

Deteksi Kematangan Buah Tomat Sayur Berdasarkan Fitur Warna HSV Menggunakan Metode Naïve Bayes

Proposal Workshop Pengolahan Citra dan Vision



Nama Anggota :

Moh. Ardias Ade Aga	(E41192114)
Nadea Ajeng Safitri	(E41191903)
Alga Saputra	(E41192329)
Muhammad Riadi Prasetyo	(E41192428)
Indra Prasetya	(E41192478)

**Proyek Workshop Pengolahan Citra dan Vision
Program Studi D4 Teknik Informatika
Jurusan Teknologi Informasi
Politeknik Negeri Jember**

2022

DAFTAR ISI

DAFTAR ISI	i
DAFTAR GAMBAR	ii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pustaka	2
BAB 2 RANCANGAN.....	5
2.1 Data.....	5
2.2 Rancangan Sistem	5
BAB 3 HASIL	7
3.1 Pengumpulan Data	7
3.2 Pelatihan Data.....	7
3.3 Proses Naïve Bayes.....	13
3.4 Pembuatan Tampilan Gui.....	19
3.5 Pembuatan Source Code dari Tampilan Gui.....	21
3.4.1 Tombol Buka Gambar.....	21
3.4.2 Tombol Proses	22
3.4.3 Tombol Ekstraksi.....	23
3.4.4 Tombol Klasifikasi.....	25
3.4.5 Tombol Reset	26
DAFTAR PUSTAKA.....	27

DAFTAR GAMBAR

Gambar 1. 1 Rumus Hue	3
Gambar 1. 2 Rumus Saturation	3
Gambar 1. 3 Rumus Value	3
Gambar 2. 1 Flowchart System.....	5
Gambar 3. 1 Data Primer Buah Tomat	7
Gambar 3. 2 Source Code untuk Membaca File	8
Gambar 3. 3 Source Code untuk Mengola Citra	9
Gambar 3. 4 Source Code untuk Melakukan Klasifikasi Naïve Bayes	10
Gambar 3. 5 Syntax Proses Naïve Bayes.....	13
Gambar 3. 6 Nilai H,S,V dari Matlab	14
Gambar 3. 7 nilai H, S, V Salinan dari Matlab	15
Gambar 3. 8 Atribut Hue dengan Target Kelas Matang.....	16
Gambar 3. 9 Nilai Mean dan Standar Deviasi	17
Gambar 3. 10 Hasil Distribusi Gaussian.....	18
Gambar 3. 11 MEnghitung Kematangan Citra Buah Tomat Sayur	19
Gambar 3. 12 Open Guide pada Matlab	20
Gambar 3. 13 Desain Guide pada Matlab.....	20
Gambar 3. 14 Tampilan Aksi Tombol Buka Gambar	21
Gambar 3. 15 Source Code Tombol Buka Gambar.....	21
Gambar 3. 16 Tampilan Aksi Tombol Proses	22
Gambar 3. 17 Source Code Tombol Proses.....	23
Gambar 3. 18 Tampilan Aksi Tombol Ekstraksi	23
Gambar 3. 19 Source Code Tombol Ekstraksi	24
Gambar 3. 20 Tampilan Aksi Tombol Klasifikasi.....	25
Gambar 3. 21 Source Code Tombol Klasifikasi	25
Gambar 3. 22 Tampilan Aksi Tombol Reset.....	26
Gambar 3. 23 Source Code Tombol Reset.....	26

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Tomat merupakan salah satu hasil pertanian yang memiliki tingkat produksi tinggi. Tingginya tingkat produksi dan distribusi tomat yang luas mengharuskan petani mampu mengklasifikasikan tingkat kematangan tomat yang dapat mengurangi resiko pembusukan tomat. Proses kematangan tomat membutuhkan waktu yang singkat. Oleh karena itu keakuratan klasifikasi level kematangan tomat sangat penting. Klasifikasi level kematangan tomat saat ini kebanyakan masih menggunakan metode manual, yaitu penilaian secara subyektif dari petani. Kelemahan dari metode ini adalah tingkat akurasi yang rendah karena penilaian yang tidak konsisten. Sehingga, perlu dikembangkan metode otomatis yang dapat meningkatkan tingkat akurasi dengan penilaian yang konsisten.

Indikator yang penting dalam klasifikasi level kematangan tomat adalah warna dari tomat tersebut. Pada pengolahan citra digital, warna memiliki peran berupa informasi yang penting karena dapat terlihat secara visual untuk merepresentasikan kualitas dari citra yang digunakan (Vibhute, 2013).

Pada penelitian ini menggunakan fitur warna *HSV*. Komponen Hue dari citra *HSV* (*Hue, Saturation, Value*) merupakan suatu komponen yang merepresentasikan warna dari panjang gelombang cahaya tampak (merah, jingga, kuning, hijau, biru, ungu). Oleh karena itu, komponen ini dapat digunakan sebagai acuan untuk melakukan pengenalan warna suatu objek pada citra digital. (pamungkas, 2021)

Dalam mengklasifikasi buah tomat menggunakan fitur warna *HSV* kita menggunakan metode Naïve Bayes. Karena metode Naive Bayes merupakan sebuah metode penggolongan berdasarkan probabilitas sederhana dan dirancang untuk dipergunakan dengan asumsi bahwa antar satu kelas dengan kelas yang lain tidak saling tergantung (*independen*). Pada klasifikasi Naive Bayes, proses pembelajaran lebih ditekankan pada mengestimasi probabilitas. Keuntungan dari pendekatan ini yaitu pengklasifikasian akan mendapatkan nilai error yang lebih kecil ketika data set berjumlah besar. (Suliztia, 2021)

1.2 Pustaka

1.2.1 Citra

Citra merupakan istilah lain untuk gambar sebagai salah satu komponen multimedia yang memegang peranan yang sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. (Rendy Pratama, 2019)

1.2.2 Model Warna RGB

RGB merupakan ruang warna yang dimiliki citra yaitu dengan tiga warna primer yaitu red, green, dan blue. Rentang nilai pada penskalaan RGB dalam setiap pixel dimulai dari skala 0 (nol) yaitu hitam hingga skala 255 (dua ratus lima puluh lima) yang mempunyai kedalaman warna yaitu dimulai dari 8 bit. RGB adalah warna dasar yang dijadikan patokan warna secara universal (primary colors) dengan citra jenis berwarna yang bisa diubah warnanya ke dalam kode-kode angka sehingga warna tersebut akan tampil universal. (Rendy Pratama, 2019)

1.2.3 Citra Biner

Citra digital yang hanya memiliki dua kemungkinan nilai piksel hitam dan putih. Citra ini hanya membutuhkan satu bit untuk mewakili nilai setiap piksel dari citra biner. Setiap piksel hanya terdiri dari warna hitam atau putih, karena hanya ada dua warna untuk setiap piksel, maka hanya perlu 1 bit .

1.2.4 Operasi Morfologi

Operasi morfologi menggunakan dua input himpunan yaitu satu citra (umumnya citra biner) dan satu kernel atau Structuring Elements (SE) (Suhandy, 2001). Secara umum terdapat sembilan bentuk structuring elements, dalam penelitian ini mengujikan empat bentuk structuring elements yang umum digunakan serta sesuai dengan citra padi. Structuring Elements (SE) tersebut kemudian digunakan untuk melakukan operasi morfologi citra.

