DAFTAR PUSTAKA

Anggriawan, M.A., Muhammad I., dan Dina B.U. 2017. *Pengenalan Tingkat Kematangan Tomat Berdasarkan Citra Warna pada Studi Kasus Pembangunan Sistem Pemilihan Otomatis*. Institut Teknologi Nasional Bandung: Bandung.

Ardiansyah, Y. 2015. *Aplikasi Identifikasi Mutu Kematangan Buah Kakao Menggunakan Image Processing dan Metode Fuzzy Logic*. Universitas Jember: Jawa Timur.

Asrul. L. 2013. *Agribisnis Kakao*. Media Bangsa: Jakarta.

Cahyan, P.A., Muhammad, A.I.M dan Ali, M.S.M. 2013. *Segmentasi Citra dengan Menggunakan Alogaritma Watershed dan Lowpass Filter sebagai Awal*. Universitas Brawijaya: Malang.

Hardiyanto, D dan Dyah, A.S. 2018. *Ekstraksi Fitur Citra Api Berbasis Ekstraksi Warna pada Ruang Warna Hsv dan Rgb*. IST AKPRIND Yogyakarta: Yogyakarta.

Indrabayu., Niswar, M dan Andryanto, A. 2016. *Sistem Pendeteksi Kesegaran Ikan Bandeng Menggunakan Citra.* Universitas Hasanuddin: Makassar.

Iqbal, M ., Triyogatama, W.W dan Bakhtiar A.A.S. 2016. *Sistem Pengendali Pengambilan Gambar pada Kamera DSLR melalui Protokol PTP*. Universitas Gajah Mada: Yogyakarta.

Kusumanto, R.D dan Alan N.T. 2011. *Pengolahan Citra Digital untuk Mendeteksi Obyek menggunakan Pengolahan Warna Model Normalisasi Rgb*. Politeknik Negeri Sriwijaya: Palembang.

Lukito, A.M., Y. Mulyono, I. Tetty, Hadi dan R. Nofiandi. 2010. *Budidaya Kakao*. AgroMedia Pustaka: Jakarta.

Permadi. Y dan Murito. 2015. *Aplikasi Pengolahan Citra untuk Identifikasi Kematangan Mentimun berdasarkan Tekstur Kulit Buah menggunkaan Metode Ekstraksi Citra Statistik*. Universitas Ahmad Dahlan: Yogyakarta.

Purwandari. E.P. 2014. *Peningkatan Kualitas Pembelajaran Pengolahan Citra Digital pada Program Studi Teknik Informatika Menggunakan Model Project Based Learning*. Universitas Bengkulu: Bengkulu.

Sulistiyanti, S.R., Arinto, S dan Muhammad K. 2016. *Pengolahan Citra Dasar dan Contoh Penerapannya*. Teknosain: Yogyakarta.

Tamrin., dan Junaidi M., dan Tahir S. 2017. *Penentuan Jumlah Butir Gabah dengan Computer Vision*. Universitas Hasanuddin: Makassar. *Jurnal Agritechno,* 10(2), 62-70. *https://doi.org/10.20956/at.v10i2.61*

Tjolleng, A. 2017. *Pengantar Pemprograman MATLAB*. PT Elex Media Komputindo: Jakarta.

Utami, A.S. 2011. *Perancangan Perangkat Lunak Sistem Temu Balik Citra Menggunakan Jarak Histogram dengan Model Warna YIQ*. Universitas Sumatera Utara: Medan.

LAMPIRAN

Lampiran 1. Perhitungan Tingkat Akurasi

Untuk mengetahui tingkat akurasi suatu sistem maka dapat dihitung sebesar sebagai berikut:

