DAFTAR LAMPIRAN

Lampiran – 1 *Listing program* Halaman Utama

```
function varargout = utama(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',
mfilename, ...
                   'gui_Singleton',  gui_Singleton,..
                   'gui_OpeningFcn',
@utama_OpeningFcn, ...
                   'gui_OutputFcn',
@utama_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
function utama_OpeningFcn(hObject,
eventdata, handles, varargin)
%Background
background =
axes('unit','normalized','position',[0 0 1 1]);
cover =
imread('latarbelakang.jpg');imagesc(cover);
set(background,'handlevisibility','off','visible'
,'off');
uistack(background,'bottom');
function halamanutama_Callback(hObject,
eventdata, handles)
close
utama
function profil_Callback(hObject, eventdata,
handles)
close
profil
function keluar_Callback(hObject,
eventdata, handles)
clc;
jawaban = keluar('Title','Konfirmasi
Keluar:');
switch jawaban
    case 'Tidak'
% NoAction
    case 'Iya'
        close
end
function
enkripsi_dekripsi_Callback(hObject,
eventdata, handles)
close
enkripsidekripsi
function RE_Callback(hObject, eventdata,
handles)
close
banding
```

Lampiran – 2 *Listing Program* Menu Profil

```
function varargout = profil(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',
mfilename, ...
                   'gui_Singleton',  gui_Singleton,..
                   'gui_OpeningFcn',
@profil_OpeningFcn, ...
                   'gui_OutputFcn',
@profil_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback =
str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
function profil_OpeningFcn(hObject,
eventdata, handles, varargin)
handles.output = hObject;
movegui(hObject,'center');
% Update handles structure
guidata(hObject, handles);
function halamanutama_Callback(hObject,
eventdata, handles)
close
utama
function keluar_Callback(hObject,
eventdata, handles)
clc;
jawaban = keluar('Title','Konfirmasi
Keluar:');

switch jawaban
   case 'Tidak'
      % NoAction
   case 'Iya'
      close
end
function
enkripsi_dekripsi_Callback(hObject,
eventdata, handles)
close
enkripsidekripsi
function RE_Callback(hObject, eventdata,
handles)
close
banding
function logo_CreateFcn(hObject, eventdata,
handles)
imshow('umpar.png');
```

Lampiran – 3 *Listing Program* Proses KSA, PRGA, dan Menu Keluar

```
function [S] = KSA (key)
key = char(key);
key = uint16(key);
key_length = size(key,2);
S=0:255;
j=0;
for i=0:1:255
   j = mod( j + S(i+1) + key(mod(i,
key_length) +1), 256);
   S([i+1 j+1]) = S([j+1 i+1]);
end
end

function [key] = PRGA(S,n)
i = 0;
j = 0;
key = uint16([]);
while n> 0
   n = n-1;
   i = mod(i+1, 256);
   j = mod(j+ S(i+1), 256);
   S([i+1 j+1]) = S([j+1 i+1]);
   K = S( mod( S(i+1) + S(j+1), 256) +1);
   key = [key, K];
end
end


function keluar_OpeningFcn(hObject,
eventdata, handles, varargin)
if(nargin > 3)
   for index = 1:2:(nargin-3),
      if nargin-3==index, break, end
      switch lower(varargin{index})
       case 'title'
        set(hObject, 'Name',
varargin{index+1});
       case 'string'
        set(handles.text1, 'String',
varargin{index+1});
      end
   end
end
% Determine the position of the dialog -
centered on the callback figure
% if available, else, centered on the screen
FigPos=get(0,'DefaultFigurePosition');
OldUnits = get(hObject, 'Units');
set(hObject, 'Units', 'pixels');
OldPos = get(hObject,'Position');
FigWidth = OldPos(3);
FigHeight = OldPos(4);
if isempty(gcbf)
   ScreenUnits=get(0,'Units');
   set(0,'Units','pixels');
   ScreenSize=get(0,'ScreenSize');
   set(0,'Units',ScreenUnits);
   FigPos(1)=1/2*(ScreenSize(3)-FigWidth);
   FigPos(2)=2/3*(ScreenSize(4)-
FigHeight);
else
   GCBFOldUnits = get(gcbf,'Units');
   set(gcbf,'Units','pixels');
   GCBFPos = get(gcbf,'Position');
   set(gcbf,'Units',GCBFOldUnits);
   FigPos(1:2) = [(GCBFPos(1) +
GCBFPos(3) / 2) - FigWidth / 2, ...
