

SI RICE

Aplikasi Rekognisi Citra Beras



ANGGOTA :

Ridan Nurfalah
Dwiza Riana
Anton



DAFTAR ISI

	Halaman
DAFTAR ISI	2
DAFTAR GAMBAR.....	3
BAB I PENDAHULUAN	4
1.1. Tentang Aplikasi	4
1.2. Cara Kerja Aplikasi	5
 BAB II AKSES APLIKASI SEBAGAI USER	 6
2.1. Tutorial Program	6
2.1.1. Halaman Login	6
2.1.2. Halaman Utama	6
2.1.3. Main Image	7
2.1.4. Segmentation	8
2.1.5. Feature Extraction	10
2.1.6. Identification	11
2.2. Algoritma Identifikasi	13
2.2.1. Algoritma Multi-SVM	14
2.1.2. Algoritma Neural Network	15
2.3. Source Code	18
DAFTAR PUSTAKA.....	28

DAFTAR GAMBAR

	Halaman
Gambar 1. Halaman Login	6
Gambar 2. Halaman Utama.....	7
Gambar 3. <i>Main Image Section</i>	8
Gambar 4. <i>K-Means Segmentation Section</i>	9
Gambar 5. <i>Binary Segmentation Section</i>	9
Gambar 6. <i>Feature Extraction Section</i>	10
Gambar 7. Identifikasi Citra Beras Basmati	12
Gambar 8. Identifikasi Citra Beras Japonica	12
Gambar 9. Identifikasi Citra Beras Jasmine	13
Gambar 10. Hasil Akurasi Algoritma Multi-SVM	14
Gambar 11. Hasil Akurasi Algoritma Neural Network	16

BAB I

PENDAHULUAN

1.1. Tentang Aplikasi

Indonesia merupakan negara dengan kebutuhan beras yang tinggi karena menjadi makanan pokok lebih dari 90% masyarakatnya. Kebutuhan tinggi memerlukan stok yang tinggi sehingga impor pun dilakukan dengan aturan-aturan sesuai Permendagri Nomor 19/M-DAG/PER/3/2014 yang menerangkan standar impor beras. Terdapat banyak jenis beras yang di impor ke Indonesia dengan berbagai kualitas, warna dan keperluan impor seperti untuk kesehatan atau stabilisasi harga. Dari sisi wana, beras putih impor merupakan beras paling banyak dikonsumsi oleh warga Indonesia. Salahsatu contohnya adalah beras melati (*jasmine*) dari Thailand. Sedangkan dari sisi keperluan impor baik untuk kesehatan dan stabilisasi harga beras japonica (Jepang) dan Basmati (Pakistan) merupakan jenis paling banyak di impor ke Indonesia. Namun masih banyak yang belum mengenal ketiga beras tersebut. Perkembangan ilmu pengetahuan dan teknologi mampu mengidentifikasi jenis beras impor tersebut baik dari kualitas, *diet value* dan ciri fisik. Salahsatunya dengan teknologi pengolahan citra yang dapat mendeteksi dari ciri fisik butir beras. Sehingga tidak perlu lagi dilakukan pengenalan jenis beras secara manual yang terkadang tidak konsisten dan hasil yang salah.

Pengenalan beras secara manual membutuhkan orang ahli dibidangnya dalam skala besar dan memerlukan waktu yang cukup lama jika dibandingkan dengan penggunaan teknologi. Dalam menangani permasalahan tersebut dan untuk menjadi sebuah terobosan dalam pengenalan jenis beras maka dibuatlah aplikasi ini dengan nama SI-RICE “Aplikasi Rekognisi Citra Beras”. Aplikasi ini dapat melakukan identifikasi pada ketiga jenis beras yang menjadi salahsatu beras impor komoditas tinggi di Indonesia, yaitu jenis beras melati (*jasmine*) dari Thailand, Basmati dari Pakistan dan Japonica dari Jepang.

