

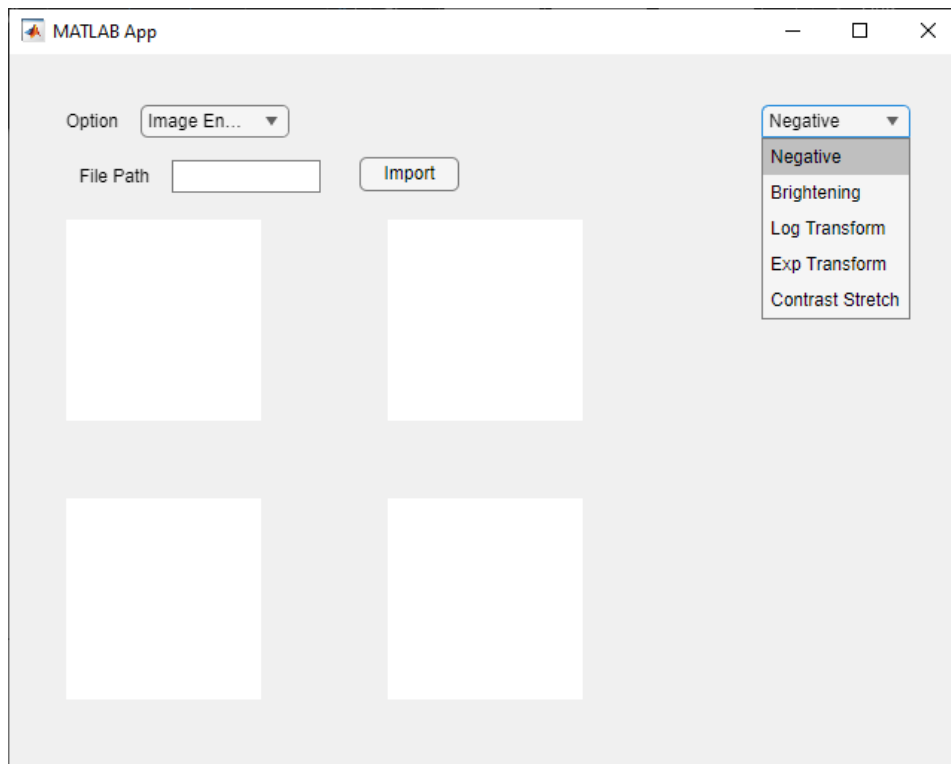
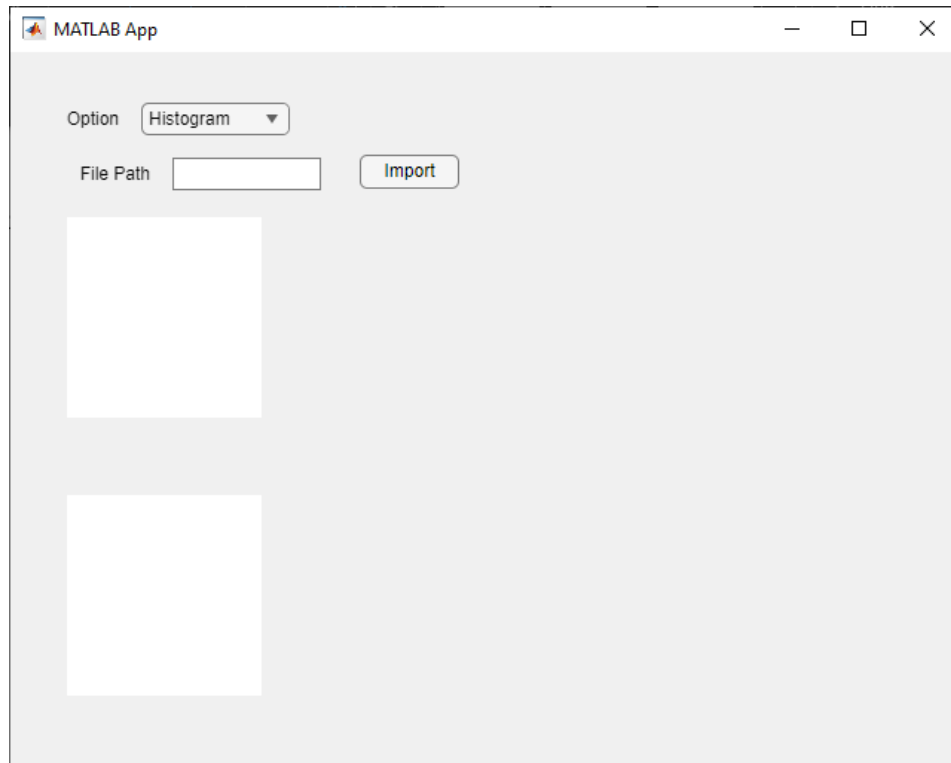
TUGAS 1

