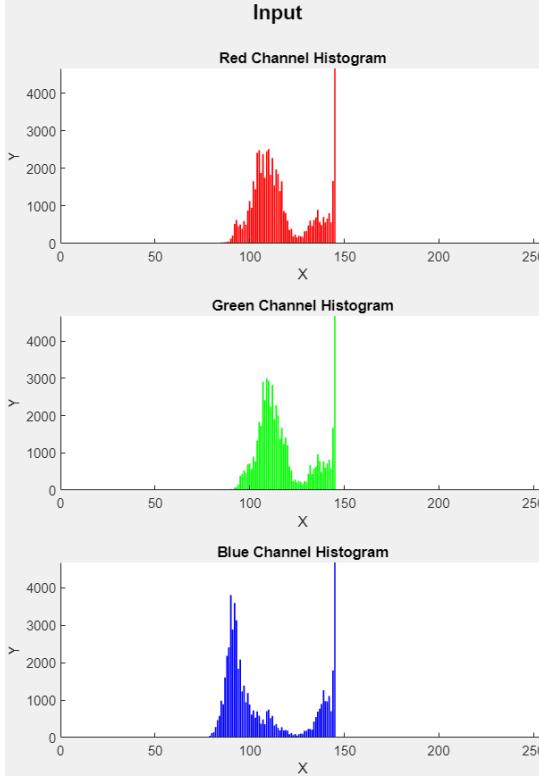
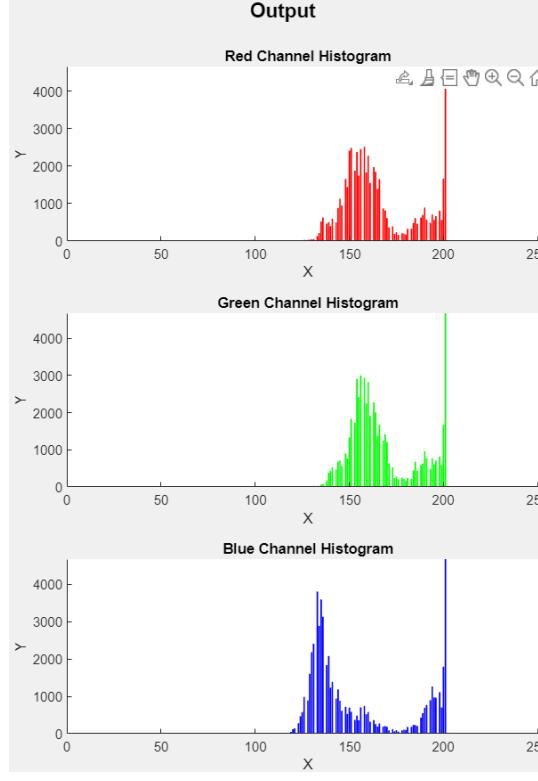


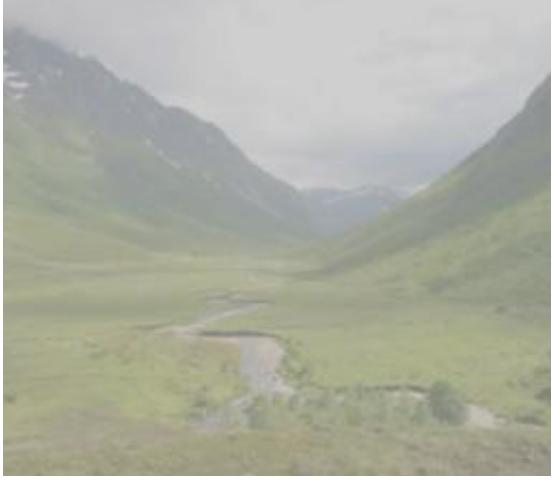
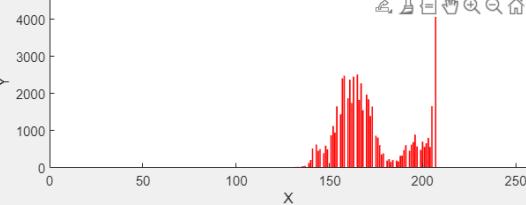
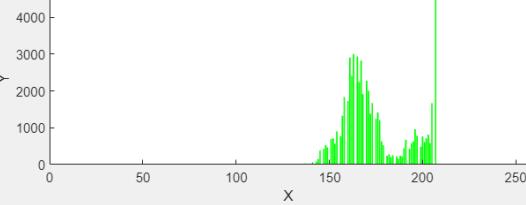
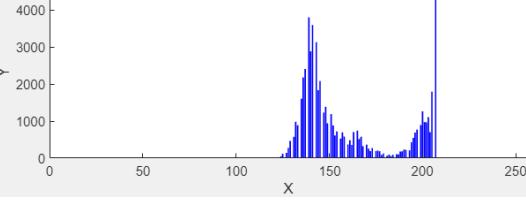
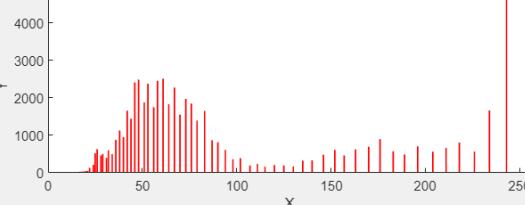
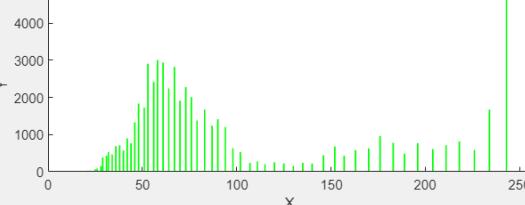
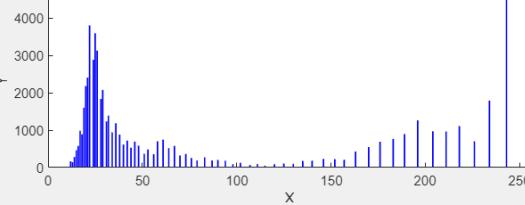
Green Channel Histogram



Blue Channel Histogram



<b>Input</b>	<b>Output (Image brightening)</b>
	 $a = 1.25; b = 20$
<b>Input</b> 	<b>Output</b> 

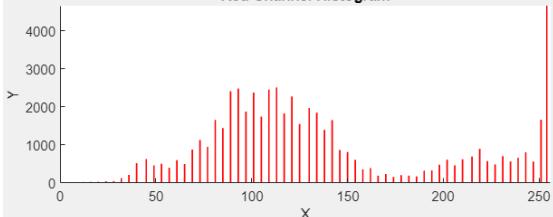
Output (Transformasi Log)	Output (Transformasi Pangkat)
 <p data-bbox="303 777 388 806"><math>c = 1.8</math></p>	 <p data-bbox="871 777 1029 806"><math>c = 4; \gamma = 4</math></p>
<p data-bbox="545 840 616 868"><b>Output</b></p> <p data-bbox="518 889 687 910"><b>Red Channel Histogram</b></p>  <p data-bbox="567 1123 698 1144"><b>Green Channel Histogram</b></p>  <p data-bbox="518 1374 687 1396"><b>Blue Channel Histogram</b></p> 	<p data-bbox="1116 840 1191 868"><b>Output</b></p> <p data-bbox="1090 889 1274 910"><b>Red Channel Histogram</b></p>  <p data-bbox="1090 1123 1286 1144"><b>Green Channel Histogram</b></p>  <p data-bbox="1090 1374 1274 1396"><b>Blue Channel Histogram</b></p> 

## Output (Peregangan Kontras)

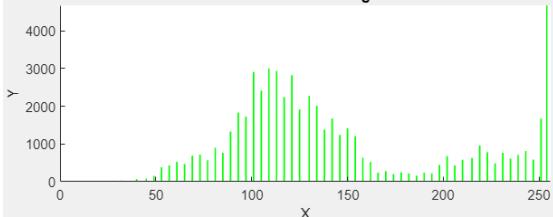


Output

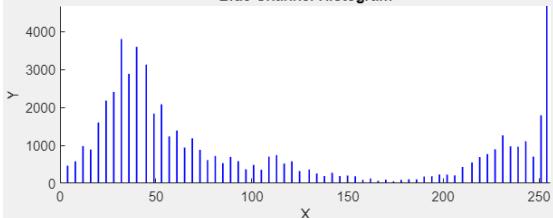
Red Channel Histogram

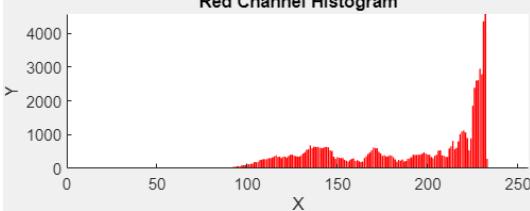
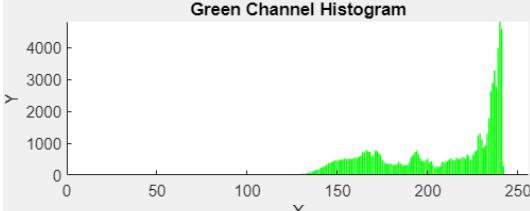
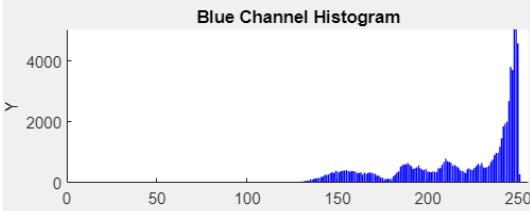
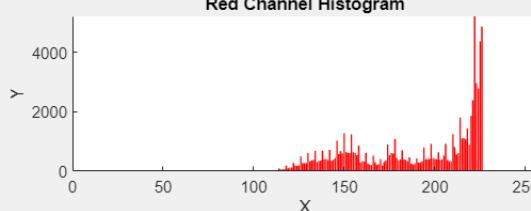
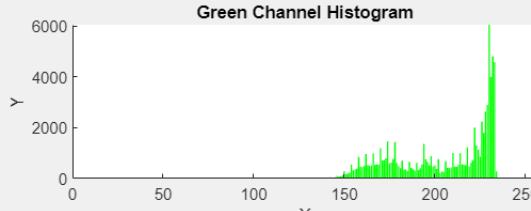
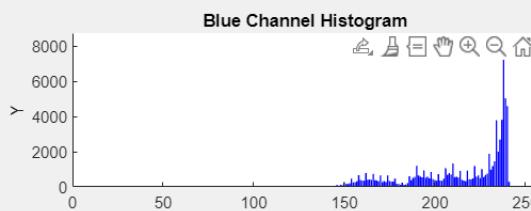


