

**PROJECT UJIAN AKHIR SEMESTER PADA MATA KULIAH TEORI PENGOLAHAN
CITRA DIGITAL**

PROGRAM PHOTO EDITOR BERBASIS MATLAB



Dosen :

Ir. Gunawan Pria Utama, M.Kom

NAMA KELOMPOK:

Andhika Wisnu Wardhana 1710511077

Muhammad Arlanda Valio 171051180

Muhammad Abi Nurhakim 1710511083

Teknik Informatika

Fakultas Ilmu Komputer

Universitas Pembangunan Nasional "Veteran" Jakarta

2019

ABSTRAK

Fotografi saat ini telah berkembang menjadi sebuah gaya hidup, hal ini dimulai semenjak munculnya era digital dan berkembangnya media sosial. Siapapun bisa melakukan kegiatan ini, karena semakin berkembangnya teknologi kamera digital yang menawarkan kemudahan dalam memotret. Bahkan saat ini, kita tidak perlu membawa kamera digital karena kamera sudah tersemat pada kamera ponsel masing-masing individu, yang hasilnya tidak kalah memuaskan dengan kamera digital. Oleh karena itu, sebagian besar orang saat ini selalu mendokumentasikan berbagai momen yang dilaluinya, karena wujud sederhananya foto bisa menyimpan banyak keindahan.

Namun terkadang hasil jepretan terlihat kurang berkualitas, dan sebagian besar orang ingin membuat foto menjadi lebih bagus dan menarik. Oleh karena itu, munculah aplikasi untuk memperbaiki kualitas citra dalam foto yaitu Photo Editor.

Di dalam aplikasi ini terdapat beberapa fitur yang ditawarkan seperti memberikan efek pada foto sesuai keinginan seperti *RGB to Grayscale*, *RGB to Black White*, *RGB to Negative*, *RGB to HSV*, *RGB to YCbCr*, *RGB to YIQ*, *RGB to YUV*. Terdapat pula fitur untuk meningkatkan *Brightness* dan *Contrast*, dan juga fitur *Flip* dan *Rotate*. Dalam aplikasi ini, gambar atau foto menjadi sebuah inputan. User menginputkan gambar, kemudian prosesnya adalah dengan menambahkan berbagai macam efek yang diinginkan untuk digunakan, sehingga didapatkan output berupa gambar atau foto yang telah diberi efek sesuai dengan selera user.

Kata Kunci : fotografi, *Photo Editor*, Matlab

BAB I

PENDAHULUAN

1.1 Latar Belakang

Era perkembangan teknologi tak berhenti, dimana selalu mengalami perkembangan pesat seiring dengan laju perkembangan Teknologi yang sangat beraneka ragam, diantaranya adalah sektor fotografi. Oleh karena itu, memanfaatkan termasuk fotografi digital telah membuka banyak pintu kesempatan dan kemungkinan, terutama karena kita tidak perlu lagi menunggu lama untuk melihat hasil jepretan dan hasil pemotretan kamera, secara instan kita dapat melihat hasil foto yang diambil. Fotografi adalah proses atau metode untuk menghasilkan gambar atau foto dari suatu objek dengan merekam pantulan cahaya yang mengenai objek tersebut pada media yang peka cahaya. Alat paling populer untuk menangkap cahaya ini adalah kamera. Tanpa cahaya, tidak ada foto yang bisa dibuat. Namun demikian, seringkali hasil jepretan kurang berkualitas. Dan aplikasi edit foto lah yang membuat kita bisa memperbaiki foto yang rusak, membuat komposisi dari banyak foto dan bahkan membuat karya seni original yang menawan. Dengan software-software seperti ini, kita bisa mengubah foto yang buruk menjadi sesuatu yang mengagumkan.

Setelah mengamati permasalahan tersebut, kami ingin membuat sebuah aplikasi untuk memperbaiki kualitas citra dalam suatu foto yaitu photo editor. Banyak penggemar fotografi yang memanfaatkan aplikasi photo editor untuk proses editing hingga menambah efek dengan tujuan menambah keindahan foto mereka.

Di dalam aplikasi ini ada beberapa fitur yang ditawarkan seperti memberikan efek pada foto sesuai keinginan seperti *RGB to Grayscale*, *RGB to Black White*, *RGB to Negative*, *RGB to CMY*, *RGB to HSI*, *RGB to YCbCr*, *RGB to YIQ*, *RGB to YUV*, *Luminosity (Brightness, Contrast)*, *Flip* dan *Rotate*. Sehingga diharapkan dengan aplikasi photo editor dapat memperbaiki kualitas dari suatu citra karena satu foto memiliki seribu cerita. Satu foto adalah data dan fakta, juga makna.

Dalam aplikasi ini, gambar atau foto menjadi sebuah inputan. User menginputkan gambar atau foto mereka, kemudian prosesnya adalah dengan menambahkan berbagai macam efek yang diinginkan untuk digunakan, sehingga didapatkan output berupa gambar atau foto yang telah diberi efek sesuai dengan selera user. Aplikasi *Photo Editor* ini dibuat dengan menggunakan Matlab. Matlab merupakan salah satu dari bahasa pemrograman yang termudah dalam hal penulisan program matematika. Maka dari itu matlab cocok untuk membuat software tentang program Photo Editor..

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah tersebut, maka rumusan masalah penelitian adalah:

- a. Bagaimana cara memberikan efek foto seperti RGB, Black White, Negative, CMY, HSI, YcbCr, TIQ, YUV, Luminosity, Flip dan Rotate?
- b. Apa saja sintaks yang digunakan pada MATLAB?

1.3 Tujuan Masalah

- a. Untuk mengetahui efek foto seperti RGB, Black White, Negative, CMY, HSI, YcbCr, TIQ, YUV, Luminosity, Flip dan Rotate.
- b. Untuk mengetahui sintaks dasar pada aplikasi MATLAB.

BAB II

TINJAU PUSTAKA

2.1 Pengolahan Citra Digital

- Pengolahan citra digital (Digital Image Processing) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer.

Secara matematis, citra merupakan fungsi kontinyu (continue) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Representasi dari fungsi kontinyu menjadi nilai-nilai diskrit disebut digitalisasi citra. Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi $f(x,y)$ yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel (pixel = picture element) atau elemen terkecil dari sebuah citra.

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Suatu citra $f(x,y)$ dalam fungsi matematis dapat dituliskan sebagai berikut:

$$\begin{aligned} 0 &\leq x \leq M-1 \\ 0 &\leq y \leq N-1 \\ 0 &\leq f(x,y) \leq G-1 \end{aligned}$$

dimana :

M = jumlah piksel baris (row) pada array citra

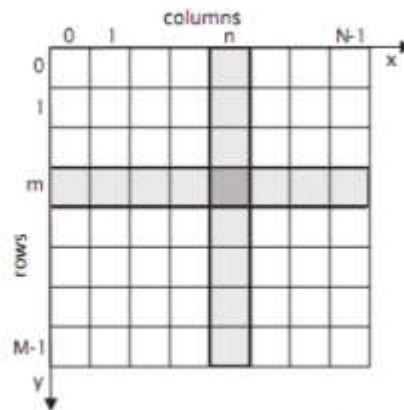
N = jumlah piksel kolom (column) pada array citra

G = nilai skala keabuan (graylevel)

Besarnya nilai M, N dan G pada umumnya merupakan perpangkatan dari dua.

$$M = 2^m ; N = 2^n ; G = 2^k$$

dimana nilai m, n dan k adalah bilangan bulat positif. Interval (0,G) disebut skala keabuan (grayscale). Besar G tergantung pada proses digitalisasinya. Biasanya keabuan 0 (nol) menyatakan intensitas hitam dan 1 (satu) menyatakan intensitas putih. Untuk citra 8 bit, nilai G sama dengan $2^8 = 256$ warna (derajat keabuan).