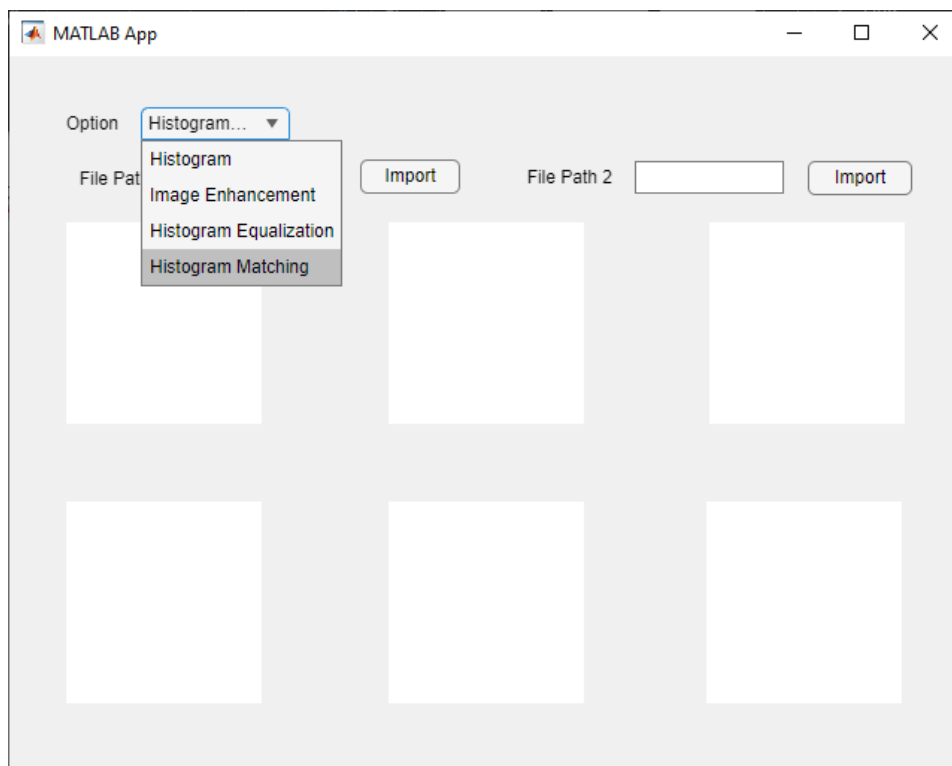
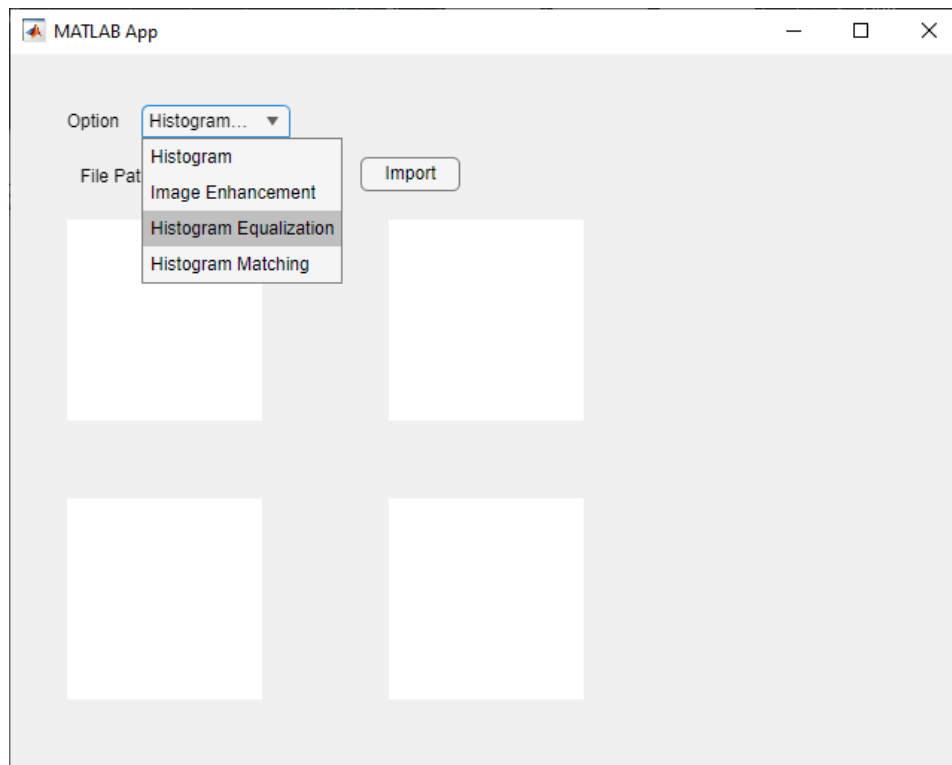
IF4073 PEMROSESAN CITRA DIGITAL