                  (GCBFPos(2) + GCBFPos(4) /
2) - FigHeight / 2];
end
FigPos(3:4)=[FigWidth FigHeight];
set(hObject, 'Position', FigPos);
set(hObject, 'Units', OldUnits);
% Show a question icon from
dialogicons.mat - variables questIconData
% and questIconMap
load dialogicons.mat
IconData=questIconData;
questIconMap(256,:) = get(handles.figure1,
'Color');
IconCMap=questIconMap;
Img=image(IconData, 'Parent',
handles.axes1);
set(handles.figure1, 'Colormap', IconCMap);
set(handles.axes1, ...
   'Visible', 'off', ...
   'YDir'   , 'reverse'       , ...
   'XLim'   , get(Img,'XData'), ...
   'YLim'   , get(Img,'YData')  ...
   );
% Make the GUI modal
set(handles.figure1,'WindowStyle','modal')
% UIWAIT makes keluar wait for user
response (see UIRESUME)
uiwait(handles.figure1);
function varargout =
keluar_OutputFcn(hObject, eventdata,
handles)
```

```
% Get default command line output from
handles structure
varargout{1} = handles.output;
% The figure can be deleted now
delete(handles.figure1);
function pushbutton1_Callback(hObject,
eventdata, handles)
% Update handles structure
guidata(hObject, handles);
% Use UIRESUME instead of delete
because the OutputFcn needs
% to get the updated handles structure.
uiresume(handles.figure1);
function pushbutton2_Callback(hObject,
eventdata, handles)
handles.output = get(hObject,'String');
% Update handles structure
guidata(hObject, handles);
% Use UIRESUME instead of delete
because the OutputFcn needs
% to get the updated handles structure.
uiresume(handles.figure1);
function figure1_CloseRequestFcn(hObject,
eventdata, handles)
if isequal(get(hObject, 'waitstatus'),
'waiting')
   % The GUI is still in UIWAIT, us
UIRESUME
   uiresume(hObject);
else
   % The GUI is no longer waiting, just
close it
   delete(hObject);
end
function figure1_KeyPressFcn(hObject,
eventdata, handles)
% Check for "enter" or "escape"
if isequal(get(hObject,'CurrentKey'),'escape')
   % User said no by hitting escape
   handles.output = 'No';

   % Update handles structure
   guidata(hObject, handles);

   uiresume(handles.figure1);
end

if isequal(get(hObject,'CurrentKey'),'return')
   uiresume(handles.figure1);
end
```

Lampiran – 4 *Listring Program* Menu Enkripsi Gabungan

```
function varargout =
enkripsidekripsi(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',
mfilename, ...
                   'gui_Singleton',  gui_Singleton,..
                   'gui_OpeningFcn',
@enkripsidekripsi_OpeningFcn, ...