Proses identifikasi dilakukan dengan perbandingan dua algoritma yang telah dilakukan pengujian, yaitu algoritma *Multi-SVM* dan *Neural Network*. Sebelum melakukan identifikasi dan tahap segmentasi pada aplikasi, terlebih dahulu dilakukan akuisisi citra yang menghasilkan 90 citra dibagi menjadi 63 citra training dan 27 citra testing. *K-Means* dipilih menjadi metode segmentasi citra pada aplikasi ini yang mengubah citra asli dengan format JPG ataupun PNG menjadi citra biner. Algoritma *K-Means* dipilih karena menghasilkan citra yang tersegmentasi dengan baik setelah dilakukan akuisisi. Tahap selanjutnya yang dilakukan aplikasi SI-RICE adalah ekstraksi fitur menggunakan morphological feature dengan metode *regionprop*

dikombinasikan dengan *Gray Level Co-Occurrence Matrix* (GLCM) menghasilkan 9 fitur yang dapat menghasilkan akurasi 96,296% untuk Multi-SVM dan Neural Network 88,89% pada citra yang dilatihkan di aplikasi. Dengan hasil percobaan yang tinggi akurasi maka aplikasi SI-RICE diduga dapat menrekognisi ketiga jenis citra beras tersebut.

1.2. Cara Kerja Aplikasi

Aplikasi identifikasi jenis beras ini dibangun dengan menggunakan MATLAB 2020 dan dapat diakses menggunakan *Personal Computer (PC)* atau Laptop dengan minimal RAM 4GB disarankan 8GB. Berikut Langkah-langkah dalam menjalankan aplikasi identifikasi jenis beras ini:

1. Akses aplikasi menggunakan PC/Laptop
2. Login dengan akun admin
3. Admin dapat input citra beras (yang telah disediakan) dengan menu browse pada bagian Main Image
4. Admin dapat melakukan cek nilai segmentasi dengan klik tombol K-Means dan Binary
5. Admin dapat menampilkan nilai *feature extraction* dengan klik tombol *Show Image* dan *Show Table*.
6. Admin dapat menampilkan hasil identifikasi dengan 2 algoritma yang digunakan pada aplikasi ini yaitu Multi-SVM dan Neural network pada menu Multisvm dan Neural Network.
7. Admin dapat melakukan logout

