Program Matlab untuk Ekstraksi Ciri Citra Daun Teh dengan Beberapa Metode

Jenis Karya:
PROGRAM KOMPUTER

Author:
Alma & Dr. Bambang Heru Iswanto, M.Si

PROGRAM STUDI FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI JAKARTA
JULI 2020

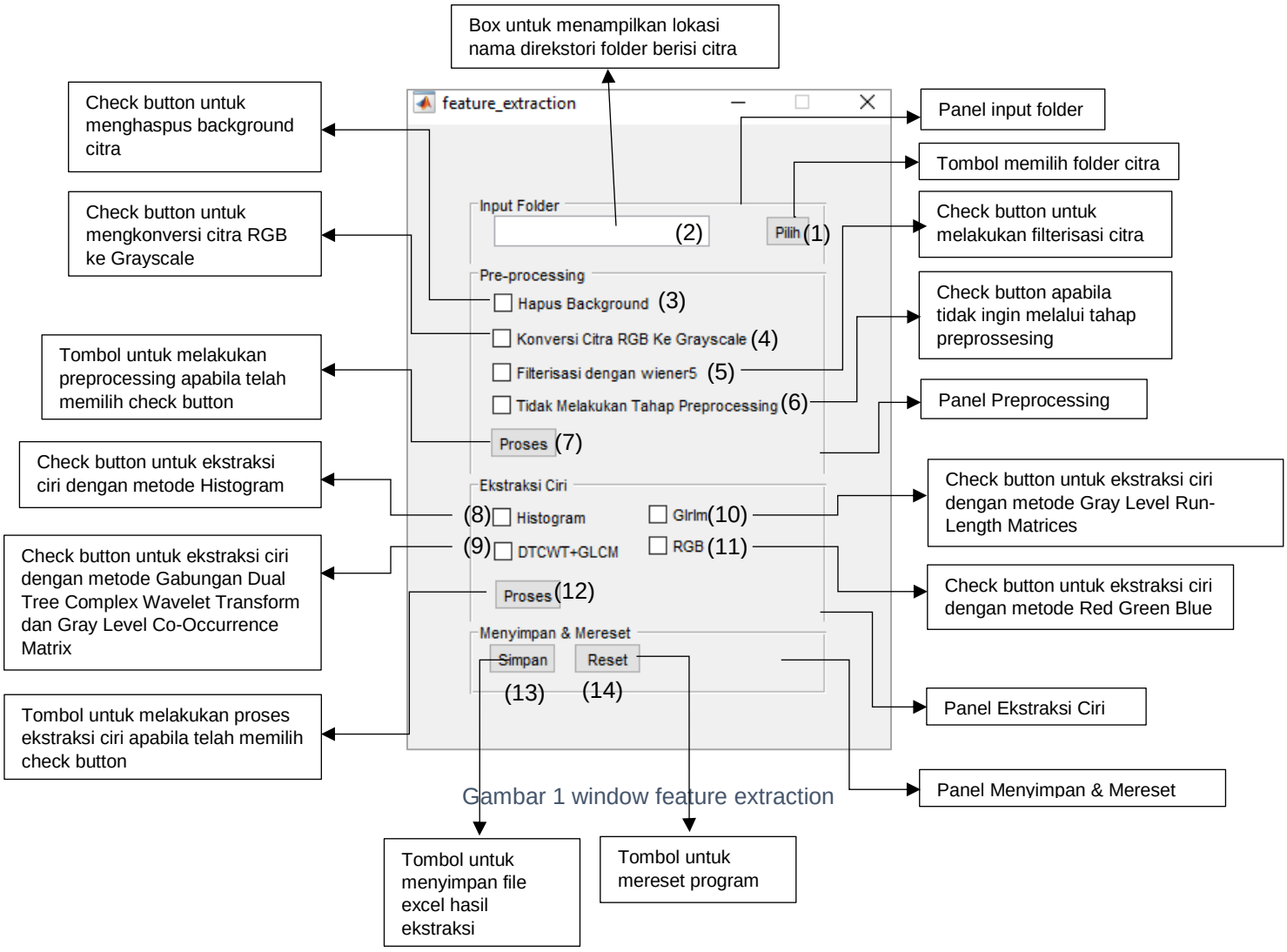
DAFTAR ISI

DAFTAR GAMBAR

I. GRAPHIC USER INTERFACE (TAB I/feature_extraction)

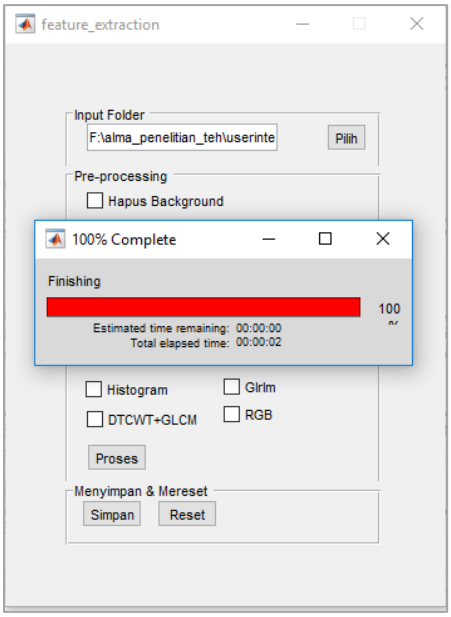
A. Deskripsi Program

Program menampilkan graphical user interface (GUI) untuk melakukan ekstraksi ciri citra daun teh sebagai mana ditampilkan pada gambar dibawah ini:



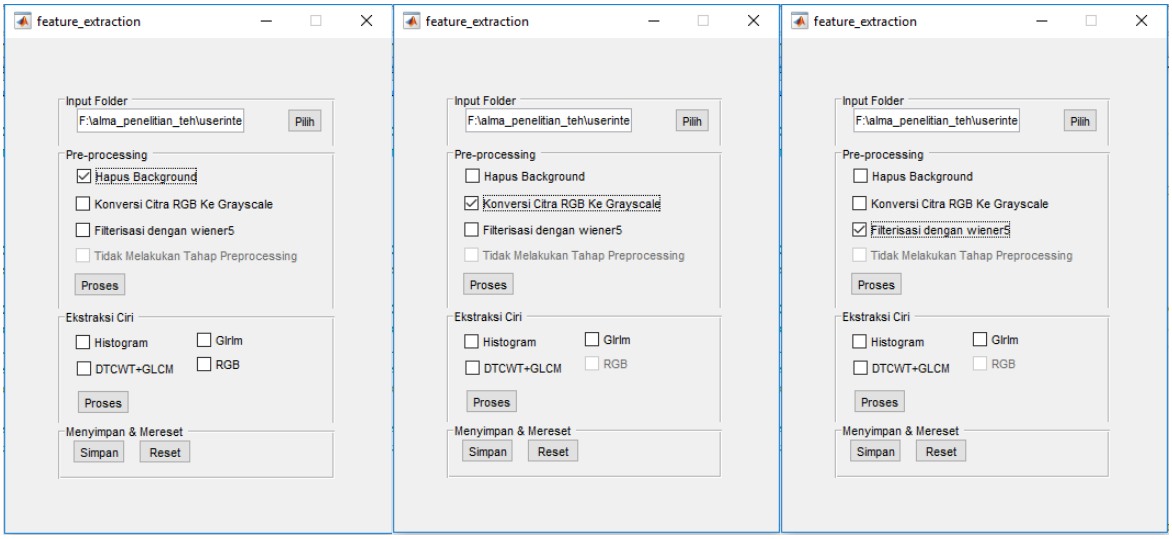
Gambar 1 window feature extraction

Program ini dijalankan dengan cara menginput folder yang berisi citra berformat **.jpg* dengan menekan tombol pilih (dengan label 1), kemudian direktori folder akan ditampilkan pada text box (dengan label 2) proses akan selesai apabila tampil popup bar untuk menunggu hingga proses selesai sebagai mana ditampilkan pada gambar dibawah ini:

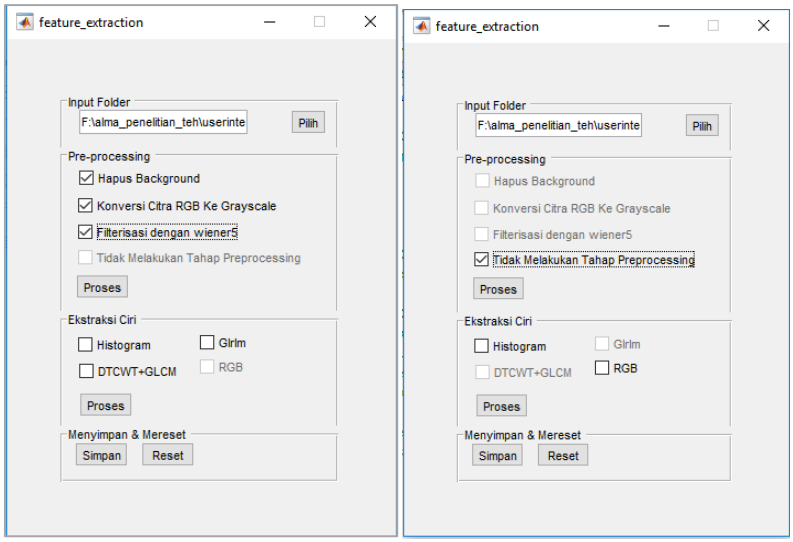


Gambar 2 window feature extraction untuk menginput folder

Setelah menunggu hingga pesan popup bar selesai ('*finishing*'), dilanjutkan tahap *PreProcessing* dengan memilih bebrapa check button (berlabel (3), (4), (5), dan (6)), namun perlu diketahui apabila menekan check button (berlabel (6)) maka check button (berlabel (3), (4), dan (5)) akan ter-*disable* (tidak dapat dipilih), begitu juga sebaliknya apabila menekan check button (berlabel (3), (4), dan (5)) maka check button (berlabel (6)) yang akan akan ter-*disable* (tidak dapat dipilih) sebagaimana ditampilkan pada gambar dibawah ini



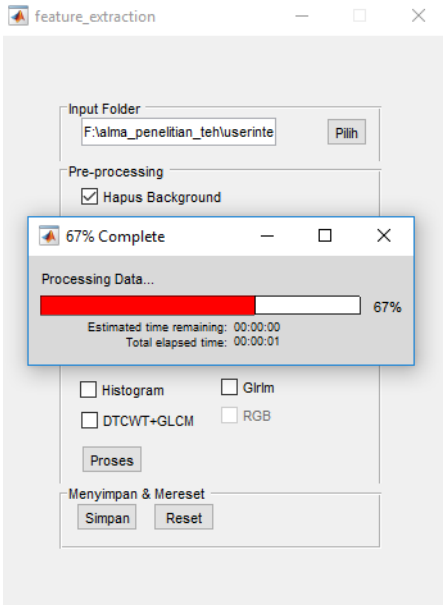
(a) (b) (c)



(d) (e)

Gambar 3 window feature extraction (a) saat memilih hanya menghapus background, (b) saat memilih hanya mengkonversi citra RGB ke citra Grayscale, (c) saat memilih hanya melakukan filterisasi , (d) saat memilih tiga checkbox secara bersamaan, dan (e) saat tidak ingin melalui tahap *preprocessing*.