Disusun Oleh:
13520052 – Gregorius Moses Marevson

Screenshot GUI



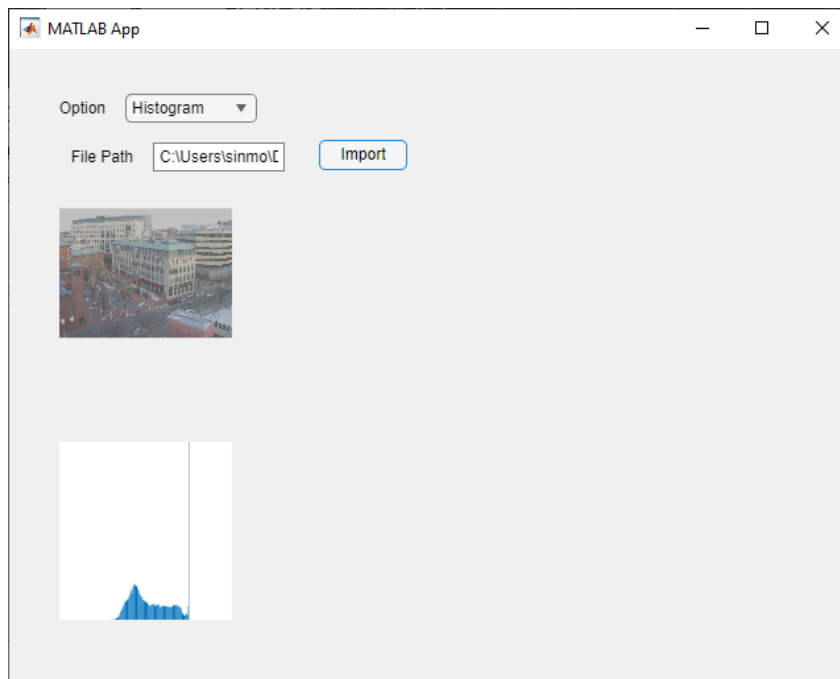


Kode Program

1. Histogram

```
1. function [histArr] = histo(img)
2.     graylevel = 256;
3.     histArr = zeros(graylevel, 1);
4.
5.     % iterate each pixel of image and count each occurrence
6.     for row = 1:size(img, 1)
7.         for col = 1:size(img, 2)
8.             for rgb = 1:size(img, 3) % size(img, 3) returns 3 if rgb or 1 if not
9.                 histArr(img(row, col, rgb) + 1) = histArr(img(row, col, rgb) + 1)
10.            + 1; % rgb indexing 1 won't produce error
10.        end
11.    end
12. end
13. end
14.
```

Program menghitung histogram pada semua pixel pada gambar (grayscale/rgb) dengan menghitung banyak kemunculan tiap graylevel dari 0 sampai 255. Asumsi input adalah gambar bertipe uint8. Berikut adalah hasil percobaan pada gambar building. Bisa dilihat program, menghasilkan histogram dengan benar.