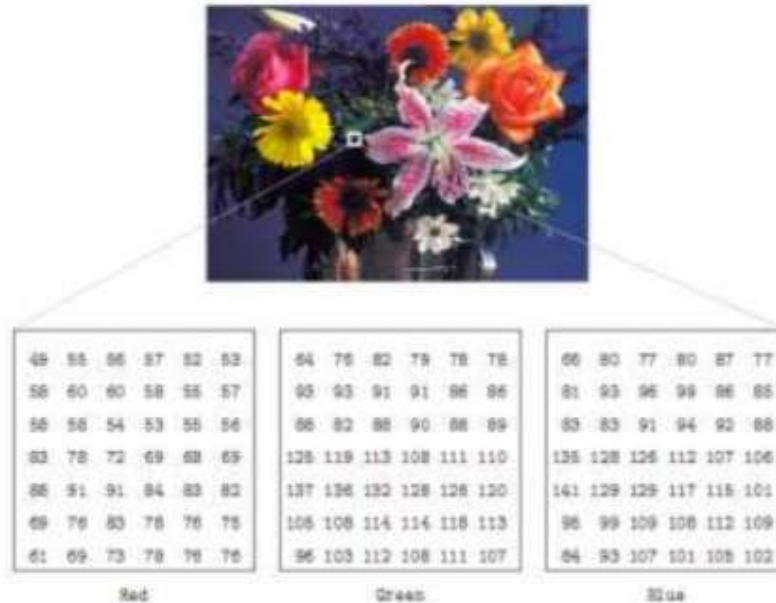


Gambar 1: Representasi citra digital dalam 2 dimensi

Obyek tertentu dapat dideteksi dengan menggunakan pengolahan citra digital ini. Salah satu metode yang digunakan adalah berdasarkan segmentasi warna. Normalisasi RGB adalah salah satu metode segmentasi warna yang memiliki kelebihan yaitu mudah, proses cepat dan efektif pada obyek traffic sign, maupun aplikasi untuk face detection.

- Jenis Citra Digital.

Pada aplikasi pengolahan citra digital pada umumnya, citra digital dapat dibagi menjadi 3, color image, black and white image dan binary image. 1. Color Image atau RGB (Red, Green, Blue). Pada color image ini masing-masing piksel memiliki warna tertentu, warna tersebut adalah merah (Red), hijau (Green) dan biru (Blue). Jika masing-masing warna memiliki range 0 - 255, maka totalnya adalah $255^3 = 16.581.375$ (16 K) variasi warna berbeda pada gambar, dimana variasi warna ini cukup untuk gambar apapun. Karena jumlah bit yang diperlukan untuk setiap pixel, gambar tersebut juga disebut gambar-bit warna. Color image ini terdiri dari tiga matriks yang mewakili nilai-nilai merah, hijau dan biru untuk setiap pikselnya, seperti yang ditunjukkan gambar 2.



Gambar 2: *Color Image*[7].

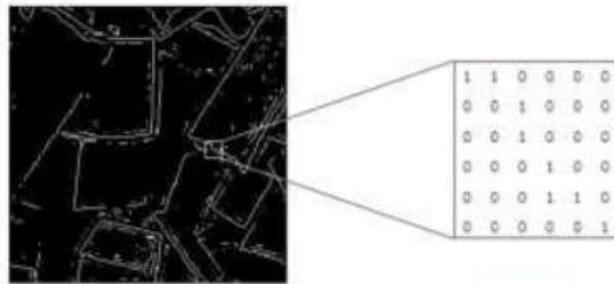
- **Black and White.** Citra digital black and white (grayscale) setiap pikselnya mempunyai warna gradasi mulai dari putih sampai hitam. Rentang tersebut berarti bahwa setiap piksel dapat diwakili oleh 8 bit, atau 1 byte. Rentang warna pada black and white sangat cocok digunakan untuk pengolahan file gambar. Salah satu bentuk fungsinya digunakan dalam kedokteran (X-ray). Black and white sebenarnya merupakan hasil rata-rata dari color image.
- **Binary Image.**
Setiap piksel hanya terdiri dari warna hitam atau putih, karena hanya ada dua warna untuk setiap piksel, maka hanya perlu 1 bit per piksel (0 dan 1) atau apabila dalam 8 bit (0 dan 255), sehingga sangat efisien dalam hal penyimpanan. Gambar yang direpresentasikan dengan biner sangat cocok untuk teks (dicetak atau tulisan tangan), sidik jari (finger print), atau gambar arsitektur. Binary image merupakan hasil pengolahan dari black and white image, dengan menggunakan fungsi sebagai berikut :

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 255 & I_{BW}(x, y) \geq T \end{cases} \quad (4)$$

dan dalam bentuk *floating point*

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 1 & I_{BW}(x, y) \geq T \end{cases} \quad (5)$$

dimana $I_{BW}(x, y)$ = nilai piksel *Gray* titik (x, y) , $I_{Bin}(x, y)$ = nilai piksel *Binary* titik (x, y) , sedangkan T adalah nilai *threshold*.



Gambar 4: *Binary Image*[7]

- Color Space

Color space adalah kemampuan reproduksi warna dalam video. Dalam video analog, biasanya disebut YUV untuk PAL atau NTSC. Untuk SECAM menggunakan YDbDr. Sedangkan Y'PbPr adalah pengelompokan warna universal yang digunakan sebagai standar kondensasi warna pada analog (pada digital disebut Y'CbCr). Y adalah satuan untuk *luma components*, Cb/Pb adalah satuan untuk *Croma blue components* dan Cr/Pr adalah satuan untuk *Chroma red components*. Ketersediaan warna dalam video tidak dihitung dalam jumlah warna, tapi jumlah ketersediaan yang direpresentasikan dalam sebuah diagram warna.

2.2 Matlab

MATLAB (Matrix Laboratory) merupakan sebuah lingkungan komputasi numerikal dan bahasa pemrograman komputer generasi keempat. Dengan menggunakan bahasa tingkat tinggi (high level language), Matlab sangat mudah untuk dioperasikan oleh penggunaanya. Matlab banyak digunakan oleh kalangan akademisi maupun industri.

MATLAB adalah sebuah bahasa dengan kemampuan tinggi untuk komputasi teknis. Ia menggabungkan komputasi, visualisasi, dan pemrograman dalam satu kesatuan yang mudah digunakan di mana masalah dan penyelesaiannya diekspresikan dalam notasi matematik yang sudah dikenal.