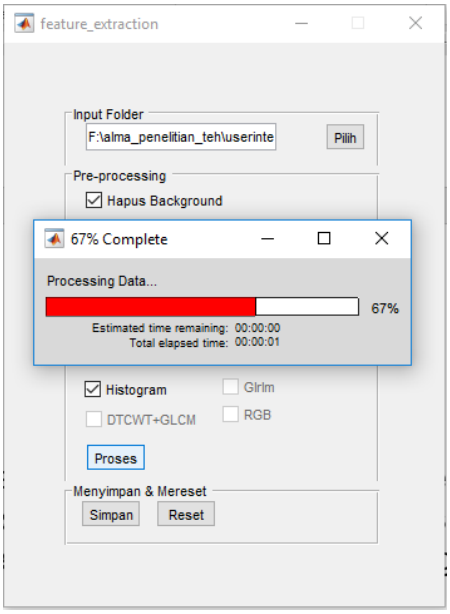
Pada bagian *preprocessing* dapat memilih hapus background, konversi citra RGB ke Grayscale, filterisasi, atau memilih ketiganya secara bersamaan, namun apabila tidak ingin melalui tahap *preprocessing* sebagaimana pada gambar diatas. Selanjutnya menekan button '*Proses*' hingga tampil popup bar menunggu hingga proses selesai, sebagaimana ditampilkan pada gambar dibawah ini



Gambar 4 window memproses tahap *preprocessing*

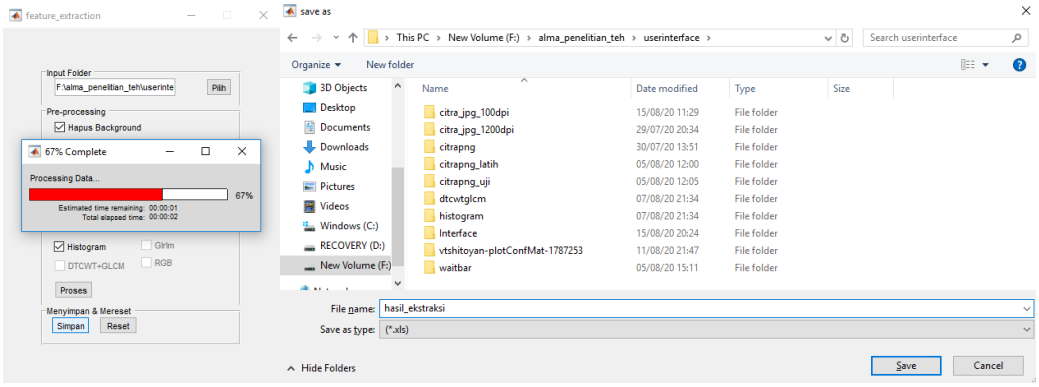
Setelah tahap *preprocessing* selesai barulah ke tahap ekstraksi ciri dengan menekan check button '*Histogram*' apabila ingin menggunakan metode histogram, menekan check button '*DTCWT+GLCM*' apabila ingin menggunakan metode gabungan dual tree complex wavelet transform dan gray level co-occurrence matriks, menekan check button '*Glrlm*' apabila inggin menggunakan metode gray level run length matrices, dan menekan check button '*RGB*' apabila ingin menggunakan metode red green blue. Pada tahap ekstraksi ciri hanya dapat memilih satu check button, dan perlu diketahui untuk metode histogram citra dapat langsung diekstraksi ciri ataupun melalui proses konversi citra RGB ke Grayscale namun untuk metode gabungan dual tree complex wavelet transform dan gray level co-occurrence matriks serta metode gray level run length matrices perlu mengkonversi citra RGB ke citra Grayscale,

dan untuk penggunaan metode red green blue tidak boleh mengkonversi citra RGB ke citra Grayscale, kemudian barulah menekan button '*Proses*' untuk memulai proses ekstraksi ciri hingga tampil popup bar menunggu hingga proses selesai sebagaimana ditampilkan pada gambar dibawah ini:



Gambar 5 window saat memproses ekstraksi ciri

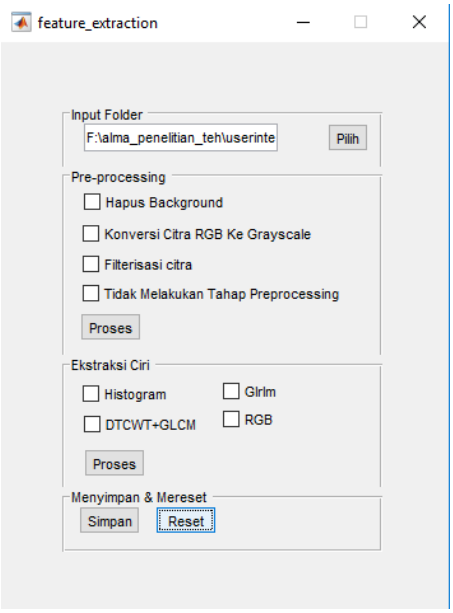
Setelah proses selesai maka hasil ekstraksi dapat disimpan kedalam format *.*xls* dengan nama file sesuai keinginan pengguna dengan menekan button '*simpan*' sebagaimana pada gambar dibawah ini:



(a) (b)

Gambar 6 window (a) saat menekan button '*simpan*' (b) saat menamai file *.*xls*

Setelah hasil disimpan pengguna dapat mereset program dengan menekan button '*reset*' sehingga sebagaimana ditampilkan pada gambar dibawah ini:



Gambar 7 window setelah mereset program

B. Code Program

Code program dibuat dengan bahasa script Matlab yang terdiri dari beberapa file program sebagai berikut.