1.2.5 HSV

HSV memiliki 3 karakteristik pokok, yaitu Hue, Saturation dan Value. 1. Hue : warna sebenarnya, seperti merah, violet, dan kuning dan untuk menentukan kemerahan (redness), kehijauan (greeness). 2. Saturation : juga disebut chroma. 3. Value : kecerahan dari warna. Nilainya berkisar antara 0-100 %. Untuk mentransformasi dari RGB ke HSV, harus menentukan koordinatkoordinat R, G, B

yang berurutan seperti merah, hijau, biru dalam ruang warna RGB, dengan max adalah nilai maksimum dari nilai red, green, blue, dan min adalah nilai minimum dari nilai red, green, blue. Untuk memperoleh sudut hue[0,360] yang tepat untuk ruang warna HSV, menggunakan rumus seperti berikut:

$$h(\text{hue}) = \begin{cases} 0, & \text{jika } \max = \min \\ 60^\circ \times \left(\frac{G-B}{\max-\min} \bmod 6 \right), & \text{jika } \max = R \\ 60^\circ \times \left(\frac{B-R}{\max-\min} + 2 \right), & \text{jika } \max = G \\ 60^\circ \times \left(\frac{R-G}{\max-\min} + 4 \right), & \text{jika } \max = B \end{cases}$$

Gambar 1. 1 Rumus Hue

Nilai-nilai untuk s dan v pada HSV didefinisikan sebagai berikut :

$$s(\text{saturation}) = \begin{cases} 0, & \text{jika } \max = \min \\ \frac{\max - \min}{\max}, & \text{otherwise} \end{cases}$$

Gambar 1. 2 Rumus Saturation

$$V(\text{value}) = \max(R, G, B)$$

Gambar 1. 3 Rumus Value

1.2.6 segmentasi

Segmentasi warna yaitu proses yang segmentasi dengan pendekatan daerah dengan menganalisis nilai warna dari tiap piksel pada citra dan membagi citra tersebut sesuai dengan fitur yang diinginkan. Pada metode segmentasi dengan deteksi warna HSV, dilakukan pemilihan sampel piksel sebagai acuan warna untuk membentuk segmen yang diinginkan. Citra digital menggunakan model warna RGB sebagai standar acuan warna, oleh karena itu proses awal pada metode ini memerlukan konversi model warna RGB ke HSV.

1.2.7 Naïve Bayes

Naive Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes. Langkah-langkah untuk pelatihan data :

1. Hitung probabilitas (prior) tiap kelas yang ada.
2. Lalu hitung rata-rata (mean) tiap fitur dan tiap kelas
3. Hitung nilai standar deviasi tiap fitur dan tiap kelas
4. hitung dentitas probabilitas
5. hitung probabilitas masing-masing kelas

Sebuah metode klasifikasi yang berakar pada teorema Bayes. Metode pengklasifikasian dengan menggunakan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dari Naïve Bayes Classifier ini adalah asumsi yang sangat kuat (naïf) akan independensi dari masing-masing kejadian. Naïve Bayes untuk setiap kelas keputusan adalah menghitung probabilitas dengan syarat bahwa kelas keputusan adalah benar. Algoritma ini mengasumsikan bahwa atribut objek adalah independen. Probabilitas yang terlibat dalam memproduksi perkiraan akhir dihitung sebagai jumlah frekuensi dari "master" tabel keputusan. Naive Bayes Classifier bekerja sangat baik dibanding model classifier lainnya. Naïve Bayes Classifier memiliki tingkat akurasi yang lebih baik dibanding model classifier lainnya". Keuntungannya bahwa metode ini hanya membutuhkan jumlah data pelatihan (training data) yang kecil untuk mengestimasi parameter yang diperlukan dalam proses klasifikasi. Karena yang diasumsikan adalah variable independent, maka hanya varian suatu variable dalam suatu kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians.

1.2.8 Algoritma naive bayes

Algoritma Naïve Bayes merupakan salah satu algoritma yang terdapat pada teknik klasifikasi. Naive Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris, yaitu memprediksi peluang dimasa depan berdasarkan pengalaman dimasa sebelumnya sehingga dikenal sebagai Teorema Bayes. Teorema tersebut dikombinasikan dengan Naïve dimana diasumsikan kondisi antar atribut saling bebas. Klasifikasi Naive Bayes diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya.

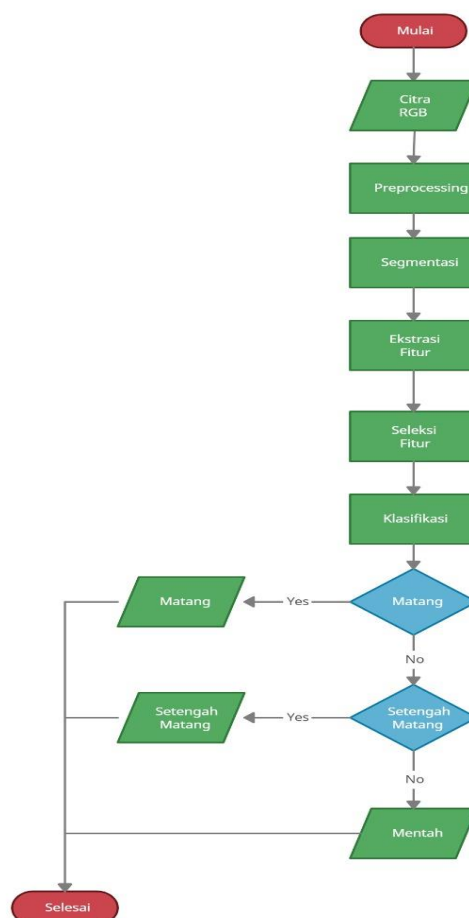
BAB 2 RANCANGAN

2.1 Data

Metode pengumpulan data dalam pelaksanaan kegiatan Proyek Akhir Workshop Pengolahan Citra dan Vision dengan judul Deteksi Kematangan Buah Tomat Sayur Berdasarkan Fitur Warna Menggunakan Metode Naïve Bayes menggunakan data primer dengan cara mengambil gambar dari 97 buah tomat sayur secara langsung. Untuk buah tomat sayur yang busuk tidak termasuk karena memiliki tekstur yang lembek.

Pengambilan gambar menggunakan studio box dan di ambil menggunakan kamera OPPO A5s dengan intensitas cahaya yang sama dan jarak antar kamera dengan buah tomat yang relatif sama. Data primer buah tomat ini dibagi menjadi 81 data latih dan 16 data uji.

2.2 Rancangan Sistem



Gambar 2. 1 Flowchart System

Secara umum sistem Deteksi Kematangan Buah Tomat Sayur menggunakan inputan berupa gambar yang diambil dengan menggunakan kamera handphone OPPO A5s. Dalam pengambilan gambar tomat sayur tersebut kami menggunakan intensitas cahaya yang sama dan jarak antar kamera terhadap buah tomat relatif sama. Beberapa gambar yang didapat kemudian dikirimkan ke komputer untuk diproses dan dianalisis.

Pada proses ini gambar buah tomat sayur akan di proses hingga gambar tersebut dapat di kenali cirinya dan dapat di bedakan. Prosesnya secara berurutan yaitu membaca citra RGB, mengkonversi citra RGB menjadi citra grayscale, mengkonversi citra grayscale menjadi citra biner, melakukan operasi morfologi filling holes untuk menyempurnakan hasil segmentasi, mengkonversi citra RGB menjadi citra HSV, melakukan ekstraksi ciri warna HSV, mengubah background menjadi 0, kemudian menghitung rata-rata nilai Hue, Saturation, dan Value.

