

LAPORAN PROYEK AKHIR
KLASIFIKASI KEMATANGAN BUAH TOMAT MENGGUNAKAN
METODE K-NEAREST NEIGHBOR (KNN)



DISUSUN OLEH :

- | | | |
|----|---------------------------|-----------|
| 1. | ANIS MAGFIROH | 123200005 |
| 2. | NOVIA ADELIA | 123200023 |
| 3. | ARDHIAN KUSUMAYUDA | 123200144 |
| 4. | MUHAMMAD RAEHAN PARIKESIT | 123200149 |

PROGRAM STUDI INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
YOGYAKARTA
2022

KATA PENGANTAR

Puji syukur kami panjatkan atas kehadiran Tuhan Yang Maha Esa, yang telah memberikan rahmat, inayah dan hidayah-Nya sehingga kami dapat menyelesaikan laporan proyek akhir mata kuliah Pengolahan Citra yang berjudul “Klasifikasi Kematangan Buah Tomat Menggunakan Metode K-Nearest Neighbor (KNN)” dengan baik dan lancar.

Tidak lupa ucapan terima kasih kepada bapak Shoffan Saifullah, S.Kom., M.Kom. selaku dosen pengampu yang selalu membimbing dan mengajari kami dalam melaksanakan kuliah dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun kami harapkan untuk menyempurnakan laporan akhir ini.

Akhir kata, kami berharap semoga hasil proyek dan laporan ini dapat bermanfaat untuk meningkatkan wawasan dan ilmu pengetahuan bagi semua pihak. Kami juga mengucapkan permohonan maaf apabila dilaporkan ini terdapat kesalahan maupun kekurangan dalam penyusunan laporan.

Yogyakarta, 22 November 2022

Penyusun

DAFTAR ISI

HALAMAN COVER	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan Proyek Akhir	1
BAB II PEMBAHASAN	2
2.1. Matlab	2
2.1. K-Nearest Neighbor (KNN)	3
BAB III METODE	4
3.1 Perencanaan dan tahap implementasi	4
3.2 Hasil dan Pembahasan	5
3.2.1 Rancangan Program	5
3.2.2 Pembahasan	6
3.2.3 Listing Program	7
3.2.2 Output Program	19
3.2.3 Uji Program	20
BAB IV KESIMPULAN DAN SARAN	21
4.1 Kesimpulan	21
4.2 Saran	21
DAFTAR PUSTAKA	22

BAB I

PENDAHULUAN

1.1. Latar Belakang

Tomat merupakan tanaman sayuran buah yang sangat dibutuhkan oleh manusia untuk memenuhi kebutuhan hidupnya. Hal ini disebabkan oleh karena kandungan gizi buah tomat yang terdiri dari vitamin dan mineral sangat berguna untuk mempertahankan kesehatan dan mencegah penyakit. Tanaman tomat termasuk tanaman semusim ordo Solanales, family Solanaceae, genus Lycopersicon, spesies Lycopersicon esculentum Mill. Tomat sangat bermanfaat bagi tubuh karena mengandung vitamin dan mineral yang diperlukan untuk pertumbuhan dan kesehatan. Buah tomat juga mengandung karbohidrat, protein, lemak dan kalori.

Prospek pengembangan tomat sangat menjanjikan, sehingga penelitian penelitian yang dilakukan dengan menggunakan tomat sebagai tanaman indikator. Namun penelitian yang dilakukan masih bersifat uji coba beberapa macam pupuk yang baru dirilis dan kontribusinya dalam meningkatkan produksi saja. Di Indonesia, kebutuhan pasar sayuran terutama buah tomat dari tahun ke tahun meningkat. Hal ini tercermin dari angka produksi tomat, berdasarkan data dari Badan Pusat Statistik dan Direktorat Jenderal Hortikultura (2016) produksi tomat pada tahun 2016 sebesar 883.233 ton. Angka produksi tersebut menggambarkan bahwa peluang bisnis tomat masih terbuka lebar.

Hal ini juga memacu perkembangan teknologi yang dapat digunakan pada buah tomat. Yaitu suatu teknologi untuk mendeteksi kematangan pada buah tomat. Hal ini dapat membantu segala pihak terutama para petani untuk mengetahui keadaan tingkat kematangan buah tomatnya. Oleh karena itu kami membangun suatu aplikasi pengolahan citra yang berjudul “Klasifikasi Kematangan Buah Tomat” dengan metode KNN (K-Nearest Neighbor).

1.2. Tujuan Proyek Akhir

Dalam penelitian untuk proyek akhir kami memiliki tujuan, yaitu:

- a. Merancang aplikasi GUI Matlab untuk mengidentifikasi dan mengklasifikasi tingkat kematangan pada citra buah tomat menggunakan metode K-Nearest Neighbor (KNN) berdasarkan warna buah. Dengan aplikasi GUI Matlab akan didapatkan informasi yang menyatakan jenis dari buah tomat apakah buah tersebut matang atau mentah.

BAB II

PEMBAHASAN

2.1. Matlab

Matlab merupakan salah satu bahasa pemrograman yang dikembangkan oleh MathWorks. Matlab tidak hanya berfungsi sebagai alat visualisasi, yang berhubungan langsung dengan ilmu Matematika. Oleh karena itu Matlab semakin banyak digunakan oleh para programmer yang menghendaki kepraktisan dalam membuat program (Paulus, 2007).