halamanutama.m

Nama File	:	Feature_extraction.m
Deskripsi	:	Menciptakan GUI pada software Matlab yang terdiri dari lima botton diantaranya button untuk memilih folder citra yang berisi citra berformat *.*jpg*,button untuk tahap *preprocessing*, button untuk mengekstraksi ciri, button untuk menyimpan file hasil ekstraksi ciri berformat *.*xls*, dan button untuk mereset program. Satu kolom nama lokasi direktori folder citra, empat check button untuk tahap *preprocessing*, dan empat check button untuk tahap ekstraksi ciri.
Platform	:	Matlab
Copyright	:	© 2020 Alma, Dr. Bambang Heru Iswanto

```
function varargout = feature_extraction(varargin)
% FEATURE_EXTRACTION MATLAB code for feature_extraction.fig
%      FEATURE_EXTRACTION, by itself, creates a new FEATURE_EXTRACTION or raises the
existing
%      singleton*.
%
%      H = FEATURE_EXTRACTION returns the handle to a new FEATURE_EXTRACTION or the
handle to
%      the existing singleton*.
%
%      FEATURE_EXTRACTION('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in FEATURE_EXTRACTION.M with the given input
arguments.
%
```

```
%      FEATURE_EXTRACTION('Property','Value',...) creates a new FEATURE_EXTRACTION or
raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before feature_extraction_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to feature_extraction_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 feature_extraction

% Last Modified by GUIDE v2.5 15-Aug-2020 12:38:21

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @feature_extraction_OpeningFcn, ...
                   'gui_OutputFcn',  @feature_extraction_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 feature_extraction is made visible.
function feature_extraction_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 feature_extraction (see VARARGIN)

% Choose default command line output for feature_extraction
handles.output = hObject;
movegui('center');
% Update handles structure
guidata(hObject, handles);

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


% --- Outputs from this function are returned to the command line.
function varargout = feature_extraction_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)

folder_data= uigetdir('*.*')
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.edit1,'String',folder_data);
handles.folder_data = folder_data;
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
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)
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
data = handles.data;
%% Menginisiasi Save as
[filenames, pathname] = uiputfile('*.xls','save as');
%% Data Ke .xls
xlswrite(filenames, data, 1);
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject, handles)

% --- 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)
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.checkbox1,'enable','on','Value',0);
set(handles.checkbox2,'enable','on','Value',0);
set(handles.checkbox3,'enable','on','Value',0);
set(handles.checkbox4,'enable','on','Value',0);
set(handles.checkbox5,'enable','on','Value',0);
set(handles.checkbox6,'enable','on','Value',0);
set(handles.checkbox7,'enable','on','Value',0);
set(handles.checkbox8,'enable','on','Value',0);
%set(handles.edit1,'String', '');

waitbar(1,waktu,'Finishing');
```

```
pause(1)
close(waktu);
guidata(hObject, handles);

% --- Executes on button press in checkbox5.
function checkbox5_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox5 (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 checkbox5
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.checkbox5,'enable','on');
set(handles.checkbox6,'enable','off');
set(handles.checkbox7,'enable','off');
set(handles.checkbox8,'enable','off');
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject,handles);

% --- Executes on button press in checkbox6.
function checkbox6_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox6 (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 checkbox6
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.checkbox5,'enable','off');
set(handles.checkbox6,'enable','on');
set(handles.checkbox7,'enable','off');
set(handles.checkbox8,'enable','off');
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject,handles);

% --- Executes on button press in checkbox7.
function checkbox7_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox7 (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 checkbox7
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.checkbox5,'enable','off');
set(handles.checkbox6,'enable','off');
set(handles.checkbox7,'enable','on');
```

```
set(handles.checkbox8,'enable','off');
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject,handles);

% --- Executes on button press in checkbox8.
function checkbox8_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox8 (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 checkbox8
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.checkbox5,'enable','off');
set(handles.checkbox6,'enable','off');
set(handles.checkbox7,'enable','off');
set(handles.checkbox8,'enable','on');
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject,handles);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)

new_folder = handles.new_folder
file = dir(fullfile(new_folder, '*.jpg'))
jumlah= numel(file);

val5 = get(handles.checkbox5,'Value');
val6 = get(handles.checkbox6,'Value');
val7 = get(handles.checkbox7,'Value');
val8 = get(handles.checkbox8,'Value');
if val5
    set(handles.checkbox5,'enable','on');
    for k = 1:jumlah
        full= fullfile(new_folder, file(k).name);
        Img = imread(full);
        H = imhist(Img)';
        H = H/sum(H);
        I = [0:255]/255;
        [P G] = imhist(Img);
        nm = sum(P);
        f_energy = sum((P/nm).^2);
        f_mean = I*H';
        f_entrop = -H*log2(H+eps)';
        f_variance = (I-f_mean).^2*H';
        f_skewness = (I-f_mean).^3*H'/f_variance^1.5;
        f_kurtosis = (I-f_mean).^4*H'/f_variance^2-3;
```

```
        data{k,1} = f_energy;
        data{k,2} = f_mean;
        data{k,3} = f_entrop;
        data{k,4} = f_variance;
        data{k,5} = f_skewness;
        data{k,6} = f_kurtosis;
        %folder_xx= strcat(new_folder,'\original\',file(k).name);
        %imwrite(Img,folder_xx);
    end
elseif val6
    set(handles.checkbox6,'enable','on');
    for k = 1:jumlah
        full= fullfile(new_folder, file(k).name);
        Img = imread(full);
        level = 1;
        biort = 'near_sym_b';
        pixel_dist = 1;
        [C,S] = dtwavedec2(Img,level,biort,'qshift_a');
        Img = dtwaverec2(C,S,biort,'qshift_a');
        Img = uint8(Img);
        %% Analisis Tekstur
        glcm_dtcwt = graycomatrix(Img,'Offset',[0 pixel_dist; -pixel_dist pixel_dist;
-pixel_dist 0; -pixel_dist -pixel_dist]);
        stats_dtcwt_glcm =
features_glcm(glcm_dtcwt,0);%{'energ','entro','dissi','contr','corrm', 'homom',
'autoc'})

        autocorrelation = stats_dtcwt_glcm.autoc;
        contrast = stats_dtcwt_glcm.contr;
        correlation = stats_dtcwt_glcm.corrm;
        clusterprominence = stats_dtcwt_glcm.cprom;
        clustershade = stats_dtcwt_glcm.cshad;

        dissimilarity = stats_dtcwt_glcm.dissi;
        energy = stats_dtcwt_glcm.energ;
        entropy = stats_dtcwt_glcm.entro;
        homogeneity = stats_dtcwt_glcm.homop;
        maximumprobability = stats_dtcwt_glcm.maxpr;

        sumofsquares = stats_dtcwt_glcm.sosvh;
        sumaverage = stats_dtcwt_glcm.savgh;
        sumvariance = stats_dtcwt_glcm.svarh;
        sumentropy =  stats_dtcwt_glcm.senth;
        differencevariance = stats_dtcwt_glcm.dvarh;

        differenceentropy = stats_dtcwt_glcm.denth;
        informationmeasureofcorrelation1 = stats_dtcwt_glcm.inf1h;
        informationmeasureofcorrelation2 = stats_dtcwt_glcm.inf2h;
        inversedifference = stats_dtcwt_glcm.homom;
        inversedifferencenormalized = stats_dtcwt_glcm.indnc;

        inversedifferencemomentnormalized = stats_dtcwt_glcm.idmnc;

        % Pembentukan data latih
        data{k,1} = mean(autocorrelation);
        data{k,2} = mean(contrast);
        data{k,3} = mean(correlation);
        data{k,4} = mean(clusterprominence);
        data{k,5} = mean(clustershade);
        data{k,6} = mean(dissimilarity);
        data{k,7} = mean(energy);
        data{k,8} = mean(entropy);
        data{k,9} = mean(homogeneity );
        data{k,10} = mean(maximumprobability);
```

```
        data{k,11} = mean(sumofsquares);
        data{k,12} = mean(sumaverage);
        data{k,13} = mean(sumvariance);
        data{k,14} = mean(sumentropy);
        data{k,15} = mean(differencevariance);
        data{k,16} = mean(differenceentropy);
        data{k,17} = mean(informationmeasureofcorrelation1);
        data{k,18} = mean(informationmeasureofcorrelation2);
        data{k,19} = mean(inversedifference);
        data{k,20} = mean(inversedifferencenormalized);
        data{k,21} = mean(inversedifferencemomentnormalized);
        %folder_xx= strcat(new_folder,'\original\',file(k).name);
        %imwrite(Img,folder_xx);
    end
elseif val7
    set(handles.checkbox7,'enable','on');
    for k = 1:jumlah
        full= fullfile(new_folder, file(k).name);
        Img = imread(full);
        [glrlm,si] = grayrlmatrix(Img,'NumLevels',5,'G',[]);
        stats_glrlm = grayrlprops(glrlm);
        sre = mean(stats_glrlm(:,1));
        lre = mean(stats_glrlm(:,2));
        gln = mean(stats_glrlm(:,3));
        rln = mean(stats_glrlm(:,4));
        rp = mean(stats_glrlm(:,5));
        lgre = mean(stats_glrlm(:,6));
        hgre = mean(stats_glrlm(:,7));
        sglge = mean(stats_glrlm(:,8));
        srhge = mean(stats_glrlm(:,9));
        lrlge = mean(stats_glrlm(:,10));
        lrhge = mean(stats_glrlm(:,11));

        data{k,1} = sre;
        data{k,2} = lre;
        data{k,3} = gln;
        data{k,4} = rln;
        data{k,5} = rp;
        data{k,6} = lgre;
        data{k,7} = hgre;
        data{k,8} = sglge;
        data{k,9} = srhge;
        data{k,10} = lrlge;
        data{k,11} = lrhge;
        %folder_xx= strcat(new_folder,'\original\',file(k).name);
        %imwrite(Img,folder_xx);
    end
elseif val8
    set(handles.checkbox8,'enable','on');
    for k = 1:jumlah
        full= fullfile(new_folder, file(k).name);
        Img = imread(full);
        %Img =
imread('F:\alma_penelitian_teh\userinterface\citra_jpg_100dpi\hapus_background\1_1.jp
g');
        Img_hsv = rgb2hsv(Img);
        H = Img_hsv(:,:,1);
        S = Img_hsv(:,:,2);
        V = Img_hsv(:,:,3);
        bw = im2bw(S,.25);
        bw = bwareaopen(bw,50);
        R = Img(:,:,1);
        G = Img(:,:,2);
        B = Img(:,:,3);
```

```
        R(~bw)=0;
        G(~bw)=0;
        B(~bw)=0;
        RGB = cat(3,R,G,B);
        [a,b]=find(bw==1);
        Red = 0;
        Green = 0;
        Blue = 0;
        for m = 1:numel(a);
            Red = Red + double(R(a(m),b(m)));
            Green = Green + double(G(a(m),b(m)));
            Blue = Blue + double(B(a(m),b(m)));
        end
        Red = Red/numel(a);
        Green = Green/numel(a);
        Blue = Blue/numel(a);
        %folder_xx= strcat(new_folder,'\original\',file(k).name);
        %imwrite(Img,folder_xx);
        data{k,1} = (Red);
        data{k,2} = (Green);
        data{k,3} = (Blue);
    end
end
handles.data = data;
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject,handles);

% --- Executes on button press in checkbox1.
function checkbox1_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox1 (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 checkbox1
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.checkbox1,'enable','on');
set(handles.checkbox4,'enable','off');
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject,handles);

% --- Executes on button press in checkbox2.
function checkbox2_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.checkbox2,'enable','on');
set(handles.checkbox4,'enable','off');
set(handles.checkbox8,'enable','off');
waitbar(1,waktu,'Finishing');
```

```
pause(1)
close(waktu);
guidata(hObject,handles);
% Hint: get(hObject,'Value') returns toggle state of checkbox2


% --- Executes on button press in checkbox4.
function checkbox4_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox4 (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 checkbox4
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.checkbox1,'enable','off');
set(handles.checkbox2,'enable','off');
set(handles.checkbox3,'enable','off');
set(handles.checkbox4,'enable','on');
set(handles.checkbox6,'enable','off');
set(handles.checkbox7,'enable','off');
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject,handles);

% --- Executes on button press in checkbox3.
function checkbox3_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox3 (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 checkbox3
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.checkbox3,'enable','on');
set(handles.checkbox4,'enable','off');
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject,handles);

% --- 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)
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu ,'Loading Data...');
pause(1)
waitbar(.67,waktu ,'Processing Data...');
pause(1)

folder_data = handles.folder_data;
file_data = dir(fullfile(folder_data, '*.jpg'));
```

```
jumlah_data = numel(file_data);

val1 = get(handles.checkbox1,'Value');
val2 = get(handles.checkbox2,'Value');
val3 = get(handles.checkbox3,'Value');
val4 = get(handles.checkbox4,'Value');

if val1 && val2 && val3 % Hapus Background, grayscale
    set(handles.checkbox1,'enable','on');
    set(handles.checkbox2,'enable','on');
    set(handles.checkbox3,'enable','on');
    for k = 1:jumlah_data
        full_data= fullfile(folder_data, file_data(k).name);
        Img = imread(full_data);
        HSV = rgb2hsv(Img);
        H = HSV(:,:,1);
        S = HSV(:,:,2);
        V = HSV(:,:,3);
        bw = im2bw(S, .4);
        bw = imfill(bw, 'holes');
        bw = bwareaopen(bw,1000);
        str = strel('disk',12);
        bw = imopen(bw,str);
        R = Img(:,:,1);
        G = Img(:,:,2);
        B = Img(:,:,3);
        R(~bw) = 0;
        G(~bw) = 0;
        B(~bw) = 0;
        Img = cat(3,R,G,B);
        Img = rgb2gray(Img);
        Img = wiener2(Img, [5 5]);
        folder=
strcat(folder_data,'\hapus_background_grayscale_filter\',file_data(k).name);
        new_folder= strcat(folder_data,'\hapus_background_grayscale_filter\');
        imwrite(Img,folder);
    end
elseif val1 && val2 % Hapus Background, grayscale
    set(handles.checkbox1,'enable','on');
    set(handles.checkbox2,'enable','on');
    for k = 1:jumlah_data
        full_data= fullfile(folder_data, file_data(k).name);
        Img = imread(full_data);
        HSV = rgb2hsv(Img);
        H = HSV(:,:,1);
        S = HSV(:,:,2);
        V = HSV(:,:,3);
        bw = im2bw(S, .4);
        bw = imfill(bw, 'holes');
        bw = bwareaopen(bw,1000);
        str = strel('disk',12);
        bw = imopen(bw,str);
        R = Img(:,:,1);
        G = Img(:,:,2);
        B = Img(:,:,3);
        R(~bw) = 0;
        G(~bw) = 0;
        B(~bw) = 0;
        Img = cat(3,R,G,B);
        Img = rgb2gray(Img);
        folder= strcat(folder_data,'\hapus_background_grayscale\',file_data(k).name);
        new_folder= strcat(folder_data,'\hapus_background_grayscale\');
        imwrite(Img,folder);
    end
```