BAB II

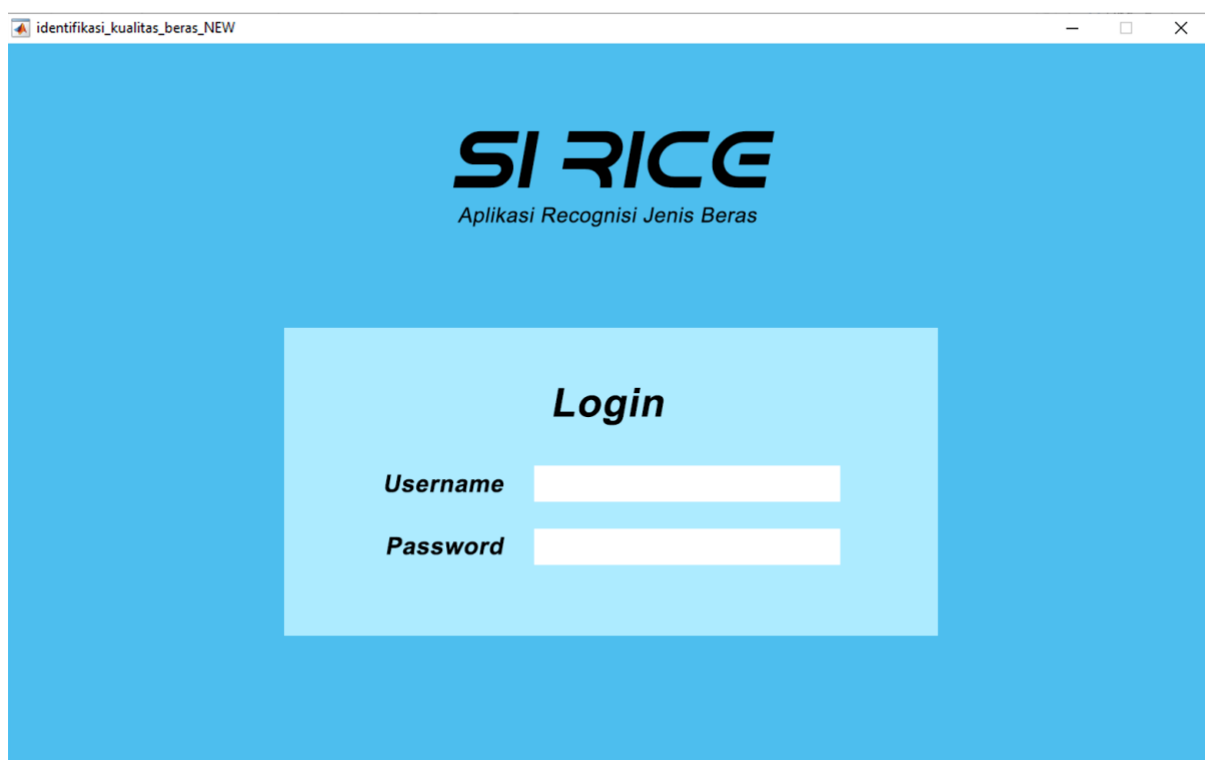
AKSES APLIKASI SEBAGAI USER

2.1. Tutorial Program

Berikut adalah tutorial dan penjelasan masing-masing menu yang ada pada Aplikasi Rekognisi Citra Beras (SI-RICE):

