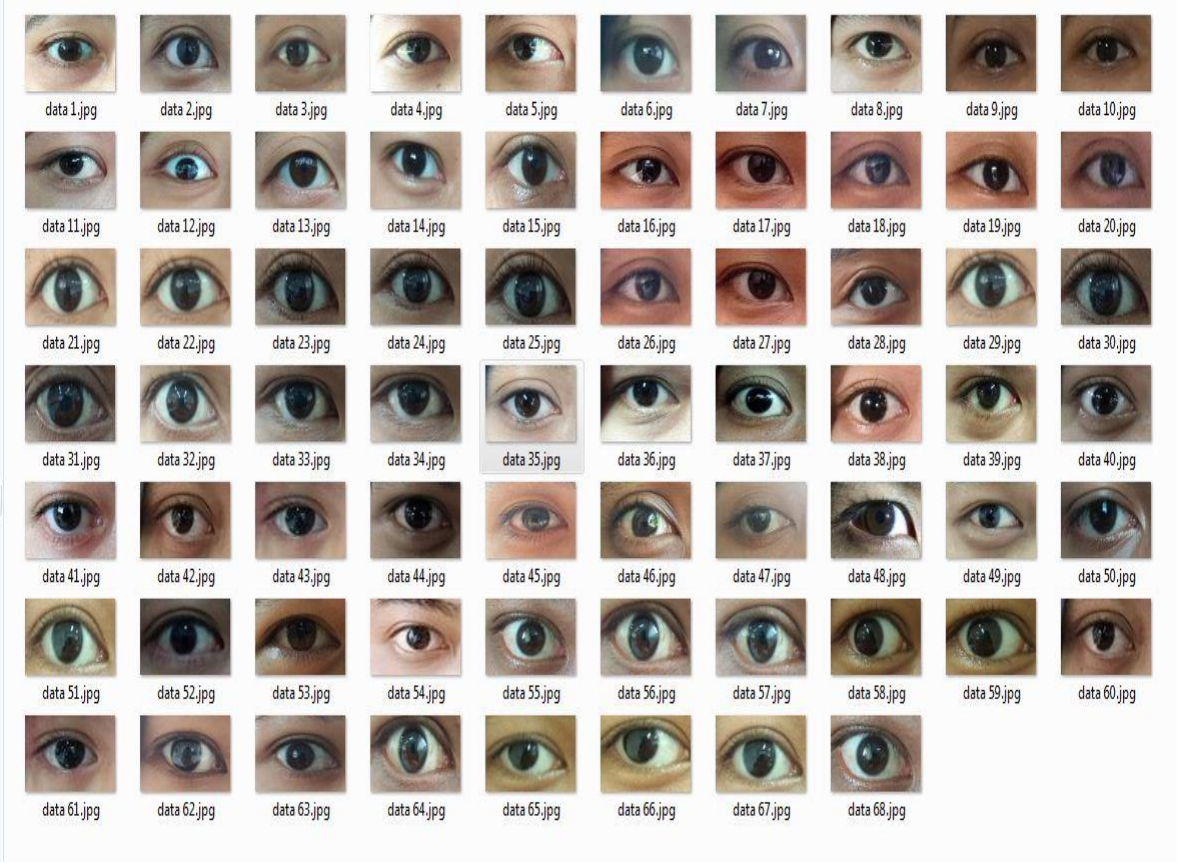
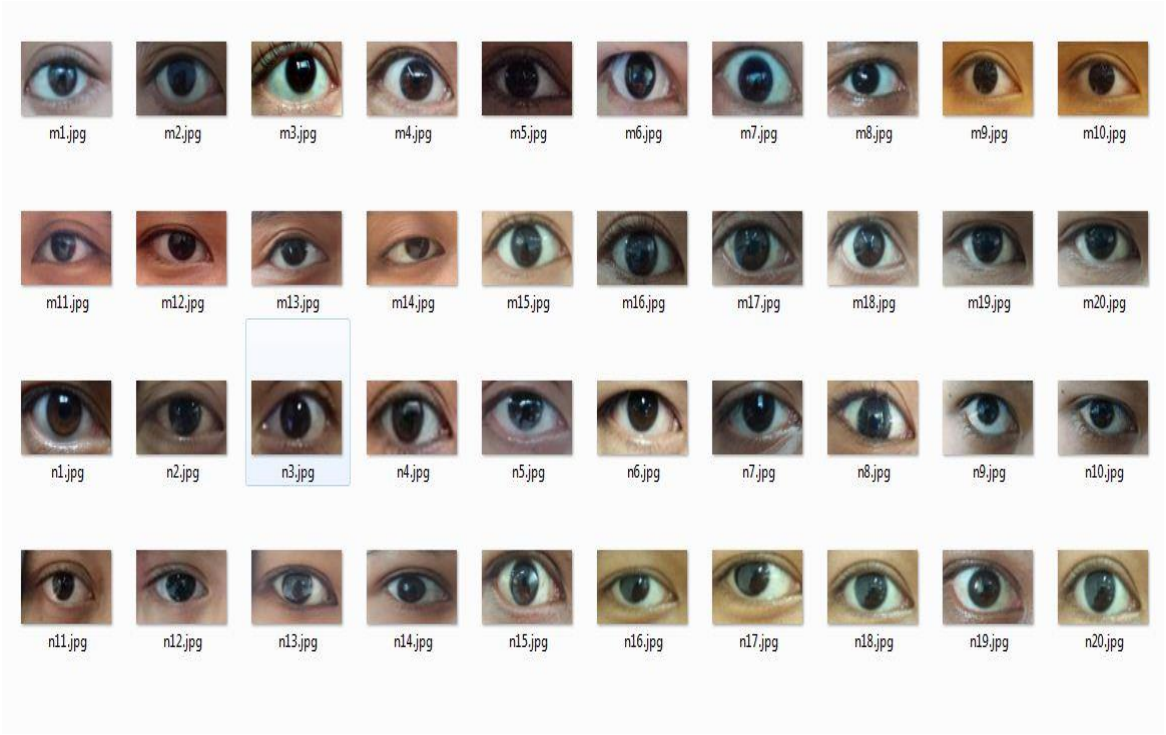
LAMPIRAN