2. Perbaikan kualitas citra

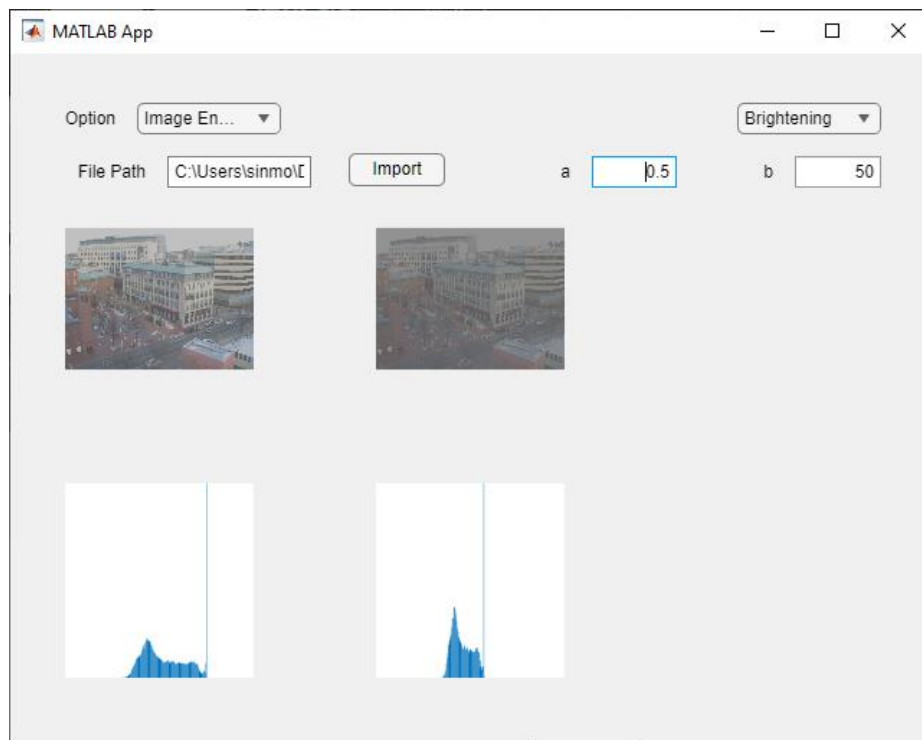
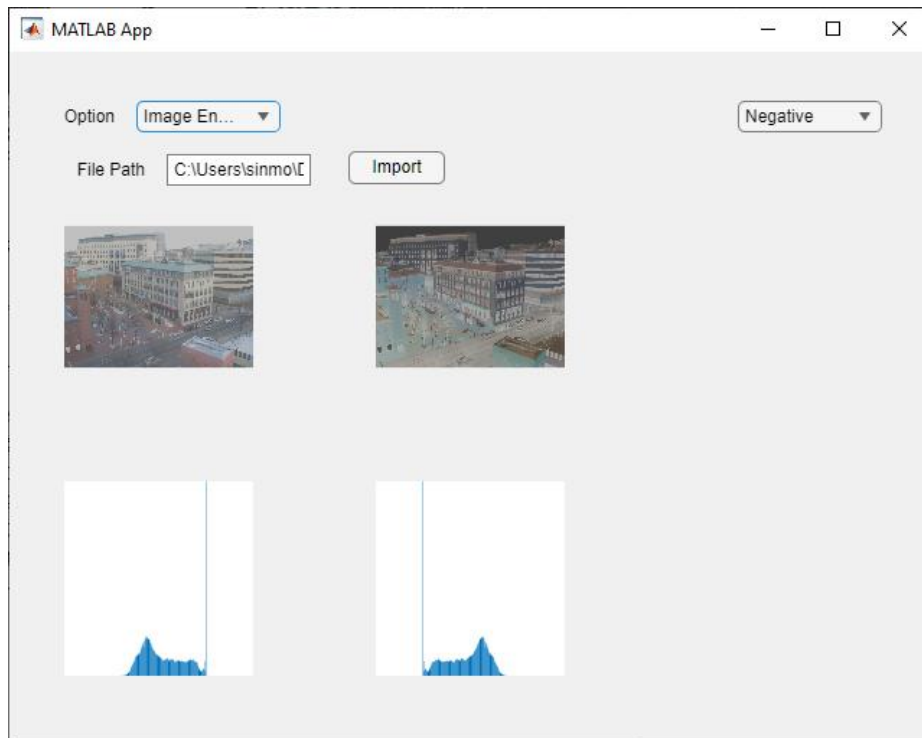
```
1. % brightening
2. function [imgMod] = brightening(img, b, a)
3.     if nargin < 3
4.         a = 1;
5.     end
6.
7.     imgMod = img * a + b;
```