a. Pengujian dengan nilai RGB

$= \frac{105}{125}$ x 100%

= 84%

b. Pengujian dengan nilai HSI

$= \frac{112}{125}$ x 100%

= 89.6%

Lampiran 2. Bahasa Program

```
function varargout = Main(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Main_OpeningFcn, ...
                   'gui_OutputFcn',  @Main_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

function Main_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = Main_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function Tombol_InputGambar_Callback(hObject, eventdata, handles)
set(handles.Output,'string','Memilih gambar untuk diinput...');
set(handles.Tombol_InputGambar,'Enable','off');
set(handles.Tombol_Hapus,'Enable','off');
set(handles.Tombol_Training,'Enable','off');
set(handles.Tombol_Deteksi,'Enable','off');
drawnow;

[filename,pathname] = uigetfile('*.jpg');
```

```
if filename == 0
    set(handles.Output,'string','Input gambar dibatalkan.');
    set(handles.Tombol_InputGambar,'Enable','on');
    set(handles.Tombol_Hapus,'Enable','on');
    set(handles.Tombol_Training,'Enable','on');
    set(handles.Tombol_Deteksi,'Enable','on');
    return;
end

set(handles.Output,'string','Membaca data gambar...');
drawnow;

%menampilkan gambar
Img = imread(fullfile(pathname,filename));
handles.Im = Img;
axes(handles.Bingkai);
imshow(Img);
[mask,masked] = createMask(Img);
mask = bwareafilt(mask, 1);
mask = imfill(mask,'holes');
masked = Img .* cast(mask, 'like', Img);
handles.Msk = masked;
axes(handles.Masked);
imshow(masked);
guidata(hObject,handles);
HSV     = rgb2hsv(im2double(Img));
Hue     = HSV(:,:,1);
Sat     = HSV(:,:,2);
Val     = HSV(:,:,3);

MeanH = mean(Hue(mask));
MeanS = mean(Sat(mask));
MeanV = mean(Val(mask));

set(handles.Hue,'string',MeanH*360);
set(handles.Val,'string',MeanS*255);
set(handles.Sat,'string',MeanV*255);

set(handles.Output,'string','Gambar berhasil diinput.');
set(handles.Tombol_InputGambar,'Enable','on');
set(handles.Tombol_Hapus,'Enable','on');
set(handles.Tombol_Training,'Enable','on');
set(handles.Tombol_Deteksi,'Enable','on');

function Tombol_Hapus_Callback(hObject, eventdata, handles)
if ~isfield(handles,'Im')
    return;
end
cla(handles.Bingkai);
handles = rmfield(handles,'Im');
guidata(hObject,handles);
set(handles.Output,'string','Gambar dihapus.');

function Tombol_Deteksi_Callback(hObject, eventdata, handles)
if ~isfield(handles,'Im')
    set(handles.Output,'string','Silakan muat gambar terlebih
dahulu.');
```

```
    return
end
if ~exist('Data Training.xls','file')
    set(handles.Output,'string','Data Training tidak ditemukan!
Silakan lakukan Training Data terlebih dahulu.');
    return
end
set(handles.Output,'string','Sedang memproses...');
set(handles.Tombol_InputGambar,'Enable','off');
set(handles.Tombol_Hapus,'Enable','off');
set(handles.Tombol_Training,'Enable','off');
set(handles.Tombol_Deteksi,'Enable','off');
drawnow;

% pisah rgb
%Img = handles.Im;
Img = handles.Msk;

[mask,~] = createMask(Img);
mask = bwareafilt(mask, 1);
mask = imfill(mask,'holes');

%cari nilai HSI
HSV     = rgb2hsv(im2double(Img));
Hue     = HSV(:,:,1);
Sat     = HSV(:,:,2);
Val     = HSV(:,:,3);

%{
%Hue
atas=1/2*((Red-Green)+(Red-Blue));
bawah=((Red-Green).^2+((Red-Blue).*(Green-Blue))).^0.5;
teta = acosd(atas./(bawah));
if Blue >= Green
    H = 360 - teta;
else
    H = teta;
end

H = H/360;
[r c] = size(H);
for i=1 : r
    for j=1 : c
        z = H(i,j);
        z(isnan(z)) = 0; %isnan adalah is not none artinya jika
bukan angka dia akan memberi 0
        H(i,j) = z;
    end
end
%S
S=1-(3./(sum(RGB,3))).*min(RGB,[],3);
[r c] = size(S);
for i=1 : r
    for j=1 : c
        z = S(i,j);
        z(isnan(z)) = 0;
        S(i,j) = z;
    end
```

```
end

%I
I=(Red+Green+Blue)/3;
%}

MeanH = mean(Hue(mask));
MeanS = mean(Sat(mask));
MeanV = mean(Val(mask));