Lampiran 1. Data Citra Latih



Lampiran 2. Data Citra Uji



Lampiran 3. Spesifikasi *Webcam*, Laptop, dan *Software* MATLAB

1. Webcam

   Nama : HP-Webcam 101

   Resolusi : 640 x 480 piksels

2. Laptop

   Merk : HP Compaq Presario CQ43

   Processor :Intel

   Tipe Processor :Intel(R) Celeron (R) CPU B815 @1.60GHz

   Operating System :Windows 7 Ultimate 32 – bit

   RAM : 2,00GB

3. Software MATLAB

   Versi : R2012a

Lampiran 4. Petunjuk Pengoperasian Sistem

1. Menyalakan laptop
2. Membuka software MATLAB R2012a
3. Membuka program HOMEbaru.m dan tunggu sampai tampilan home muncul
4. Menekan tombol “START” untuk memulai dan tunggu sampai tampilan sistem muncul
5. Menakan tombol “Kamera ON” untuk menyalakan kamera
6. Posisikan mata dengan jarak 5 – 10 cm dari webcam dengan mata dalam keadaan terbuka dan pupil terlihat jelas
7. Menakan tombol “Capture” untuk mengambil citra mata yang sudah sesuai dengan posisi
8. Jika ingin menggunakan citra dari folder yang ada di PC, tekan tombol “Buka Citra” lalu pilih citra yang akan digunakan
9. Tunggu hingga citra ditampilkan
10. Menekan tombol “Crop” lalu akan muncul gambar persegi panjang. Ubah ukuran persegi panjang tersebut sesuai ukuran citra yang akan digunakan. Setelah ukuran disesuaikan, klik 2x di dalam persegi panjang, kemudian tunggu hingga citra hasil cropping ditampilkan
11. Menekan tombol “Simpan Hasil Crop” untuk menyimpan citra hasil cropping
12. Menekan tombol “Resize” untuk menyesuaikan ukuran piksel citra menjadi 250 x 144 piksels
13. Menekan tombol “Simpan Hasil Resize” untuk menyimpan citra hasil resizing
14. Menekan tombol “Ekstrasi Ciri” untuk menampilkan nilai ciri yang berupa metric dan eccentricity. Nilai ciri akan ditampilkan pada tabel
15. Menekan tombol “Identifikasi Hasil” untuk mengetahui hasil deteksi sistem. Hasil yang ditampilkan berupa keputusan deteksi
16. Untuk melakukan proses deteksi pada citra lain, menekan tombol “RESET” dan ulangi langkah 5 – 15
17. Tombol “EXIT” untuk keluar dari sistem