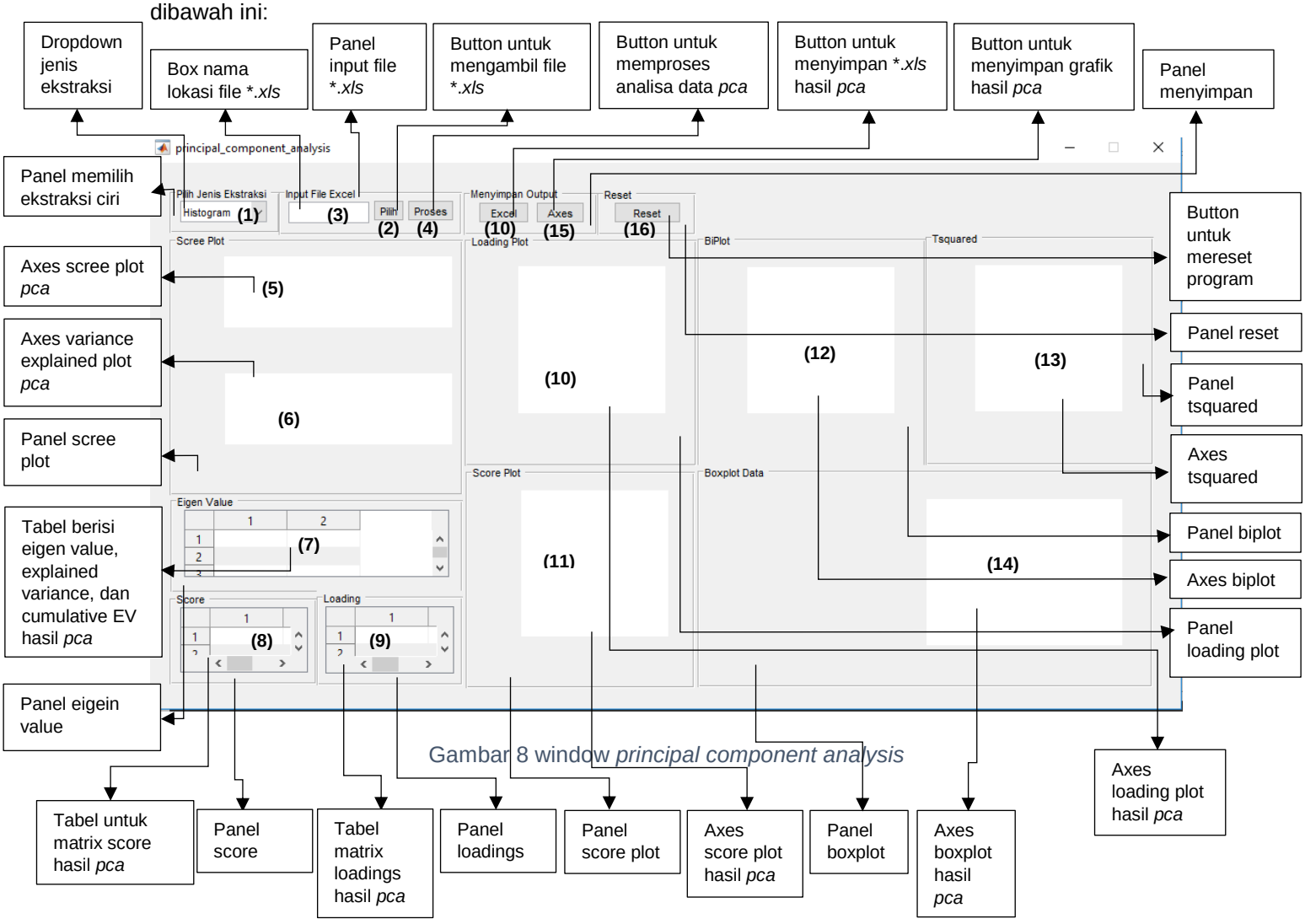
```
elseif val1 && val3 % Hapus background, filter
    set(handles.checkbox1,'enable','on');
    set(handles.checkbox3,'enable','on');
    for k = 1:jumlah_data
        full_data= fullfile(folder_data, file_data(k).name);
        Img = imread(full_data);
        folder= strcat(folder_data,'\hapus_background_filter\',file_data(k).name);
        new_folder= strcat(folder_data,'\hapus_background_filter\');
        imwrite(Img,folder);
    end
    waitbar(.7,waktu,'IMAGE MUST BE CONVERTED, RESET PROGRAM');
    pause(1)
elseif val1 % Hapus_background
    set(handles.checkbox1,'enable','on');
    for k = 1:jumlah_data
        full_data= fullfile(folder_data, file_data(k).name);
        Img = imread(full_data);
        HSV = rgb2hsv(Img);
        H = HSV(:,:,1);
        S = HSV(:,:,2);
        V = HSV(:,:,3);
        bw = im2bw(S, .4);
        bw = imfill(bw, 'holes');
        bw = bwareaopen(bw,1000);
        str = strel('disk',12);
        bw = imopen(bw,str);
        R = Img(:,:,1);
        G = Img(:,:,2);
        B = Img(:,:,3);
        R(~bw) = 0;
        G(~bw) = 0;
        B(~bw) = 0;
        Img = cat(3,R,G,B);
        folder= strcat(folder_data,'\hapus_background\',file_data(k).name);
        new_folder= strcat(folder_data,'\hapus_background\');
        imwrite(Img,folder);
    end
elseif val2 && val3 % Grayscale, filter
    set(handles.checkbox2,'enable','on');
    set(handles.checkbox3,'enable','on');
    for k = 1:jumlah_data
        full_data= fullfile(folder_data, file_data(k).name);
        Img = imread(full_data);
        Img = rgb2gray(Img);
        Img = wiener2(Img, [5 5]);
        folder= strcat(folder_data,'\grayscale_filter\',file_data(k).name);
        new_folder = strcat(folder_data,'\grayscale_filter\');
        imwrite(Img,folder);
    end
elseif val2 % grayscale
    set(handles.checkbox2,'enable','on');
    for k = 1:jumlah_data
        full_data= fullfile(folder_data, file_data(k).name);
        Img = imread(full_data);
        Img = rgb2gray(Img);
        folder= strcat(folder_data,'\grayscale\',file_data(k).name);
        new_folder= strcat(folder_data,'\grayscale\');
        imwrite(Img,folder);
    end
elseif val3 % Filter
    set(handles.checkbox3,'enable','on');
    for k = 1:jumlah_data
        full_data= fullfile(folder_data, file_data(k).name);
        Img = imread(full_data);
```

```
        folder= strcat(folder_data,'\filter\',file_data(k).name);
        new_folder= strcat(folder_data,'\filter\');
    end
    %msgbox({'Image Must Be Converted'; 'Reset Program'},'error','error')
    waitbar(.7,waktu,'IMAGE MUST BE CONVERTED, RESET PROGRAM');
    pause(1)
elseif val4 % Original
    set(handles.checkbox4,'enable','on');
    for k = 1:jumlah_data
        full_data= fullfile(folder_data, file_data(k).name);
        Img = imread(full_data);
        folder= strcat(folder_data,'\original\',file_data(k).name);
        new_folder = strcat(folder_data,'\original\');
        imwrite(Img,folder);
    end
end
new_folder
handles.new_folder = new_folder;
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
%handles.jumlah_data = jumlah_data;
guidata(hObject,handles);


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
```