Pemakaian MATLAB meliputi :

- Matematika dan komputasi
- Pengembangan algoritma
- Akuisisi data

- Pemodelan, simulasi dan prototype
- Grafik saintifik dan engineering
- Perluasan pemakaian, seperti graphical user interface (GUI).

maka itu, sistem interaktif yang mempunyai basis data array yang tidak membutuhkan dimensi. Ini memungkinkan kita dapat menyelesaikan banyak masalah komputasi teknis, khususnya yang berkaitan dengan formulasi matrik dan vector.

Nama MATLAB merupakan singkatan dari matrix laboratory. Matlab juga memiliki bahasa pemrograman tinggi, tertutup, dan case sensitive dalam lingkungan komputasi numerik yang dikembangkan oleh MathWorks. Salah satu kelebihan yang paling populer adalah kemampuan membuat grafik dengan visualisasi terbaik. MATLAB mempunyai banyak tools yang dapat membantu berbagai disiplin ilmu. Ini merupakan salah satu penyebab industri menggunakan MATLAB. Selain itu MATLAB mempunyai banyak library yang sangat membantu untuk menyelesaikan permasalahan matematika seperti membuat simulasi fungsi, pemodelan matematika dan perancangan GUI.

BAB III

METODE PENELITIAN

Pengembangan aplikasi ini dilakukan dengan menggunakan software Matlab, oleh karena itu kami menggunakan sintaks-sintaks yang ada pada Matlab. Pada aplikasi *Photo Editor* yang kami buat terdapat efek-efek untuk mengedit foto. Sintaks yang digunakan adalah :

RGB to Grayscale

```
% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton1
set(handles.radiobutton1,'value',0);
citra = handles.data1;
axes(handles.axes2)
gray = rgb2gray(citra);
imshow(citra)
imshow(gray)
title('Grayscale');
axes(handles.axes3);
imhist(gray);
```

RGB to Black & White

```
127 % --- Executes on button press in radiobutton2.
128 function radiobutton2_Callback(hObject, eventdata, handles)
129 % hObject    handle to radiobutton2 (see GCBO)
130 % eventdata  reserved - to be defined in a future version of MATLAB
131 % handles    structure with handles and user data (see GUIDATA)
132 - set(handles.radiobutton2,'value',0);
133 - citra = handles.data1;
134 - axes(handles.axes2)
135 - bw = im2bw(citra);
136 - imshow(citra)
137 - imshow(bw)
138 - title('Black & White');
139 - axes(handles.axes3);
140 - imhist(bw);
141 % Hint: get(hObject,'Value') returns toggle state of radiobutton2
142
```

RGB to Negative

```
% --- Executes on button press in radiobutton3.  
function radiobutton3_Callback(hObject, eventdata, handles)  
% hObject    handle to radiobutton3 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
% Hint: get(hObject,'Value') returns toggle state of radiobutton3  
set(handles.radiobutton3,'value',0);  
citra = handles.data1;  
axes(handles.axes2)  
comp = imcomplement(citra);  
imshow(citra)  
imshow(comp)  
title('Negative');
```

RGB to YCbCr

```
% --- Executes on button press in radiobutton4.  
function radiobutton4_Callback(hObject, eventdata, handles)  
% hObject    handle to radiobutton4 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
% Hint: get(hObject,'Value') returns toggle state of radiobutton4  
set(handles.radiobutton4,'value',0);  
citra = handles.data1;  
axes(handles.axes2)  
ycbcr = rgb2ycbcr(citra);  
imshow(citra)  
imshow(ycbcr)  
title('YCbCr');
```

RGB to YIQ

```
% --- Executes on button press in radiobutton5.  
function radiobutton5_Callback(hObject, eventdata, handles)  
% hObject    handle to radiobutton5 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
% Hint: get(hObject,'Value') returns toggle state of radiobutton5  
set(handles.radiobutton5,'value',0);  
citra = handles.data1;  
YIQ = rgb2ntsc(citra);  
axes(handles.axes2)  
imshow(YIQ);  
title('YIQ');
```

RGB to HSV

```
% --- Executes on button press in radiobutton6.  
function radiobutton6_Callback(hObject, eventdata, handles)  
% hObject    handle to radiobutton6 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
% Hint: get(hObject,'Value') returns toggle state of radiobutton6  
set(handles.radiobutton6,'value',0);  
citra = handles.data1;  
HSV = rgb2hsv(citra);  
axes(handles.axes2)  
imshow(HSV);  
title('HSV');
```

RGB to YUV

```
function radiobutton7_Callback(hObject, eventdata, handles)  
% hObject    handle to radiobutton7 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
% Hint: get(hObject,'Value') returns toggle state of radiobutton7  
set(handles.radiobutton7,'value',0);  
RGB=handles.data1;  
R = RGB(:,1,1);  
G = RGB(:,1,2);  
B = RGB(:,1,3);  
Y = 0.299 * R + 0.587 * G + 0.114 * B;  
U = -0.14713 * R - 0.28886 * G + 0.436 * B;  
V = 0.615 * R - 0.51499 * G - 0.18001 * B;  
YUV = cat(3,Y,U,V);  
axes(handles.axes2);  
imshow(YUV);  
title('YUV');
```

Dikarenakan tidak adanya sintaks pada Matlab untuk RGB to YUV, maka kami menggunakan source code untuk menghasilkan gambar YUV. Sinyal YUV diciptakan dari sumber RGB (red, green, blue), dengan diberi bobot tertentu, nilai R, G dan B ditambahkan untuk menghasilkan sinyal Y, yang menyatakan terang-gelap secara keseluruhan, atau *luminance* dari titik tersebut. Sinyal U lalu dibentuk dengan mengurangi sinyal blue dari RGB dengan Y, dan V dengan mengurangi red.

Selain dapat memberikan efek pada foto, aplikasi Photo Editor ini juga terdapat slider untuk meningkatkan atau mengurangi Brightness dan Contrast pada citra dengan menggunakan source code sebagai berikut :

Slider Brightness

```
function slider1_Callback(hObject, eventdata, handles)
% hObject handle to slider1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%       get(hObject,'Min') and get(hObject,'Max') to determine range of slider
proyek=guidata(gcf);
global cerah
set(handles.slider1,'Max',200);
set(handles.slider1,'Min',-200);
valC = get(handles.slider1,'Value');
valK = get(handles.slider2,'Value');
set(handles.edit1,'String',valC);
citra = handles.data1;
kontras=valK*(citra+valC);
cerah=kontras+valC;
imshow(cerah,'Parent',handles.axes2);
```

Dimulai dengan menentukan nilai batas Max dan Min pada Slider, kami menempatkan nilai 200 untuk nilai Max dan -200 untuk nilai Min. Untuk mendapatkan nilai Brightness dan Contrast dari slider digunakan sintaks :

```
valC = get(handles.slider1,'Value');
valK = get(handles.slider2,'Value');
```

Brightness merupakan penjumlahan dari nilai Contrast yang sudah di set dengan sintaks :
kontras=valK*(citra+valC);

Dijumlahkan dengan value dari slider Brightness.