                   'gui_OutputFcn',
@enkripsidekripsi_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
function buka1_Callback(hObject,
eventdata, handles)
clc;
[namafile,
namapath]=uigetfile({'*.bmp';'*.*'},
'ambilgambar');
ambilgambar=imread([namapath,
namafile]);
[rows, columns, numberOfColorChannels] =
size(ambilgambar);
if rows>300 || columns>300
   msgbox('Ukuran Gambar Terlalu Besar
!','Terjadi Kesalahan','warn');
else
axes(handles.input1);
imshow(ambilgambar);
end
function pixel1_Callback(hObject,
eventdata, handles)
clc;
if isempty(getimage(handles.input1))
   msgbox('Silahkan Masukkan Gambar
!','Terjadi Kesalahan','warn');
else
ambil=getimage(handles.input1);
impixelregion(handles.input1);
end
function enkripsi1_Callback(hObject,
eventdata, handles)
clc;
%Input RC4
key = get(handles.edit1, 'String');
%Input Elgamal
p = 257; %Privasi
g = str2double(get(handles.edit2, 'String'));
x = str2double(get(handles.edit3, 'String'));
k = str2double(get(handles.edit17, 'String'));
%Privasi
if isempty(getimage(handles.input1))
   msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
elseif isempty(get(handles.edit2,
'String'))||isempty(get(handles.edit3,
'String'))||isempty(get(handles.edit1,
'String'))||isempty(get(handles.edit17,
'String'))
   msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
elseif(g>=p-1)
   msgbox('Nilai g Harus Lebih Kecil Dari
256 !','Terjadi Kesalahan','warn');
        set(handles.edit2, 'String','');
elseif (x>=p-1)
   msgbox('Nilai x Harus Lebih Kecil Dari
256 !','Terjadi Kesalahan','warn');
        set(handles.edit3, 'String','');
elseif (x>=g)
   msgbox('Nilai x Harus Lebih Kecil Dari
Nilai g !','Terjadi Kesalahan','warn');
        set(handles.edit3, 'String','');
elseif (k>=g)
   msgbox('Nilai k Harus Lebih Kecil Dari
Nilai g !','Terjadi Kesalahan','warn');
        set(handles.edit17, 'String','');
elseif (k>=x)
   msgbox('Nilai k Harus Lebih Kecil Dari
Nilai x !','Terjadi Kesalahan','warn');
        set(handles.edit17, 'String','');
else
%Ambil Nilai Pixel
plainimage = getimage(handles.input1);
red = plainimage (:,:,1);
green = plainimage (:,:,2);
blue = plainimage (:,:,3);
```

```
%Proses Enkripsi RC4
Z = uint8(PRGA(KSA(key),
size(plainimage, 2)));
P = uint8(char(plainimage));
hasil = bitxor(Z, P);
%Proses Enkripsi Elgamal
plainimage=double(hasil);
y = mod(sym(g).^x,p);
%display(y);
set(handles.edit4, 'String', double(y));
enkripsi = sym(y).^k;
enkripsi1 = mod(sym(enkripsi)*plainimage,
p);
axes(handles.hasil1);
imshow(uint8(enkripsi1));
end
function pixel2_Callback(hObject,
eventdata, handles)
clc;
if isempty(getimage(handles.hasil1))
  msgbox('Silahkan Masukkan Gambar
!','Terjadi Kesalahan','warn');
else
ambil=getimage(handles.hasil1);
impixelregion(handles.hasil1);
end
function simpan1_Callback(hObject,
eventdata, handles)
clc;
K = str2double(get(handles.edit1,'String'));
g = str2double(get(handles.edit2,'String'));
x = str2double(get(handles.edit3,'String'));
k = str2double(get(handles.edit17,'String'));
namafile = get(handles.edit15,'String');
if isempty(getimage(handles.hasil1))
  msgbox('Lakukan Enkripsi !','Terjadi
Kesalahan','warn');
elseif isempty(get(handles.edit15, 'String'));
  msgbox('Lengkapi Nama File Yang
Disimpan !','Terjadi Kesalahan','warn');
else
%Gambar
folderr = 'E:\irfan surya
sulnas\Skripsi\Gambar\';
gabung = strcat(folderr ,namafile, '.bmp');
gabung2 = strcat(namafile, '.txt');
simpangambar = getimage(handles.hasil1);
imwrite(simpangambar, gabung, 'bmp')
%Teks
directory = 'E:\irfan surya
sulnas\Skripsi\Gambar';
simpang = fullfile(directory , gabung2);
KunciPublik = table(['g =';'x =';'k =';'K
='],[g;x;k;K]);