%{

% VarRed = var(Hue(:)); VarGreen = var(Sat(:)); VarBlue =
var(Val(:));
% VarH = var(H(:)); VarS = var(S(:)); VarI = var(I(:));
% RangeR = ((max(max(Hue)))-(min(min(Hue))));
% RangeG = ((max(max(Sat)))-(min(min(Sat))));
% RangeB = ((max(max(Val)))-(min(min(Val))));
% RangeH = ((max(max(H)))-(min(min(H))));
% RangeS = ((max(max(S)))-(min(min(S))));
% RangeI = ((max(max(I)))-(min(min(I))));
%}

training1 = xlsread('Data Training');
group = training1(:,4);
training = [training1(:,1) training1(:,2) training1(:,3)];
Z=[MeanH MeanS MeanV];

knnmodel = fitcknn(training,group);
hasil1 = predict(knnmodel,Z);
if hasil1==1
    x='Hasil Deteksi: MATANG';
else
    x='Hasil Deteksi: MENTAH';
end
set(handles.Output,'string',x);

set(handles.Tombol_InputGambar,'Enable','on');
set(handles.Tombol_Hapus,'Enable','on');
set(handles.Tombol_Training,'Enable','on');
set(handles.Tombol_Deteksi,'Enable','on');

function Tombol_Training_Callback(hObject, eventdata, handles)

image_folder = 'Training\matang';
filenames = dir(fullfile(image_folder,'*.jpg'));
total_images = numel(filenames);

if total_images == 0
    set(handles.Output,'string','Tidak ditemukan satupun gambar
sampel (matang). Pastikan adanya folder Training berisi dua folder
(matang dan mentah) berisi gambar sampel.');
    return
end

set(handles.Tombol_InputGambar,'Enable','off');
set(handles.Tombol_Hapus,'Enable','off');
```

```
set(handles.Tombol_Training,'Enable','off');
set(handles.Tombol_Deteksi,'Enable','off');

Z1=[];
for n = 1:total_images
  set(handles.Output,'string',['Training Data (Sampel Matang)... '
num2str(100*n/total_images) '%']);
  drawnow;
  full_name= fullfile(image_folder, filenames(n).name);
  Img = imread(full_name);

%Masked gambar
% [imh,imw,~] = size(Img);
% roipos = [imw/2-imw/10,imh/2-imh/10,imh/5,imh/5];
% cropped = imcrop(Img,roipos);

[mask,~] = createMask(Img);
mask = bwareafilt(mask, 1);
mask = imfill(mask,'holes');

%cari nilai HSI
HSV     = rgb2hsv(im2double(Img));
Hue     = HSV(:,:,1);
Sat     = HSV(:,:,2);
Val     = HSV(:,:,3);

%Hue
%{
atas=1/2*((Red-Green)+(Red-Blue));
bawah=((Red-Green).^2+((Red-Blue).*(Green-Blue))).^0.5;
teta = acosd(atas./(bawah));
if Blue >= Green
    H = 360 - teta;
else
    H = teta;
end

H = H/360;
[r c] = size(H);
for i=1 : r
    for j=1 : c
        z = H(i,j);
        z(isnan(z)) = 0;
        H(i,j) = z;
    end
end

%S
S=1-(3./(sum(RGB,3))).*min(RGB,[],3);
[r c] = size(S);
for i=1 : r
    for j=1 : c
        z = S(i,j);
        z(isnan(z)) = 0;
        S(i,j) = z;
    end
end
```

```
%I
I=(Red+Green+Blue)/3;
%}

MeanH = mean(Hue(mask));
MeanS = mean(Sat(mask));
MeanV = mean(Val(mask));

%{
VarRed = var(Red(:));
VarGreen = var(Green(:));
VarBlue = var(Blue(:));
VarH = var(H(:));
VarS = var(S(:));
VarI = var(I(:));
RangeR = ((max(max(Red)))-(min(min(Red))));
RangeG = ((max(max(Green)))-(min(min(Green))));
RangeB = ((max(max(Blue)))-(min(min(Blue))));
RangeH = ((max(max(H)))-(min(min(H))));
RangeS = ((max(max(S)))-(min(min(S))));
RangeI = ((max(max(I)))-(min(min(I))));

sdR = std2(Red);
sdG = std2(Green);
sdB = std2(Blue);
sdH = std2(H);
sdS = std2(S);
sdI = std2(I);
%}

Z=[MeanH MeanS MeanV 1];
Z1=[Z1;Z];
end

image_folder = 'Training\mentah';
filenames = dir(fullfile(image_folder, '*.jpg'));
total_images = numel(filenames);

if total_images == 0
    set(handles.Output,'string','Tidak ditemukan satupun gambar
sampel (mentah). Pastikan adanya folder Training berisi dua folder
(matang dan mentah) berisi gambar sampel.');
    return
end

for n = 1:total_images
  set(handles.Output,'string',['Training Data (Sampel Mentah)... '
num2str(100*n/total_images) '%']);
  drawnow;
  full_name= fullfile(image_folder, filenames(n).name);
  Img = imread(full_name);

  %Masked gambar
% [imh,imw,~] = size(Img);
% roipos = [imw/2-imw/10,imh/2-imh/10,imh/5,imh/5];