Lampiran 5. List Program Pelatihan Jaringan

```
clc; clear; close all;

image_folder = 'latih';
filenames = dir(fullfile(image_folder, '*.jpg'));
total_images = numel(filenames);

area = zeros(1,total_images);
perimeter = zeros(1,total_images);
metric = zeros(1,total_images);
eccentricity = zeros(1,total_images);

for n = 1:total_images
    full_name= fullfile(image_folder, filenames(n).name);
    I = imread(full_name);
    J = I(:,:,1);
    K = im2bw(J,.6);
    L = imcomplement(K);
    str = strel('disk',5);
    M = imclose(L,str);
    N = imfill(M,'holes');
    O = bwareaopen(N,2200);
    stats =
regionprops(O,'Area','Perimeter','Eccentricity');
    area(n) = stats.Area;
    perimeter(n) = stats.Perimeter;
    metric(n) = 4*pi*area(n)/(perimeter(n)^2);
    eccentricity(n) = stats.Eccentricity;
end

input = [metric;eccentricity];
target = zeros(1,68);
target(:,1:34) = 1; % Batasan terendah tingkat miopia yang
digunakan sebesar 0,25D
target(:,35:68) = 0;

net = newff(input,target,[20
10],{'logsig','logsig'},'trainlm');
net.trainParam.epochs = 1000;
net.trainParam.goal = 1e-6;
net = train(net,input,target);
output = round(sim(net,input));
save netlatih

[m,n] = find(output==target);
akurasi = sum(m)/total_images*100
```

Lampiran 6. List Program Pengujian Jaringan

```
rsz = handles.rsz;
J = rsz(:,:,1);
K = im2bw(J,.6);
L = imcomplement(K);
str = strel('disk',5);
M = imclose(L,str);
N = imfill(M,'holes');
O = bwareaopen(N,2200);
stats = regionprops(O,'Area','Perimeter','Eccentricity');
area = stats.Area;
perimeter = stats.Perimeter;
metric = 4*pi*area/(perimeter^2);
eccentricity = stats.Eccentricity;
input = [metric;eccentricity];
load netlatih
output = round(sim(net,input));
if output == 1;
hsl = 'Miopia';
else
hsl = 'Tidak MIopia';
end
set(handles.edit1,'String',hsl);
```