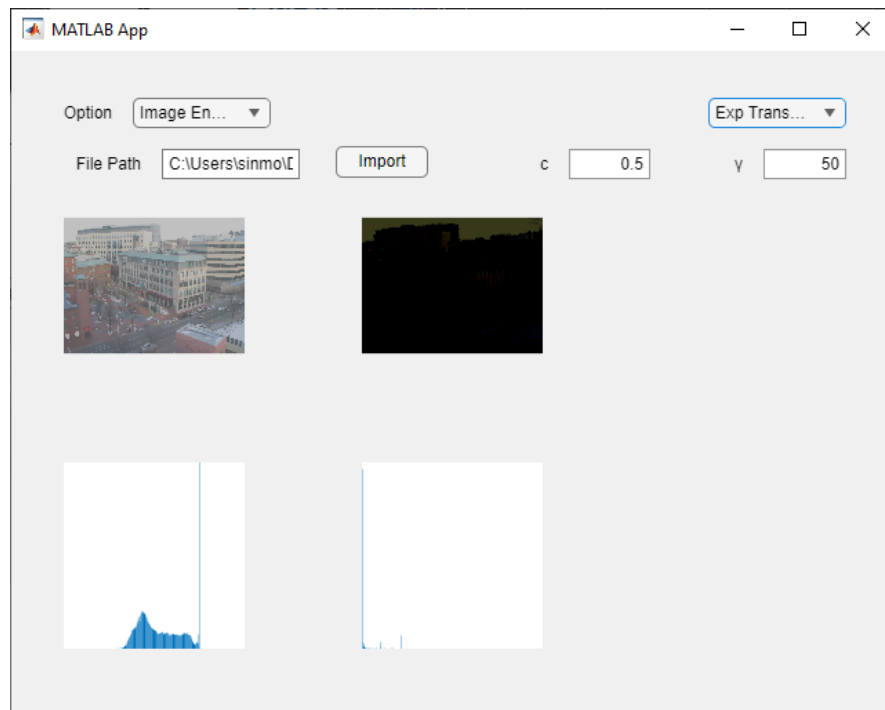
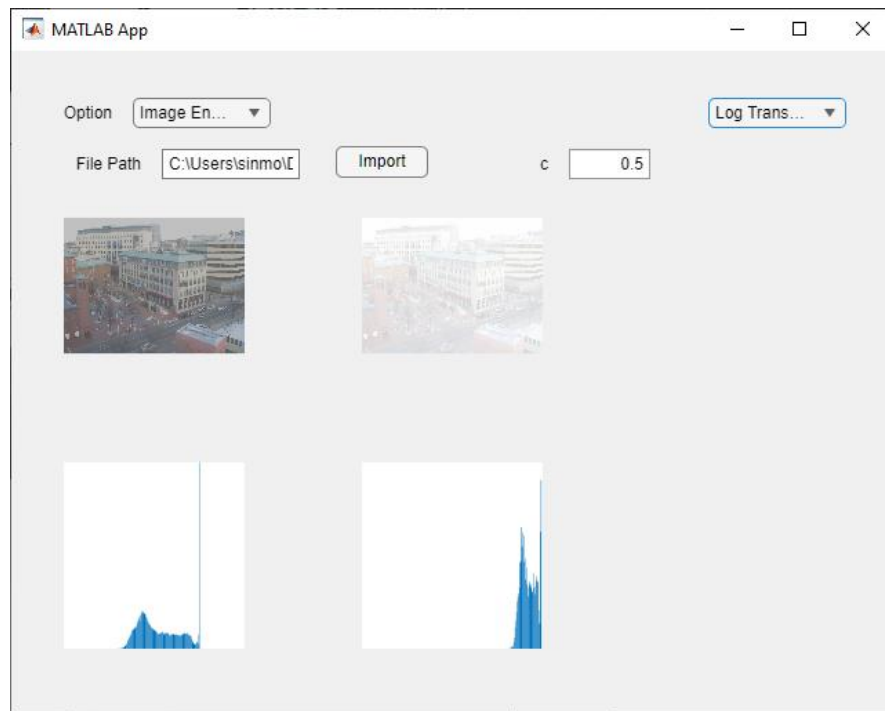
```

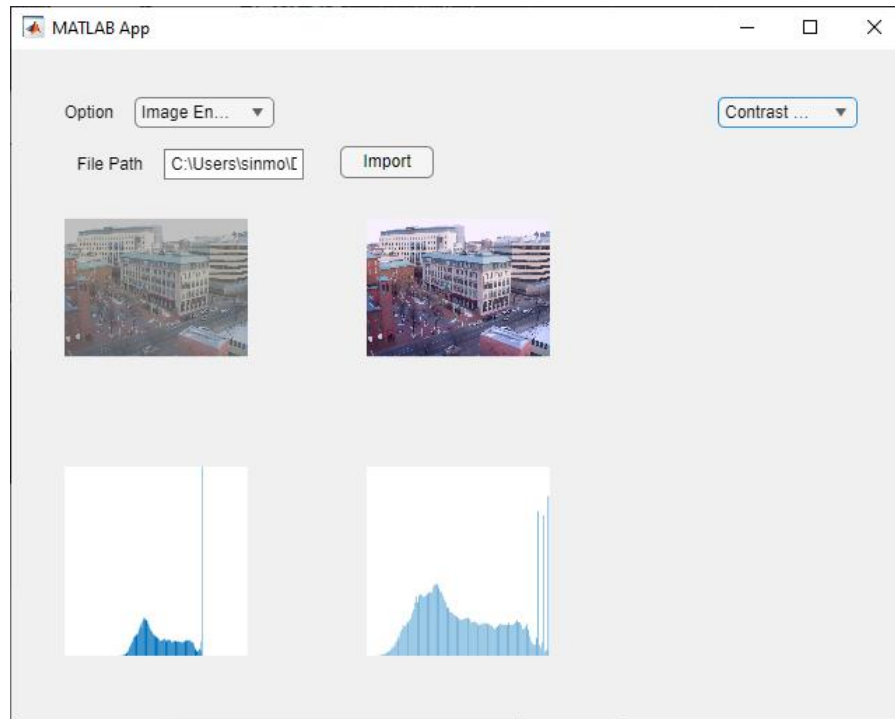
8. end
9.
10.% negative
11.function [imgMod] = negative(img)
12.    greylevel = 256;
13.    imgMod = (greylevel - 1) - img;
14.end
15.
16.% log
17.function [imgMod] = logtrans(img, c)
18.    imgMod = c * log(double(img) + 1);
19.    imgMod = imgMod./max(max(imgMod)); % normalization
20.end
21.
22.% pangkat
23.function [imgMod] = powtrans(img, c, gamma)
24.    imgMod = c * (double(img) .^ gamma);
25.    imgMod = imgMod./max(max(imgMod)); % normalization
26.end
27.
28.% contrast streteching
29.function [imgMod] = contraststretching(img)
30.    rmin = min(min(img));
31.    rmax = max(max(img));
32.    imgMod = (img - rmin) .* (255 / (rmax - rmin));
33.end
34.