Matlab adalah sebuah bahasa (pemrograman dengan unjuk kerja tinggi (high-performance) untuk komutasi teknis, yang mengintegrasikan komputasi, visualisasi, dan pemrograman di dalam lingkungan yang mudah penggunaanya dalam memecahkan persoalan dengan solusinya yang dinyatakan dengan notasi matematika.(Wijaya, Marvin Ch,2007). Penggunaan Matlab, yaitu :

1. Matematika dan komputasi
2. Pengembangan algoritma
3. Pemodelan, simulasi, dan pembuatan ‘prototipe’
4. Analisis data, eksplorasi dan visualisasi
5. Grafik untuk sains dan teknik
6. Pengembangan aplikasi, termasuk pembuatan antarmuka grafis untuk pengguna (Graphical User Interface).

Matlab adalah sebuah sistem interaktif yang menggunakan elemen data dasarnya adalah array yang tidak membutuhkan dimensi. Nama Matlab merupakan singkatan dari ‘matrix laboratory’. Pada awalnya Matlab dibuat untuk mempermudah pengembangan perangkat lunak berbasis matriks oleh proyek LINPACK dan EISPACK. Fitur – fitur Matlab untuk penyelesaian spesifik disebut ‘toolboxes’. Toolboxes adalah koleksi komprehensif dari fungsi – fungsi Matlab (M-file) yang memperlebar lingkungan Matlab dalam menyelesaikan kelas – kelas tertentu dari permasalahan. Beberapa toolbar yang tersedia meliputi bidang : pengolahan sinyal, sistem kendali, jaringan syaraf (neural network), logika ‘fuzzy’, wavelet, simulasi dan lain sebagainya (Wijaya, Marvin Ch,2007).

2.1. K-Nearest Neighbor (KNN)

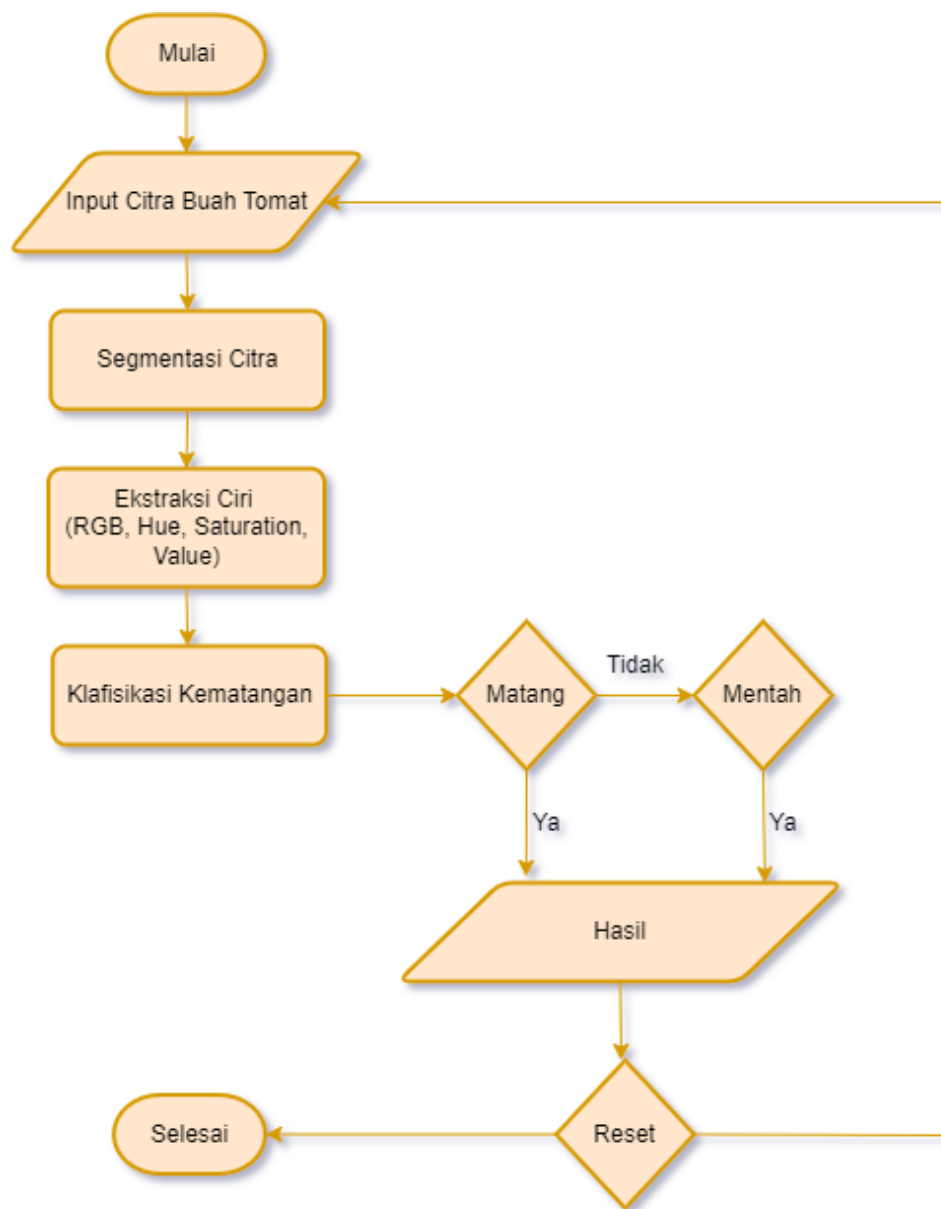
Algoritma K-Nearest Neighbor (KNN) merupakan sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pelatihan yang jaraknya paling dekat dengan objek. Data pelatihan diproyeksikan ke ruang berdimensi banyak, dimana masing – masing dimensi merepresentasikan fitur dari data. Algoritma KNN mencakup metode yang menggunakan algoritma terkontrol. Perbedaan antara pelatihan terawasi dan pelatihan tidak terawasi adalah bahwa tujuan pelatihan terbimbing adalah menemukan pola baru dalam data dengan menggabungkan pola data yang ada dengan data baru. Di sisi lain, dalam pelatihan tanpa pengawasan, informasi masih belum memiliki pola, dan tujuan pelatihan tanpa pengawasan adalah untuk menemukan pola dalam informasi tersebut. Tujuan dari algoritma KNN adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan contoh pelatihan. Saat hasil sampel uji baru diklasifikasikan berdasarkan kategori KNN terbanyak. Dalam proses klasifikasi, algoritma ini tidak menggunakan model untuk pencocokan dan hanya mengandalkan memori. Algoritma KNN menggunakan klasifikasi tetangga sebagai nilai prediksi dari sampel data uji baru. Jarak yang digunakan adalah jarak Euclidean. Jarak Euclidean adalah jarak yang paling umum digunakan dalam data numerik. Algoritma KNN adalah algoritma yang menentukan suatu nilai jarak saat data testing dengan data training didasarkan pada nilai terkecil dari nilai tetangga terdekat