% cropped = imcrop(Img,roipos);
```

```
[mask,~] = createMask(Img);
mask = bwareafilt(mask, 1);
mask = imfill(mask,'holes');

%cari nilai HSI
HSV     = rgb2hsv(im2double(Img));
Hue     = HSV(:,:,1);
Sat     = HSV(:,:,2);
Val     = HSV(:,:,3);

%{
%Hue
atas=1/2*((Red-Green)+(Red-Blue));
bawah=((Red-Green).^2+((Red-Blue).*(Green-Blue))).^0.5;
teta = acosd(atas./(bawah));
if Blue >= Green
    H = 360 - teta;
else
    H = teta;
end

H = H/360;
[r c] = size(H);
for i=1 : r
    for j=1 : c
        z = H(i,j);
        z(isnan(z)) = 0;
        H(i,j) = z;
    end
end

%S
S=1-(3./(sum(RGB,3))).*min(RGB,[],3);
[r c] = size(S);
for i=1 : r
    for j=1 : c
        z = S(i,j);
        z(isnan(z)) = 0;
        S(i,j) = z;
    end
end
%I
I=(Red+Green+Blue)/3;
%}

MeanH = mean(Hue(mask));
MeanS = mean(Sat(mask));
MeanV = mean(Val(mask));

%{
VarRed = var(Red(:));
VarGreen = var(Green(:));
VarBlue = var(Blue(:));
VarH = var(H(:));
VarS = var(S(:));
VarI = var(I(:));
RangeR = ((max(max(Red)))-(min(min(Red))));
RangeG = ((max(max(Green)))-(min(min(Green))));
```

```
RangeB = ((max(max(Blue)))-(min(min(Blue))));
RangeH = ((max(max(H)))-(min(min(H))));
RangeS = ((max(max(S)))-(min(min(S))));
RangeI = ((max(max(I)))-(min(min(I))));

sdR = std2(Red);
sdG = std2(Green);
sdB = std2(Blue);
sdH = std2(H);
sdS = std2(S);
sdI = std2(I);
%}

Z=[MeanH MeanS MeanV 2];
Z1=[Z1;Z];
end

set(handles.Output,'string','Menyimpan Data Training... ');
xlswrite('Data Training.xls',Z1);
set(handles.Output,'string','Training Selesai!');

set(handles.Tombol_InputGambar,'Enable','on');
set(handles.Tombol_Hapus,'Enable','on');
set(handles.Tombol_Training,'Enable','on');
set(handles.Tombol_Deteksi,'Enable','on');

function Output_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all
properties.
function Hue_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

% --- Executes during object creation, after setting all
properties.
function Val_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (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 Sat_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 [BW,maskedRGBImage] = createMask(RGB)
%createMask  Threshold RGB image using auto-generated code from
colorThresholder app.
%  [BW,MASKEDRGBIMAGE] = createMask(RGB) thresholds image RGB
using
%  auto-generated code from the colorThresholder app. The
colorspace and
%  range for each channel of the colorspace were set within the
app. The
%  segmentation mask is returned in BW, and a composite of the
mask and
%  original RGB images is returned in maskedRGBImage.
% Auto-generated by colorThresholder app on 01-Nov-2020
%------------------------------------------------------
% Convert RGB image to chosen color space
I = rgb2hsv(RGB);
% Define thresholds for channel 1 based on histogram settings
channel1Min = 0.000;
channel1Max = 0.070;
% Secondary threshold for channel 1
chan1Min    = 0.800;
chan1Max    = 1.000;
% Define thresholds for channel 2 based on histogram settings
channel2Min = 0.100;
channel2Max = 1.000;
% Define thresholds for channel 3 based on histogram settings
channel3Min = 0.000;
channel3Max = 1.000;
% Create mask based on chosen histogram thresholds
sliderBW = ((I(:,:,1) >= channel1Min ) & (I(:,:,1) <= channel1Max)
| ...
    (I(:,:,1) >= chan1Min) & (I(:,:,1) <= chan1Max)) & ...
    (I(:,:,2) >= channel2Min ) & (I(:,:,2) <= channel2Max) & ...
    (I(:,:,3) >= channel3Min ) & (I(:,:,3) <= channel3Max);
```

```
BW = sliderBW;
% Initialize output masked image based on input image.
maskedRGBImage = RGB;
% Set background pixels where BW is false to zero.
maskedRGBImage(repmat(~BW,[1 1 3])) = 0;

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

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

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