Slider Contrast

```
function slider2_Callback(hObject, eventdata, handles)
% hObject handle to slider2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%       get(hObject,'Min') and get(hObject,'Max') to determine range of slider
proyek=guidata(gcf);
global cerah
set(handles.slider2,'Max',5);
set(handles.slider2,'Min',-5);
valCerah = get(handles.slider1,'Value');
valKontras = get(handles.slider2,'Value');
set(handles.edit2,'String',valKontras);
citra = handles.data1;
cerah=citra+valCerah;
kontras=valKontras*cerah;
imshow(kontras,'Parent',handles.axes2);
```

Dimulai dengan menentukan nilai batas Max dan Min pada Slider, kami menempatkan nilai 5 untuk nilai Max dan -5 untuk nilai Min. Untuk mendapatkan nilai Brightness dan Contrast dari slider digunakan sintaks :

```
valCerah = get(handles.slider1,'Value');
valKontras = get(handles.slider2,'Value');
```

Contrast merupakan perkalian dari nilai Brightness yang sudah di set dengan sintaks :
cerah=citra+valCerah;

Dikalikan dengan value dari slider Contrast.

Untuk memutar gambar terdapat 2 tombol, yaitu 'Clockwise' dan 'C-Clockwise'.

C-Clockwise

```
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
IM = getimage(handles.axes2);
rotate=imrotate(IM,90);
axes(handles.axes2)
imshow(rotate);
```

Dengan sintaks tersebut, gambar akan diputar berlawanan arah jarum jam sebanyak 90° pada setiap klik.

Clockwise

```
% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
IM = getimage(handles.axes2);
rotate=imrotate(IM,-90);
axes(handles.axes2)
imshow(rotate);
```

Dengan sintaks tersebut, gambar akan diputar searah jarum jam sebanyak 90° pada setiap klik.

Untuk membalikkan gambar terdapat tombol 'Flip Image' dengan menggunakan sintaks :

```
flip=fliplr(IM);
```

Flip Image

```
% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
IM = getimage(handles.axes2);
flip=fliplr(IM);
axes(handles.axes2)
imshow(flip);
```

Untuk menampilkan histogram RGB dapat menggunakan sintaks :

```

I = citra;
R = I(:,:,1);
G = I(:,:,2);
B = I(:,:,3);
axes(handles.axes3)
hist = histogram(R(:));
hist.FaceColor = [1 0 0];
xlim([0 255])
grid on
hold on
hist = histogram(G(:));
hist.FaceColor = [0 1 0];
xlim([0 255])
hist = histogram(B(:));
hist.FaceColor = [0 0 1];
xlim([0 255])
hold off

```

untuk menentukan histogram Red :

```

hist = histogram(R(:));
hist.FaceColor = [1 0 0];
xlim([0 255])

```

untuk menentukan histogram Green :

```

hist = histogram(G(:));
hist.FaceColor = [0 1 0];
xlim([0 255])

```

untuk menentukan histogram Blue :

```

hist = histogram(B(:));
hist.FaceColor = [0 0 1];
xlim([0 255])

```

Untuk menampilkan informasi pada image dapat menggunakan sintaks :

```

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global info
imageinfo(info);

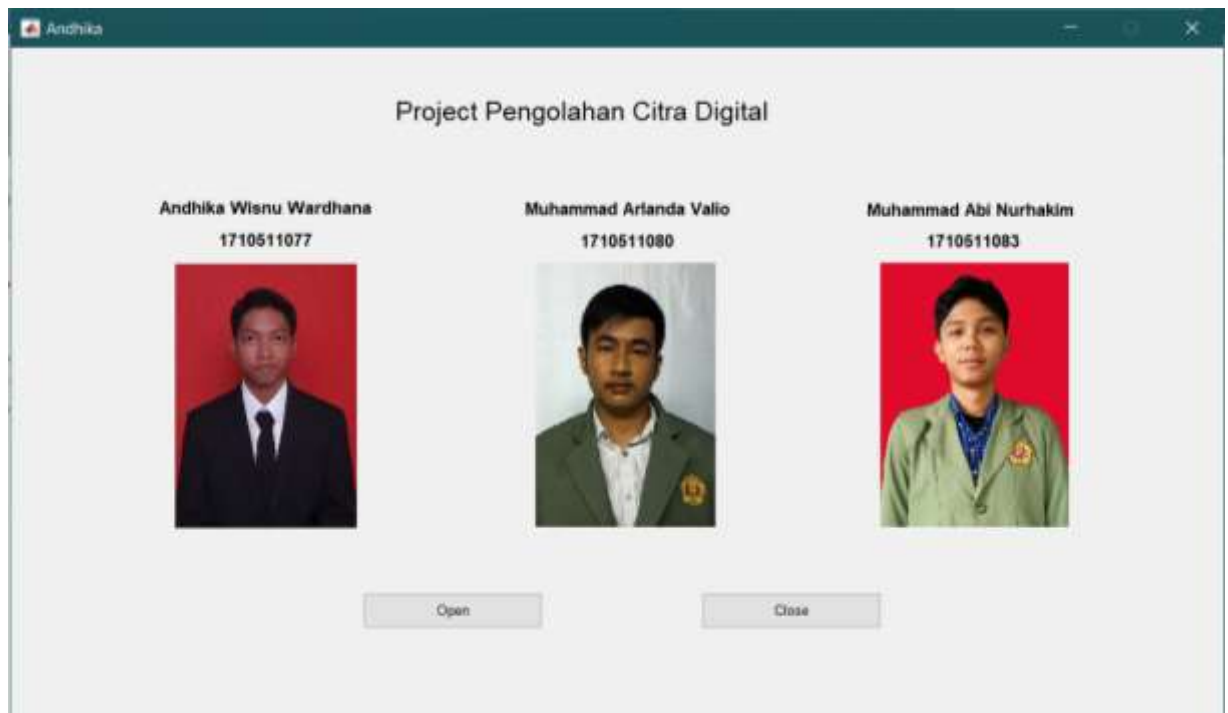
```

BAB IV

HASIL DAN PEMBAHASAN

Aplikasi *Photo Editor* menggunakan Matlab ini terdiri dari dua form kerja user. Pada form pertama terdapat foto dari ketiga anggota kelompok, dan terdapat dua tombol yaitu 'Open' dan 'Close'.

Gambar 1 berikut ini merupakan tampilan awal pada saat aplikasi pertama dijalankan.