BAB III

METODE

3.1 Perencanaan dan tahap implementasi

Tahap implementasi pada identifikasi kematangan buah tomat dengan menggunakan metode K-Nearest Neighbor (KNN) memiliki beberapa urutan tahapan sebagai berikut



Gambar 3.1.1 Tahapan Identifikasi

Diagram alir diatas merupakan gambaran penjelasan dari tahapan identifikasi buah tomat. Citra yang sudah diinputkan nantinya akan memiliki hasil identifikasi berupa buah tomat matang atau buah tomat mentah.

Proses metode K-Nearest Neighbor (KNN) dilakukan dengan membandingkan data uji dengan data latih (*training*). Tahapan awal yang dilakukan adalah dengan menentukan jumlah nilai K. Berikut merupakan gambar diagram penjelasannya

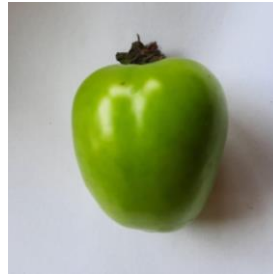


Gambar 3.1.2 Metode KNN

3.2 Hasil dan Pembahasan

3.2.1 Rancangan Program

Program ini dibuat dengan tujuan untuk mengidentifikasi kematangan buah tomat dengan tepat menggunakan metode K-Nearest Neighbor (KNN). Adapun data citra yang dibutuhkan untuk di input ke dalam program ini sebagai berikut



Gambar 3.2.1 Citra Tomat Mentah

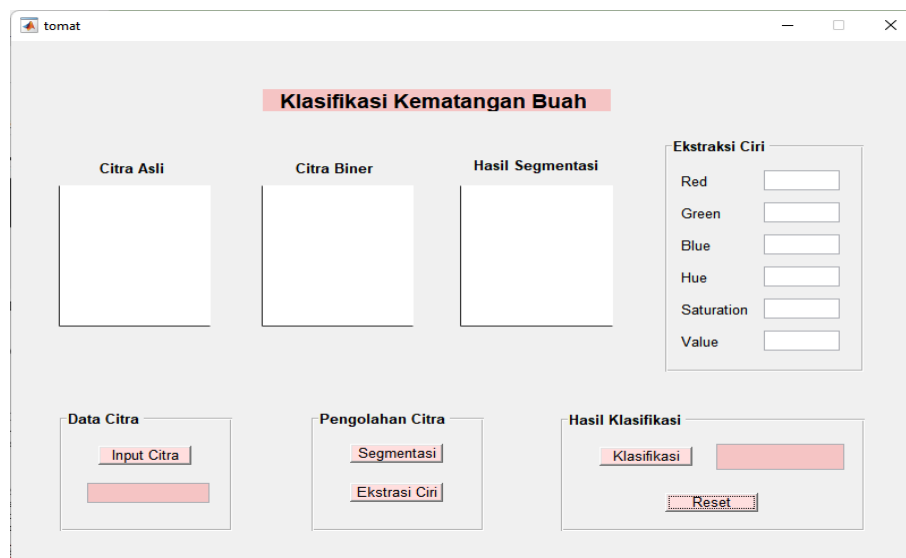
Setelah citra pada gambar 3.2.1 diinputkan selanjutnya citra akan di segmentasi dan dihitung nilai ekstraksi cirinya. Kemudian diklasifikasikan berdasar pada nilai ekstraksi cirinya dan dioutputkan hasil “Tomat Mentah”.



Gambar 3.2.2 Hasil Segmentasi & Ekstraksi Ciri

3.2.2 Pembahasan

Penjelasan mengenai tampilan dan macam - macam push button yang terdapat pada GUI.



Gambar 3.2.3 Tampilan GUI

Keterangan push button :

- a) Input Citra : User menginputkan citra.
- b) Segmentasi : Program melakukan segmentasi pada citra yang diinputkan.
- c) Ekstraksi Ciri : Hasil segmentasi citra di hitung ekstraksi cirinya.
- d) Klasifikasi : Citra diklasifikasikan berdasar pada nilai ekstrak cirinya.
- e) Reset : Mereset ulang program dan menginputkan kembali citra yang ingin di identifikasi.