2.1.1. Halaman Login

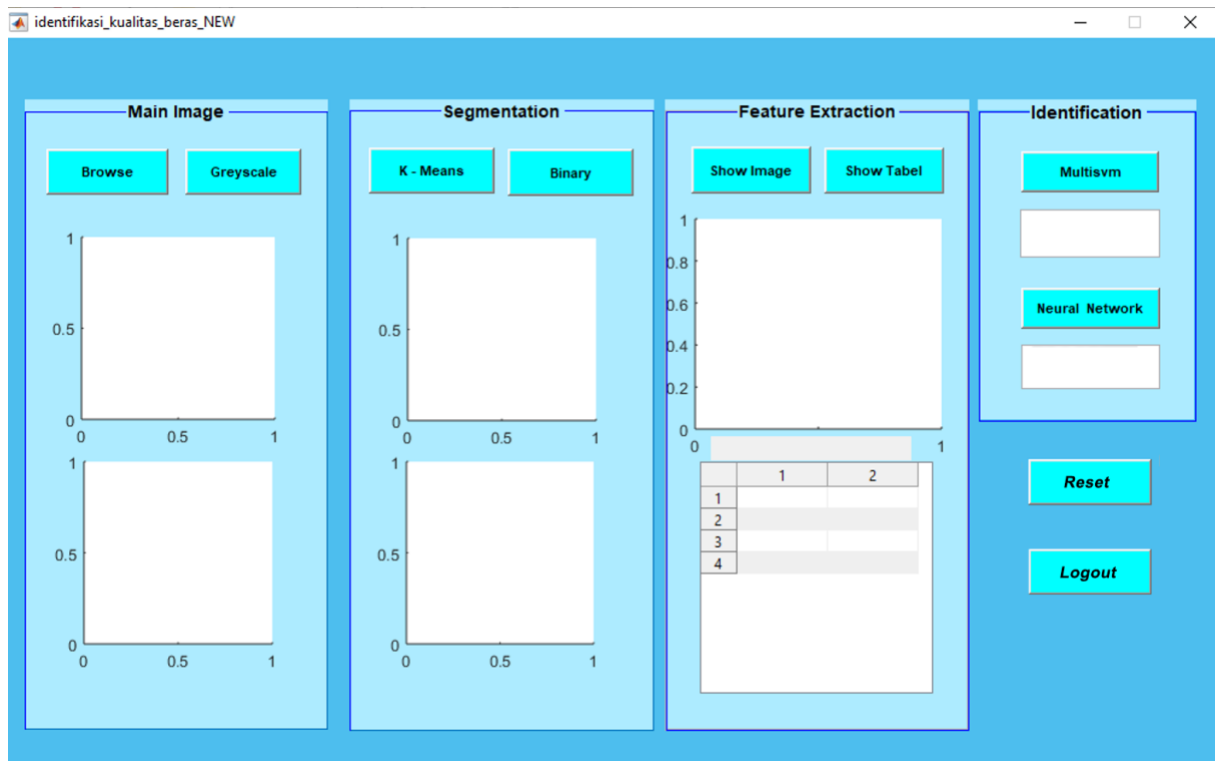
Halaman pertama yang muncul ketika membuka aplikasi adalah halaman login. Default username untuk admin adalah “admin” dan password “admin”.