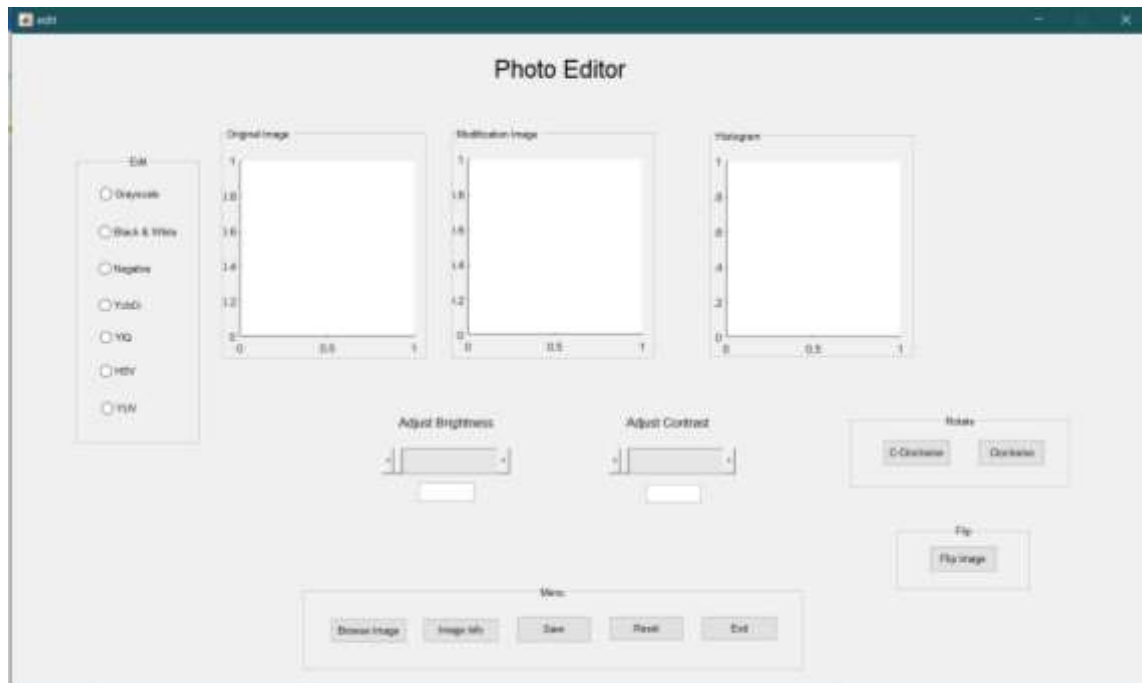
Gambar 1. Tampilan Awal Aplikasi

Untuk memulai proses pengeditan, user diharuskan untuk menekan tombol Open untuk diarahkan ke form 2, yaitu form Pengeditan atau *Photo Editor*. Pada tampilan awal *Photo Editor* terdapat

1. Panel 'Edit' untuk memberikan berbagai efek pada citra.
2. Panel 'Original Image' untuk menampilkan citra atau gambar asli yang belum diedit.
3. Panel 'Modification Image' untuk menampilkan citra atau gambar setelah diedit.
4. Panel 'Histogram' untuk menampilkan perubahan histogram dari citra atau gambar tersebut.
5. Slider 'Adjust Brightness' untuk mengubah kecerahan pada citra atau gambar.
6. Slider 'Adjust Contrast' untuk mengubah kontras pada citra atau gambar.

7. Panel 'Rotate' untuk mengubah rotasi citra atau gambar searah jarum jam ataupun berlawanan jarum jam.
8. Panel 'Flip' untuk membalik gambar atau citra.
9. Panel 'Menu' berisi menu-menu dasar seperti Browse Image, Image Info, Save, Reset, dan Exit.

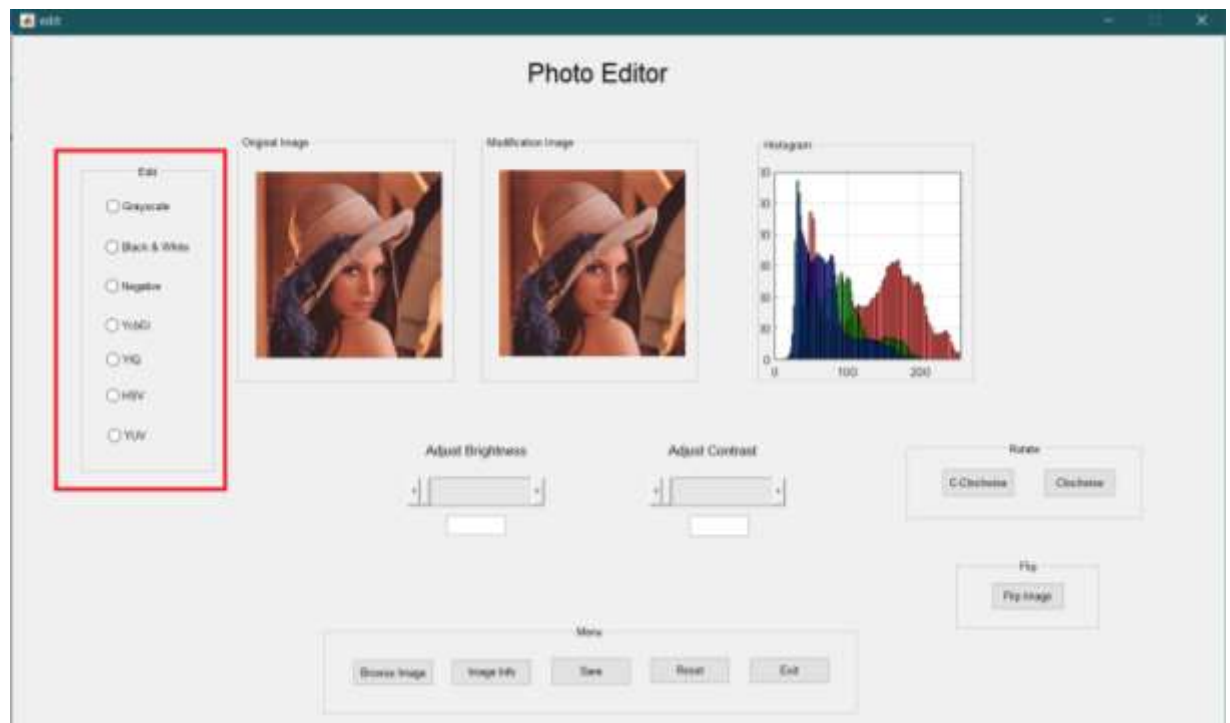
Gambar 2 berikut ini merupakan tampilan awal pada saat aplikasi *Photo Editor* dijalankan.



Gambar 2. Tampilan Awal Aplikasi *Photo Editor*

Untuk dapat memulai proses pengeditan, user diharuskan untuk menginputkan gambar atau citra terlebih dahulu dengan menekan tombol 'Browse Image'.