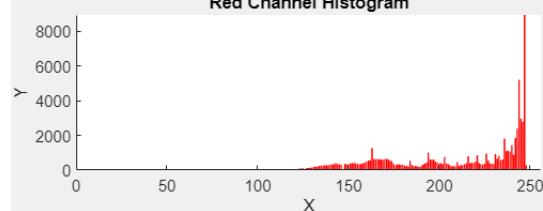
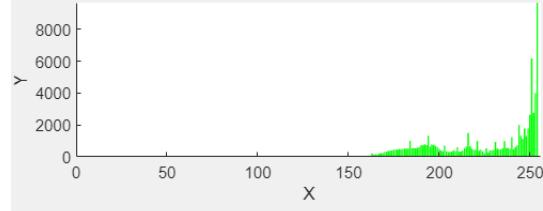
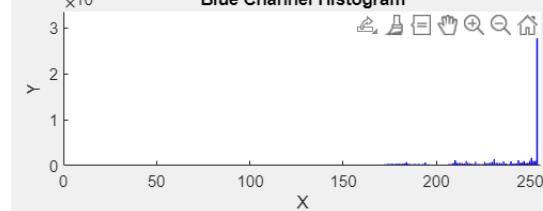
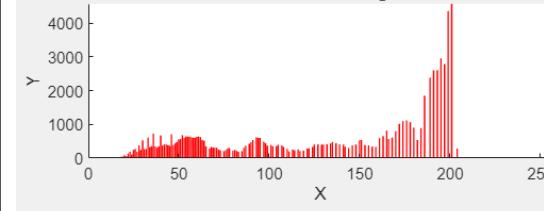
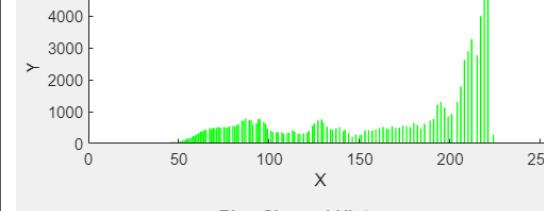
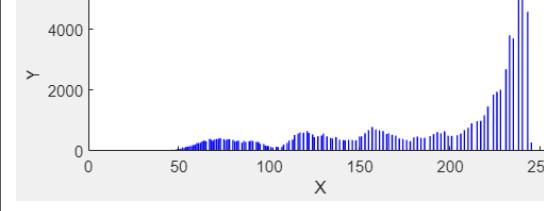
Green Channel Histogram



Blue Channel Histogram



<b>Input</b>	<b>Output (Image brightening)</b>
	 $a = 0.8; b = 40$
<b>Input</b>   	<b>Output</b>   

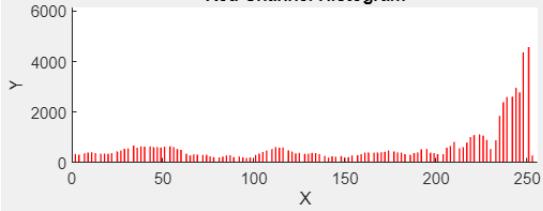
Output (Transformasi Log)	Output (Transformasi Pangkat)
	
$c = 1.5$	$c = 1; \gamma = 2.5$
<p style="text-align: center;"><b>Output</b></p> <p><b>Red Channel Histogram</b></p>  <p><b>Green Channel Histogram</b></p>  <p><b>Blue Channel Histogram</b></p> 	<p style="text-align: center;"><b>Output</b></p> <p><b>Red Channel Histogram</b></p>  <p><b>Green Channel Histogram</b></p>  <p><b>Blue Channel Histogram</b></p> 

### Output (Peregangan Kontras)

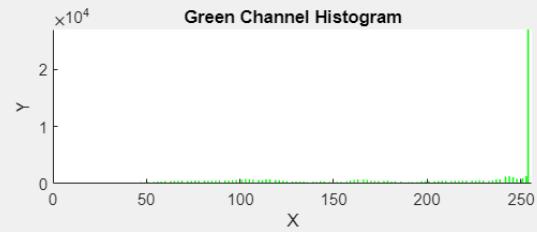


Output

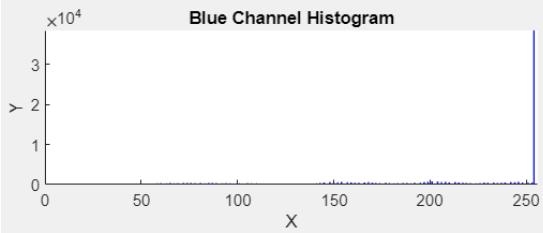
Red Channel Histogram



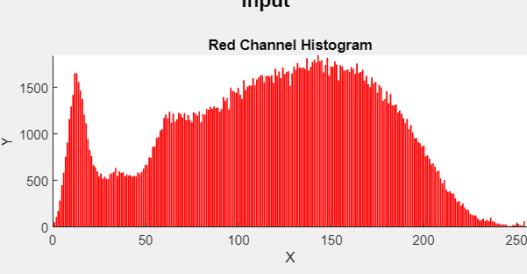
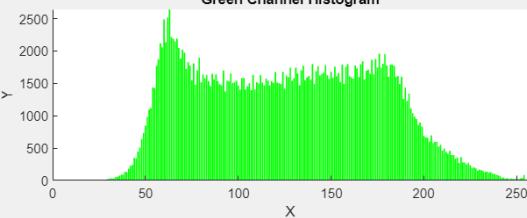
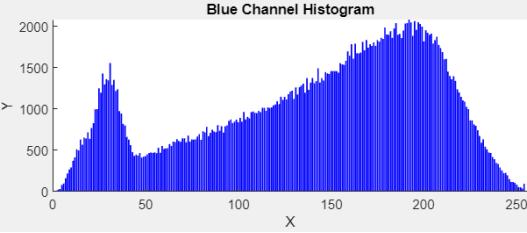
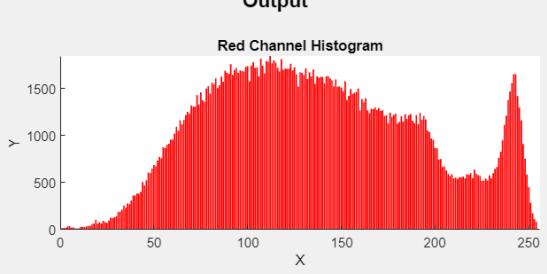
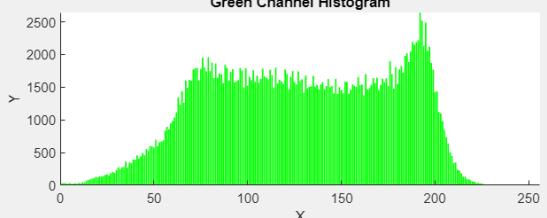
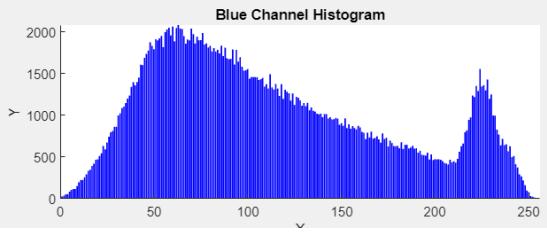
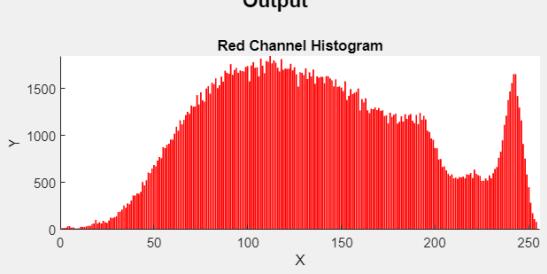
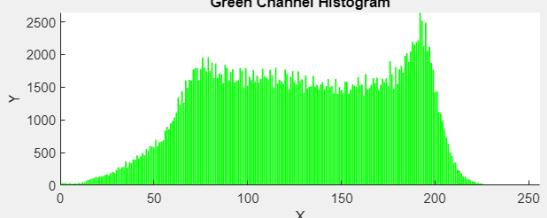
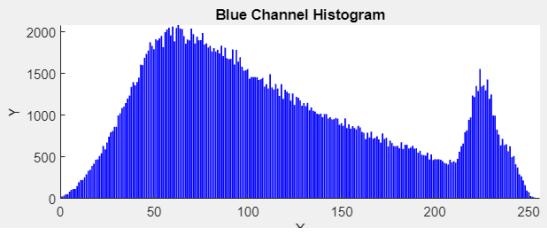
Green Channel Histogram