Setelah nilai-nilai fitur ekstraksi telah didapat. Nilai tersebut akan diklasifikasi dengan Naïve Bayes. Klasifikasi dengan Naïve Bayes merupakan klasifikasi berdasar teorema Bayes dengan asumsi antar variable penjelas independen. Cara ini dapat diasumsikan sebagai keadaan atau ketiadaan dari sebuah kejadian tertentu dari suatu grup yang tidak berhubungan dengan keadaan atau ketiadaan kejadian lain. Teknik ini dapat dipergunakan untuk berbagai masalah klasifikasi dan dapat diterapkan pada klasifikasi buah tomat sayur.

Pada penelitian ini proses pelatihan menggunakan 81 buah tomat sayur yang terdiri dari 14 tomat sayur matang, 28 tomat sayur mentah, dan 39 tomat sayur setengah matang. Setelah melakukan pelatihan terhadap 81 buah tomat sayur mendapatkan akurasi pelatihan yaitu 100% yang berarti sistem ini sudah layak untuk di lanjutkan ketahap pengujian dengan sampel tomat sayur yang lainnya.

Output atau hasil yang di harapkan dari serangkaian proses di atas yaitu agar aplikasi yang kami buat mampu untuk mengklasifikasikan tingkat kematangan buah tomat sayur berdasarkan fitur warna HSV.

BAB 3 HASIL

Berdasarkan pelaksanaan proyek akhir semester 5 yang telah kelompok kami lakukan, maka diperoleh hasil seperti berikut:

3.1 Pengumpulan Data

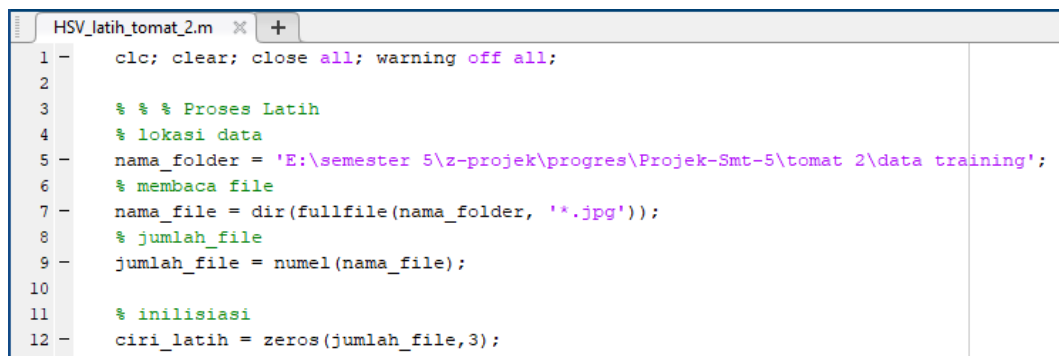
Langkah awal yang kami lakukan dalam pembuatan proyek semester 5 ini yaitu dengan pengumpulan data primer buah tomat sayur. Pengumpulan data dilakukan dengan cara memfoto sebanyak 97 buah tomat sayur yang kami beli di pasar. Gambar 3.1 berikut merupakan beberapa contoh data primer buah tomat sayur yang telah kami ambil. Data primer buah tomat ini dibagi menjadi 81 data latih dan 16 data uji.



Gambar 3. 1 Data Primer Buah Tomat

3.2 Pelatihan Data

Setelah melakukan pengumpulan data langkah selanjutnya yaitu melakukan pelatihan terhadap 81 gambar buah tomat sayur. Source code dari pelatihan ini bisa di lihat pada Gambar 3.2, Gambar 3.3, dan Gambar 3.4 berikut ini:

A screenshot of a MATLAB script editor window titled 'HSV_latih_tomat_2.m'. The script contains 12 lines of code. Line 1: 'clc; clear; close all; warning off all;'. Line 2: empty. Line 3: '% % % Proses Latih'. Line 4: '% lokasi data'. Line 5: 'nama_folder = 'E:\semester 5\z-projek\progres\Projek-Smt-5\tomat 2\data training';'. Line 6: '% membaca file'. Line 7: 'nama_file = dir(fullfile(nama_folder, '*.jpg'));'. Line 8: '% jumlah_file'. Line 9: 'jumlah_file = numel(nama_file);'. Line 10: empty. Line 11: '% iniliasi'. Line 12: 'ciri_latih = zeros(jumlah_file,3);'.

```
1 - clc; clear; close all; warning off all;
2
3 % % % Proses Latih
4 % lokasi data
5 - nama_folder = 'E:\semester 5\z-projek\progres\Projek-Smt-5\tomat 2\data training';
6 % membaca file
7 - nama_file = dir(fullfile(nama_folder, '*.jpg'));
8 % jumlah_file
9 - jumlah_file = numel(nama_file);
10
11 % iniliasi
12 - ciri_latih = zeros(jumlah_file,3);
```

Gambar 3. 2 Source Code untuk Membaca File

```

13     % mengolah citra
14     for n = 1:jumlah_file
15         % membaca file rgb
16         Img = imread(fullfile(nama_folder, nama_file(n).name));
17         % figure, imshow(Img)
18
19         % rgb to grayscale
20         Img_gray = rgb2gray(Img);
21         % figure, imshow(Img_gray)
22
23         % grayscale to biner
24         bw = imbinarize(Img_gray);
25         % figure, imshow(bw)
26
27         % operasi komplemen
28         bw = imcomplement(bw);
29         % figure, imshow(bw)
30
31         % morfologi filling holes
32         bw = imfill(bw, 'holes');
33         % figure, imshow(bw)
34
35         % ekstraksi ciri
36
37         %RGB to HSV
38         HSV = rgb2hsv(Img);
39         % figure, imshow(HSV)
40
41         % ekstrak H S V
42         H = HSV(:, :, 1);
43         S = HSV(:, :, 2);
44         V = HSV(:, :, 3);
45
46         %mengubah background menjadi 0
47         H(~bw) = 0;
48         S(~bw) = 0;
49         V(~bw) = 0;
50
51         %menghitung rata2 h,s,v
52         Hue = sum(sum(H))/sum(sum(bw));
53         Saturation = sum(sum(S))/sum(sum(bw));
54         Value = sum(sum(V))/sum(sum(bw));
55
56
57
58         %menghitung luas objek
59         % Luas = sum(sum(bw));
60
61         %mengisi variable
62         ciri_latih(n,1) = Hue;
63         ciri_latih(n,2) = Saturation;
64         ciri_latih(n,3) = Value;
65     end

```

Gambar 3. 3 Source Code untuk Mengola Citra

```

67 % menyusun variabel kelas_latih
68 kelas_latih = cell(jumlah_file, 1);
69
70 % mengisi =ma2 tomat pada variabel kelas_latih
71 for k = 1:14
72     kelas_latih{k} = 'Matang';
73 end
74
75 for k = 15:42
76     kelas_latih{k} = 'Mentah';
77 end
78
79 for k = 43:81
80     kelas_latih{k} = 'Setengah Matang';
81 end
82
83 % klasifikasi citra menggunakan algoritma naive bayes
84 Mdl = fitcnb(ciri_latih, kelas_latih);
85
86 % membaca kelas keluaran hasil pelatihan
87 hasil_latih = predict(Mdl, ciri_latih);
88
89 % menghitung akurasi latih
90 jumlah_benar = 0;
91 for k = 1:jumlah_file
92     if isequal(hasil_latih{k}, kelas_latih{k})
93         jumlah_benar = jumlah_benar+1;
94     end
95 end
96
97 akurasi_latih = jumlah_benar/jumlah_file*100;
98
99 save Mdl Mdl
100

```

Gambar 3. 4 Source Code untuk Melakukan Klasifikasi Naïve Bayes

Gambar 3.2, Gambar 3.3, dan Gambar 3.4 di atas merupakan tampilan source code pada matlab untuk pelatihan citra, untuk penjelasannya akan di jelaskan berikut ini.