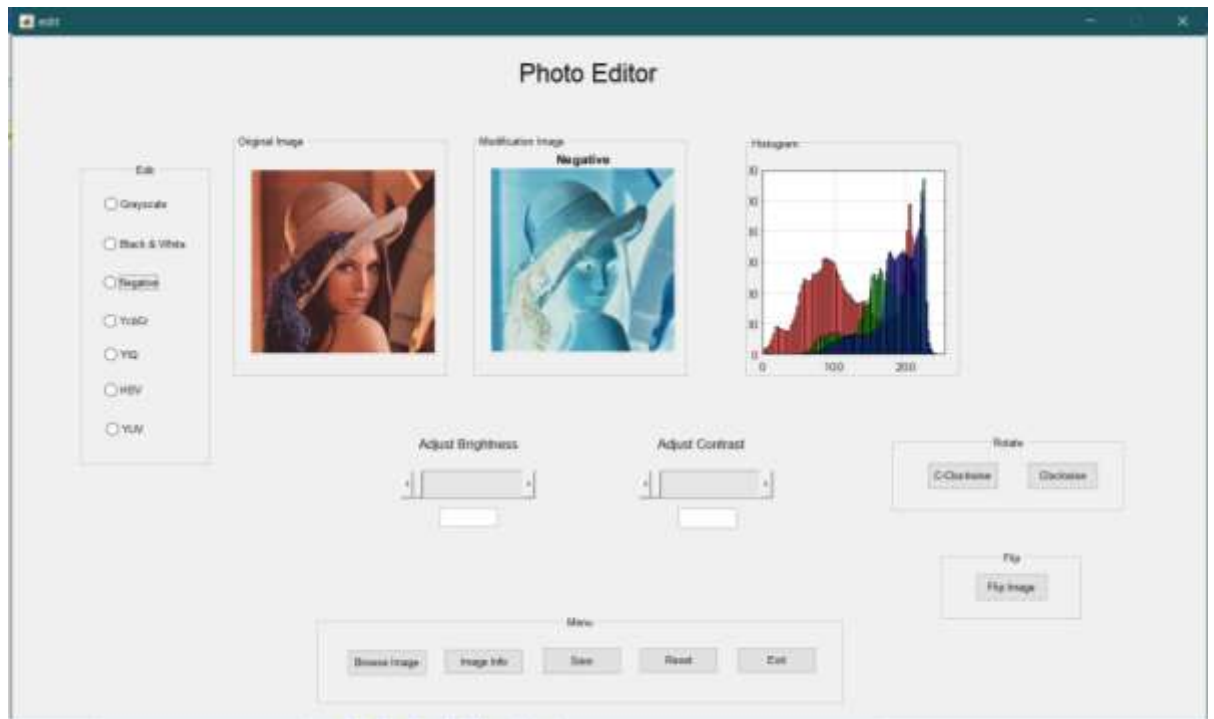
Setelah memilih dan menginputkan gambar atau citra, selanjutnya adalah memilih efek yang akan digunakan dalam proses pengeditan, efek terdapat pada Panel 'Edit', dapat ditunjukkan pada gambar 3 berikut.



Gambar 3. Pemilihan Efek

Beberapa efek dasar yang disediakan dalam aplikasi ini antara lain *RGB to Grayscale*, *RGB to Black & White*, *RGB to Negative*, *RGB to YCbCr*, *RGB to YIQ*, *RGB to HSV*, *RGB to YUV*.

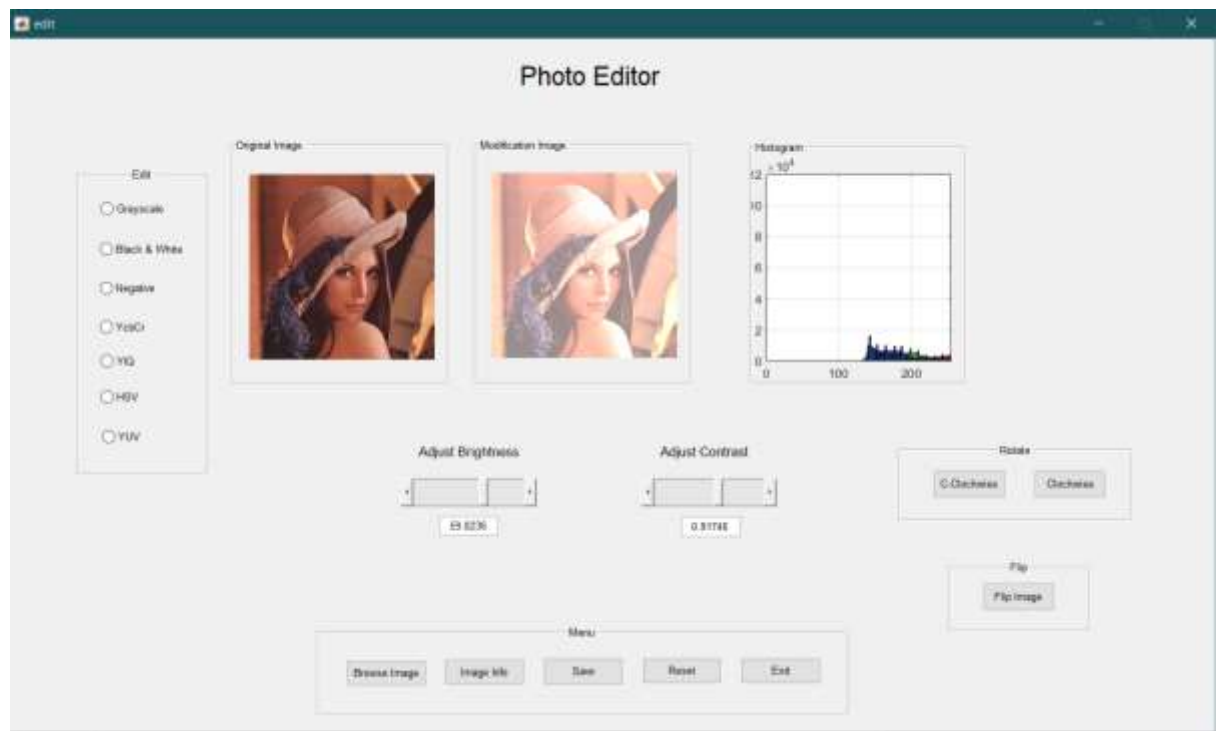
Gambar 4 berikut ini merupakan tampilan apabila user memilih untuk menggunakan efek *RGB to Negative*.



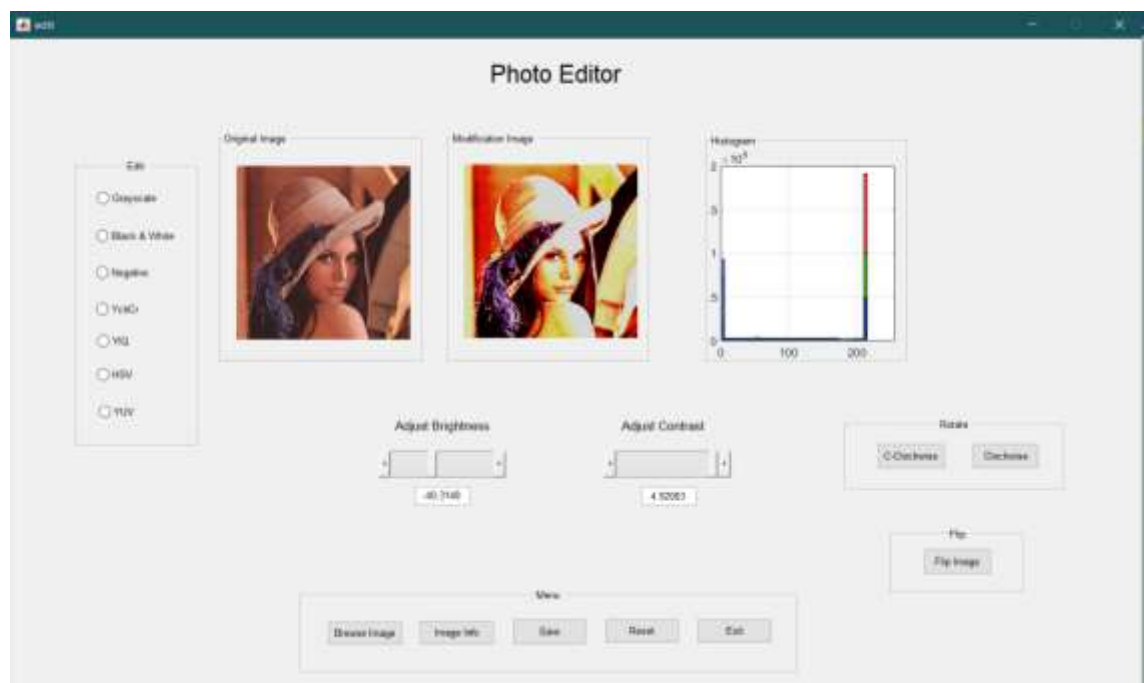
Gambar 4. Pemberian Efek Negative pada Citra