Lampiran 7. List Program GUI Home

```
function varargout = HOMEbaru(varargin)
% HOMEbaru MATLAB code for HOMEbaru.fig
%      HOMEbaru, by itself, creates a new HOMEbaru or raises the
existing
%      singleton*.
%
%      H = HOMEbaru returns the handle to a new HOMEbaru or the
handle to
%      the existing singleton*.
%
%      HOMEbaru('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in HOMEbaru.M with the given input
arguments.
%
%      HOMEbaru('Property','Value',...) creates a new HOMEbaru or
raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before HOMEbaru_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to HOMEbaru_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help HOMEbaru

% Last Modified by GUIDE v2.5 28-Apr-2012 22:01:06

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @HOMEbaru_OpeningFcn, ...
                   'gui_OutputFcn',  @HOMEbaru_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

% --- Executes just before HOMEbaru is made visible.
function HOMEbaru_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to HOMEbaru (see
VARARGIN)
% Choose default command line output for HOMEbaru
%axes(handles.axes1)
%a = imread('uny.jpg');
%imshow(a);
handles.output = hObject;
axes(handles.axes1)
a = imread('uny.jpg');
imshow(a);
gifPlayer_zgp(handles, 'HOMEbaru');
%axes(handles.axes1)
%gifPlayer_zgp(handles, 'HOMEbaru');
% Update handles structure
guidata(hObject, handles);

function gifPlayer_zgp(handles, fname)
    %# read all GIF frames
    axes(handles.axes2);
    info = imfinfo('mata.gif');%'mata.gif', 'Logo_uny.gif');
    delay = ( info(1).DelayTime ) / 100;
    [img,map] = imread('mata.gif', 'frames','all');
    [imgH,imgW,~,numFrames] = size(img);
       %axes(handles.axes2);
      hImg = imshow(img(:,:,:,1), map);
      pause(delay)
  %
      %# loop over frames continuously
      counter = 1;
      while ishandle(hImg)
  %         # increment counter circularly
          counter = rem(counter, numFrames) + 1;
%         # update frame
        set(hImg, 'CData',img(:,:,:,counter))
%         # update colormap
        n = max(max( img(:,:,:,counter) ));
        colormap( info(counter).ColorTable(1:n,:) );
%         # pause for the specified delay
        pause(delay)
      end
%b = ycbcr2rgb(a);
%imshow(a);
%hback=axes('units','normalized','position',[0 0 1 1]);
%uistack(hback,'bottom');
%[back map]=imread('normal4.jpg');
```

```
      %image(back)
      %colormap(map)
%background=imread('normal4.jpg');
%set(hback,'handlevisibility','off','visible','off')
%guidata(hObject,handles);
% UIWAIT makes HOMEbaru wait for user response (see UIRESUME)
%uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = HOMEbaru_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;


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

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (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,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on slider movement.
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


% --- Executes during object creation, after setting all
properties.
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (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,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of
edit3 as a double

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (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,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of
edit5 as a double

% --- Executes during object creation, after setting all
properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(~, eventdata, handles)
% hObject    handle to edit6 (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,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of
edit6 as a double

% --- Executes during object creation, after setting all
properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function text2_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to text2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete(handles.figure1)

% --- Executes during object creation, after setting all
properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: place code in OpeningFcn to populate axes2
```