% lokasi data

nama_folder = 'E:\semester 5\z-projek\progres\Projek-Smt-5\tomat 2\data training';

keterangan: menunjukkan lokasi folder yang akan kita olah datanya

% membaca file

nama_file = dir(fullfile(nama_folder, '*.jpg'));

keterangan: membaca semua data dengan ekstensi .jpg

% jumlah_file

jumlah_file = numel(nama_file);

keterangan: membaca jumlah file dari folder data training dengan ekstensi .jpg

```
% iniliasi
```

```
ciri_latih = zeros(jumlah_file,3);
```

keterangan: membuat variable ciri_latih yang isinya berupa 0 semua dengan jumlah baris sesuai dengan jumlah file yang ada, sedangkan jumlah kolomnya berjumlah 3

```
% mengolah citra
```

```
for n = 1:jumlah_file
```

keterangan: perulangan for sebanyak n (jumlah data)

```
    % membaca file rgb
```

```
    Img = imread(fullfile(nama_folder, nama_file(n).name));
```

```
    % figure, imshow(Img)
```

keterangan: membaca citra yang berada pada folder “data training”

```
    % rgb to grayscale
```

```
    Img_gray = rgb2gray(Img);
```

```
    % figure, imshow(Img_gray)
```

keterangan: melakukan konversi citra RGB menjadi citra Grayscale

```
    % grayscale to biner
```

```
    bw = imbinarize(Img_gray);
```

```
    % figure, imshow(bw)
```

keterangan: melakukan konversi citra Grayscale menjadi citra Biner

```
    % operasi komplemen
```

```
    bw = imcomplement(bw);
```

```
    % figure, imshow(bw)
```

keterangan: melakukan operasi komplemen agar nilai pada background menjadi 0 atau hitam

```
    % morfologi filling holes
```

```
    bw = imfill(bw,'holes');
```

```
    % figure, imshow(bw)
```

keterangan: melakukan operasi morfologi filling holes untuk menyempurnakan hasil segmentasi

```
    % ekstraksi ciri
```

```
    %RGB to HSV
```

```
    HSV = rgb2hsv(Img);
```

```
    % figure, imshow(HSV)
```

keterangan: mengubah citra RGB menjadi citra HSV

```

% ekstrak H S V
H = HSV(:,:,1);
S = HSV(:,:,2);
V = HSV(:,:,3);
    keterangan: mengekstrak kompone Hue, Saturation, and Value

%mengubah background menjadi 0
H(~bw) = 0;
S(~bw) = 0;
V(~bw) = 0;
    keterangan: mengubah nilai piksel backgown menjad nol

%menghitung rata2 h,s,v
Hue = sum(sum(H))/sum(sum(bw));
Saturation = sum(sum(S))/sum(sum(bw));
Value = sum(sum(V))/sum(sum(bw));
    keterangan: menghitung rata-rata dari setiap komponen dengan nilai
    backgown dari setiap citra

%mengisi variable
ciri_latih(n,1) = Hue;
ciri_latih(n,2) = Saturation;
ciri_latih(n,3) = Value;
    keterangan: mengisi variable ciri_latih dengan ciri hasil ekstraksi
end

% menyusun variabel kelas_latih
kelas_latih = cell(jumlah_file, 1);
    keterangan: membuat variable dengan kelas cell yang bernama kelas_latih
    dengan jumlah baris sebanyak jumlah file dan berbentuk string

% mengisi nama2 tomat pada variabel kelas_latih
for k = 1:14
    kelas_latih{k} = 'Matang';
end

for k = 15:42
    kelas_latih{k} = 'Mentah';
end

for k = 43:81
    kelas_latih{k} = 'Setengah Matang';
end
    keterangan: mengisi nama-nama pada data latih sesuai dengan urutan pada folder
    "data training"

```

```
% klasifikasi citra menggunakan algoritma naive bayes
```

```
Mdl = fitcnb(ciri_latih, kelas_latih);
```

keterangan: melakukan pelatihan dengan menggunakan metode Naïve Bayes yang sudah di sediakan oleh Matlab

```
% membaca kelas keluaran hasil pelatihan
```

```
hasil_latih = predict(Mdl,ciri_latih);
```

keterangan: membaca hasil pelatihan yang telah dilakukan

```
% menghitung akurasi latih
```

```
jumlah_benar = 0;
```

```
for k = 1:jumlah_file
```

```
    if isequal(hasil_latih{k}, kelas_latih{k})
```

```
        jumlah_benar = jumlah_benar+1;
```

```
    end
```

```
end
```

```
akurasi_latih = jumlah_benar/jumlah_file*100;
```

keterangan: menghitung akurasi pelatihan dengan cara melihat apakah nilai pada variable hasil_latih pada urutan ke-k sama dengan nilai pada variable kelas_latih pada urutan ke-k. akurasinya diambil dari jumlah benar di bagi dengan jumlah file dikali dengan 100

```
save Mdl Mdl
```

keterangan: menyimpan hasil pelatihan dengan nama *Mdl.mat*

3.3 Proses Naïve Bayes

Pada proses klasifikasi Naïve Bayes paramater yang kelompok kami gunakan yaitu nilai Hue, Saturation dan Value dari masing-masing citra. Masing-masing nilai ini kami dapat dari pengolahan citra yang sudah di lakukan pada aplikasi Matlab yang sebelumnya sudah kami buat dengan syntax seperti Gambar 3.5 berikut.

```
% klasifikasi citra menggunakan algoritma naive bayes
Mdl = fitcnb(ciri_latih, kelas_latih);
```

Gambar 3. 5 Syntax Proses Naïve Bayes

Untuk Proses di dalamnya kami jelaskan dalam bentuk perhitungan menggunakan aplikasi Microsoft Excel. Langkah pertama nilai HSV hasil pengolahan citra dari aplikasi Matlab kita salin ke dalam Microsoft Excel.

ciri_latih					
81x3 double					
	1	2	3	4	
1	0.0505	0.7234	0.7122		
2	0.0647	0.7508	0.7026		
3	0.0541	0.7179	0.7333		
4	0.0408	0.7194	0.7121		
5	0.0611	0.7123	0.7454		
6	0.0378	0.7346	0.7121		
7	0.0464	0.7277	0.7394		
8	0.0534	0.7099	0.7234		
9	0.0481	0.7302	0.7173		
10	0.0403	0.7239	0.7230		
11	0.0469	0.7172	0.7464		
12	0.0361	0.7671	0.6884		
13	0.0423	0.7307	0.7024		
14	0.0503	0.7293	0.7225		
15	0.2801	0.5737	0.5691		
16	0.2406	0.5208	0.5989		
17	0.2547	0.5294	0.5999		
18	0.2524	0.5217	0.6038		