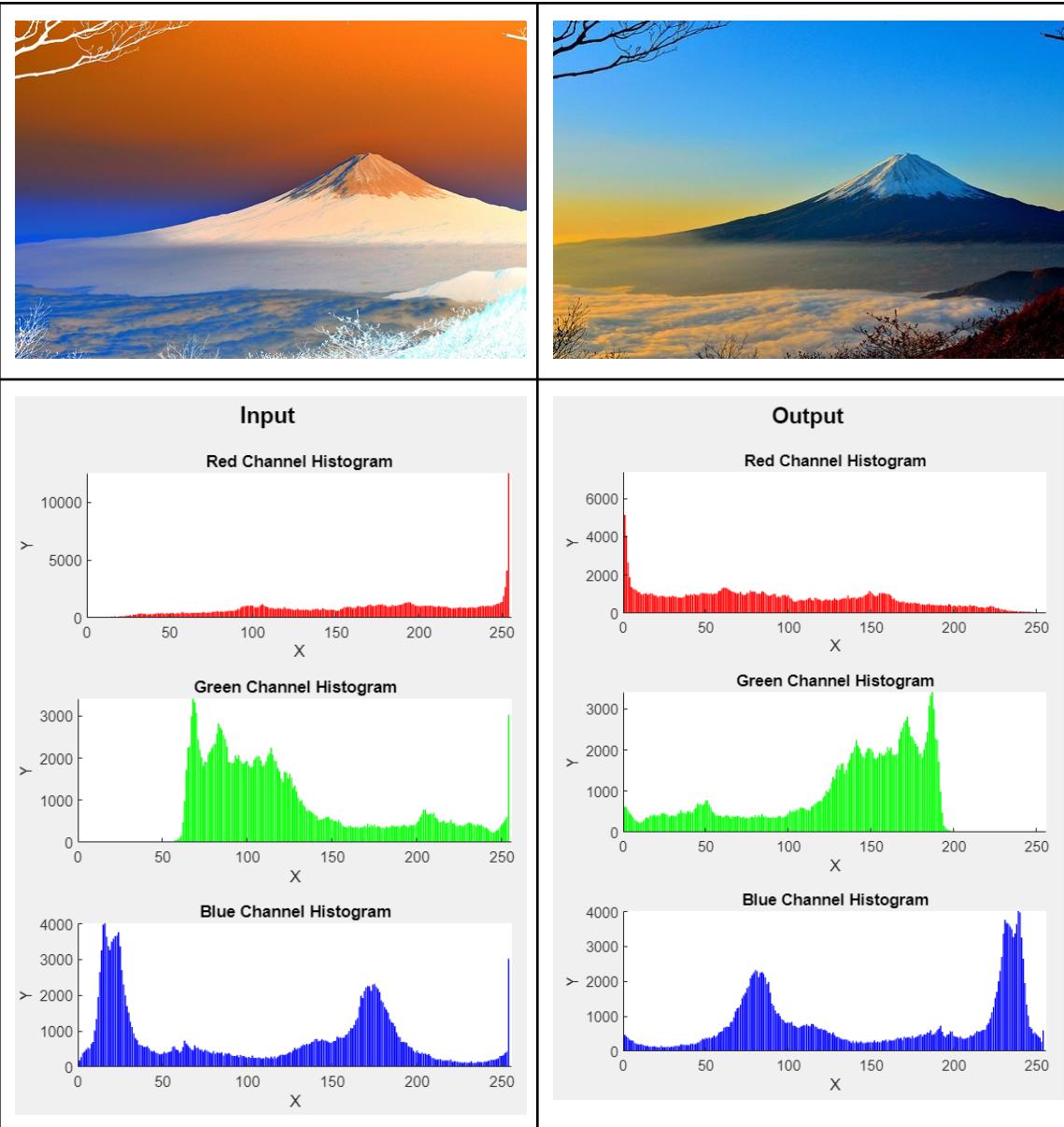
Blue Channel Histogram



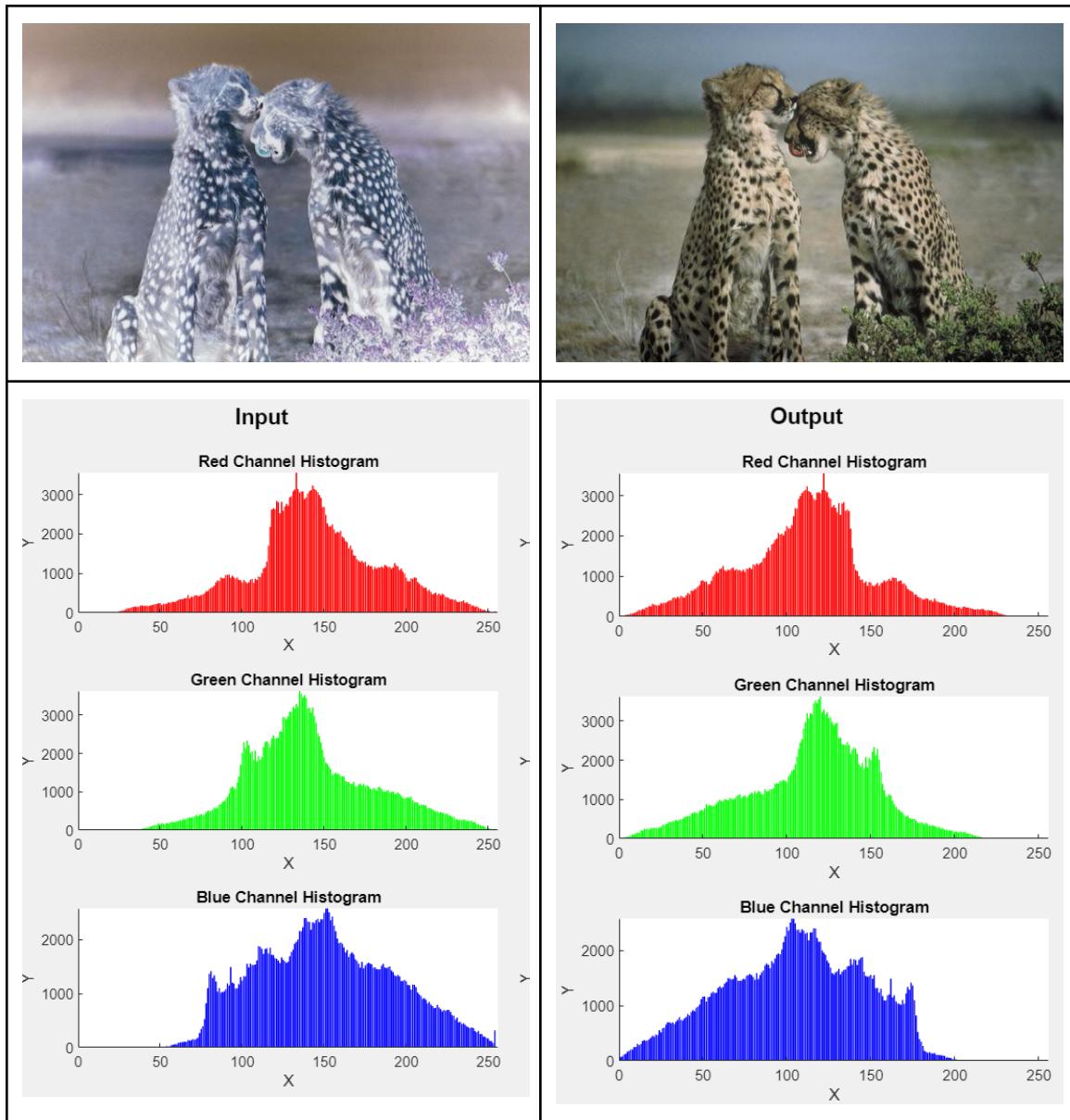
### 3.2.3. Testing Negative Image

Input	Output
	
<b>Input</b>     <b>Output</b>   	 <b>Output</b>   

Input	Output
-------	--------



<b>Input</b>	<b>Output</b>
--------------	---------------



### 3.2.4. Analisis

Seluruh transformasi, yaitu *image brightening*, *negative*, *log transformation*, *power transformation*, dan *image stretching*, dapat berfungsi dengan baik. Seluruh transformasi dapat bekerja pada gambar grayscale maupun RGB.

### 3.3. Histogram Equalization

Program ini melakukan perataan histogram untuk citra yang terlalu gelap, terlalu terang, dan citra kontras-rendah, baik untuk citra grayscale maupun untuk citra berwarna. Program tersebut dapat menampilkan citra masukan, histogram citra masukan, citra luaran, dan histogram citra luaran