Lampiran 8. List Program Sistem

```
function varargout = coba(varargin)
% COBA MATLAB code for coba.fig
%      COBA, by itself, creates a new COBA or raises the existing
%      singleton*.
%
%      H = COBA returns the handle to a new COBA or the handle to
%      the existing singleton*.
%
%      COBA('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in COBA.M with the given input
arguments.
%
%      COBA('Property','Value',...) creates a new COBA or raises
the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before coba_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to coba_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help coba

% Last Modified by GUIDE v2.5 21-May-2012 05:58:23

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @coba_OpeningFcn, ...
                   'gui_OutputFcn',  @coba_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
```

```
% --- Executes just before coba is made visible.
function coba_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to coba (see VARARGIN)

% Choose default command line output for coba
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
movegui(hObject,'center');
clc;clear;

% UIWAIT makes coba wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = coba_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
%axes(handles.axes1)
%[FileName, PathName] = uigetfile('*.jpg','buka');
%Img = imread([PathName,FileName]);
%axes(handles.axes1)
%imshow(Img)
%handles.Img = Img;
%guidata(hObject, handles)
global VidObj
VidObj = videoinput('winvideo',1,'YUY2_640x480');
handles.VidObj = VidObj;
axes(handles.axes1)
%hImage= image(zeros(vidRes(2),vidRes(1),nBands));
hImage= image(zeros(300,300,3),'Parent',handles.axes1); %tampilan
capture dan ukuran
preview(VidObj,hImage);

% --- 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)
%Img = handles.Img;
%[m,n,~] = size(Img);
gambarRGB = handles.gambarRGB;
[m,n,~] = size(gambarRGB);
axes(handles.axes2)
imshow(gambarRGB);
h = imrect(gca,[n/2 m/2 0.2*n 0.2*m]);%membentuk persegi panjang
wait(h);
mask = createMask(h);
[B,~] = bwboundaries(mask,'noholes');
k = 1:length(B)
boundary = B{k};
axes(handles.axes1)
imshow(gambarRGB);
hold on
plot(boundary(:,2), boundary(:,1), 'b', 'LineWidth', 2)
hold off
    R = gambarRGB(:,:,1);
    G = gambarRGB(:,:,2);
    B = gambarRGB(:,:,3);
    R(~mask) = 0;
    G(~mask) = 0;
    B(~mask) = 0;
    RGB = cat(3,R,G,B);
    [row,col] = find(mask==1);
    gbrcrop = imcrop(RGB,[min(col) min(row) max(col)-min(col)
max(row)-min(row)]);%memotong citra
    handles.gbrcrop = gbrcrop;
    %save asli.jpg
    axes(handles.axes2)
    imshow(gbrcrop);
    guidata(hObject, handles);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
%global asli
%asli = handles.asli;
%imshow(asli);
Img = handles.Img;
%b = handles.b;
gray = rgb2gray(Img);
%gray = (asli(:,:,1)+asli(:,:,2)+asli(:,:,3))/3;
axes(handles.axes3)
imshow(gray);
handles.gray = gray;
guidata(hObject, handles

% --- 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)
Img = handles.Img;
biner = im2bw(Img);
axes(handles.axes5)
imshow(biner);
%handles.thresh = thresh;
handles.biner = biner;
guidata(hObject, handles);

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
%asli = handles.asli;
gray = handles.gray;
segmen = edge(gray,'canny');
axes(handles.axes6)
imshow(segmen);
handles.segmen = segmen;
guidata(hObject, handles);

% --- 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)
segmen = handles.segmen;
[name_file_save,path_save] = uiputfile({'*.jpg'}, 'Save Image');
if ~isequal (name_file_save, 0)
    imwrite (segmen,fullfile(path_save, name_file_save));
else
    return
end

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
rsz = handles.rsz;
J = rsz(:,:,1);
K = im2bw(J,.6);
L = imcomplement(K);
str = strel('disk',5);
M = imclose(L,str);
N = imfill(M,'holes');
O = bwareaopen(N,2200);
stats = regionprops(O,'Area','Perimeter','Eccentricity');
```

```
area = stats.Area;
perimeter = stats.Perimeter;
metric = 4*pi*area/(perimeter^2);
eccentricity = stats.Eccentricity;
input = [metric;eccentricity];
load netlatih
output = round(sim(net,input));
if output == 1;
hsl = 'Miopia';
else
hsl = 'Tidak MIopia';
end
set(handles.edit1,'String',hsl);

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (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,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton11 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
gray = handles.gray;
[name_file_save,path_save] = uiputfile({'*.jpg'}, 'Save Image');
if ~isequal (name_file_save, 0)
    imwrite (gray,fullfile(path_save, name_file_save));
else
    return
end

% --- Executes on button press in pushbutton12.
```

```
function pushbutton12_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton12 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
biner = handles.biner;
[name_file_save,path_save] = uiputfile({'*.jpg'}, 'Save Image');