II. GRAPHIC USER INTERFACE (TAB II/principal_component_analysis)

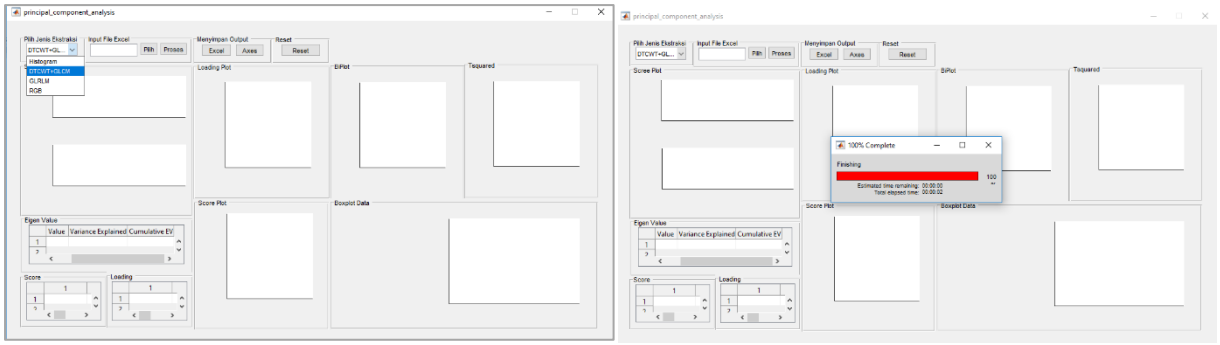
A. Deskripsi Program

Program menampilkan graphical user interface (GUI) untuk melakukan analisa data hasil ekstraksi ciri citra daun teh dengan metode principal component analysis sebagaimana ditampilkan pada gambar dibawah ini:



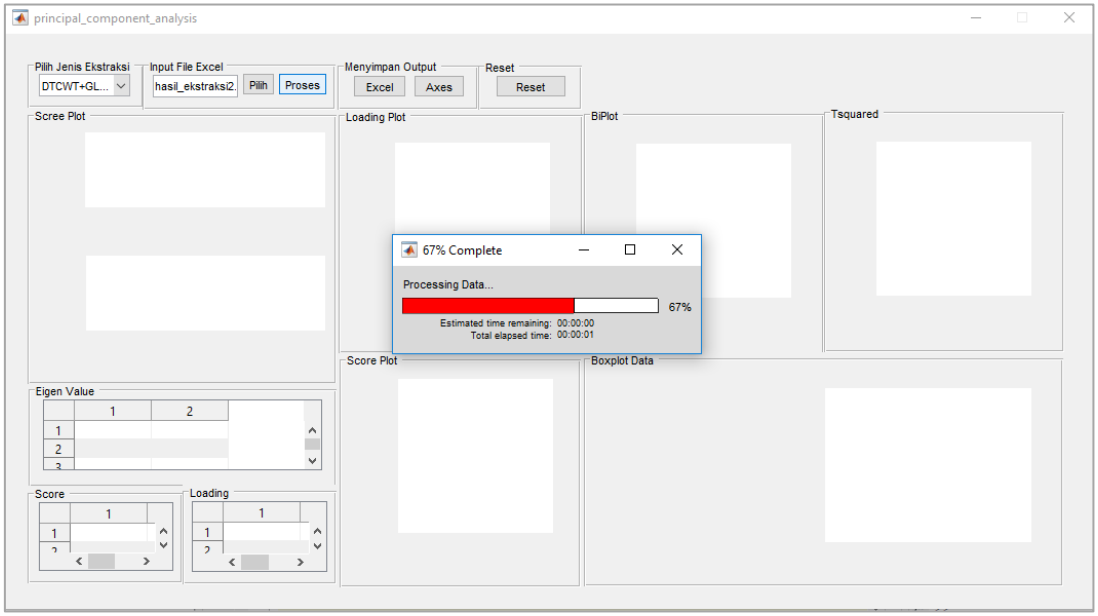
Gambar 8 window *principal component analysis*

Untuk melakukan analisa data hasil ekstraksi ciri dengan program *principal component analysis* pertama dengan memilih jenis ekstraksi ciri (berlabel (1)), hingga tampil popup bar menunggu hingga selesai sebagaimana ditampilkan pada gambar dibawah ini:



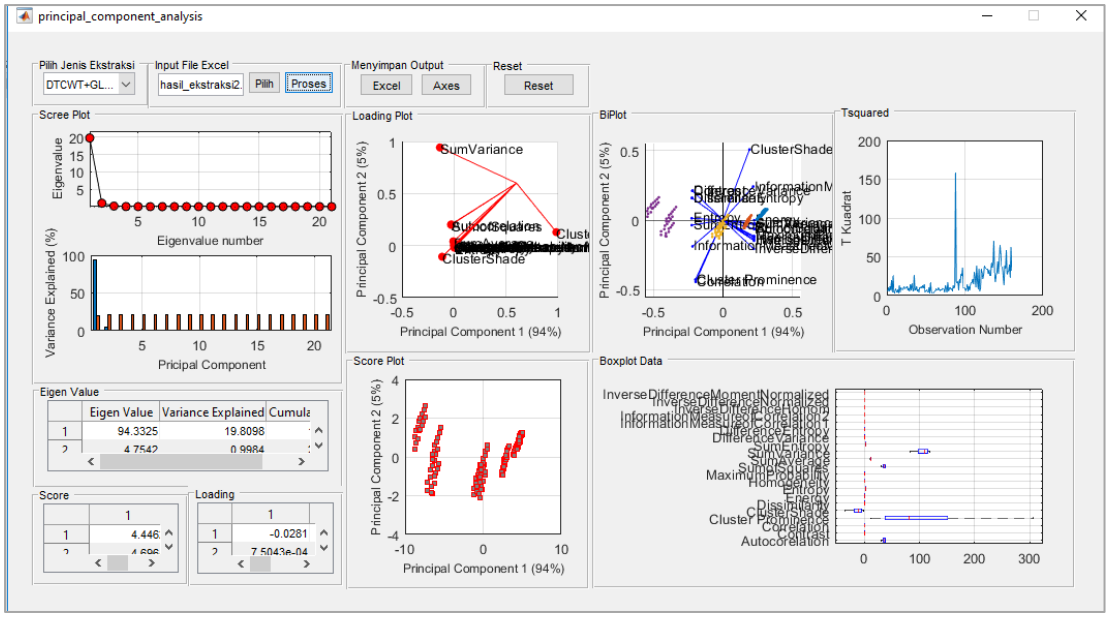
Gambar 9 window saat memilih jenis ekstraksi ciri

Kemudian memilih file berformat *.*xls* hasil ekstraksi ciri sesuai dengan jenis ekstraksi yang telah dipilih dengan menekan button '*pilih*' (berlabel (2)) dan lokasi nama file akan tampil pada box name (berlabel (3)) sebagaimana ditampilkan pada gambar dibawah ini, namun jika tidak sesuai proses analisa data dengan program ini tidak akan berjalan.



Gambar 10 window *principal component analysis*

Kemudian menekan button untuk memulai analisa deta dengan metode *principal component analysis* sebagaimana ditampilkan pada gambar diatas kemudian hasil analisa data tampil secara bersamaan seperti gambar dibawah ini:



Gambar 11 window hasil analisa data dengan metode *principal component analysis*

B. Code Program

Code program dibuat dengan bahasa script Matlab yang terdiri dari beberapa file program sebagai berikut.