#### 3.3.1. Kode Program

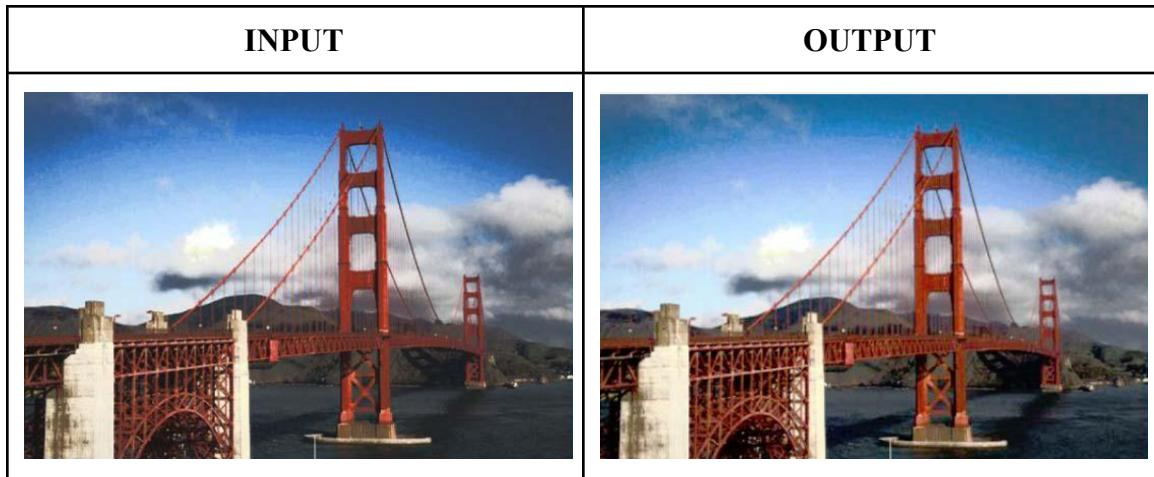
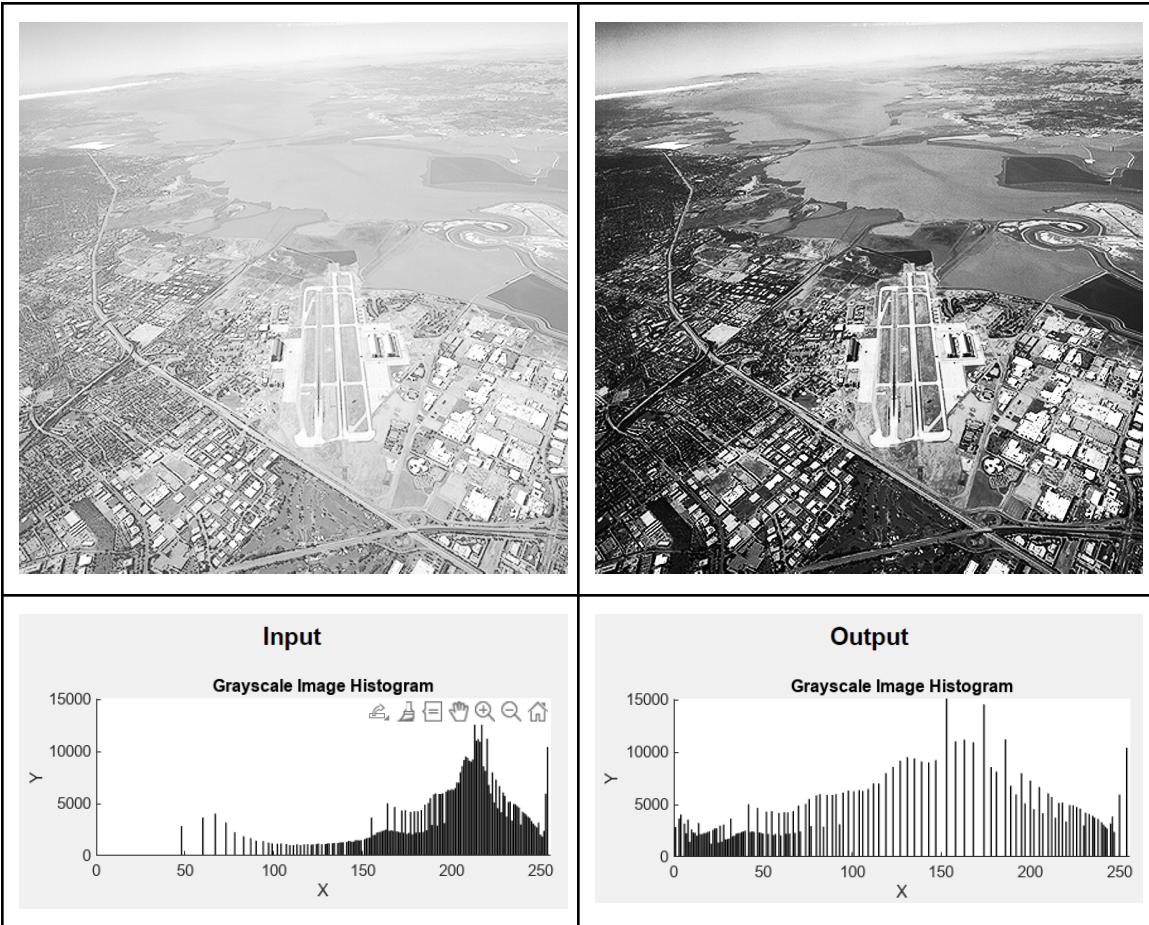
Berikut merupakan potongan kode program untuk melakukan perataan histogram (*histogram equalization*)

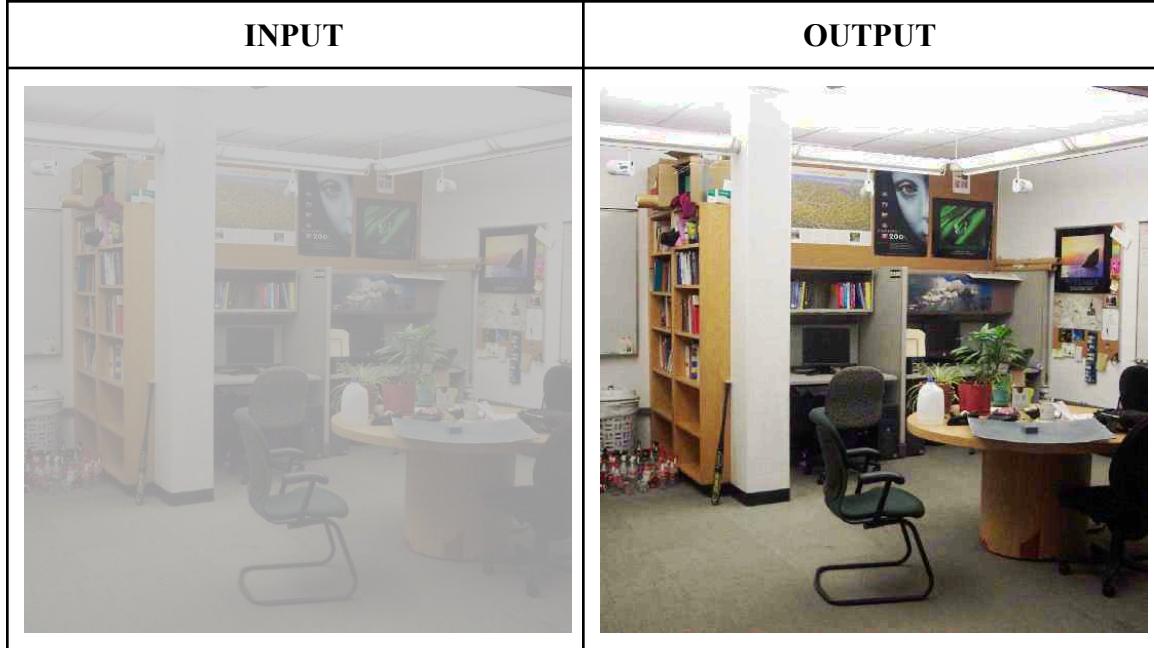
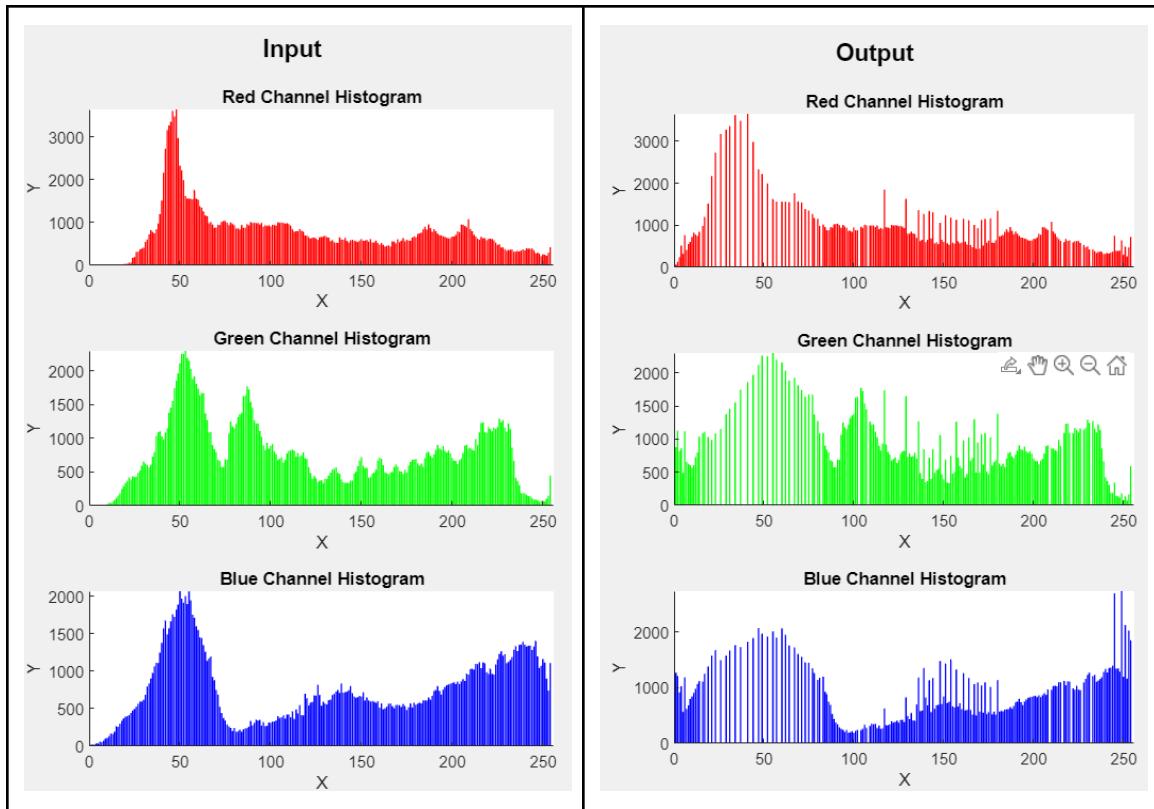
```
function result = histogramEqualization(imagedata)
    [N, M] = size(imagedata);
    % Hitung histogram citra
    HistInput = myhist(imagedata, true);
    HistOutput = zeros(256, 1);
    result = imagedata;
    % Hitung histogram citra
    sum = 0.0;
    for i = 1:256
        sum = sum + HistInput(i);
        HistOutput(i) = floor(255 * sum);
    end
    % Update citra sesuai histogram hasil perataan
    for i = 1:N
        for j = 1:M
            result(i,j) = HistOutput(imagedata(i, j) + 1);
        end
    end
end
```