3.2.3 Listing Program

A. Listing program untuk tahapan pelatihan (*training*)

```
clc; clear; close all; warning off all;

%%% Matang
% membaca file citra
nama_folder = 'Data Latih/Matang';
nama_file = dir(fullfile(nama_folder, '*.jpg'));
jumlah_file = numel(nama_file);
% inisialisasi variabel ciri_matang
ciri_matang = zeros(jumlah_file,6);
target_matang = cell(jumlah_file,1);

for n = 1:jumlah_file
    % membaca citra RGB
    Img =
im2double(imread(fullfile(nama_folder,nama_file(n).name)));
    % konversi citra RGB menjadi grayscale
    Img_gray = rgb2gray(Img);
    % konversi citra grayscale menjadi biner
    bw = imbinarize(Img_gray);
    % operasi morfologi
    bw = imcomplement(bw);
    bw = imfill(bw, 'holes');
    bw = bwareaopen(bw,100);
    % ekstraksi ciri warna RGB
    R = Img(:,:,1);
    G = Img(:,:,2);
    B = Img(:,:,3);
    R(~bw) = 0;
    G(~bw) = 0;
    B(~bw) = 0;
    RGB = cat(3,R,G,B);
    Red = sum(sum(R))/sum(sum(bw));
    Green = sum(sum(G))/sum(sum(bw));
    Blue = sum(sum(B))/sum(sum(bw));
    % ekstraksi ciri warna HSV
    HSV = rgb2hsv(Img);
    H = HSV(:,:,1);
    S = HSV(:,:,2);
    V = HSV(:,:,3);
    H(~bw) = 0;
    S(~bw) = 0;
    V(~bw) = 0;
```

```

Hue = sum(sum(H))/sum(sum(bw));
Saturation = sum(sum(S))/sum(sum(bw));
Value = sum(sum(V))/sum(sum(bw));
% ekstraksi ciri tekstur GLCM
Img_gray(~bw) = 0;
% mengisi hasil ekstraksi ciri pada variabel ciri_matang
ciri_matang(n,1) = Red;
ciri_matang(n,2) = Green;
ciri_matang(n,3) = Blue;
ciri_matang(n,4) = Hue;
ciri_matang(n,5) = Saturation;
ciri_matang(n,6) = Value;
% mengisi nilai target pada variabel target_matang
target_matang{n} = 'matang';
end

%%% Mentah
% membaca file citra
nama_folder = 'Data Latih/Mentah';
nama_file = dir(fullfile(nama_folder, '*.jpg'));
jumlah_file = numel(nama_file);
% inisialisasi variabel ciri_mentah
ciri_mentah = zeros(jumlah_file,6);
target_mentah = cell(jumlah_file,1);

for n = 1:jumlah_file
    % membaca citra RGB
    Img =
im2double(imread(fullfile(nama_folder,nama_file(n).name)));
    % konversi citra RGB menjadi grayscale
    Img_gray = rgb2gray(Img);
    % konversi citra grayscale menjadi biner
    bw = imbinarize(Img_gray);
    % operasi morfologi
    bw = imcomplement(bw);
    bw = imfill(bw, 'holes');
    bw = bwareaopen(bw,100);
    % ekstraksi ciri warna RGB
    R = Img(:,:,1);
    G = Img(:,:,2);
    B = Img(:,:,3);
    R(~bw) = 0;
    G(~bw) = 0;
    B(~bw) = 0;
    RGB = cat(3,R,G,B);
    Red = sum(sum(R))/sum(sum(bw));
    Green = sum(sum(G))/sum(sum(bw));
    Blue = sum(sum(B))/sum(sum(bw));
    % ekstraksi ciri warna HSV
    HSV = rgb2hsv(Img);
    H = HSV(:,:,1);
    S = HSV(:,:,2);
    V = HSV(:,:,3);
    H(~bw) = 0;
    S(~bw) = 0;
    V(~bw) = 0;
    Hue = sum(sum(H))/sum(sum(bw));
    Saturation = sum(sum(S))/sum(sum(bw));
    Value = sum(sum(V))/sum(sum(bw));

```

```

    % mengisi hasil ekstraksi ciri pada variabel ciri_mentah
    ciri_mentah(n,1) = Red;
    ciri_mentah(n,2) = Green;
    ciri_mentah(n,3) = Blue;
    ciri_mentah(n,4) = Hue;
    ciri_mentah(n,5) = Saturation;
    ciri_mentah(n,6) = Value;

    % mengisi nilai target pada variabel target_mentah
    target_mentah{n} = 'mentah';
end

% menyusun variabel ciri_latih dan target_latih
ciri_latih = [ciri_matang;ciri_mentah];
target_latih = [target_matang;target_mentah];

% melakukan pelatihan menggunakan algoritma K-NN
Model = fitcknn(ciri_latih, target_latih,'NumNeighbors', 10);

% membaca kelas keluaran hasil penelitian
hasil_latih = predict(Model, ciri_latih);

save Model Model