writetable(KunciPublik,simpang);
simpan = msgbox('Berhasil Disimpan
!','Berhasil','help');
      set(handles.edit1, 'String','');
      set(handles.edit2, 'String','');
      set(handles.edit3, 'String','');
      set(handles.edit4, 'String','');
      set(handles.edit17, 'String','');
      set(handles.edit15, 'String','');
      %cla(handles.input1,'reset');
      cla(handles.hasil1,'reset');
end
function dekripsi1_Callback(hObject,
eventdata, handles)
clc;

%Input Elgamal
p = 257; %Privasi
g = str2double(get(handles.edit6, 'String'));
x = str2double(get(handles.edit7, 'String'));
k = str2double(get(handles.edit18, 'String'));
%Privasi
%Input RC4
key = get(handles.edit5, 'String');
if isempty(getimage(handles.input2))
  msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
elseif isempty(get(handles.edit6,
'String'))||isempty(get(handles.edit7,
'String'))||isempty(get(handles.edit5,
'String'))
  msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
else
%Proses Dekripsi Elgamal
y = mod(sym(g).^x,p);
a = mod(sym(g).^k,p);
impers = mod((sym(a).^(p-1-x)),p);
display(impers);
%Ambil Nilai Pixel
plainimage = getimage(handles.input2);
red = plainimage (:,:,1);
green = plainimage (:,:,2);
blue = plainimage (:,:,3);
plainimage=double(plainimage);
hasil = mod((sym(plainimage)*impers),p);
%Proses Dekripsi RC4
hasil1 = double(hasil);
```

```
Z = uint8(PRGA(KSA(key), size(hasil1,
2)));
P = uint8(char(hasil1));
dekripsi = bitxor(Z, P);
axes(handles.hasil2);
imshow(dekripsi);
end
function simpan2_Callback(hObject,
eventdata, handles)
clc;
namafile = get(handles.edit16,'String');
if isempty(getimage(handles.hasil2))
  msgbox('Lakukan Dekripsi !','Terjadi
Kesalahan','warn');
elseif isempty(get(handles.edit16, 'String'));
  msgbox('Lengkapi Nama File Yang
Disimpan !','Terjadi Kesalahan','warn');
else
%Gambar
folderr = 'E:\irfan surya
sulnas\Skripsi\Gambar\';
gabung = strcat(folderr ,namafile, '.bmp');
simpangambar = getimage(handles.hasil2);
imwrite(simpangambar, gabung, 'bmp');
simpan = msgbox('Berhasil Disimpan
!','Berhasil','help');
        set(handles.edit5, 'String','');
        set(handles.edit6, 'String','');
        set(handles.edit7, 'String','');
        set(handles.edit18, 'String','');
        set(handles.edit16, 'String','');
        cla(handles.input2,'reset');
        cla(handles.hasil2,'reset');
        cla(handles.input1,'reset');
end
function hist1_Callback(hObject, eventdata,
handles)
clc;
if isempty(getimage(handles.input1))
  msgbox('Silahkan Masukkan Gambar
!','Terjadi Kesalahan','warn');
else
baca = getimage(handles.input1);
R = baca(:,:,1);
G = baca(:,:,2);
B = baca(:,:,3);
figure('NumberTitle','off','Name','Sebelum
Enkripsi');
[R_data pixel_level] = imhist(R);
[G_data pixel_level] = imhist(G);
[B_data pixel_level] = imhist(B);
bar(pixel_level, R_data, 'r');
hold on;
bar(pixel_level, G_data, 'g');
bar(pixel_level, B_data, 'b');
end
function hist2_Callback(hObject, eventdata,
handles)
clc;
if isempty(getimage(handles.hasil2))
  msgbox('Silahkan Masukkan Gambar
!','Terjadi Kesalahan','warn');
else
baca = getimage(handles.hasil2);
R = baca(:,:,1);
G = baca(:,:,2);
B = baca(:,:,3);
figure('NumberTitle','off','Name','Setelah
Dekripsi');
[R_data pixel_level] = imhist(R);
[G_data pixel_level] = imhist(G);
[B_data pixel_level] = imhist(B);
bar(pixel_level, R_data, 'r');
hold on;
bar(pixel_level, G_data, 'g');
bar(pixel_level, B_data, 'b');
end
```

Lampiran – 5 *Listing Program* Menu Algoritma RC4 dan Elgamal

```
function varargout = banding(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',
mfilename, ...
                   'gui_Singleton',  gui_Singleton,..