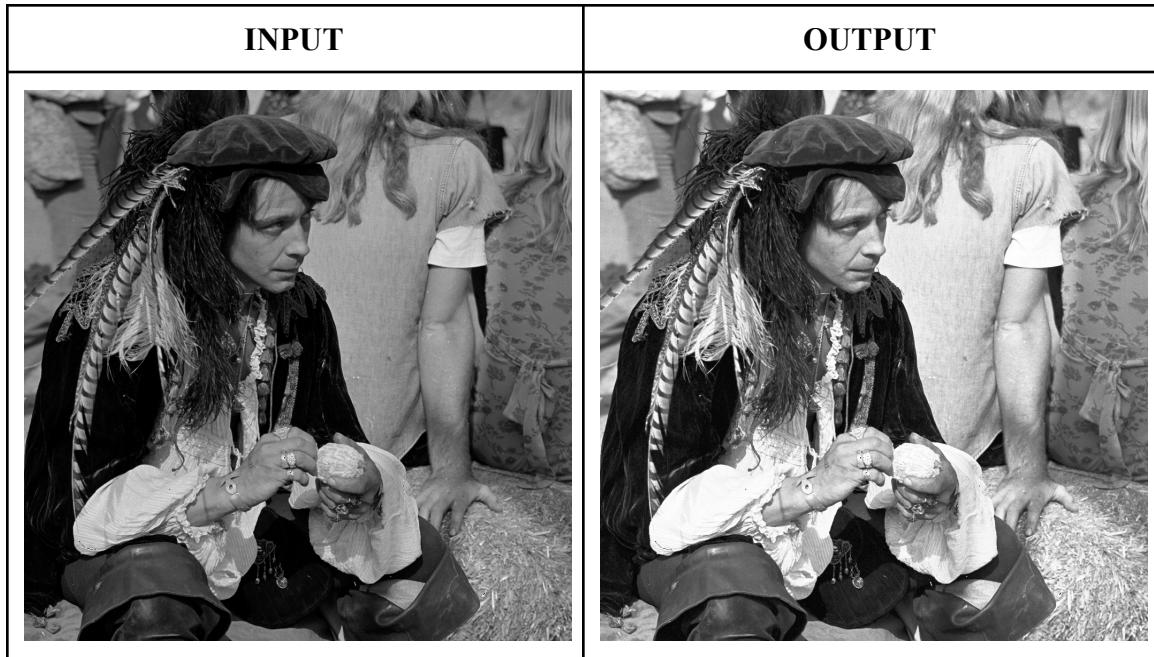
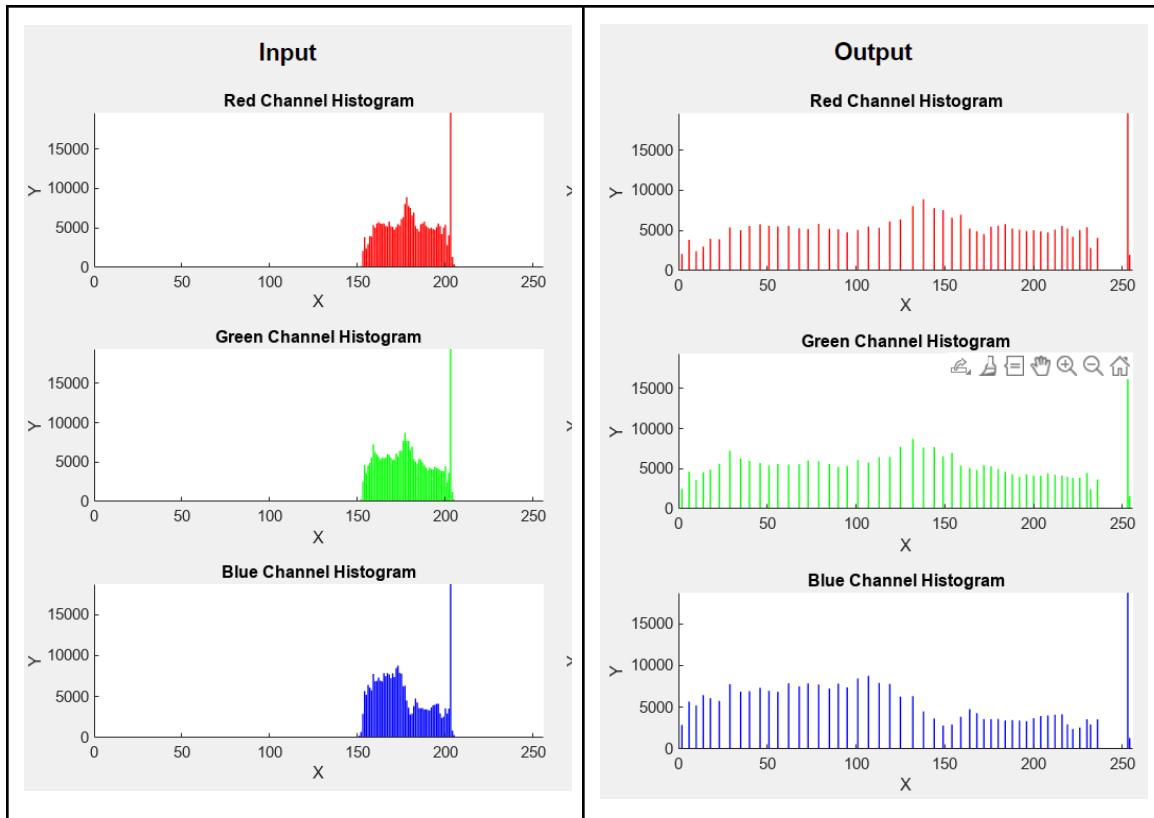
Pertama akan dihitung nilai histogram dengan normalisasi yang terdapat pada image yang diinputkan oleh pengguna. Lalu dilakukan inisialisasi variabel “HistOutput” untuk menampung histogram keluaran nantinya. Selanjutnya, dilakukan perhitungan kumulatif dari histogram input. Loop dilakukan dari nilai intensitas 0 (disini dikodekan dalam 1 karena indeks pada matlab dimulai dari 1) hingga 256 (255) untuk menghitung akumulasi histogram input dalam variabel “sum”. Nilai sum akan dilakukan floor untuk menghasilkan bilangan bulat untuk perkalian “sum” dengan 255 (intensitas tertinggi). Setelah itu, matriks input citra di update dengan nilai histogram yang baru.

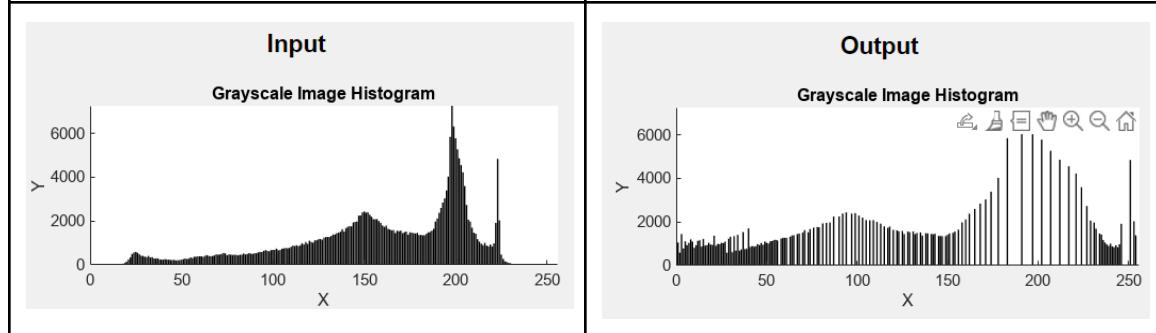
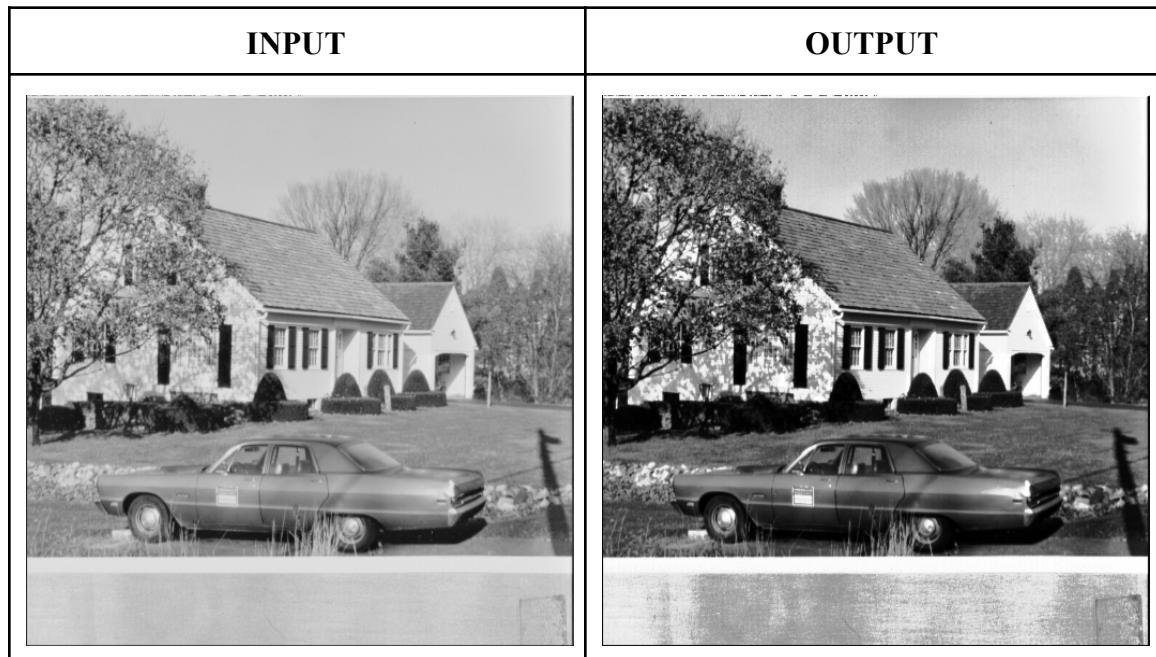
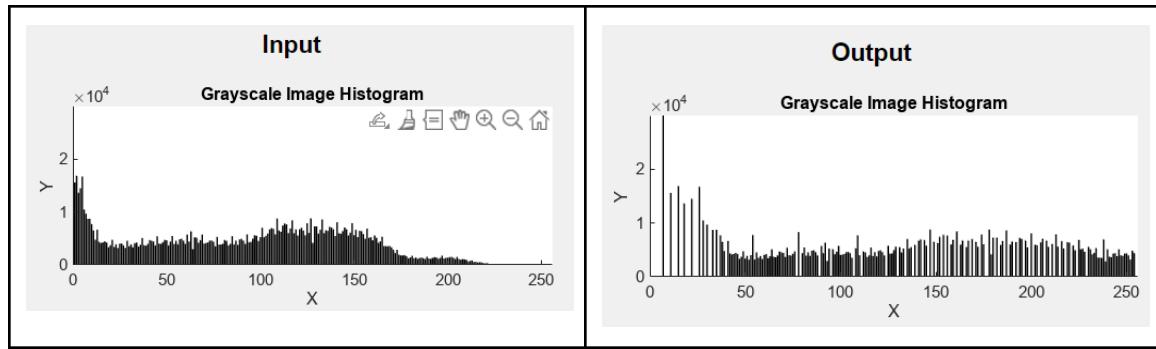
### 3.3.2. Testing