```

B. Listing program untuk tahapan pengujian (*testing*)

```

clc; clear; close all; warning off all;

%%% Matang
% membaca file citra
nama_folder = 'Data Uji/Matang';
nama_file = dir(fullfile(nama_folder,'*.jpg'));
jumlah_file = numel(nama_file);
% inisialisasi variabel ciri_matang
ciri_matang = zeros(jumlah_file,6);
target_matang = cell(jumlah_file,1);

for n = 1:jumlah_file
    % membaca citra RGB
    Img =
    im2double(imread(fullfile(nama_folder,nama_file(n).name)));
    % konversi citra RGB menjadi grayscale
    Img_gray = rgb2gray(Img);
    % konversi citra grayscale menjadi biner
    bw = imbinarize(Img_gray);
    % operasi morfologi
    bw = imcomplement(bw);
    bw = imfill(bw,'holes');
    bw = bwareaopen(bw,100);
    % ekstraksi ciri warna RGB
    R = Img(:,:,1);
    G = Img(:,:,2);
    B = Img(:,:,3);
    R(~bw) = 0;
    G(~bw) = 0;
    B(~bw) = 0;
    RGB = cat(3,R,G,B);

```

```

Red = sum(sum(R))/sum(sum(bw));
Green = sum(sum(G))/sum(sum(bw));
Blue = sum(sum(B))/sum(sum(bw));
% ekstraksi ciri warna HSV
HSV = rgb2hsv(Img);
H = HSV(:,:,1);
S = HSV(:,:,2);
V = HSV(:,:,3);
H(~bw) = 0;
S(~bw) = 0;
V(~bw) = 0;
Hue = sum(sum(H))/sum(sum(bw));
Saturation = sum(sum(S))/sum(sum(bw));
Value = sum(sum(V))/sum(sum(bw));
% ekstraksi ciri tekstur GLCM
Img_gray(~bw) = 0;
% mengisi hasil ekstraksi ciri pada variabel ciri_matang
ciri_matang(n,1) = Red;
ciri_matang(n,2) = Green;
ciri_matang(n,3) = Blue;
ciri_matang(n,4) = Hue;
ciri_matang(n,5) = Saturation;
ciri_matang(n,6) = Value;
% mengisi nilai target pada variabel target_matang
target_matang{n} = 'matang';
end

%%% Mentah
% membaca file citra
nama_folder = 'Data Latih/Mentah';
nama_file = dir(fullfile(nama_folder, '*.jpg'));
jumlah_file = numel(nama_file);
% inisialisasi variabel ciri_mentah
ciri_mentah = zeros(jumlah_file,6);
target_mentah = cell(jumlah_file,1);

for n = 1:jumlah_file
    % membaca citra RGB
    Img =
im2double(imread(fullfile(nama_folder,nama_file(n).name)));
    % konversi citra RGB menjadi grayscale
    Img_gray = rgb2gray(Img);
    % konversi citra grayscale menjadi biner
    bw = imbinarize(Img_gray);
    % operasi morfologi
    bw = imcomplement(bw);
    bw = imfill(bw,'holes');
    bw = bwareaopen(bw,100);
    % ekstraksi ciri warna RGB
    R = Img(:,:,1);
    G = Img(:,:,2);
    B = Img(:,:,3);
    R(~bw) = 0;
    G(~bw) = 0;
    B(~bw) = 0;
    RGB = cat(3,R,G,B);
    Red = sum(sum(R))/sum(sum(bw));
    Green = sum(sum(G))/sum(sum(bw));
    Blue = sum(sum(B))/sum(sum(bw));
    % ekstraksi ciri warna HSV

```

```

    HSV = rgb2hsv(Img);
    H = HSV(:,:,1);
    S = HSV(:,:,2);
    V = HSV(:,:,3);
    H(~bw) = 0;
    S(~bw) = 0;
    V(~bw) = 0;
    Hue = sum(sum(H))/sum(sum(bw));
    Saturation = sum(sum(S))/sum(sum(bw));
    Value = sum(sum(V))/sum(sum(bw));

    % mengisi hasil ekstraksi ciri pada variabel ciri_mentah
    ciri_mentah(n,1) = Red;
    ciri_mentah(n,2) = Green;
    ciri_mentah(n,3) = Blue;
    ciri_mentah(n,4) = Hue;
    ciri_mentah(n,5) = Saturation;
    ciri_mentah(n,6) = Value;

    % mengisi nilai target pada variabel target_mentah
    target_mentah{n} = 'mentah';
end

% menyusun variabel ciri_uji dan target_uji
ciri_uji = [ciri_matang;ciri_mentah];
target_uji = [target_matang;target_mentah];

%Memanggil model k-nn hasil penelitian
load Model

% membaca kelas keluaran hasil pengujian
hasil_uji = predict(Model, ciri_uji);