```

Perbaikan kualitas citra dapat dilakukan dengan melakukan transformasi: brightening, negative, log, pangkat, dan contrast streteching. Transformasi negative dihasilkan dengan membalikkan histogram citra/dengan mengurangi tiap pixel dari nilai graylevel tertinggi. Transformasi brightening menerima 2 input (a dan b), kemudian tiap pixel ditransformasi menjadi $a * \text{pixel} + b$. Transformasi log menerima 1 input (c), kemudian tiap pixel ditransformasi menjadi $c * \log(\text{pixel})$. Transformasi pangkat menerima 2 input (c dan gamma), kemudian tiap pixel ditransformasi menjadi $c * \text{pixel} ^ \text{gamma}$. Contrast stretching mentransformasi tiap pixel menjadi $(\text{pixel} - \text{rmin}) * (255 / (\text{rmax} - \text{rmin}))$.







Bisa dilihat dari beberapa gambar di atas, program berhasil melakukan transformasi perbaikan kualitas citra. Perbaikan citra paling jelas ada pada transformasi contrast stretching yang menghasilkan citra bagus dan tidak washed-out.

3. Histogram Equalization

```

1. function [imgMod, heq] = histequal(img)
2.     if size(img, 3) == 3 % rgb image histogram is splitted per channel
3.         h = histosplit(img);
4.         heq = zeros(256, 1, 3);
5.         sum = zeros(1, 1, 3);
6.     else
7.         h = histo(img);
8.         heq = zeros(256, 1);
9.         sum = zeros(1, 1);
10.    end
11.    s = size(img, 1) * size(img, 2);
12.    h = h / s;
13.    imgMod = img;
14.    for i=1:256
15.        for rgb=1:size(img, 3)
16.            sum(rgb) = sum(rgb) + h(i, 1, rgb);
17.            heq(i, 1, rgb) = floor(255 * sum(rgb));
18.        end
19.    end
20.
21.    for row = 1:size(img, 1)
22.        for col = 1:size(img, 2)
23.            for rgb = 1:size(img, 3)
24.                imgMod(row, col, rgb) = heq(imgMod(row, col, rgb) + 1, 1, rgb);
25.            end
26.        end
27.    end
28. end
  
```

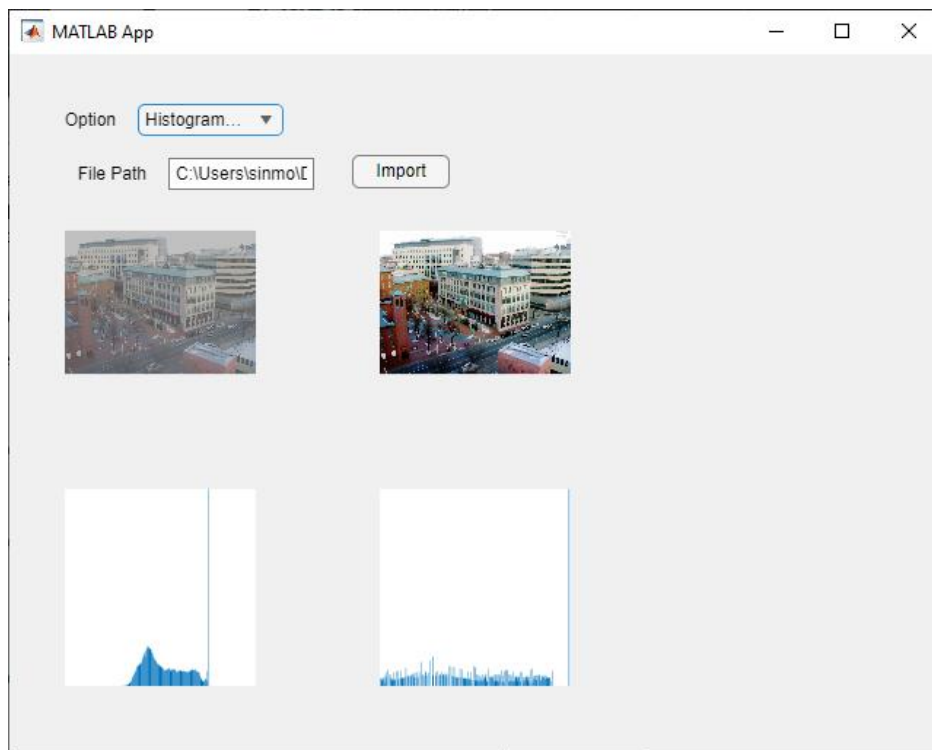


```

29.% helper function
30.function [histArr] = histosplit(img) % split histogram per channel for
    rgb image only
31.    graylevel = 256;
32.    histArr = zeros(graylevel, 1, 3);
33.
34.    % iterate each pixel of image and count each occurrence
35.    for row = 1:size(img, 1)
36.        for col = 1:size(img, 2)
37.            for rgb = 1:size(img, 3)
38.                histArr(img(row, col, rgb) + 1, 1, rgb) =
    histArr(img(row, col, rgb) + 1, 1, rgb) + 1;
39.            end
40.        end
41.    end
42.end
43.