INPUT	OUTPUT																																												
<p><b>Input</b></p> <p>Grayscale Image Histogram</p> <table border="1"><caption>Estimated Input Histogram Data</caption><thead><tr><th>Bin Range (X)</th><th>Frequency (Y)</th></tr></thead><tbody><tr><td>0-25</td><td>~15000</td></tr><tr><td>25-50</td><td>~10000</td></tr><tr><td>50-75</td><td>~5000</td></tr><tr><td>75-100</td><td>~1000</td></tr><tr><td>100-125</td><td>~500</td></tr><tr><td>125-150</td><td>~200</td></tr><tr><td>150-175</td><td>~100</td></tr><tr><td>175-200</td><td>~50</td></tr><tr><td>200-225</td><td>~20</td></tr><tr><td>225-250</td><td>~10</td></tr></tbody></table>	Bin Range (X)	Frequency (Y)	0-25	~15000	25-50	~10000	50-75	~5000	75-100	~1000	100-125	~500	125-150	~200	150-175	~100	175-200	~50	200-225	~20	225-250	~10	<p><b>Output</b></p> <p>Grayscale Image Histogram</p> <table border="1"><caption>Estimated Output Histogram Data</caption><thead><tr><th>Bin Range (X)</th><th>Frequency (Y)</th></tr></thead><tbody><tr><td>0-25</td><td>~15000</td></tr><tr><td>25-50</td><td>~10000</td></tr><tr><td>50-75</td><td>~8000</td></tr><tr><td>75-100</td><td>~6000</td></tr><tr><td>100-125</td><td>~4000</td></tr><tr><td>125-150</td><td>~3000</td></tr><tr><td>150-175</td><td>~2500</td></tr><tr><td>175-200</td><td>~2000</td></tr><tr><td>200-225</td><td>~1500</td></tr><tr><td>225-250</td><td>~1000</td></tr></tbody></table>	Bin Range (X)	Frequency (Y)	0-25	~15000	25-50	~10000	50-75	~8000	75-100	~6000	100-125	~4000	125-150	~3000	150-175	~2500	175-200	~2000	200-225	~1500	225-250	~1000
Bin Range (X)	Frequency (Y)																																												
0-25	~15000																																												
25-50	~10000																																												
50-75	~5000																																												
75-100	~1000																																												
100-125	~500																																												
125-150	~200																																												
150-175	~100																																												
175-200	~50																																												
200-225	~20																																												
225-250	~10																																												
Bin Range (X)	Frequency (Y)																																												
0-25	~15000																																												
25-50	~10000																																												
50-75	~8000																																												
75-100	~6000																																												
100-125	~4000																																												
125-150	~3000																																												
150-175	~2500																																												
175-200	~2000																																												
200-225	~1500																																												
225-250	~1000																																												
INPUT	OUTPUT																																												