`Principal_component_analysis.m`

Nama File	:	Principal_component_analysis.m
Deskripsi	:	Menciptakan GUI untuk halaman principal_component_analysis pada Matlab yang terdiri dari lima button untuk mengamfil file hasil ekstraksi ciri berformat .*xls,button untuk memproses analisa data, button simpan excel hasil analisa data, button untuk menyimpan axes, button untuk mereset program, kemudian program ini juga memiliki tujuh axes yaitu axes untuk scree plot, explained variance, axes biplot, axes loading plot, axes tsquared, axes boxplot, axes score plot, kemudian memiliki tiga tabel yaitu tabel eigen value, tabel matrix loadings, serta matrix score.
Platform	:	Matlab
Copyright	:	© 2020 Alma, Dr. Bambang Heru Iswanto

```
function varargout = principal_component_analysis(varargin)
% PRINCIPAL_COMPONENT_ANALYSIS MATLAB code for principal_component_analysis.fig
%      PRINCIPAL_COMPONENT_ANALYSIS, by itself, creates a new
PRINCIPAL_COMPONENT_ANALYSIS or raises the existing
%      singleton*.
%
%      H = PRINCIPAL_COMPONENT_ANALYSIS returns the handle to a new
PRINCIPAL_COMPONENT_ANALYSIS or the handle to
%      the existing singleton*.
%
%      PRINCIPAL_COMPONENT_ANALYSIS('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in PRINCIPAL_COMPONENT_ANALYSIS.M with the given input
arguments.
```

```
%
%      PRINCIPAL_COMPONENT_ANALYSIS('Property','Value',...) creates a new
PRINCIPAL_COMPONENT_ANALYSIS or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before principal_component_analysis_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to principal_component_analysis_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 principal_component_analysis

% Last Modified by GUIDE v2.5 11-Aug-2020 14:12:47

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @principal_component_analysis_OpeningFcn, ...
                   'gui_OutputFcn',  @principal_component_analysis_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 principal_component_analysis is made visible.
function principal_component_analysis_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 principal_component_analysis (see VARARGIN)

% Choose default command line output for principal_component_analysis
handles.output = hObject;
movegui('center');
% Update handles structure
guidata(hObject, handles);

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


% --- Outputs from this function are returned to the command line.
function varargout = principal_component_analysis_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)
folder_excel= uigetfile('*.xls');
file = xlsread(folder_excel);
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.edit1,'String',folder_excel);
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
%% Import to handles
handles.folder_excel = folder_excel;
handles.file = file;
guidata(hObject, handles);


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 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)
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
file = handles.file;
```

```
folder_excel = handles.folder_excel;
file_excel = dir(fullfile(folder_excel, '*.xls'));
jumlah_file = numel(file_excel);

%% PCA
metode = get(handles.popupmenu1,'Value')
switch metode
    case 1
        fitur = {'Mean','Entropy','Variansi','Skewness','Kurtosis','Energy'};
    case 2
        fitur = {'Autocorelation','Contrast','Correlation','Cluster Prominence',...
            'ClusterShade','Dissimilarity','Energy','Entropy','Homogeneity',...

'MaximumProbability','SumofSquares','SumAverage','SumVariance','SumEntropy',...

'DifferenceVariance','DifferenceEntropy','InformationMeasureofCorrelation1',...
            'InformationMeasureofCorrelation2','InverseDifferenceHomom',...
            'InverseDifferenceNormalized','InverseDifferenceMomentNormalized'};
    case 3
        fitur = {'ShortRunEmphasis','LongRunEmphasis','GrayLevelNonuniformity',...
            'RunLengthNonuniformity', 'RunPercentage','LowGrayLevelRunEmphasis',...

'HighGrayLevelRunEmphasis','ShortRunLowGrayLevelEmphasis','ShortRunHighGrayLevelEmpha
sis',...
            'LongRunLowGrayLevelEmphasis','LongRunHighGrayLevelEmphasis'};
    case 4
        fitur = {'Red','Green','Blue'};
end

[data_normalZ,muZ,sigmaZ] = zscore(file);
[coeff,score,latent,tsquared,explained] = pca(data_normalZ);
[loadings, scores, ~, tscores]=princomp(file)
jenis_daun = zeros(jumlah_file,1);
jenis_daun(1:40,:) = 1;
jenis_daun(41:80,:) = 2;
jenis_daun(81:120,:) = 3;
jenis_daun(121:160,:) = 4;

%% Informasi EigenValue
eigentabel = [explained, latent, cumsum(latent)];
set(handles.uitable1,'Data',eigentabel,'ForegroundColor',[0 0 0],'ColumnName',{'Eigen
Value','Variance Explained','Cumulative EV'})

%% Information Loadings
set(handles.uitable2,'Data',loadings,'ForegroundColor',[0 0 0])

%% Information Score
set(handles.uitable3,'Data',score,'ForegroundColor',[0 0 0])

%% Plotting Eigenvalue
axes(handles.axes1);
plot(1:length(latent),latent,'ko-','MarkerFaceColor','r');
line([1,length(latent)],[0 0],'LineStyle',':','XLimInclude','off',...
     'Color',[.0 .0 .0])
axis([1,length(latent),min(latent),max(latent)*1.1]);
xlabel('Eigenvalue number','FontSize',9);
ylabel('Eigenvalue','FontSize',9);
grid on

%% Plotting Variance Explained
axes(handles.axes2);
explain = [explained cumsum(latent)]
bar(explain);
xlabel('Pricipal Component','FontSize',9);
```

```
ylabel('Variance Explained (%)','FontSize',9);
grid on

%% Ploting Loading
axes(handles.axes3);
scatter(loadings(:,1), loadings(:,2),'ro','filled')
for y = 1:length(fitur)
    line([0.6,loadings(y,1)],[0.6,loadings(y,2)],'Color','r')
    text(loadings(y,1), loadings(y,2),fitur{y})
end
xlabel('Principal Component 1 ('+string(round(explained(1)))+'%)','FontSize',9)
ylabel('Principal Component 2 ('+string(round(explained(2)))+'%)','FontSize',9)
grid on

%% Plotting Score
axes(handles.axes4);
plot(score(:,1), score(:,2),
's','LineWidth',1,'MarkerSize',5,'MarkerEdgeColor','r','MarkerFaceColor',[0.5,0.5,0.5
])
xlabel('Principal Component 1 ('+string(round(explained(1)))+'%)','FontSize',9)
ylabel('Principal Component 2 ('+string(round(explained(2)))+'%)', 'FontSize',9)
grid on

%% Plotting Biplot
axes(handles.axes5);
h = biplot(coeff(:,1:2),'Scores',score(:,1:2),'varlabels',fitur);
hID = get(h, 'tag');
hPt = h(strcmp(hID,'obsmarker'));
grp = findgroups(jenis_daun);
grp(isnan(grp)) = max(grp(~isnan(grp)));
grpID = 1:max(grp);
clrMap = lines(length(unique(grp)));
for i = 1:max(grp)
    set(hPt(grp==i),'Color',clrMap(i,:),'DisplayName',sprintf('Cluster
%d',grpID(i)));
end
xlabel('Principal Component 1 ('+string(round(explained(1)))+'%)','FontSize',9)
ylabel('Principal Component 2 ('+string(round(explained(2)))+'%)','FontSize',9)
[~, unqIdx] = unique(grp)
%legend(hPt(unqIdx),'Location', 'SE', )
grid on

axes(handles.axes6)
boxplot(file,'Orientation','horizontal','Labels',fitur);
grid on

axes(handles.axes7)
plot(tsquared);
xlabel('Observation Number','FontSize',9)
ylabel('T Kuadrat','FontSize',9)
grid on

handles.eigentabel = eigentabel;
handles.loadings = loadings;
handles.score = score;

waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject, handles);

% --- 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)
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])
axes(handles.axes1)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])
axes(handles.axes2)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])
axes(handles.axes3)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])
axes(handles.axes4)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])
axes(handles.axes5)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])
axes(handles.axes6)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])
axes(handles.axes7)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])

set(handles.uitable1,'Data',cell(size(get(handles.uitable1,'Data'))));
set(handles.uitable2,'Data',cell(size(get(handles.uitable2,'Data'))));
set(handles.uitable3,'Data',cell(size(get(handles.uitable2,'Data'))))

set(handles.edit1,'String', '');

waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject, handles);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
eigentabel = handles.eigentabel;
loadings = handles.loadings;
score = handles.score;
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
```

```
pause(1)
%% Menginisiasi Save as
[filenames, pathname] = uiputfile('*.xls','save as');
%% Data Ke .xls
xlswrite(filenames, eigentabel, 1);
xlswrite(filenames, loadings, 2);
xlswrite(filenames, score, 3);
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
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)
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)

xlabel(handles.axes1,'Eigenvalue number','FontSize',9);
ylabel(handles.axes1,'Eigenvalue','FontSize',9);
frame_data_1=getframe(handles.axes1);

frame_data_2=getframe(handles.axes2);
frame_data_3=getframe(handles.axes3);
frame_data_4=getframe(handles.axes4);
frame_data_5=getframe(handles.axes5);
frame_data_6=getframe(handles.axes6);
frame_data_7=getframe(handles.axes7);

image_frame_1=frame2im(frame_data_1);
image_frame_2=frame2im(frame_data_2);
image_frame_3=frame2im(frame_data_3);
image_frame_4=frame2im(frame_data_4);
image_frame_5=frame2im(frame_data_5);
image_frame_6=frame2im(frame_data_6);
image_frame_7=frame2im(frame_data_7);

metode = get(handles.popupmenu1,'Value')
switch metode
    case 1
        fitur = {'Mean','Entropy','Variansi','Skewness','Kurtosis','Energy'};
    case 2
        fitur = {'Autocorelation','Contrast','Correlation','Cluster Prominence',...
            'ClusterShade','Dissimilarity','Energy','Entropy','Homogeneity',...

'MaximumProbability','SumofSquares','SumAverage','SumVariance','SumEntropy',...

'DifferenceVariance','DifferenceEntropy','InformationMeasureofCorrelation1',...
            'InformationMeasureofCorrelation2','InverseDifferenceHomom',...
            'InverseDifferenceNormalized','InverseDifferenceMomentNormalized'};
end

%[nama_file, nama_path] = uiputfile('*.png');
%imwrite([image_frame_1,image_frame_2,image_frame_3,image_frame_4,image_frame_5,image
_frame_6,image_frame_7],fullfile(nama_path,nama_file));

imwrite([image_frame_1],['axes1_',num2str(metode),'.jpg'])%'axes1','jpg');
imwrite([image_frame_2],['axes2_',num2str(metode),'.jpg']);
```