```

C. Listing program untuk membuat tampilan GUI (*Graphical User Interface*)

```

function varargout = tomat(varargin)
% TOMAT MATLAB code for tomat.fig
%     TOMAT, by itself, creates a new TOMAT or raises the existing
%     singleton*.
%
%     H = TOMAT returns the handle to a new TOMAT or the handle to
%     the existing singleton*.
%
%     TOMAT('CALLBACK',hObject,eventData,handles,...) calls the
local
%     function named CALLBACK in TOMAT.M with the given input
arguments.
%
%     TOMAT('Property','text11',...) creates a new TOMAT or raises
the
%     existing singleton*. Starting from the left, property
text11 pairs are
%     applied to the GUI before tomat_OpeningFcn gets called. An
%     unrecognized property name or invalid text11 makes property
application
%     stop. All inputs are passed to tomat_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 tomat

% Last Modified by GUIDE v2.5 22-Nov-2022 08:14:19

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

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

% Update handles structure
guidata(hObject, handles);

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

% --- Outputs from this function are returned to the command line.
function varargout = tomat_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 edit3_Callback(hObject, eventdata, handles)
% hObject      handle to class (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 class as text
%         str2double(get(hObject,'String')) returns contents of
class as a double

% --- Executes during object creation, after setting all
properties.
function class_CreateFcn(hObject, eventdata, handles)
% hObject      handle to class (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 class.
function class_Callback(hObject, eventdata, handles)
% hObject      handle to class (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
ciri_uji = handles.ciri_uji;
% memanggil model K-NN hasil pelatihan
load Model

% membaca kelas keluaran hasil penelitian
hasil_uji = predict(Model, ciri_uji);

set(handles.klasifikasi, 'String', hasil_uji);

% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata, handles)
% hObject      handle to reset (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
set(handles.file, 'String', "");
set(handles.red, 'String', "");
set(handles.green, 'String', "");
set(handles.blue, 'String', "");
set(handles.hue, 'String', "");
set(handles.saturation, 'String', "");
set(handles.value, 'String', "");
set(handles.klasifikasi, 'String', "");

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',[])

% --- Executes on button press in segmentasi.
function segmentasi_Callback(hObject, eventdata, handles)
% hObject      handle to segmentasi (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% memanggil variabel Img yang ada di lokasi handles
Img = handles.Img;
% konversi citra RGB menjadi grayscale
Img_gray = rgb2gray(Img);
% konversi citra grayscale menjadi biner
bw = imbinarize(Img_gray);
% operasi morfologi
bw = imcomplement(bw);
bw = imfill(bw,'holes');
bw = bwareaopen(bw,100);
% menampilkan citra biner hasil segmentasi pada axes2
axes(handles.axes2)
imshow(bw)

% ekstraksi komponen RGB
R = Img(:,:,1);
G = Img(:,:,2);
B = Img(:,:,3);
% mengubah nilai background menjadi nol
R(~bw) = 0;
G(~bw) = 0;
B(~bw) = 0;
RGB = cat(3,R,G,B);
% menampilkan citra RGB hasil segmentasi pada axes3
axes(handles.axes3)
imshow(RGB)

% menyimpan variabel Img_gray & bw pada lokasi handles
handles.Img_gray = Img_gray;
handles.bw = bw;
guidata(hObject, handles)

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

% memanggil variabel Img, Img_gray, & bw yang ada di lokasi handles
Img = handles.Img;
Img_gray = handles.Img_gray;
bw = handles.bw;
% ekstraksi ciri warna RGB
R = Img(:,:,1);
G = Img(:,:,2);

```

```

B = Img(:,:,3);
R(~bw) = 0;
G(~bw) = 0;
B(~bw) = 0;
Red = sum(sum(R))/sum(sum(bw));
Green = sum(sum(G))/sum(sum(bw));
Blue = sum(sum(B))/sum(sum(bw));

% ekstraksi ciri warna HSV
HSV = rgb2hsv(Img);
H = HSV(:,:,1);
S = HSV(:,:,2);
V = HSV(:,:,3);
H(~bw) = 0;
S(~bw) = 0;
V(~bw) = 0;
Hue = sum(sum(H))/sum(sum(bw));
Saturation = sum(sum(S))/sum(sum(bw));
Value = sum(sum(V))/sum(sum(bw));

% menyusun variabel ciri uji
ciri_uji = [Red, Green, Blue, Hue, Saturation, Value];

set(handles.red, 'String', num2str(Red));
set(handles.green, 'String', num2str(Green));
set(handles.blue, 'String', num2str(Blue));
set(handles.hue, 'String', num2str(Hue));
set(handles.saturation, 'String', num2str(Saturation));
set(handles.value, 'String', num2str(Value));

% menyimpan variabel ciri_uji pada lokasi handles
handles.ciri_uji = ciri_uji;
guidata(hObject, handles)

% --- Executes on button press in input_citra.
function input_citra_Callback(hObject, eventdata, handles)
% hObject      handle to input_citra (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% memanggil menu "browser file"
[nama_file, nama_folder] = uigetfile('*.jpg');

% jika ada nama file yang dipilih maka akan mengeksekusi perintah
di bawah
if ~isequal(nama_file,0)
    % membaca file citra rgb
    Img = im2double(imread(fullfile(nama_folder, nama_file)));

    axes(handles.axes1)
    imshow(Img)

    set(handles.file, 'String', nama_file)

    handles.Img = Img;
    guidata(hObject, handles)
else
    return
end

```

```

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

% --- Executes during object creation, after setting all
properties.
function file_CreateFcn(hObject, eventdata, handles)
% hObject      handle to file (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 red_Callback(hObject, eventdata, handles)
% hObject      handle to red (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 red as text
%         str2double(get(hObject,'String')) returns contents of red
as a double

% --- Executes during object creation, after setting all
properties.
function red_CreateFcn(hObject, eventdata, handles)
% hObject      handle to red (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 green_Callback(hObject, eventdata, handles)
% hObject      handle to text7 (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 text7 as text

```

```

%          str2double(get(hObject,'String')) returns contents of
text7 as a double

% --- Executes during object creation, after setting all
properties.
function text7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text7 (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 blue_Callback(hObject, eventdata, handles)
% hObject    handle to text8 (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 text8 as text
%          str2double(get(hObject,'String')) returns contents of
text8 as a double

% --- Executes during object creation, after setting all
properties.
function text8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text8 (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 hue_Callback(hObject, eventdata, handles)
% hObject    handle to text9 (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 text9 as text
%          str2double(get(hObject,'String')) returns contents of
text9 as a double

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