Gambar 3. 6 Nilai H,S,V dari Matlab

3					
4	No. ▼	Hue ▼	Saturation ▼	Value ▼	Kategori ▼
5	1	0,0505	0,7234	0,7122	Matang
6	2	0,0647	0,7508	0,7026	Matang
7	3	0,0541	0,7179	0,7333	Matang
8	4	0,0408	0,7194	0,7121	Matang
9	5	0,0611	0,7123	0,7454	Matang
10	6	0,0378	0,7346	0,7121	Matang
11	7	0,0464	0,7277	0,7394	Matang
12	8	0,0534	0,7099	0,7234	Matang
13	9	0,0481	0,7302	0,7173	Matang
14	10	0,0403	0,7239	0,7230	Matang
15	11	0,0469	0,7172	0,7464	Matang
16	12	0,0361	0,7671	0,6884	Matang
17	13	0,0423	0,7307	0,7024	Matang
18	14	0,0503	0,7293	0,7225	Matang
19	15	0,2801	0,5737	0,5691	Mentah
20	16	0,2406	0,5208	0,5989	Mentah
21	17	0,2547	0,5294	0,5999	Mentah
22	18	0,2524	0,5217	0,6038	Mentah
23	19	0,2917	0,5914	0,5462	Mentah
<div> ◀ ▶ Naive Bayes ⊕ </div>					

Gambar 3. 7 nilai H, S, V Salinan dari Matlab

Langkah selanjutnya yaitu mencari Probabilitas dari data yang ada. Berikut tahapan mencari probabilitas dari setiap data melalui aplikasi Microsoft Excel.

1. Menentukan atribut dan target kelas

Terdapat tiga atribut yaitu Hue, Saturation dan value serta terdapat tiga target kelas yaitu Matang, Mentah dan Setengah Matang. Pada contoh ini kita gunakan atribut Hue dengan target kelas Matang.

No. ▼	Hue ▼	Saturation ▼	Value ▼	Kategori ▼
1	0,0505	0,7234	0,7122	Matang
2	0,0647	0,7508	0,7026	Matang
3	0,0541	0,7179	0,7333	Matang
4	0,0408	0,7194	0,7121	Matang
5	0,0611	0,7123	0,7454	Matang
6	0,0378	0,7346	0,7121	Matang
7	0,0464	0,7277	0,7394	Matang
8	0,0534	0,7099	0,7234	Matang
9	0,0481	0,7302	0,7173	Matang
10	0,0403	0,7239	0,7230	Matang
11	0,0469	0,7172	0,7464	Matang
12	0,0361	0,7671	0,6884	Matang
13	0,0423	0,7307	0,7024	Matang
14	0,0503	0,7293	0,7225	Matang

Gambar 3. 8 Atribut Hue dengan Target Kelas Matang

2. Mencari Mean dan Standar Deviasi

Kemudian nilai Hue dengan target kelas Matang kita cari nilai Mean dan Standar Deviasi-nya dengan rumus

$$\text{Mean } (\mu) = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\text{Standar Deviasi } (S) = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n - 1}}$$

Kemudian diperoleh nilai Mean dan nilai Standar Deviasi seperti pada Gambar 3.9 berikut ini

88					
89	Atribut: Hue				
90	Target kelas: Matang				
91	No.	Hue	(Hue-vHue)	(Hue-vHue)^2	
92	1	0,0505	0,0024069809	0,0000057936	
93	2	0,0647	0,0166199398	0,0002762224	
94	3	0,0541	0,0060431783	0,0000365200	
95	4	0,0408	-0,0072257340	0,0000522112	
96	5	0,0611	0,0130670158	0,0001707469	
97	6	0,0378	-0,0102984997	0,0001060591	
98	7	0,0464	-0,0016405332	0,0000026913	
99	8	0,0534	0,0053044093	0,0000281368	
100	9	0,0481	0,0000467303	0,0000000022	
101	10	0,0403	-0,0077619169	0,0000602474	
102	11	0,0469	-0,0011143861	0,0000012419	
103	12	0,0361	-0,0119307531	0,0001423429	
104	13	0,0423	-0,0057129356	0,0000326376	
105	14	0,0503	0,0021965044	0,0000048246	
106	Σ	0,67		0,0009196778	
107					
108	menghitung mean			0,048057392	
109	menghitung standar deviasi			0,008410972	

Gambar 3. 9 Nilai Mean dan Standar Deviasi

3. Mencari Distribusi Gaussian

Tahap selanjutnya yaitu mencari nilai Distribusi Gaussian menggunakan rumus berikut

$$\text{Distribusi Gaussian } P(X = x_i | Y = y_j) = \frac{1}{\sqrt{2\pi}S} \text{Exp}^{-\frac{(x_i - \mu_{ij})^2}{S^2_{ij}}}$$

Kemudian diperoleh nilai Gaussian seperti Gambar 3.10 berikut

H-Matang	H-Mentah	H-Setengah Matang	S-Matang	S-Mentah	S-Setengah Matang	V-Matang	V-Mentah	V-Setengah Matang
4,009	0,000	0,182	2,927	0,000	0,000	2,480	0,000	0,093
0,088	0,000	0,752	0,356	0,000	0,000	1,066	0,000	0,224
2,597	0,000	0,276	2,042	0,000	0,000	1,662	0,000	0,008
2,080	0,000	0,051	2,328	0,000	0,000	2,457	0,000	0,095
0,389	0,000	0,556	1,090	0,000	0,000	0,326	0,000	0,001
0,972	0,000	0,032	2,709	0,000	0,000	2,470	0,000	0,094
4,189	0,000	0,110	3,232	0,000	0,000	0,832	0,000	0,004
2,923	0,000	0,255	0,768	0,000	0,000	2,951	0,000	0,028
4,351	0,000	0,137	3,181	0,000	0,000	2,988	0,000	0,055
1,857	0,000	0,047	2,990	0,000	0,000	2,977	0,000	0,029
4,275	0,000	0,118	1,929	0,000	0,000	0,269	0,000	0,001
0,582	0,000	0,025	0,005	0,000	0,000	0,093	0,000	0,633
2,743	0,000	0,063	3,145	0,000	0,000	1,036	0,000	0,229
4,064	0,000	0,178	3,216	0,000	0,000	3,004	0,000	0,031
0,000	1,490	0,000	0,000	1,863	1,917	0,000	1,840	0,013
0,000	0,362	0,000	0,000	0,159	0,165	0,000	1,537	0,286
0,000	2,037	0,000	0,000	0,468	0,314	0,000	1,438	0,308
0,000	1,689	0,000	0,000	0,180	0,177	0,000	1,044	0,414
0,000	0,295	0,000	0,000	0,447	1,921	0,000	0,167	0,000

Gambar 3. 10 Hasil Distribusi Gaussian

Ketiga proses di atas di ulang sesuai dengan atribut dan target kelas yang sesuai hingga di dapat semua nilai yang ada pada Gambar 3.10 di atas.