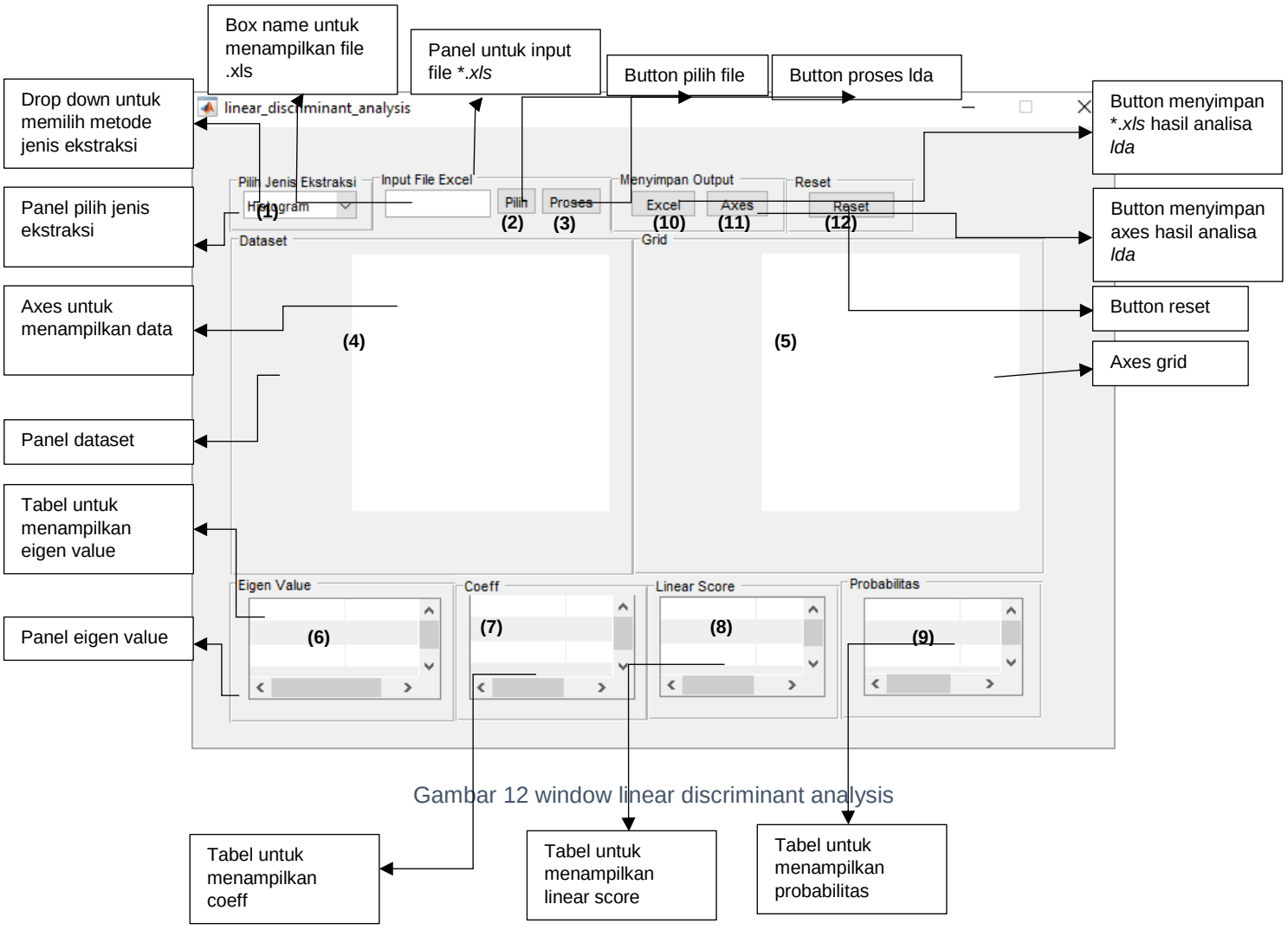
```
imwrite([image_frame_3],['axes3_',num2str(metode),'.jpg']);
imwrite([image_frame_4],['axes4_',num2str(metode),'.jpg']);
imwrite([image_frame_5],['axes5_',num2str(metode),'.jpg']);
imwrite([image_frame_6],['axes6_',num2str(metode),'.jpg']);
imwrite([image_frame_7],['axes7_',num2str(metode),'.jpg']);

%saveas(frame_data_1,'axes1.jpg')
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject, handles);

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as
cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
metode = get(handles.popupmenu1,'Value')
switch metode
    case 1
        fitur = {'Mean','Entropy','Variansi','Skewness','Kurtosis','Energy'};
    case 2
        fitur = {'Autocorelation','Contrast','Correlation','Cluster Prominence',...
            'ClusterShade','Dissimilarity','Energy','Entropy','Homogeneity',...

'MaximumProbability','SumofSquares','SumAverage','SumVariance','SumEntropy',...

'DifferenceVariance','DifferenceEntropy','InformationMeasureofCorrelation1',...
            'InformationMeasureofCorrelation2','InverseDifferenceHomom',...
            'InverseDifferenceNormalized','InverseDifferenceMomentNormalized'};
    case 3
        fitur = {'ShortRunEmphasis','LongRunEmphasis','GrayLevelNonuniformity',...
            'RunLengthNonuniformity', 'RunPercentage','LowGrayLevelRunEmphasis',...

'HighGrayLevelRunEmphasis','ShortRunLowGrayLevelEmphasis','ShortRunHighGrayLevelEmpha
sis',...
            'LongRunLowGrayLevelEmphasis','LongRunHighGrayLevelEmphasis'};
    case 4
        fitur = {'Red','Green','Blue'};
end
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu 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
```

III. GRAPHIC USER INTERFACE (TAB III/ linear_discriminant_analysis)

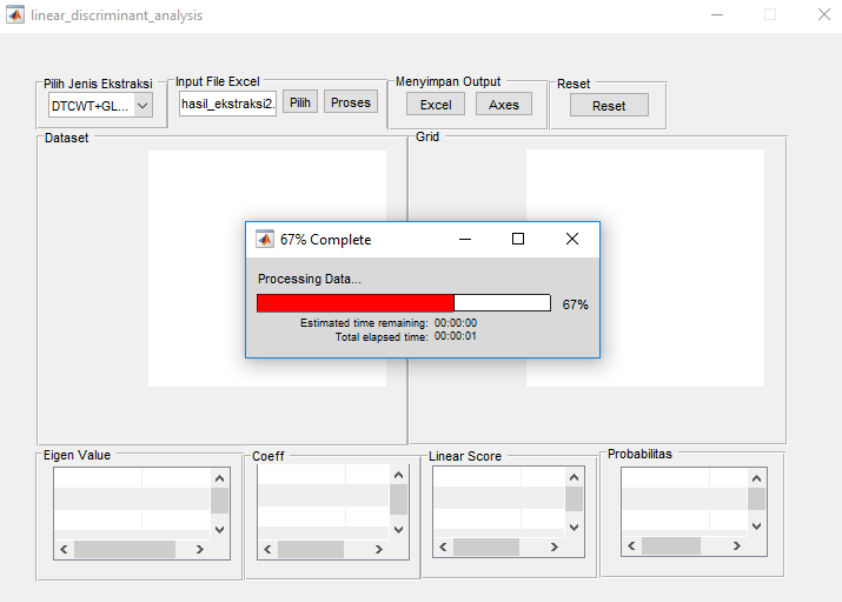
A. Deskripsi Program

Program menampilkan graphical user interface (GUI) untuk melakukan analisa data hasil ekstraksi ciri citra daun teh dengan metode linear disciminant analysis sebagai mana ditampilkan pada gambar dibawah ini:



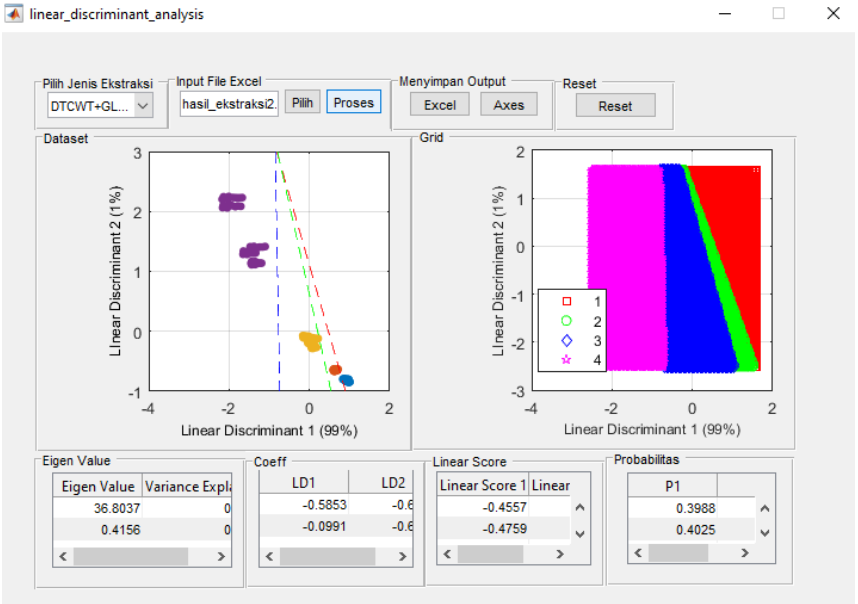
Gambar 12 window linear discriminant analysis

Untuk menganalisa data dengan metode *linear discriminant analysis* pada program ini hal pertama yang dilakukan yaitu memilih jenis ekstraksi untuk mengetahui jumlah fitur yang akan dipakai kemudian mengambil file **.xls* dengan menekan button pilih kemudian menunggu proses selesai sampai popup bar selesai seperti yang ditampilkan pada gambar dibawah ini



Gambar 13 window *linear discriminant analysis* saat mengambil file *.*xls*

Kemudian menekan button '*proses*' untuk memproses analisa data dengan metode *linear discriminant analysis* hingga popup menunggu selesai sebagaimana ditampilkan pada gambar dibawah ini, hasil analisa data akan ditampilkan pada axes, serta tabel hasil analisis data dengan metode *linear discriminant analysis.*



Gambar 14 window hasil analisa data *linear discriminant analysis*

Kemudian apabila pengguna ingin menyimpan file hasil analisa data dengan menggunakan metode *linear discriminant analysis* pengguna dapat menekan button '*excel*' untuk menyimpan file *.*xls* dan button '*axes*' untuk menyimpan axes hasil analisa tersebut.Kemudian apabila ingin mereset program dapat menekan button '*reset*'.