```

```

function text9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to text9 (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 saturation_Callback(hObject, eventdata, handles)
% hObject      handle to text10 (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 text10 as text
%         str2double(get(hObject,'String')) returns contents of
text10 as a double

% --- Executes during object creation, after setting all
properties.
function text10_CreateFcn(hObject, eventdata, handles)
% hObject      handle to text10 (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 value_Callback(hObject, eventdata, handles)
% hObject      handle to text11 (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 text11 as text
%         str2double(get(hObject,'String')) returns contents of
text11 as a double

% --- Executes during object creation, after setting all
properties.
function text11_CreateFcn(hObject, eventdata, handles)
% hObject      handle to text11 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

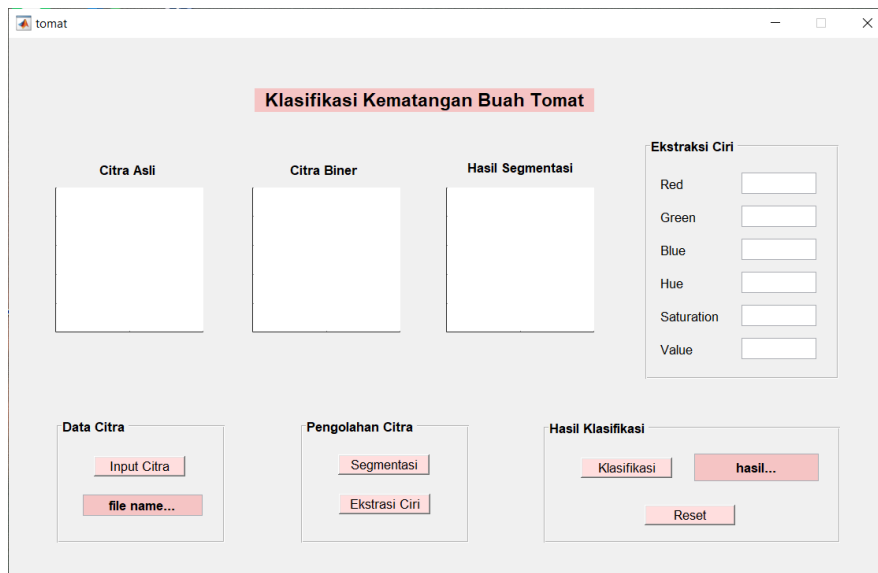
% Hint: edit controls usually have a white background on Windows.

```

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

3.2.4 Output Program

Program yang dibangun membutuhkan data input, yaitu sebuah citra berwarna. Hasil dari listing program yang telah dibangun adalah sebagai berikut.



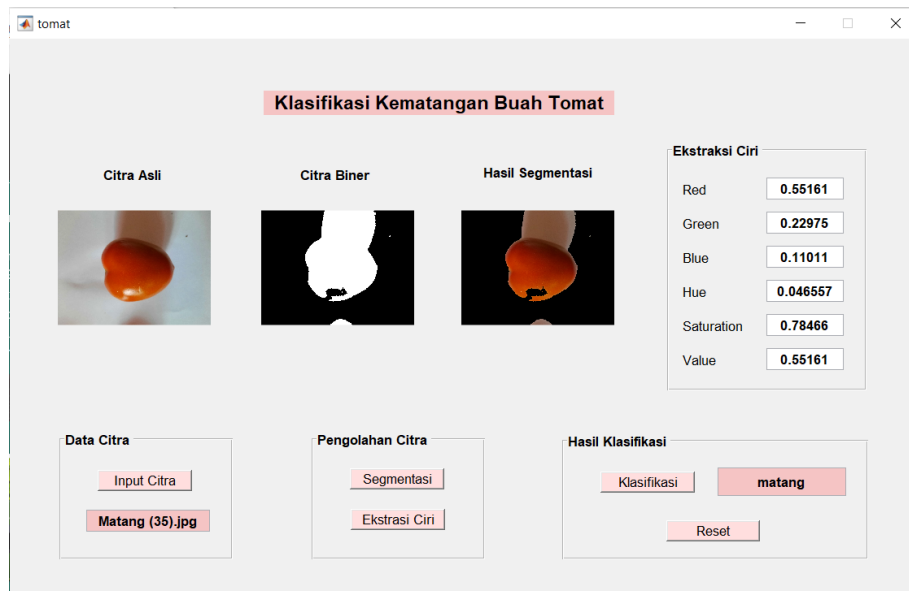
3.2.2.1 Gambar tampilan GUI

Pada program di atas, terdapat beberapa tombol kontrol sebagai berikut.