4. Menghitung tingkat kematangan dari setiap citra

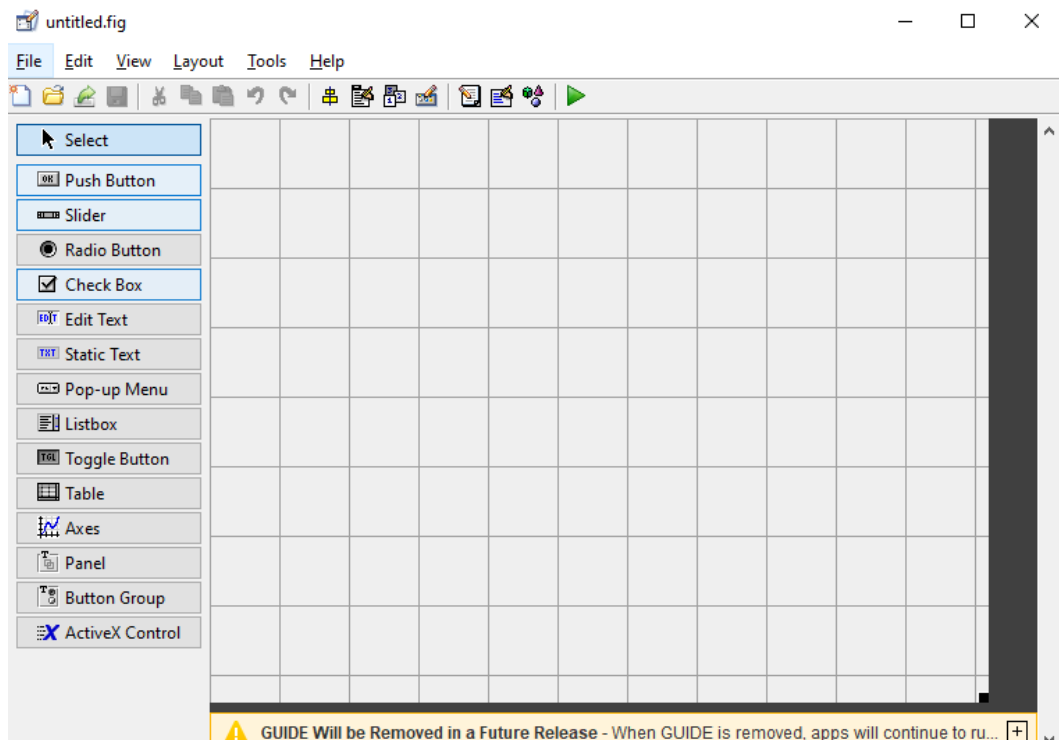
Tahap selanjutnya yaitu menentukan tingkat kematangan setiap buah tomat yaitu dengan mengalikan nilai HSV untuk target kelas matang, nilai HSV untuk target kelas Mentah dan nilai HSV untuk target kelas Setengah Matang dari masing-masing citra. Kemudian menentukan tingkat kematangannya berdasarkan nilai tertingginya. Seperti contoh Gambar 3.11 berikut, karena pada citra nomor 1 nilai tertinggi pada target kelas matang maka citra tersebut termasuk dalam golongan buah yang matang.

Matang	Mentah	Setengah Matang	Hasil Uji
29,105	0,000	0,000	Matang
0,033	0,000	0,000	Matang
8,815	0,000	0,000	Matang
11,900	0,000	0,000	Matang
0,138	0,000	0,000	Matang
6,502	0,000	0,000	Matang
11,258	0,000	0,000	Matang
6,628	0,000	0,000	Matang
41,354	0,000	0,000	Matang
16,527	0,000	0,000	Matang
2,217	0,000	0,000	Matang
0,000	0,000	0,000	Matang
8,937	0,000	0,000	Matang
39,261	0,000	0,000	Matang
0,000	5,108	0,000	Mentah
0,000	0,088	0,000	Mentah
0,000	1,372	0,000	Mentah
0,000	0,318	0,000	Mentah

Gambar 3. 11 MEnghitung Kematangan Citra Buah Tomat Sayur

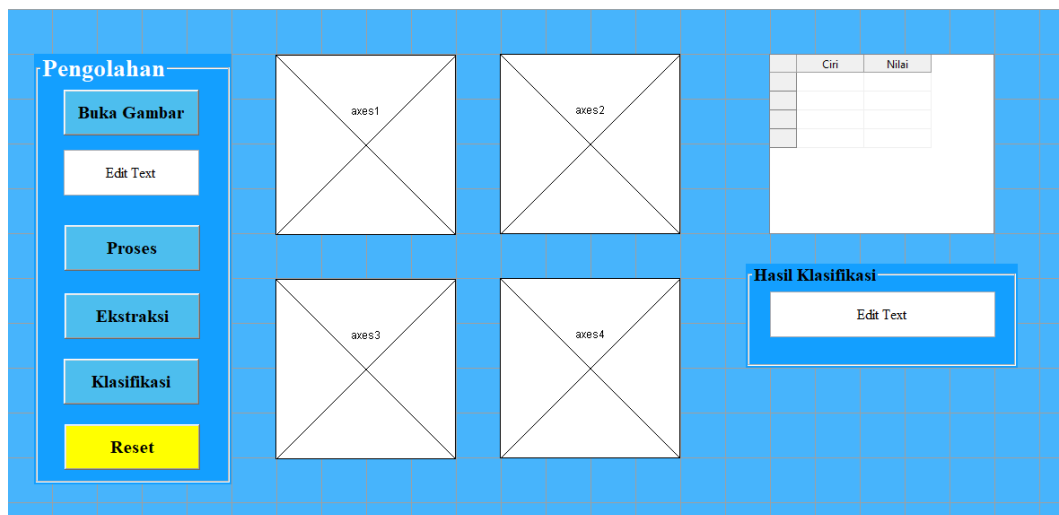
3.4 Pembuatan Tampilan Gui

Langkah selanjutnya yaitu membuat tampilan aplikasi. Langkah pertama dalam pembuatan GUI yaitu dengan mengetikkan perintah *guide* pada command window di Matlab kemudian tekan enter. Kemudian akan muncul tampilan guide seperti Gambar 3.5 berikut.



Gambar 3. 12 Open Guide pada Matlab

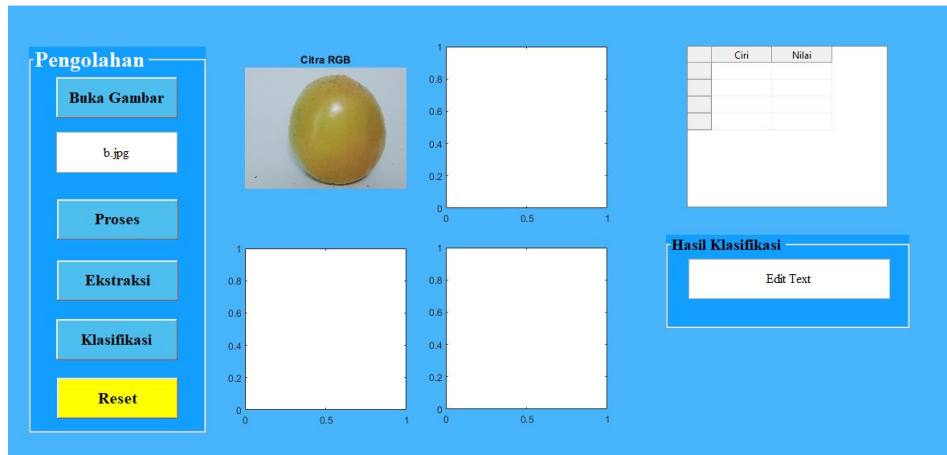
Langkah yang selanjutnya kita lakukan yaitu membuat tampilan programnya seperti Gambar 3.6 berikut ini kemudian kita simpan agar kita dapat melakukan proses selanjutnya.