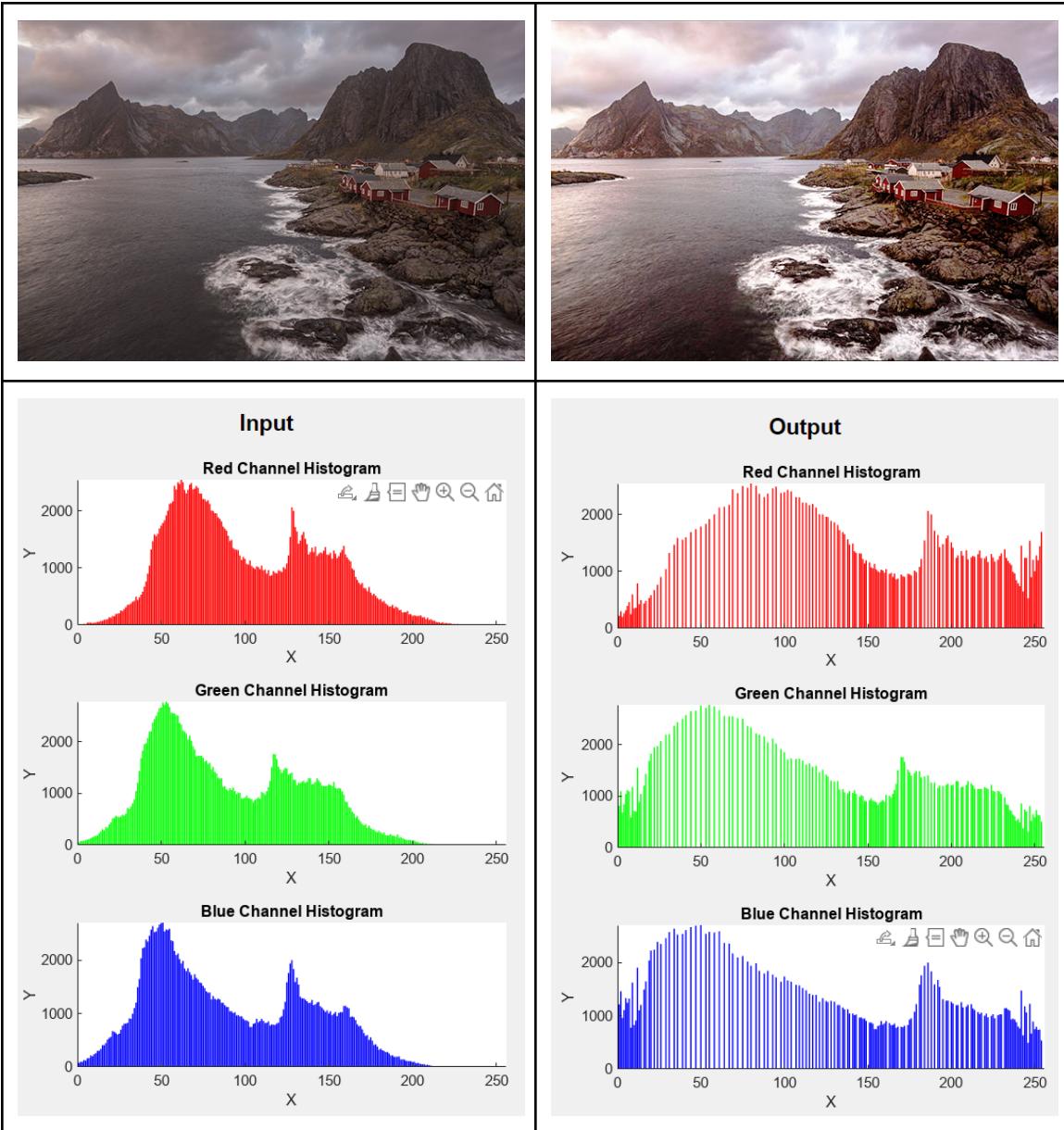




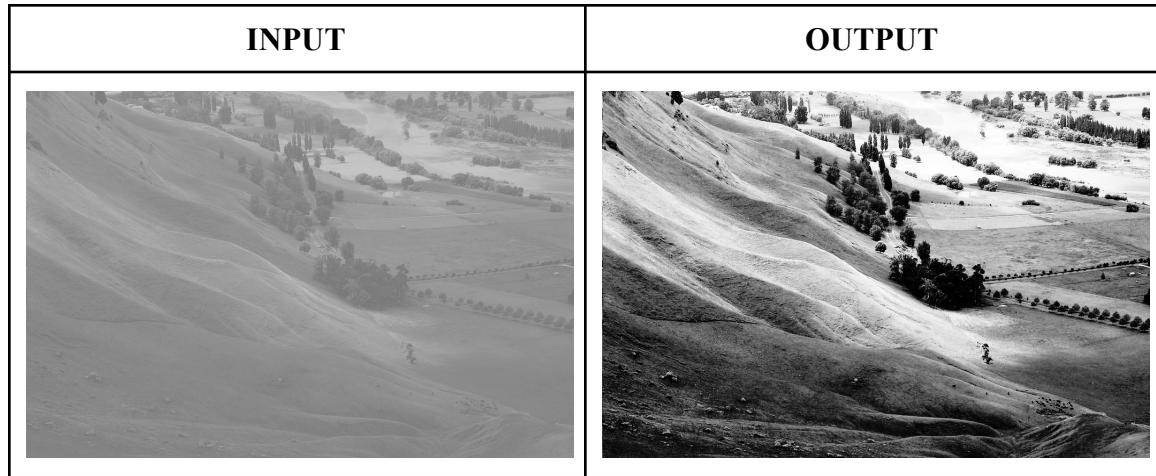
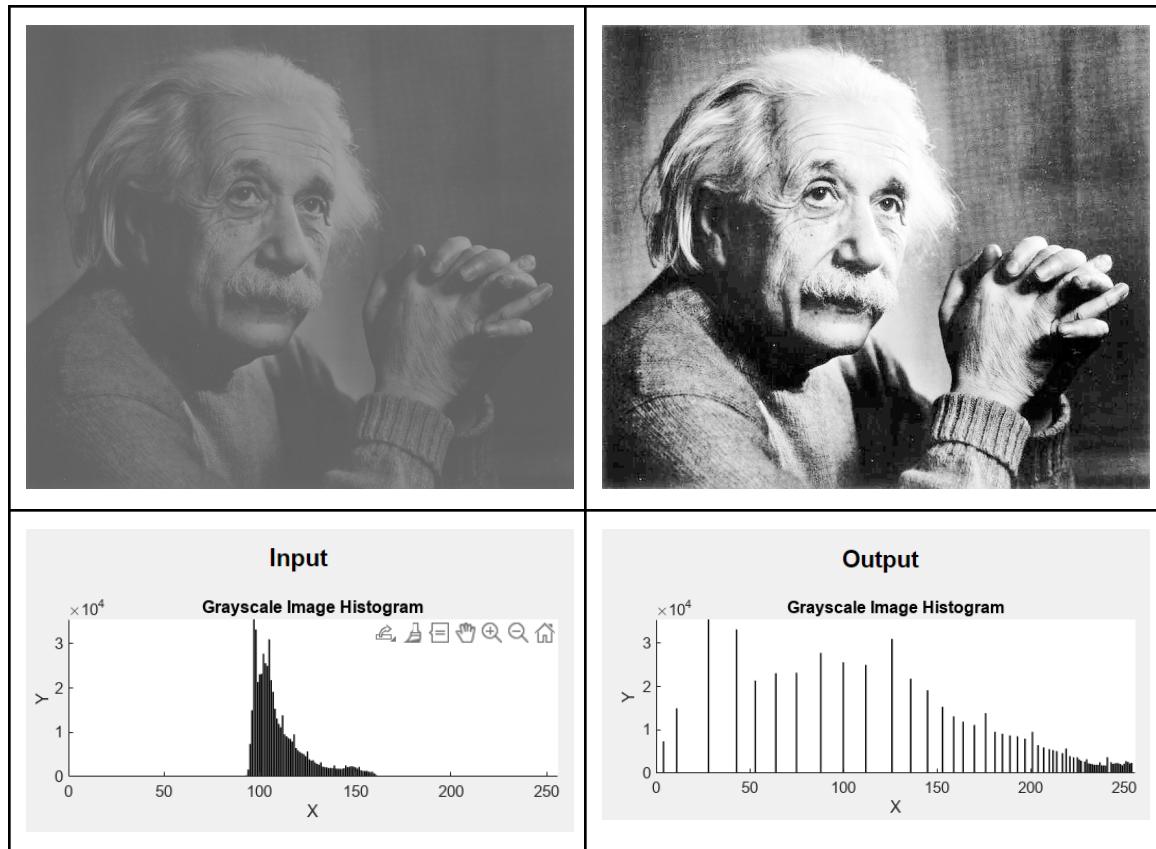


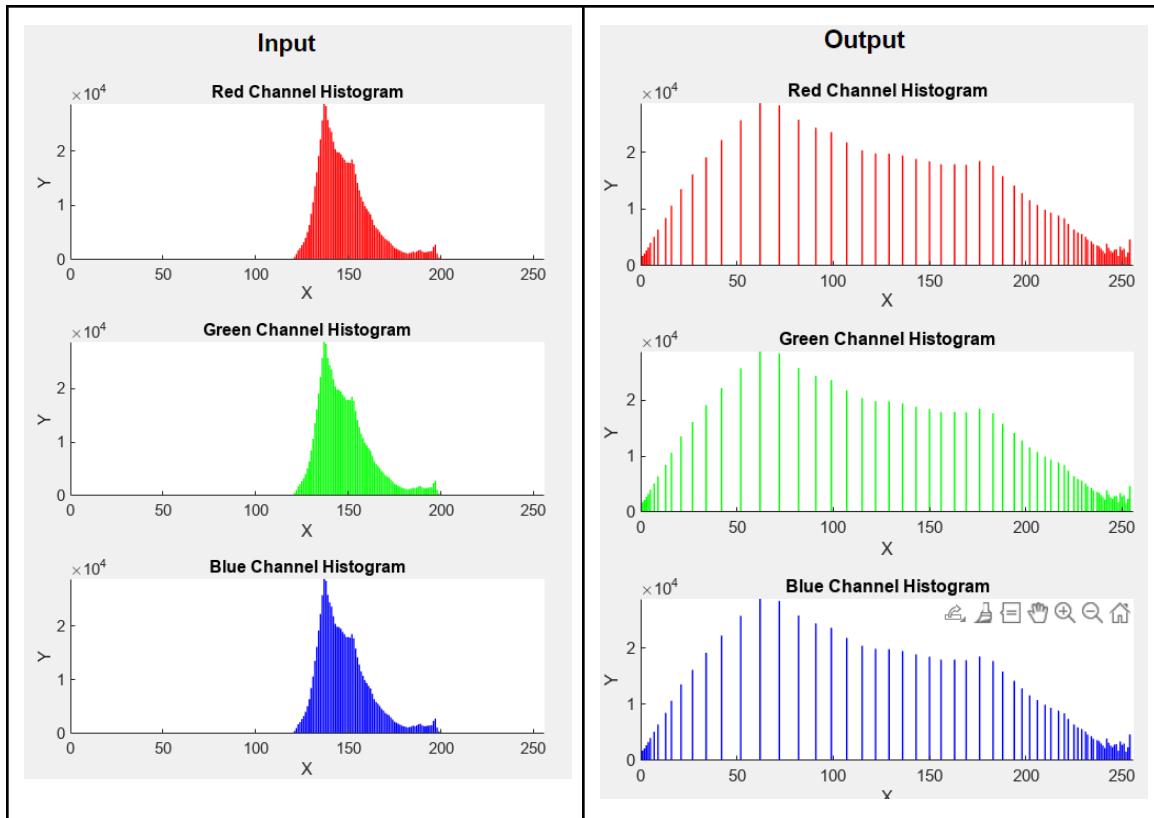


INPUT	OUTPUT
-------	--------



INPUT	OUTPUT
-------	--------





### 3.3.3. Analisis

Berdasarkan testing yang telah dilakukan. program mampu menampilkan output gambar yang telah dilakukan perataan histogram beserta histogram hasil perataan dari masing-masing saluran dengan baik.

## 3.4. Histogram Specification

Pada program ini akan dilakukan *histogram specification*, yaitu mencocokkan histogram citra masukan agar menjadi mirip dengan histogram citra referensi.