                   'gui_OpeningFcn',
@banding_OpeningFcn, ...
                   'gui_OutputFcn',
@banding_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
function buka1_Callback(hObject,
eventdata, handles)
clc;
[namafile,
namapath]=uigetfile({'*.bmp';'*.*'},
'ambilgambar');
ambilgambar=imread([namapath,
namafile]);
[rows, columns, numberOfColorChannels] =
size(ambilgambar);
if rows>300 || columns>300
   msgbox('Ukuran Gambar Terlalu Besar
!','Terjadi Kesalahan','warn');
else
axes(handles.axes5);
imshow(ambilgambar);
end
function pixel1_Callback(hObject,
eventdata, handles)
clc;
if isempty(getimage(handles.axes5))
   msgbox('Silahkan Masukkan Gambar
!','Terjadi Kesalahan','warn');
else
ambil=getimage(handles.axes5);
impixelregion(handles.axes5);
end
function enkripsi2_Callback(hObject,
eventdata, handles)
clc;
%Input
p = 257; %Privasi
g = str2double(get(handles.edit3, 'String'));
x = str2double(get(handles.edit4, 'String'));
k = str2double(get(handles.edit17, 'String'));
%Privasi
if isempty(getimage(handles.axes5))
   msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
elseif isempty(get(handles.edit3,
'String'))||isempty(get(handles.edit4,
'String'))||isempty(get(handles.edit17,
'String'))
   msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
elseif (g>=p-1)
   msgbox('Nilai g Harus Lebih Kecil Dari
256 !','Terjadi Kesalahan','warn');
        set(handles.edit3, 'String','');
elseif (x>=p-1)
   msgbox('Nilai x Harus Lebih Kecil Dari
256 !','Terjadi Kesalahan','warn');
        set(handles.edit4, 'String','');
elseif (x>=g)
   msgbox('Nilai x Harus Lebih Kecil Dari
Nilai g !','Terjadi Kesalahan','warn');
        set(handles.edit4, 'String','');
elseif (k>=g)
   msgbox('Nilai k Harus Lebih Kecil Dari
Nilai g !','Terjadi Kesalahan','warn');
        set(handles.edit17, 'String','');
elseif (k>=x)
   msgbox('Nilai k Harus Lebih Kecil Dari
Nilai x !','Terjadi Kesalahan','warn');
        set(handles.edit17, 'String','');
else
%Proses Enkripsi
y = mod(sym(g).^x,p);
set(handles.edit5, 'String', double(y));
%Proses Ambil Pixel
plainimage = getimage(handles.axes5);
red = plainimage (:,:,1);
green = plainimage (:,:,2);
blue = plainimage (:,:,3);
plainimage=double(plainimage);
```

```
enkripsi = sym(y).^k;
enkripsi1 = mod(sym(enkripsi)*plainimage,
p);
axes(handles.axes7);
imshow(uint8(enkripsi1));
end
function simpan3_Callback(hObject,
eventdata, handles)
clc;
g = str2double(get(handles.edit3,'String'));
x = str2double(get(handles.edit4,'String'));
k = str2double(get(handles.edit17,'String'));
namafile = get(handles.edit15,'String');
if isempty(getimage(handles.axes7))
  msgbox('Lakukan Enkripsi !','Terjadi
Kesalahan','warn');
elseif isempty(get(handles.edit15, 'String'));
  msgbox('Lengkapi Nama File Yang
Disimpan !','Terjadi Kesalahan','warn');
else
%Gambar
folderr = 'E:\irfan surya
sulnas\Skripsi\Gambar\';
gabung = strcat(folderr ,namafile, '.bmp');
gabung2 = strcat(namafile, '.txt');
simpangambar = getimage(handles.axes7);
imwrite(simpangambar, gabung, 'bmp')
%Teks
directory = 'E:\irfan surya
sulnas\Skripsi\Gambar';
simpang = fullfile(directory, gabung2);
KunciPublik = table(['g =';'x =';'k
='],[g;x;k]);
writetable(KunciPublik,simpang);
simpan = msgbox('Berhasil Disimpan
!','Berhasil','help');
        set(handles.edit3, 'String','');