Gambar 3. 13 Desain Guide pada Matlab

3.5 Pembuatan Source Code dari Tampilan Gui

3.4.1 Tombol Buka Gambar



Gambar 3. 14 Tampilan Aksi Tombol Buka Gambar

Tombol “Buka Gambar” di gunakan untuk membuka file gambar yang terdapat di dalam komputer kita dan kemudian di tampilkan di *axes1* sedangkan nama file akan ditampilkan pada *edit1* dengan judul Citra RGB. Untuk *source code* bisa di lihat pada Gambar 3.8 berikut ini.

```
% --- 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)

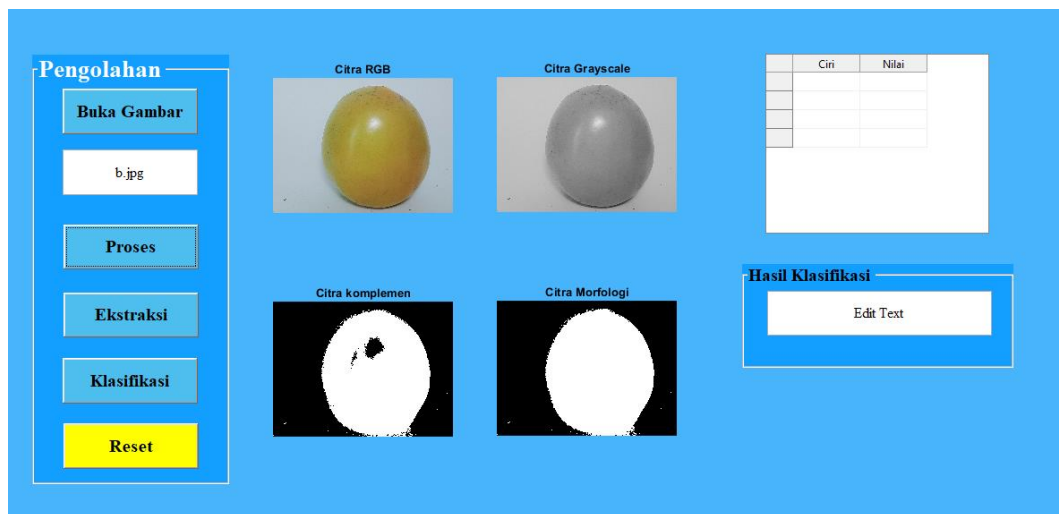
% memanggil menu "pencarian file"
[nama_file, nama_folder] = uigetfile('*.jpg');

% jika ada nama file yg dipilih maka akan mengeksekusi perintah di bawah
% ini
if ~isequal(nama_file,0)
    % membaca file rgb
    Img = imread(fullfile(nama_folder, nama_file));
    % menampilkan citra rgb pada axes
    axes(handles.axes1)
    imshow(Img)
    title('Citra RGB')
    % menampilkan nama file pada edit text
    set(handles.edit1,'String', nama_file)

    % menyimpan variable img pada lokasi handles agar dapat dipanggil oleh
    % pushbutton yang lain
    handles.Img = Img;
    guidata(hObject, handles)
else
    % jika tidak ada nama file yg dipilih maka akan kembali
    return
end
```

Gambar 3. 15 Source Code Tombol Buka Gambar

3.4.2 Tombol Proses



Gambar 3. 16 Tampilan Aksi Tombol Proses

Tombol Proses digunakan untuk memproses gambar yang sebelumnya sudah di pilih. Urutan prosesnya yaitu mengubah citra RGB menjadi citra Grayscale yang kemudian akan ditampilkan pada *axes2* dengan judul Citra Grayscale, kemudian mengubahnya menjadi citra biner dan di buat agar nilai backgrown menjadi 0 yang kemudian akan di tampilkan pada *axes3* dengan judul Citra Komplemen, kemudian menyempurnakannya dengan operasi morfologi filling holes yang akan ditampilkan pada *axes4* dengan judul Citra Morfologi. Untuk *source code* bisa di lihat pada Gambar 3.10 berikut ini.

```

% --- Executes on button press in pushbutton3.
function pushbutton3_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)

% memanggil variable Img yang ada di lokasi handles
Img = handles.Img;

% rgb to grayscale
Img_gray = rgb2gray(Img);
% menampilkan citra biner pada axes
axes(handles.axes2)
imshow(Img_gray)
title('Citra Grayscale')

% grayscale to biner
bw = imbinarize(Img_gray);

% operasi komplemen
bw = imcomplement(bw);

% menampilkan citra biner pada axes
axes(handles.axes3)
imshow(bw)
title('Citra komplemen')

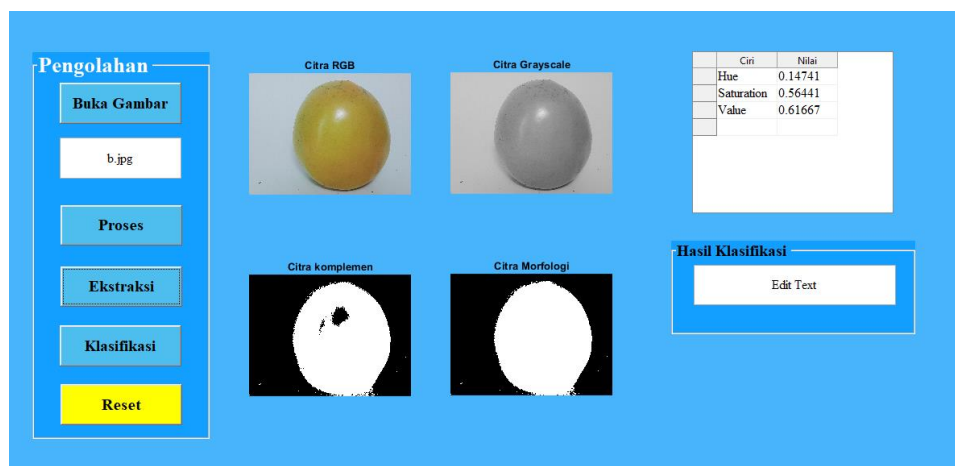
% morfologi filling holes
bw = imfill(bw,'holes');
% menampilkan citra biner pada axes
axes(handles.axes4)
imshow(bw)
title('Citra Morfologi')

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