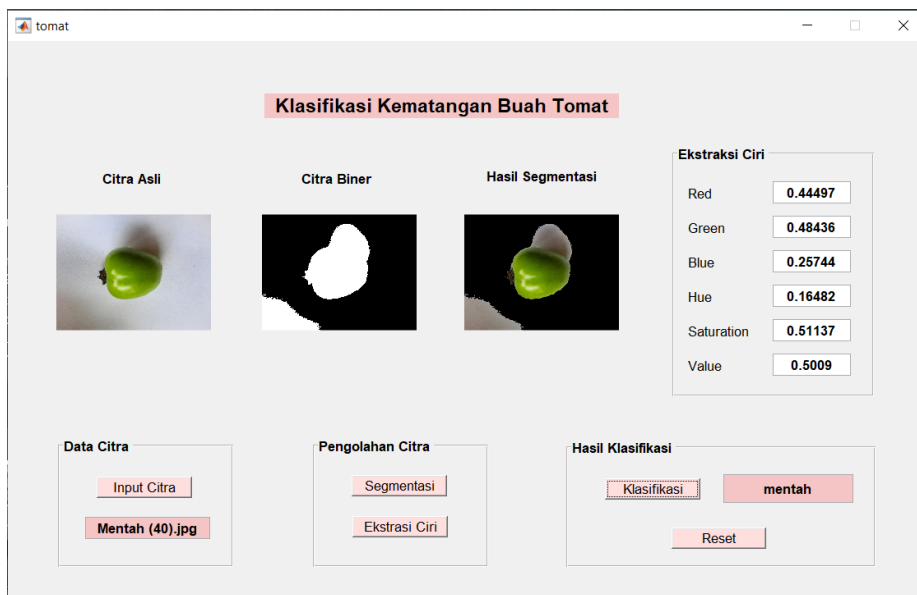
1. Tombol input citra yang berfungsi untuk memasukkan citra yang akan diuji.
2. Tombol segmentasi yang berfungsi untuk memproses segmentasi dari citra yang telah diinput.
3. Tombol ekstraksi ciri yang berfungsi untuk memproses ekstraksi ciri dari citra yang telah diinput.
4. Tombol klasifikasi yang berfungsi untuk menampilkan hasil klasifikasi citra yang telah diinput.
5. Tombol reset yang berfungsi untuk mengatur ulang semua tampilan.

3.2.5 Uji Program

Berikut adalah uji program terhadap beberapa data citra berwarna.



3.2.3.1 Gambar hasil running program pada citra berwarna Matang (35).jpg



3.2.3.1 Gambar hasil running program pada citra berwarna Mentah (40).jpg

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Aplikasi yang dirancang dengan menggunakan MATLAB GUI dan dengan metode K-Nearest Neighbor ini mampu mengidentifikasi dan mengklasifikasi tingkat kematangan buah tomat dengan tingkat mentah dan matang. Aplikasi ini juga mampu mengolah citra yang diinput menjadi citra biner dan citra hasil segmentasi serta ekstraksi cirinya. Dengan begitu, pengguna aplikasi ini dapat secara pasti tanpa ragu-ragu untuk mengetahui apakah buah tomat yang dipilih telah matang atau masih mentah.

4.2 Saran

Saran untuk aplikasi pengolahan citra menggunakan metode KNN berikutnya adalah agar bisa memproses dengan lebih baik dan akurat. Dan selain itu juga agar aplikasi ini bisa dibuat GUI yang lebih baik dan interaktif kembali agar dapat digunakan oleh banyak orang dengan segala kemudahannya.

DAFTAR PUSTAKA

- Liantoni, F. (2016). Klasifikasi Daun Dengan Perbaikan Fitur Citra Menggunakan Metode K-Nearest Neighbor. Ultimatics : Jurnal Teknik Informatika, 7(2), 98-104.**
<https://doi.org/10.31937/ti.v7i2.356>
- Pratama R, Fuad A, Tempola F (2019). DETEKSI KEMATANGAN BUAH TOMAT BERDASARKAN FITUR WARNA MENGGUNAKAN METODE TRANSFORMASI RUANG WARNA HIS. Jurnal Informatika dan Ilmu Komputer (JIKO), 2(2), 81-86.**
<http://dx.doi.org/10.33387/jiko>

PEMBAGIAN TUGAS

NAMA	TUGAS
Anis Magfiroh	Pembuatan main, pembuatan data model dan pembuatan laporan
Novia Adelia	Pembuatan data latih, data uji dan GUI dan pembuatan laporan
Ardhian Kusumayuda	Pembuatan main, pembuatan data model dan pembuatan laporan
Muhammad Raehan Parikesit	Pembuatan main program, pembuatan GUI dan pembuatan laporan

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : **ANIS MAGFIROH**

NIM : 123200005

Kelas : Pengolahan Citra IF - A

Tugas : Proyek Akhir

Dengan ini menyatakan mahasiswa tersebut **telah** menyelesaikan project akhir Mata Kuliah Pengolahan Citra sesuai tugasnya masing-masing. Demikian surat ini saya buat untuk dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 22 November 2022

Mahasiswa,



(ANIS MAGHFIROH)

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : **NOVIA ADELIA**

NIM : 123200023

Kelas : Pengolahan Citra IF - A

Tugas : Proyek Akhir

Dengan ini menyatakan mahasiswa tersebut **telah** menyelesaikan project akhir Mata Kuliah Pengolahan Citra sesuai tugasnya masing-masing. Demikian surat ini saya buat untuk dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 22 November 2022

Mahasiswa,

A handwritten signature in black ink, appearing to read 'Novia Adelia', with a horizontal line drawn underneath it.

(NOVIA ADELIA)

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : **ARDHIAN KUSUMAYUDA**

NIM : 123200144

Kelas : Pengolahan Citra IF - A

Tugas : Proyek Akhir

Dengan ini menyatakan mahasiswa tersebut **telah** menyelesaikan project akhir Mata Kuliah Pengolahan Citra sesuai tugasnya masing-masing. Demikian surat ini saya buat untuk dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 22 November 2022

Mahasiswa,



(**ARDHIAN KUSUMAYUDA**)

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : **MUHAMMAD RAEHAN PARIKESIT**

NIM : 123200149

Kelas : Pengolahan Citra IF - A

Tugas : Proyek Akhir

Dengan ini menyatakan mahasiswa tersebut **telah** menyelesaikan project akhir Mata Kuliah Pengolahan Citra sesuai tugasnya masing-masing. Demikian surat ini saya buat untuk dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 22 November 2022

Mahasiswa,



(**MUHAMMAD RAEHAN PARIKESIT**)