

## DAFTAR LAMPIRAN

### Pembuatan GUI Matlab

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

% Last Modified by GUIDE v2.5 28-Jul-2017 09:36:39

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

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

% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the command line.
function varargout = konversi_biner_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)
[name_file,name_path] = uigetfile(...
    {'*.bmp;*.jpg;*.tif','Files of type (*.bmp,*.jpg*.tif)';
    '*.bmp','File Bitmap (*.bmp)';...
    '*.jpg','File Jpeg (*.jpg)';
    '*.tif','File Tif (*.tif)';
    '*.*','All Files (*.*)'},...
    'Pilih Gambar');

```

```

if ~isequal (name_file,0)
    handles.data1 = imread(fullfile(name_path,name_file));
    guidata(hObject,handles);
    axes(handles.axes1);
    imshow(handles.data1);

else
    return
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)
gray = handles.data2;
value = get (handles.slider1,'value');
thresh = imcomplement (im2bw(gray,value/255));
assignin('base', 'thresh', thresh); %simpan data ke base dengan
nama thresh dg data thresh
axes(handles.axes2);
imshow(thresh);
handles.data3 = thresh;
guidata(hObject,handles);
set(handles.tablehasil, 'Data', thresh);
set(handles.edit1, 'string',value)

% 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

% --- 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)

```

```

image = handles.data1;
gray = rgb2gray(image);
axes(handles.axes2);
imshow(gray);
handles.data2 = gray;
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)

thresh = handles.data3;
[name_file_save,path_save] = uiputfile(...
    {'*.bmp','File Bitmap (*.bmp)';...
    '*.jpg','File Jpeg (*.jpg)';
    '*.tif','File Tif (*.tif)';
    '*.*','All File (*.*)'},...
    'Simpan');

if ~isequal (name_file_save,0)
    imwrite(thresh,fullfile(path_save,name_file_save));
else
    return
end

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 tes_apply.
function tes_apply_Callback(hObject, eventdata, handles)
% hObject    handle to tes_apply (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
X = xlsread('Tabel.xlsx'); %membaca excel dan memasukkan variable X
assignin(X, 'base', X);
set(handles.tabelhasil, 'Data', X); %menampilkan data ke table gui
matlab

% --- 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)

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (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 edit2 as text
%        str2double(get(hObject, 'String')) returns contents of
edit2 as a double

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (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 when entered data in editable cell(s) in
tablehasil.
function tablehasil_CellEditCallback(hObject, eventdata, handles)
% hObject    handle to tablehasil (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)
```

## Pelatihan Sistem

```
%data input dan target
A=[1 0 1 0 1 0; 1 0 0 1 0 1; 1 1 0 1 1 1; 0 1 1 1 0 0; 1 0 1 0 1
1; 1 1 1 1 0 0];
B=[0 0 1 1 0 1; 1 0 1 0 0 0; 0 1 0 0 1 1; 1 0 1 0 1 0; 1 1 0 0 0
0; 1 1 1 0 1 0];
C=[1 1 0 1 1 0; 0 1 0 1 1 0; 0 0 1 1 0 0; 1 1 1 1 1 0; 0 0 1 1 1
1; 0 0 1 1 0 1];
Z=[1 1 1 1 1 1];

%Membangun jaringan syaraf feedforward
net = newff(minmax(A), [3 1], {'tansig' 'purelin'}, 'traingdm');
net = newff(minmax(B), [3 1], {'tansig' 'purelin'}, 'traingdm');
net = newff(minmax(C), [3 1], {'tansig' 'purelin'}, 'traingdm');

%melihat bobot-bobot awal input, lapisan, dan bias
BobotAwal_Input      = net.IW{1,1}
BobotAwal_Bias_Input = net.b{1,1}
BobotAwal_Lapisan    = net.LW{2,1}
BobotAwal_Bias_Lapisan = net.b{2,1}

%set max epoch, goal, learning rate, momentum, show step
net.trainParam.epoch = 50;
net.trainParam.goal   = 1e-3;
net.trainParam.lr     = 0.1;
net.trainParam.mc     = 0.3;
net.trainParam.show   = 10;

%melakukan pembelajaran
net = train(net,A,Z);

%melihat kembali bobot penghitungan
BobotAwal_Input      = net.IW{1,1}
BobotAwal_Bias_Input = net.b{1,1}
BobotAwal_Lapisan    = net.LW{2,1}
BobotAwal_Bias_Lapisan = net.b{2,1}

%melakukan simulasi
y = sim(net,A)

%menggambar plot grafik output
pause;
subplot (611)
plot(A(1,:),Z,'bo',A(1,:),y,'r*');
title('Perbandingan antara target (o) dan Output jaringan (*)');
xlabel('input pertama');
ylabel('Target atau Output');
grid;
subplot (612)
plot(A(2,:),Z,'bo',A(2,:),y,'r*');
title('Perbandingan antara target (o) dan Output jaringan (*)');
xlabel('input kedua');
ylabel('Target atau Output');
```

```

grid;
subplot (613)
plot(A(1,:),Z,'bo',A(1,:),Y,'r*');
title('Perbandingan antara target (o) dan Output jaringan (*)');
xlabel('input ketiga');
ylabel('Target atau Output');
grid;
subplot (614)
plot(A(2,:),Z,'bo',A(2,:),Y,'r*');
title('Perbandingan antara target (o) dan Output jaringan (*)');
xlabel('input keempat');
ylabel('Target atau Output');
grid;
subplot (615)
plot(A(2,:),Z,'bo',A(2,:),Y,'r*');
title('Perbandingan antara target (o) dan Output jaringan (*)');
xlabel('input kelima');
ylabel('Target atau Output');
grid;
subplot (616)
plot(A(2,:),Z,'bo',A(2,:),Y,'r*');
title('Perbandingan antara target (o) dan Output jaringan (*)');
xlabel('input keenam');
ylabel('Target atau Output');
grid;

```