Selain itu, terdapat pengaturan *Lumonisity* yang terdiri dari *Brightness* dan *Contrast*. Pengaturan ini menggunakan slider sehingga user dapat dengan mudah mengatur tingkat kecerahan dan kontras pada gambar atau citra

Gambar 5 dan 6 berikut ini merupakan tampilan apabila user memilih untuk meningkatkan *Brightness* dan *Contrast* pada citra.



Gambar 5. Penambahan *Brightness* pada Citra



Gambar 6. Penambahan *Contrast* pada Citra

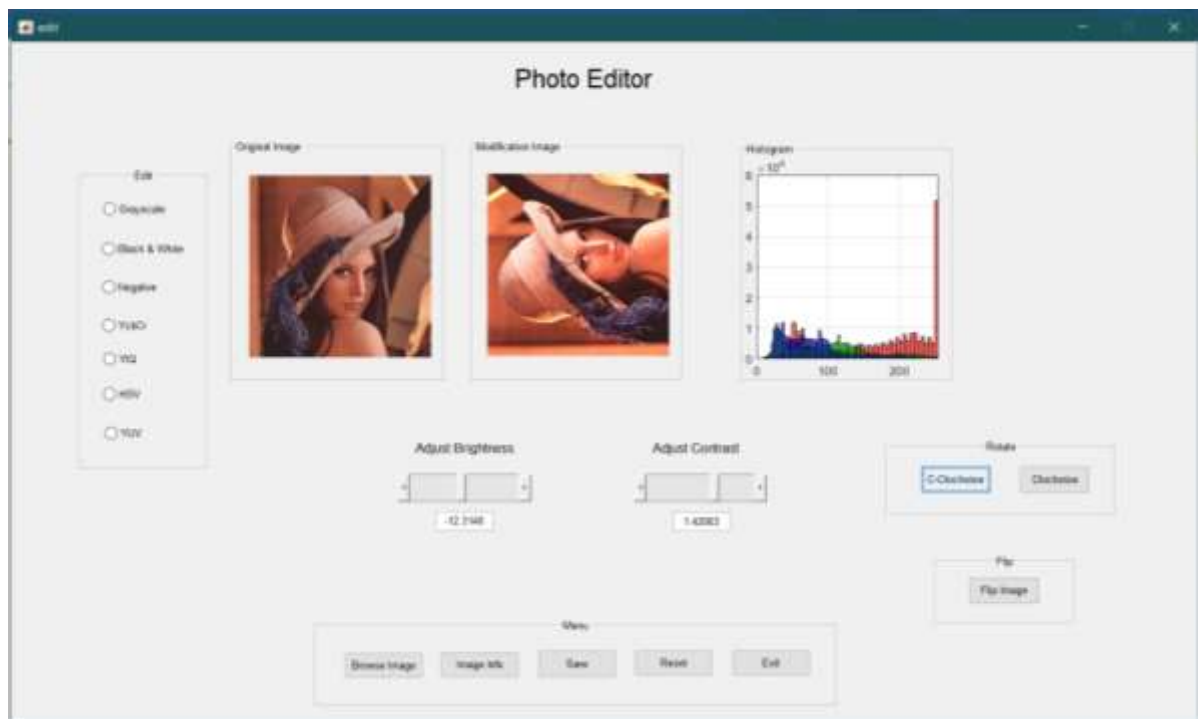
Aplikasi Photo Editor ini juga dapat melakukan flip dan rotate pada gambar atau citra yang diinputkan tersebut. Apabila user menekan tombol 'Clockwise' maka

gambar akan rotate 90° searah jarum jam, dan apabila user menekan tombol ‘C-Clockwise’ maka gambar akan rotate 90° berlawanan arah jarum jam. Sedangkan apabila user menekan tombol ‘Flip Image’ maka gambar akan terbalik secara horizontal.

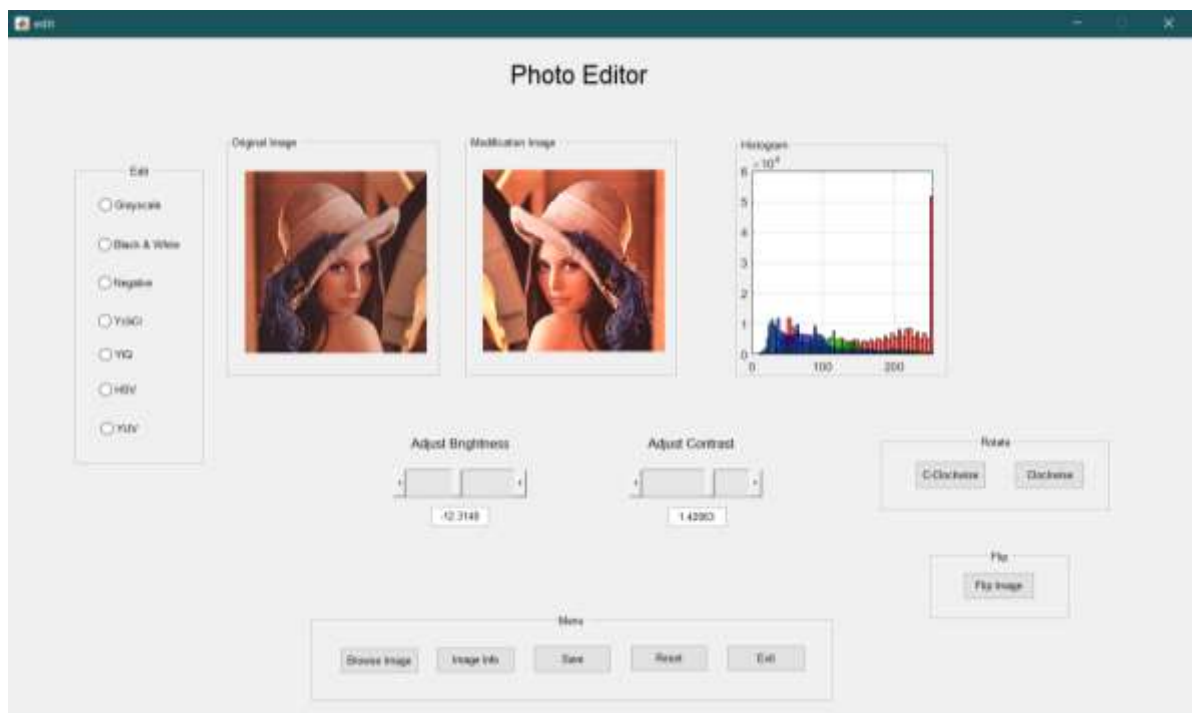
Gambar 7, 8, dan 9 berikut ini merupakan tampilan apabila user memilih untuk rotate ‘Clockwise’, ‘C-Clockwise’, dan ‘Flip Image’ pada citra atau gambar.