if ~isequal (name_file_save, 0)
    imwrite (biner,fullfile(path_save, name_file_save));
else
    return
end

% --- Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton13 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
segmen = handles.segmen;
[name_file_save,path_save] = uiputfile({'*.jpg'}, 'Save Image');
if ~isequal (name_file_save, 0)
    imwrite (segmen,fullfile(path_save, name_file_save));
else
    return
end

% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton14 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
gbrcrop = handles.gbrcrop;
[name_file_save,path_save] = uiputfile({'*.jpg'}, 'Save Image');
if ~isequal (name_file_save, 0)
    imwrite (gbrcrop,fullfile(path_save,
name_file_save));%menyimpan data pada folder
else
    return
end

% --- Executes on button press in pushbutton15.
function pushbutton15_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton15 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

axes(handles.axes1)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])

axes(handles.axes2)
```

```
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])

axes(handles.axes7)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])

%set(handles.text2,'String',[])
set(handles.uitable1,'Data',[])
set(handles.edit1,'String',[])

% --- Executes on button press in pushbutton16.
function pushbutton16_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton16 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
%asli = handles.asli;
gbrcrop = handles.gbrcrop;
rsz = imresize(gbrcrop,[144 255]);%resize dengan ukuran 255 x 144
axes(handles.axes7)
imshow(rsz);
handles.rsz = rsz;
guidata(hObject, handles);

% --- Executes on button press in pushbutton17.
function pushbutton17_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton17 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
rsz = handles.rsz;
[name_file_save,path_save] = uiputfile({'*.jpg'}, 'Save Image');
if ~isequal (name_file_save, 0)
    imwrite (rsz,fullfile(path_save, name_file_save));
else
    return
end

% --- Executes on button press in pushbutton18.
function pushbutton18_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton18 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
%selection = questdlg(['Are you sure you want to
exit?'],['Yes','No']);
%if strcmp(selection,'No' && 'cancel')
 %   return;
%end
delete(handles.figure1);

% --- Executes on button press in pushbutton19.
```

```
function pushbutton19_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton19 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
%rsz = handles.rsz;
gambarRGB = handles.gambarRGB;
J = gambarRGB(:,:,1);
    K = im2bw(J,.6);
    L = imcomplement(K);
    str = strel('disk',5);
    M = imclose(L,str);
    N = imfill(M,'holes');
    O = bwareaopen(N,2200);
stats = regionprops(O,'Area','Perimeter','Eccentricity');
area = stats.Area;
perimeter = stats.Perimeter;
metric = 4*pi*area/(perimeter^2);
eccentricity = stats.Eccentricity;

ciri = cell(2,2);
ciri{1,1} = 'Metric';
ciri{2,1} = 'Eccentricity';
ciri{1,2} = num2str(metric);
ciri{2,2} = num2str(eccentricity);

handles.ciri = ciri;
guidata(hObject, handles)

%set(handles.text2,'String','Ekstraksi Ciri')
set(handles.uitable1,'Data',ciri,'RowName',1:2)

% ------------------------------------------------------------------
----
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton20.
function pushbutton20_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton20 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
global VidObj
gambar = getsnapshot(VidObj);%capture image
gambarRGB = ycbcr2rgb(gambar);%konversi citra ke RGB
imshow(gambarRGB);
handles.gambarRGB = gambarRGB;
axes(handles.axes1)
guidata(hObject, handles);

% --- Executes on button press in pushbutton21.
```

```
function pushbutton21_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton21 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename,pathname] = uigetfile('*.jpg');%membuka file

if ~isequal(filename,0)
    gambarRGB= imread(fullfile(pathname,filename));
    axes(handles.axes1)
    imshow(gambarRGB)
    title('Citra RGB')
else
    return
end

handles.gambarRGB = gambarRGB;
guidata(hObject, handles);

% --- Executes when entered data in editable cell(s) in uitable1.
function uitable1_CellEditCallback(hObject, eventdata, handles)
% hObject    handle to uitable1 (see GCBO)
% eventdata  structure with the following fields (see UITABLE)
%   Indices: row and column indices of the cell(s) edited
%   PreviousData: previous data for the cell(s) edited
%   EditData: string(s) entered by the user
%   NewData: EditData or its converted form set on the Data
property. Empty if Data was not changed
%   Error: error string when failed to convert EditData to
appropriate value for Data
% handles    structure with handles and user data (see GUIDATA)
```