Gambar 1. Halaman Login

2.1.2. Halaman Utama

Setelah melakukan login maka akan muncul tampilan halaman utama yang bersisi 4 bagian utama, yaitu *Main Image Section*, *Segmentation Section*, *Feature Extraction Section* dan *Identification Section* dimana setiap bagian memiliki fungsi masing-masing dalam tahapan identifikasi jenis citra beras.



Gambar 2. Halaman Utama

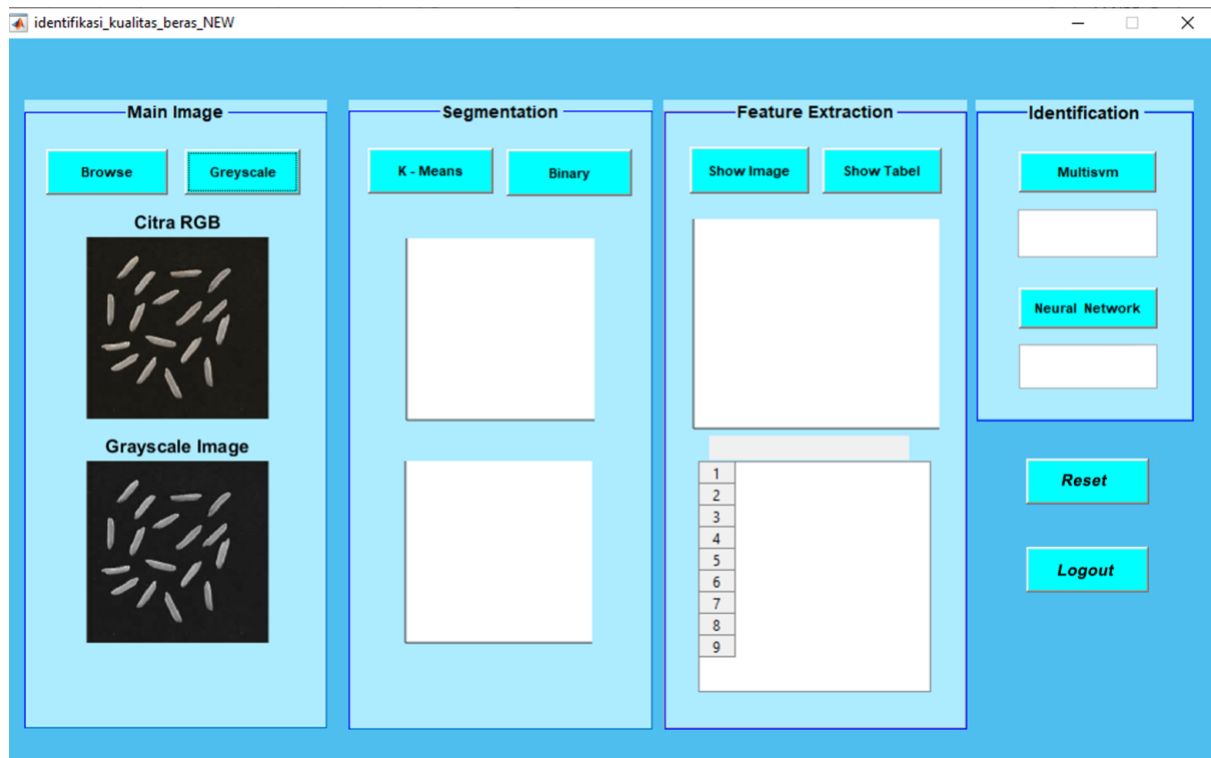
2.1.3. Main Image

Halaman main image menampilkan langkah pertama dalam pengenalan atau identifikasi jenis citra beras pada aplikasi SI-RICE. Pada *Main Image Section* terdapat 2 button yaitu:

- Browse : untuk input citra beras yang akan diidentifikasi jenisnya menggunakan aplikasi SI-RICE
- Greyscale : untuk mengubah citra dengan format awal baik itu .jpg ataupun .png dengan kombinasi warna RGB (Red Green Blue) menjadi citra skala abu-abu (Grayscale) langkah ini sudah masuk pada pengolahan citra dan pengubahan jenis citra sesuai kebutuhan langkah selanjutnya.

Selain 3 button yang telah dijelaskan diatas terdapat juga 2 tampilan atau jendela axes dalam main image ini, yaitu:

- Citra RGB : untuk menampilkan citra yang telah diinput pada aplikasi SI-RICE dengan tombol Browse.
- Greyscale Image : untuk menampilkan figure citra yang telah diubah menjadi citra skala abu-abu dari citra RGB dengan klik tombol greyscale.