Gambar 7. Rotate Clockwise pada Citra



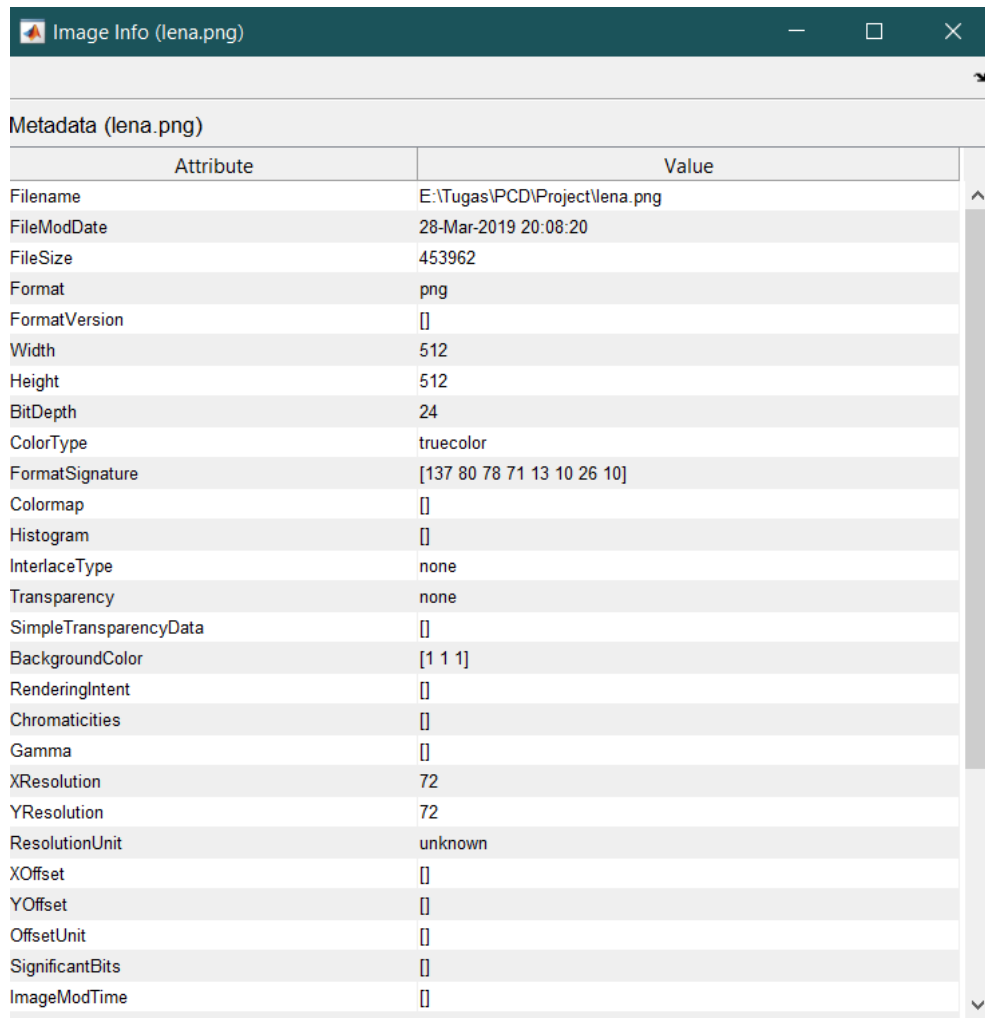
Gambar 8. Rotate C-Clockwise pada Citra



Gambar 9. Flip pada Citra

Aplikasi Photo Editor ini juga dapat menampilkan informasi mengenai gambar secara detil, user dapat melihat informasi gambar tersebut dengan menekan tombol 'Image Info'.

Gambar 10 berikut ini merupakan tampilan apabila user ingin melihat informasi mengenai gambar.



Metadata (lena.png)	
Attribute	Value
Filename	E:\Tugas\PCD\Project\lena.png
FileModDate	28-Mar-2019 20:08:20
FileSize	453962
Format	png
FormatVersion	[]
Width	512
Height	512
BitDepth	24
ColorType	truecolor
FormatSignature	[137 80 78 71 13 10 26 10]
Colormap	[]
Histogram	[]
InterlaceType	none
Transparency	none
SimpleTransparencyData	[]
BackgroundColor	[1 1 1]
RenderingIntent	[]
Chromaticities	[]
Gamma	[]
XResolution	72
YResolution	72
ResolutionUnit	unknown
XOffset	[]
YOffset	[]
OffsetUnit	[]
SignificantBits	[]
ImageModTime	[]

Gambar 10. Informasi mengenai Gambar

BAB V

KESIMPULAN DAN SARAN

Kesimpulan

Kesimpulan yang didapatkan dari Hasil dan Pembahasan aplikasi *Photo Editor* adalah sebagai berikut :

1. Aplikasi *Photo Editor* Berbasis Matlab adalah sistem yang dapat melakukan pengeditan terhadap gambar atau citra yang berasal dari input yang diberikan oleh user.
2. Aplikasi ini menawarkan beberapa pilihan efek yang dapat digunakan oleh user dalam proses pengeditan, antara lain *RGB to Grayscale*, *RGB to Black & White*, *RGB to Negative*, *RGB to YCbCr*, *RGB to YIQ*, *RGB to HSV*, *RGB to YUV*.
3. Aplikasi ini juga menawarkan fitur untuk menambahkan dan mengurangi *Brightness* dan *Contrast*, memutar citra (*Rotate*), dan membalik citra (*Flip*).

Saran

Menyadari aplikasi masih jauh dari kata sempurna, kedepannya akan dilakukan pengembangan lebih lanjut mengenai Aplikasi *Photo Editor* ini, berupa

1. Penambahan fitur –fitur pada aplikasi
2. Perbaikan terhadap bug

DAFTAR PUSTAKA

- 'Apa Itu MATLAB - Tutorial Bahasa Pemrograman MATLAB' (2017).
- 'COLOR SPACE & BIT(s) PER PIXEL' (2009).
<https://tentangvideo.wordpress.com/2009/10/09/5-color-space-bits-per-pixel/>.
- Kusumanto, R. D. and Tompunu, A. N. (2011) 'PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB', 2011(Semantik).
- 'Matlab - Wikipedia bahasa Indonesia, ensiklopedia bebas' (2018).
<https://id.wikipedia.org/wiki/MATLAB>.
- 'Seminar Nasional Teknologi Informasi & Komunikasi Terapan. 2011.
<https://docplayer.info/40671836-Seminar-nasional-teknologi-informasi-komunikasi-terapan-2011-semantik-2011-isbn.html>.
- Wijayaningrum, Vivi N., Dyah P., Jasicka I., Mareta R., Rofika, dan Durrotul F.
Aplikasi Photo Editor Menggunakan Matlab.