B. Code Program

Code program dibuat dengan bahasa script Matlab yang terdiri dari file program sebagai berikut.

linear_discriminant_analysis.m

Nama File	:	inear_discriminant_analysis.m
Deskripsi	:	Menciptakan GUI untuk menganalisa data hasil ekstraksi ciri citra dengan metode *linear discriminant analysis* yang terdiri dari lima button yaitu, button untuk memilih file **.xls*, button untuk memproses analisa data, button untuk menyimpan file **.xls* hasil analisa data, button untuk menyimpan axes, button reset, kemudian terdapat dua axes yaitu axes untuk menampilkan batas data, dan axes untuk menampilkan wilayah kelompok daun, serta terdapat empat tabel yaitu tabel untuk menampilkan eigen value, tabel score, tabel coeff, serta tabel probabilita
Platform	:	Matlab
Copyright	:	© 2020 Alma, Dr. Bambang Heru Iswanto

```
function varargout = linear_discriminant_analysis(varargin)
% LINEAR_DISCRIMINANT_ANALYSIS MATLAB code for linear_discriminant_analysis.fig
%      LINEAR_DISCRIMINANT_ANALYSIS, by itself, creates a new
LINEAR_DISCRIMINANT_ANALYSIS or raises the existing
%      singleton*.
%
%      H = LINEAR_DISCRIMINANT_ANALYSIS returns the handle to a new
LINEAR_DISCRIMINANT_ANALYSIS or the handle to
%      the existing singleton*.
%
%      LINEAR_DISCRIMINANT_ANALYSIS('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in LINEAR_DISCRIMINANT_ANALYSIS.M with the given input
arguments.
%
%      LINEAR_DISCRIMINANT_ANALYSIS('Property','Value',...) creates a new
LINEAR_DISCRIMINANT_ANALYSIS or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before linear_discriminant_analysis_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to linear_discriminant_analysis_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 linear_discriminant_analysis

% Last Modified by GUIDE v2.5 14-Aug-2020 14:40:40

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @linear_discriminant_analysis_OpeningFcn, ...
                   'gui_OutputFcn',  @linear_discriminant_analysis_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 linear_discriminant_analysis is made visible.
function linear_discriminant_analysis_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 linear_discriminant_analysis (see VARARGIN)

% Choose default command line output for linear_discriminant_analysis
handles.output = hObject;
movegui('center');
% Update handles structure
guidata(hObject, handles);

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


% --- Outputs from this function are returned to the command line.
function varargout = linear_discriminant_analysis_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;



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 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)
folder_excel= uigetfile('*.xls');
file = xlsread(folder_excel);
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
set(handles.edit1,'String',folder_excel);
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
%% Import to handles
handles.folder_excel = folder_excel;
handles.file = file;
guidata(hObject, handles);

% --- 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)
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
file = handles.file;
folder_excel = handles.folder_excel;
file_excel = dir(fullfile(folder_excel, '*.xls'));
jumlah_file = numel(file_excel);

%% PCA
metode = get(handles.popupmenu1,'Value')
switch metode
    case 1
        fitur = {'Mean','Entropy','Variansi','Skewness','Kurtosis','Energy'};
    case 2
        fitur = {'Autocorelation','Contrast','Correlation','Cluster Prominence',...
            'ClusterShade','Dissimilarity','Energy','Entropy','Homogeneity',...

'MaximumProbability','SumofSquares','SumAverage','SumVariance','SumEntropy',...

'DifferenceVariance','DifferenceEntropy','InformationMeasureofCorrelation1',...
            'InformationMeasureofCorrelation2','InverseDifferenceHomom',...
            'InverseDifferenceNormalized','InverseDifferenceMomentNormalized'};
    case 3
        fitur = {'ShortRunEmphasis','LongRunEmphasis','GrayLevelNonuniformity',...
            'RunLengthNonuniformity', 'RunPercentage','LowGrayLevelRunEmphasis',...

'HighGrayLevelRunEmphasis','ShortRunLowGrayLevelEmphasis','ShortRunHighGrayLevelEmpha
sis',...
            'LongRunLowGrayLevelEmphasis','LongRunHighGrayLevelEmphasis'};
    case 4
```

```
        fitur = {'Red','Green','Blue'};
end

[data_normalZ,muZ,sigmaZ] = zscore(file);
[coeff,score,latent,tsquared,explained] = pca(data_normalZ);
[loadings, scores, variances, tscores]=princomp(file)
%% Target kelas
jenis_daun = zeros(jumlah_file,1);
jenis_daun(1:40,:) = 1;
jenis_daun(41:80,:) = 2;
jenis_daun(81:120,:) = 3;
jenis_daun(121:160,:) = 4;
%% LDA

M = [data_normalZ(:,1),data_normalZ(:,2)];
[S, W, lambda] = LDA([M], jenis_daun)
[X,Y] = meshgrid(linspace(-2.5,1.6),linspace(-2.5,1.6));
X = X(:);Y = Y(:);
Prob = exp(S)./repmat(sum(exp(S),2),[1,2])
[A,err,P,lopgp,coeff] = classify([X Y], M, jenis_daun,'linear');
K = coeff(1,2).const;
L = coeff(1,2).linear;
f = @(Y,Z) K+[Y,Z]*L;
K1 = coeff(2,3).const;
L1 = coeff(2,3).linear;
f1 = @(Y,Z) K1+[Y,Z]*L1
K2 = coeff(3,4).const;
L2 = coeff(3,4).linear;
f2 = @(Y,Z) K2+[Y,Z]*L2;
%% Informasi Eigen value
explainLda = (lambda./sum(lambda));
eigentabel = [lambda, explainLda];
set(handles.uitable1,'Data',eigentabel,'ForegroundColor',[0 0 0],'ColumnName',{'Eigen
Value','Variance Explained'});
%% Informasi W (LD)
set(handles.uitable2,'Data',W,'ForegroundColor',[0 0 0],'ColumnName',{'LD1','LD2'});
%% Informasi Score
set(handles.uitable3,'Data',S,'ForegroundColor',[0 0 0],'ColumnName',{'Linear Score
1','Linear Score 2'});
%% Informasi Probability
set(handles.uitable4,'Data',Prob,'ForegroundColor',[0 0 0],'ColumnName',{'P1','P2'});
color = lines(7)
%% Plotting
axes(handles.axes1);
h1 = gscatter(data_normalZ(:,1),data_normalZ(:,2),jenis_daun,color([1 2 3
4],:),'....',[],'off')
hold on
h2 = fimplicit(f,[-4 2 -1 3],'--r')
h3 = fimplicit(f1,[-4 2 -1 3],'--g')
h4 = fimplicit(f2,[-4 2 -1 3],'--b')
xlabel('Linear Discriminant 1
('+string(round(explainLda(1).*100))+'%)','FontSize',9,'Color','k')
ylabel('LInear Discriminant 2
('+string(round(explainLda(2).*100))+'%)','FontSize',9,'Color','k')
grid on

axes(handles.axes2);
gscatter(X,Y,A,'rgbm','sodp');
xlabel('Linear Discriminant 1 ('+string(round(explainLda(1).*100))+'%)','FontSize',9)
ylabel('LInear Discriminant 2 ('+string(round(explainLda(2).*100))+'%)','FontSize',9)
grid on

waitbar(1,waktu,'Finishing');
pause(1)
```

```
close(waktu);
handles.eigentabel = eigentabel;
handles.W = W;
handles.S = S;
handles.Prob = Prob;
guidata(hObject, handles);

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as
cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
metode = get(handles.popupmenu1,'Value')
switch metode
    case 1
        fitur = {'Mean','Entropy','Variansi','Skewness','Kurtosis','Energy'};
    case 2
        fitur = {'Autocorelation','Contrast','Correlation','Cluster Prominence',...
            'ClusterShade','Dissimilarity','Energy','Entropy','Homogeneity',...

'MaximumProbability','SumofSquares','SumAverage','SumVariance','SumEntropy',...

'DifferenceVariance','DifferenceEntropy','InformationMeasureofCorrelation1',...
            'InformationMeasureofCorrelation2','InverseDifferenceHomom',...
            'InverseDifferenceNormalized','InverseDifferenceMomentNormalized'};
    case 3
        fitur = {'ShortRunEmphasis','LongRunEmphasis','GrayLevelNonuniformity',...
            'RunLengthNonuniformity', 'RunPercentage','LowGrayLevelRunEmphasis',...

'HighGrayLevelRunEmphasis','ShortRunLowGrayLevelEmphasis','ShortRunHighGrayLevelEmpha
sis',...
            'LongRunLowGrayLevelEmphasis','LongRunHighGrayLevelEmphasis'};
    case 4
        fitur = {'Red','Green','Blue'};
end
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu 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 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)
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])
axes(handles.axes1)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])
axes(handles.axes2)

set(handles.uitable1,'Data',cell(size(get(handles.uitable1,'Data'))));
set(handles.uitable2,'Data',cell(size(get(handles.uitable2,'Data'))));
set(handles.uitable3,'Data',cell(size(get(handles.uitable2,'Data'))));
set(handles.uitable4,'Data',cell(size(get(handles.uitable2,'Data'))));

set(handles.edit1,'String', '');

waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject, handles);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
eigentabel = handles.eigentabel;
W = handles.W;
S = handles.S;
Prob = handles.Prob;
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)
%% Menginisiasi Save as
[filenames, pathname] = uiputfile('*.xls','save as');
%% Data Ke .xls
xlswrite(filenames, eigentabel, 1);
xlswrite(filenames, W, 2);
xlswrite(filenames, S, 3);
xlswrite(filenames, Prob, 4);
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
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)
```

```
waktu = waitbar(0,'Please Wait...');
pause(.5)
waitbar(.33,waktu,'Loading Data...');
pause(1)
waitbar(.67,waktu,'Processing Data...');
pause(1)

frame_data_1=getframe(handles.axes1);
frame_data_2=getframe(handles.axes2);

image_frame_1=frame2im(frame_data_1);
image_frame_2=frame2im(frame_data_2);

metode = get(handles.popupmenu1,'Value')
switch metode
    case 1
        fitur = {'Mean','Entropy','Variansi','Skewness','Kurtosis','Energy'};
    case 2
        fitur = {'Autocorelation','Contrast','Correlation','Cluster Prominence',...
            'ClusterShade','Dissimilarity','Energy','Entropy','Homogeneity',...

'MaximumProbability','SumofSquares','SumAverage','SumVariance','SumEntropy',...

'DifferenceVariance','DifferenceEntropy','InformationMeasureofCorrelation1',...
            'InformationMeasureofCorrelation2','InverseDifferenceHomom',...
            'InverseDifferenceNormalized','InverseDifferenceMomentNormalized'};
end

%[nama_file, nama_path] = uiputfile('*.png');
%imwrite([image_frame_1,image_frame_2,image_frame_3,image_frame_4,image_frame_5,image
_frame_6,image_frame_7],fullfile(nama_path,nama_file));

imwrite([image_frame_1],['axes1Lda_',num2str(metode),'.jpg'])%'axes1','jpg');
imwrite([image_frame_2],['axes2Lda_',num2str(metode),'.jpg']);

%saveas(frame_data_1,'axes1.jpg')
waitbar(1,waktu,'Finishing');
pause(1)
close(waktu);
guidata(hObject, handles);
```