```

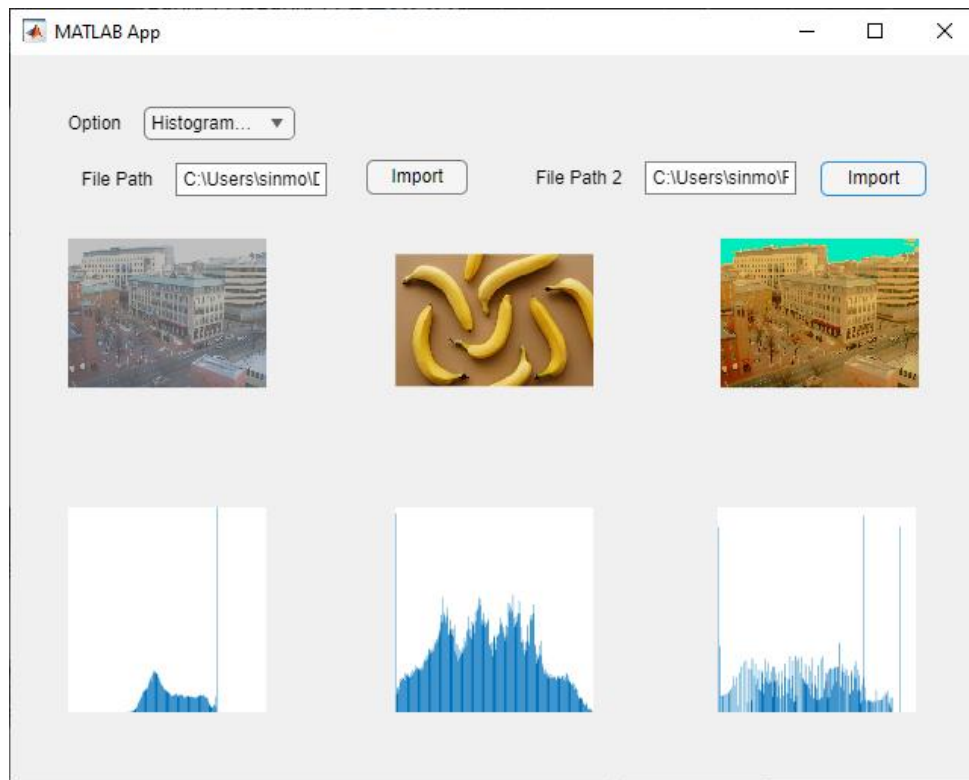
Histogram equalization adalah pemerataan histogram agar gambar terlihat jelas. Bisa dilihat pada gambar di bawah, program berhasil membuat historam menjadi rata sehingga kontras gambar terlihat bagus.



4. Histogram Matching

```
1. function [imgMod, heq] = histmatch(img, imgref)
2.     [~, heqori] = histequal(img);
3.     [~, heqref] = histequal(imgref);
4.
5.     if size(img, 3) == 3
6.         heq = zeros(256, 1, 3);
7.     else
8.         heq = zeros(256, 1);
9.     end
10.
11.     for i=1:256
12.         for rgb=1:size(img, 3)
13.             minval = 999;
14.             for j=1:256
15.                 range = abs(heqori(i, 1, rgb) - heqref(j, 1, rgb));
16.                 if range < minval
17.                     minval = range;
18.                 elseif range > minval
19.                     heq(i, 1, rgb) = j;
20.                     break
21.                 end
22.             end
23.         end
24.     end
25.
26.     imgMod = img;
27.     for row = 1:size(img, 1)
28.         for col = 1:size(img, 2)
29.             for rgb = 1:size(img, 3)
30.                 imgMod(row, col, rgb) = heq(imgMod(row, col, rgb) + 1, 1, rgb);
31.             end
32.         end
33.     end
34. end
35.
```

Histogram matching adalah pencocokan histogram yang dilakukan dengan mengubah histogram pada citra original menjadi mirip dengan histogram pada citra target. Gambar di bawah menunjukkan demonstrasi program dengan melakukan matching pada gambar building yang washed-out dengan gambar banana dengan kontras bagus. Hasil dari histogram matching adalah gambar building dengan kontras yang bagus dengan warna kuning seperti pada gambar banana.



Github: <https://github.com/Moses-ui/tugas-1-citra>