### 3.4.1. Kode Program

Berikut merupakan potongan kode program untuk melakukan *histogram specification*.

```
function result = histogramSpecification(InputImagedata, SpecTargetImagedata)
    inputHist = myhist(InputImagedata, true);
    targetHist = myhist(SpecTargetImagedata, true);

    % Calculate input equalization
    inputEq = inputHist;
    sum = 0.0;
```

```

for i = 1:256
    sum = sum + inputHist(i);
    inputEq(i) = floor(255 * sum);
end

% Calculate target equalization
targetEq = targetHist;
sum = 0.0;
for i = 1:256
    sum = sum + targetHist(i);
    targetEq(i) = floor(255 * sum);
end

% Do inverse transformation
invMapping = zeros(256);
for i = 1:256
    minval = abs(inputEq(i) - targetEq(1));
    minj = 0;
    for j = 1:256
        if (abs(inputEq(i) - targetEq(j)) < minval)
            minval = abs(inputEq(i) - targetEq(j));
            minj = j;
        end
    end
    invMapping(i) = minj;
end

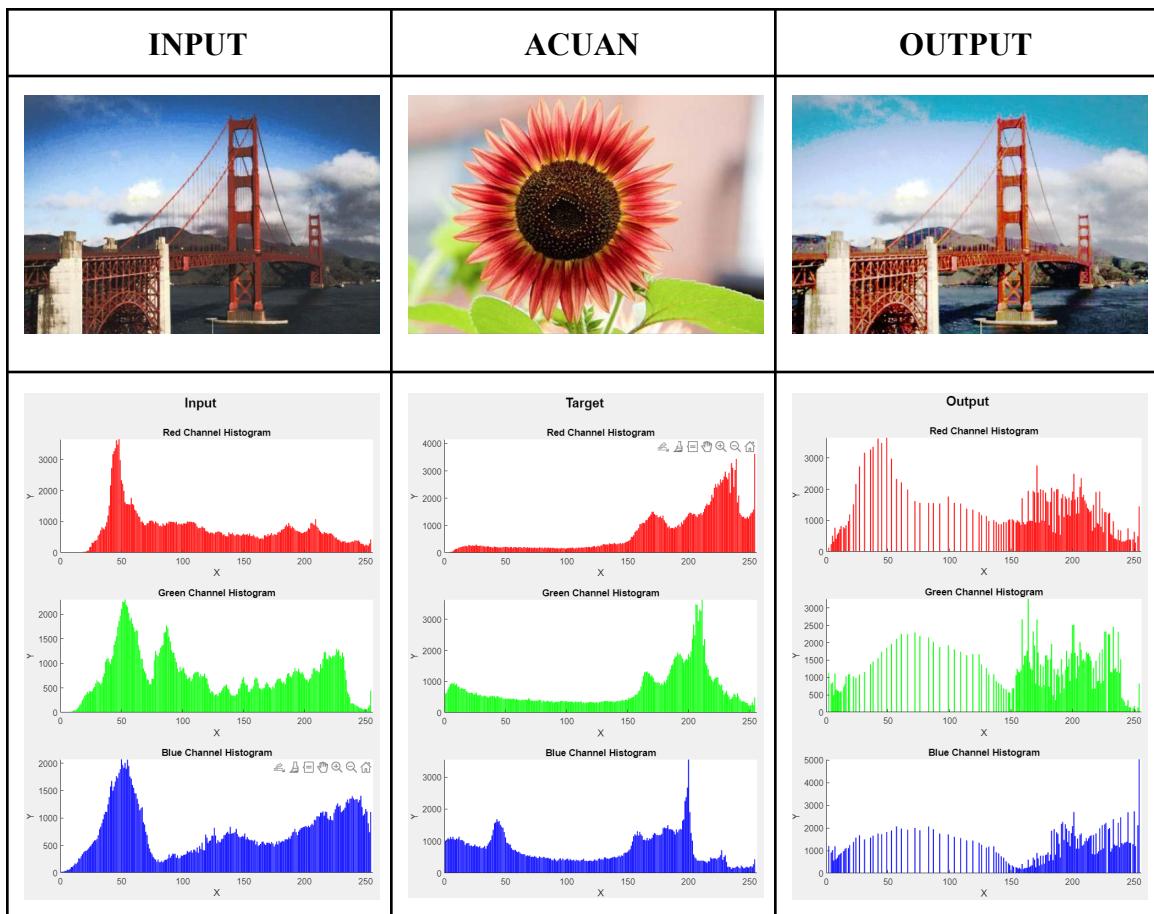
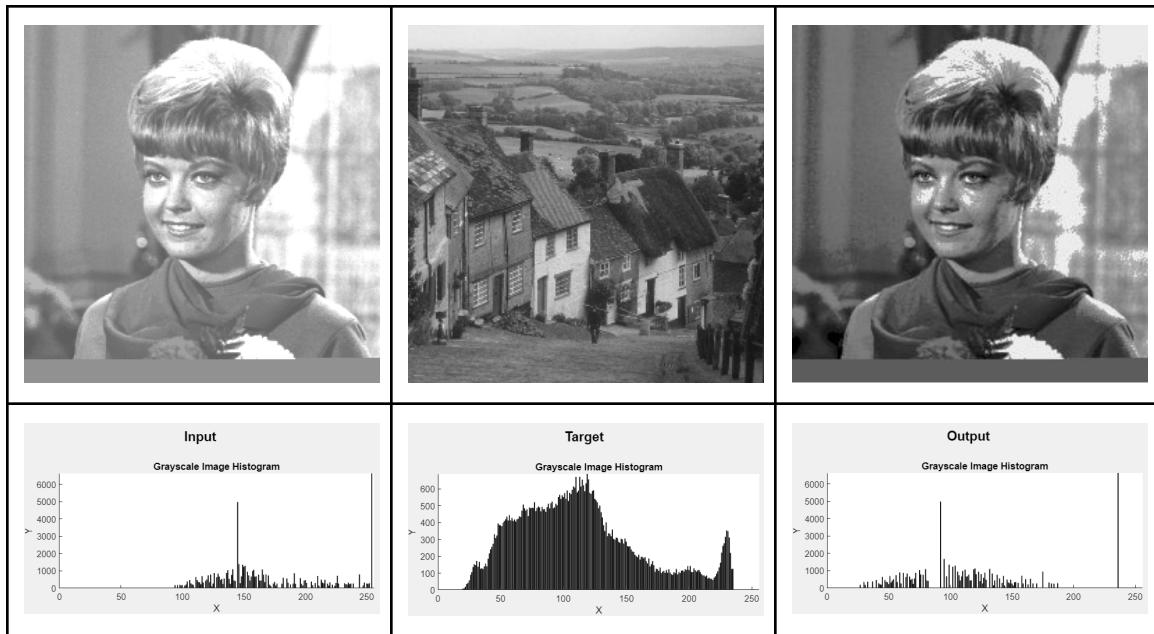
% Apply the inverse to input image, resulting output image
result = Inputimagedata;
[M, N] = size(Inputimagedata);
for i = 1:M
    for j = 1:N
        result(i,j) = invMapping(Inputimagedata(i,j) + 1);
    end
end
end

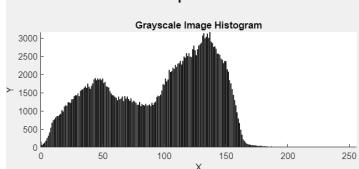
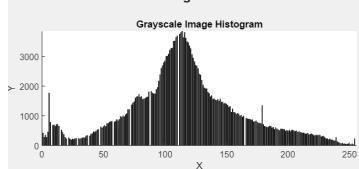
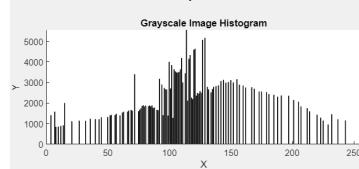
```

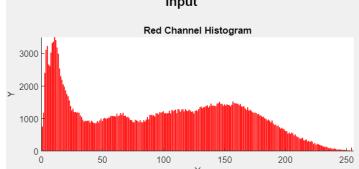
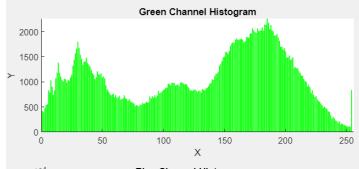
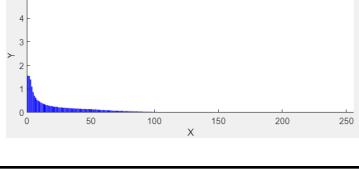
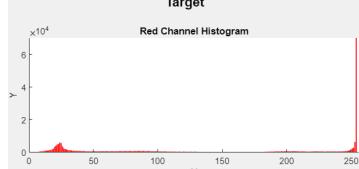
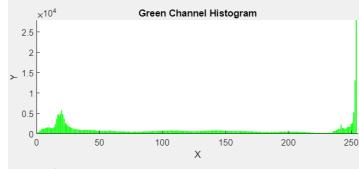
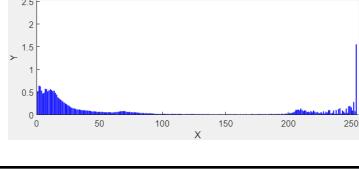
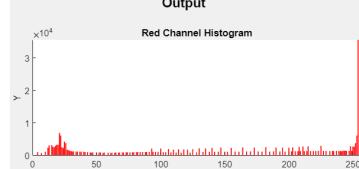
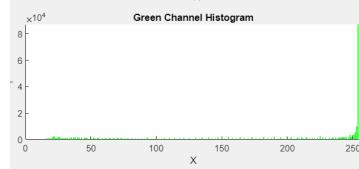
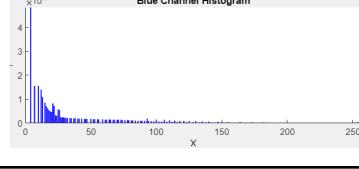
### 3.4.2. Testing

Berikut testing program Histogram Citra menggunakan beberapa image berikut ini.

INPUT	ACUAN	OUTPUT
-------	-------	--------



INPUT	ACUAN	OUTPUT
		
<p>Input</p> <p>Grayscale Image Histogram</p> 	<p>Target</p> <p>Grayscale Image Histogram</p> 	<p>Output</p> <p>Grayscale Image Histogram</p> 

INPUT	ACUAN	OUTPUT
		
<p>Input</p> <p>Red Channel Histogram</p>  <p>Green Channel Histogram</p>  <p>Blue Channel Histogram</p> 	<p>Target</p> <p>Red Channel Histogram</p>  <p>Green Channel Histogram</p>  <p>Blue Channel Histogram</p> 	<p>Output</p> <p>Red Channel Histogram</p>  <p>Green Channel Histogram</p>  <p>Blue Channel Histogram</p> 

### 3.4.3. Analisis

Fungsi `histogramSpecification` menerima masukan berupa matriks data citra masukan dan matriks data citra acuan. Fungsi mengembalikan matriks data citra keluaran yang merupakan hasil *histogram specification*. Langkah-langkah fungsi dalam melakukan *histogram specification* adalah:

1. membuat histogram citra masukan,
2. membuat histogram citra acuan,
3. melakukan perataan histogram citra masukan,
4. melakukan perataan histogram citra acuan,
5. membuat pemetaan intensitas piksel masukan ke intensitas piksel keluaran, dan
6. mengaplikasikan pemetaan ke citra masukan menghasilkan citra keluaran.

Pemetaan dilakukan dengan cara untuk tiap intensitas piksel masukan, dicari nilai intensitas yang berkorespondensi pada histogram seragamnya. Lalu, dicari nilai pada histogram seragam citra acuan yang paling mendekati nilai tersebut. Akhirnya, didapatkan nilai intensitas piksel yang bersesuaian dengan citra acuan.

Berdasarkan testing yang telah dilakukan dengan menggunakan 2 gambar *grayscale* dan 2 gambar RGB, program mampu membuat histogram citra masukan agar mirip dengan histogram citra acuan.

## **Alamat GitHub Program**

<https://github.com/lizardyy/Tugas-1-Citra>

## **Referensi**

PPT      IF4073      Interpretasi      dan      Pengolahan      Citra      2023/2024  
[\(https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2023-2024/citra23-24.htm \)](https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2023-2024/citra23-24.htm).