        set(handles.edit4, 'String','');
        set(handles.edit5, 'String','');
        set(handles.edit17, 'String','');
        set(handles.edit15, 'String','');
        %cla(handles.axes5,'reset');
        cla(handles.axes7,'reset');
end
function dekripsi2_Callback(hObject,
eventdata, handles)
clc;
%Input
p = 257; %Privasi
g = str2double(get(handles.edit6, 'String'));
x = str2double(get(handles.edit7, 'String'));
k = str2double(get(handles.edit18, 'String'));
%Privasi
if isempty(getimage(handles.axes8))
  msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
elseif isempty(get(handles.edit6,
'String'))||isempty(get(handles.edit7,
'String'))
  msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
else
%Proses Dekripsi
y = mod(sym(g).^x,p);
a = mod(sym(g).^k,p);
impers = mod(sym(a).^(p-1-x),p);
%Ambil Nilai Pixel
plainimage = getimage(handles.axes8);
red = plainimage (:,:,1);
green = plainimage (:,:,2);
blue = plainimage (:,:,3);
%plainimage=double(plainimage);
dekripsi =
mod((sym(plainimage)*impers),p);
axes(handles.axes9);
imshow(uint8(dekripsi));
end
function enkripsi1_Callback(hObject,
eventdata, handles)
clc;
%Input
key = get(handles.edit1, 'String');
if isempty(getimage(handles.axes1))
  msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
elseif isempty(get(handles.edit1, 'String'))
  msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
else
%Proses Amibl Pixel
plainimage = getimage(handles.axes1);
red = plainimage (:,:,1);
green = plainimage (:,:,2);
blue = plainimage (:,:,3);
%Proses Enkripsi
Z = uint8(PRGA(KSA(key),
size(plainimage, 2)));
P = uint8(char(plainimage));
hasil = bitxor(Z, P);
axes(handles.axes2);
imshow(hasil);
```

```
end
function dekripsi1_Callback(hObject,
eventdata, handles)
clc;
%Input
key = get(handles.edit2, 'String');
if isempty(getimage(handles.axes3))
  msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
elseif isempty(get(handles.edit2, 'String'))
  msgbox('Data Tidak Boleh Kosong
!','Terjadi Kesalahan','warn');
else
%Ambil Nilai Pixel
plainimage = getimage(handles.axes3);
red = plainimage (:,:,1);
green = plainimage (:,:,2);
blue = plainimage (:,:,3);
%Proses Dekripsi
Z = uint8(PRGA(KSA(key),
size(plainimage, 2)));
P = uint8(char(plainimage));
hasil = bitxor(Z, P);
axes(handles.axes4);
imshow(hasil);
end
function hist1_Callback(hObject, eventdata,
handles)
clc;
if isempty(getimage(handles.axes5))
  msgbox('Silahkan Masukkan Gambar
!','Terjadi Kesalahan','warn');
else
baca = getimage(handles.axes5);
R = baca(:,:,1);
G = baca(:,:,2);
B = baca(:,:,3);
figure('NumberTitle','off','Name','Sebelum
Enkripsi');
[R_data pixel_level] = imhist(R);
[G_data pixel_level] = imhist(G);
[B_data pixel_level] = imhist(B);
bar(pixel_level, R_data, 'r');
hold on;
bar(pixel_level, G_data, 'g');
bar(pixel_level, B_data, 'b');
end
function hist2_Callback(hObject, eventdata,
handles)
clc;
if isempty(getimage(handles.axes9))
  msgbox('Silahkan Masukkan Gambar
!','Terjadi Kesalahan','warn');
else
baca = getimage(handles.axes9);
R = baca(:,:,1);
G = baca(:,:,2);
B = baca(:,:,3);
figure('NumberTitle','off','Name','Setelah
Dekripsi');
[R_data pixel_level] = imhist(R);
[G_data pixel_level] = imhist(G);
[B_data pixel_level] = imhist(B);
bar(pixel_level, R_data, 'r');
hold on;
bar(pixel_level, G_data, 'g');
bar(pixel_level, B_data, 'b');
end
```