```

Gambar 3. 17 Source Code Tombol Proses

3.4.3 Tombol Ekstraksi



Gambar 3. 18 Tampilan Aksi Tombol Ekstraksi

Tombol Ekstraksi digunakan untuk mengekstrak ciri dari gambar yang telah dipilih. Nilai ekstraksi yang di ambil yaitu nilai Hue, Saturation, dan Value dimana nilai ini nantinya akan ditampilkan pada *uitable1*. Untuk *source code* bisa di lihat pada Gambar 3.12 berikut ini.

```
% --- Executes on button press in pushbutton5.
function pushbutton4_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)
Img = handles.Img;
bw = handles.bw;

% ekstraksi ciri

    %RGB to HSV
    HSV = rgb2hsv(Img);
%     figure, imshow(HSV)

    % ekstrak H S V
    H = HSV(:,:,1);
    S = HSV(:,:,2);
    V = HSV(:,:,3);

    %mengubah background menjadi 0
    H(~bw) = 0;
    S(~bw) = 0;
    V(~bw) = 0;

%     figure, imshow(H)
%     figure, imshow(S)
%     figure, imshow(V)

    %menghitung rata2 h,s,v
    Hue = sum(sum(H))/sum(sum(bw));
    Saturation = sum(sum(S))/sum(sum(bw));
    Value = sum(sum(V))/sum(sum(bw));

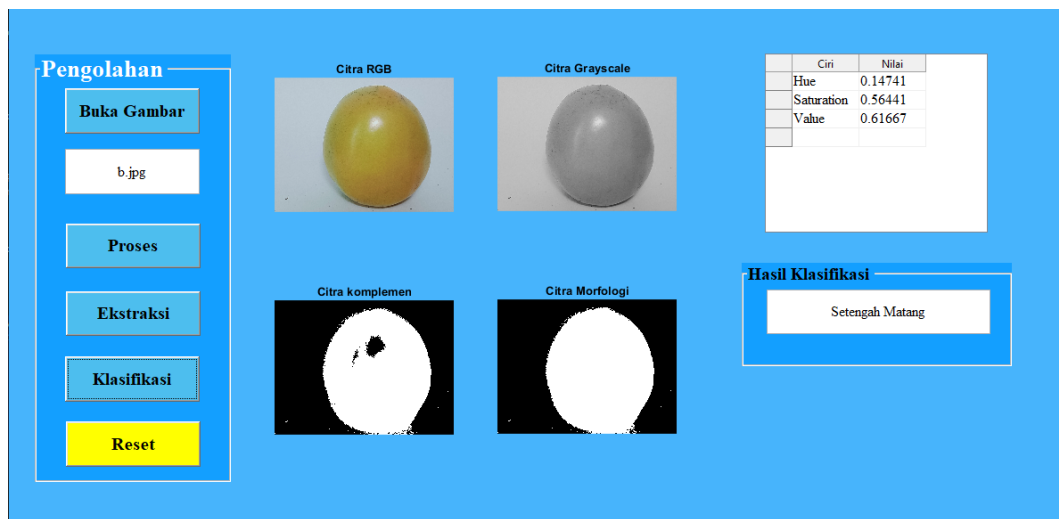
    %mengisi variable
    ciri_uji(1,1) = Hue;
    ciri_uji(1,2) = Saturation;
    ciri_uji(1,3) = Value;

    % menampilkan ciri hasil ekstraksi pada tabel
    ciri_tabel = cell(3,2);
    ciri_tabel{1,1} = 'Hue';
    ciri_tabel{2,1} = 'Saturation';
    ciri_tabel{3,1} = 'Value';
    ciri_tabel{1,2} = num2str(Hue);
    ciri_tabel{2,2} = num2str(Saturation);
    ciri_tabel{3,2} = num2str(Value);
    set(handles.uitable1, 'Data', ciri_tabel)

% menyimpan variable ciri_uji pada lokasi handles agar dapat dipanggil oleh
% pushbutton yang lain
handles.ciri_uji = ciri_uji;
guidata(hObject, handles)
```

Gambar 3. 19 Source Code Tombol Ekstraksi

3.4.4 Tombol Klasifikasi



Gambar 3. 20 Tampilan Aksi Tombol Klasifikasi

Tombol Klasifikasi digunakan untuk mengklasifikasi gambar yang sebelumnya telah di pilih dimana proses pengklasifikasiannya berdasarkan nilai Hue, Saturation, dan Value. Kemudian ketiga nilai tersebut akan di lakukan pengujian menggunakan metode Naïve Bayes dari file *Mdl.mat* yang sebelumnya sudah kita buat pada proses pelatihan data. Hasil pengujian ini kemudian akan ditampilkan pada *edit2* dan untuk *source code* bisa di lihat pada Gambar 3.14 berikut ini.

```
% --- 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)
ciri_uji = handles.ciri_uji;

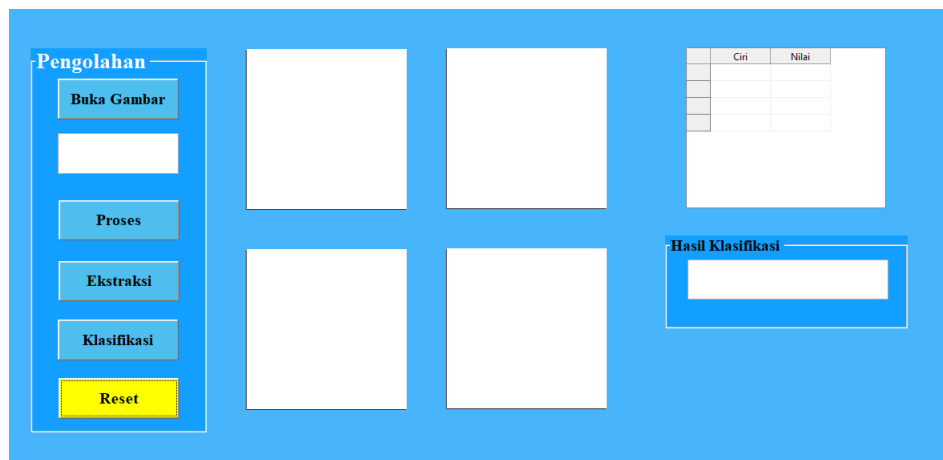
% klasifikasi citra menggunakan algoritma naive bayes
load Mdl;

% membaca kelas keluaran hasil pelatihan
hasil_uji = predict(Mdl,ciri_uji);

% menampilkan kelas keluaran hasil pelatihan
set(handles.edit2, 'String', hasil_uji{1})
```

Gambar 3. 21 Source Code Tombol Klasifikasi

3.4.5 Tombol Reset



Gambar 3. 22 Tampilan Aksi Tombol Reset

Tombol Reset di gunakan untuk menghapus semua tampilan yang muncul pada *axes1*, *axes2*, *axes3*, *axes4*, *edit1*, *edit2*, dan *uitable1*. Untuk *source code* bisa di lihat pada Gambar 3.16 berikut ini.

```
% --- 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)

% mereset tampilan GUI
set(handles.edit1, 'String', [])
set(handles.edit2, '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', [])

axes(handles.axes4)
cla reset
set(gca, 'XTick', [])
set(gca, 'YTick', [])

set(handles.uitable1, 'Data', [])
```

Gambar 3. 23 Source Code Tombol Reset

DAFTAR PUSTAKA

- Adhi Wibowo, d. (2021). Deteksi Kematangan Buah Jambu Kristal Berdasarkan Fitur Warna Menggunakan Metode Transformasi Ruang Warna Hsv (Hue Saturation Value) Dan K-Nearest Neighbor. *Journal of Informatic and Computer Science Engineering* , 76-88.
- M Taufiq Tamam, d. (2020). Rancang Bangun Purwarupa Sistem Deteksi Tingkat Kematangan Buah Jeruk Berdasarkan Warna Kulitnya . *JURNAL Riset REKAYASA ELEKTRO*, Vol.2, No.2.
- pamungkas, a. (2021, 12 20). Retrieved from Pemrograman Matlab:
<https://pemrogramanmatlab.com/tag/citra-hsv/>
- Rahmat Karim, d. (2020). SISTEM CERDAS DALAM KLASIFIKASI KEMATANGAN BUAH JERUK. *Jurnal Teknologi & Manajemen Informatika* , Vol.6 No.1.
- Rendy Pratama, d. (2019). DETEKSI KEMATANGAN BUAH TOMAT BERDASARKAN FITUR WARNA MENGGUNAKAN METODE TRANSFORMASI RUANG WARNA HIS. *JIKO (Jurnal Informatika dan Komputer)* , 81-86.
- Suliztia, M. L. (2021, 12 20). Retrieved from
<https://medium.com/@megaluna/comparing-naive-bayes-and-k-nearest-neighbor-classification-methods-of-breast-cancer-in-coimbra-bba254f04860>