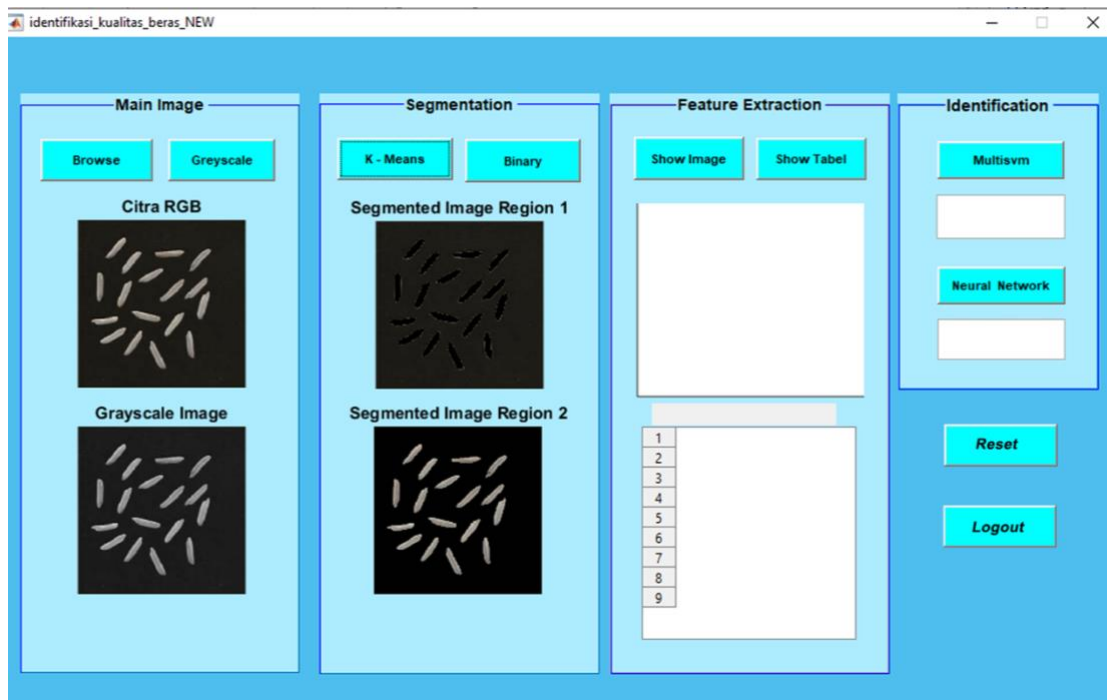


Gambar 3. *Main Image Section*

2.1.4. *Segmentation*

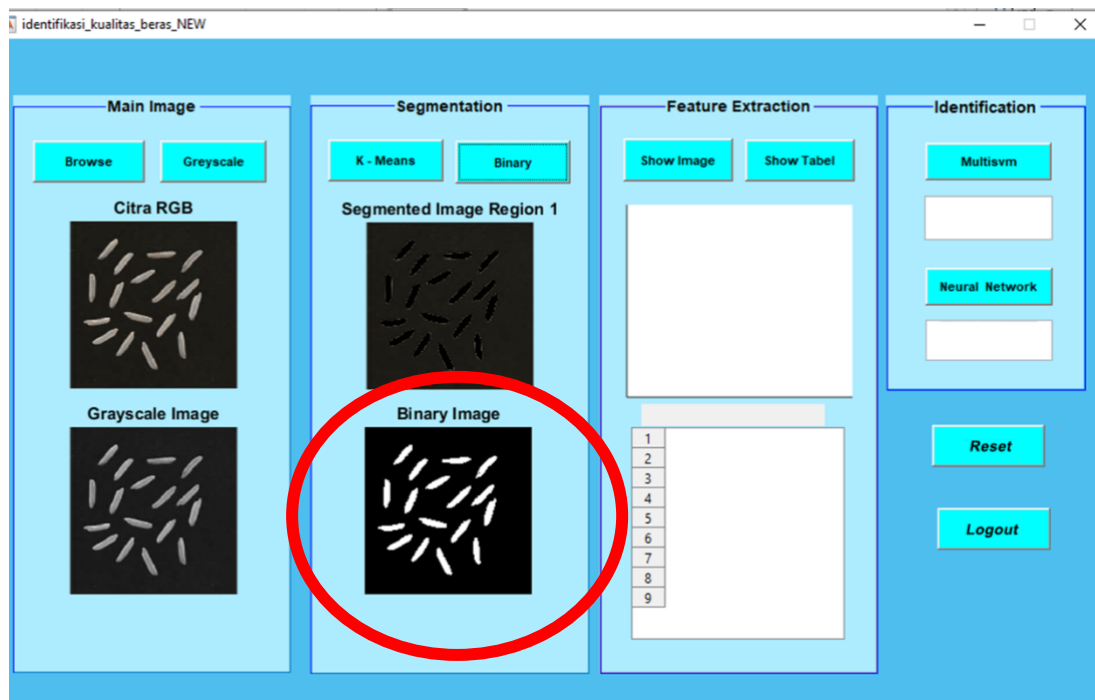
Section kedua dalam aplikasi SI-RICE ini adalah *segmentation section* merupakan tahapan segmentasi yang dilakukan dalam identifikasi jenis citra beras. Terdapat dua button pada *segmentation section* ini yang masing-masing memiliki fungsi untuk menampilkan hasil prose segmentasi pada ciitra beras dengan algoritma masing-masing, diantaranya :

- a. Algoritma K-Means yang diwakili dengan tombol “K-Means”, pada proses segmentasi menggunakan algoritma K-Means ini, citra diproses dan ditampilkan dengan 2 klaster citra hasil segmentasi yang ditunjukan pada axes 1 dan axes 2 pada *segmentation section* Aplikasi SI-RICE . Dari hasil segmentasi tahap pertama tersebut, klaster 2 dinyatakan lebih baik dibandingkan klaster 1 karena memudahkan pada proses konversi biner dan memisahkan dengan baik antara objek yang akan diteliti pada citra dengan bukan objek penelitian



Gambar 4. *K-Means Segmentation Section*

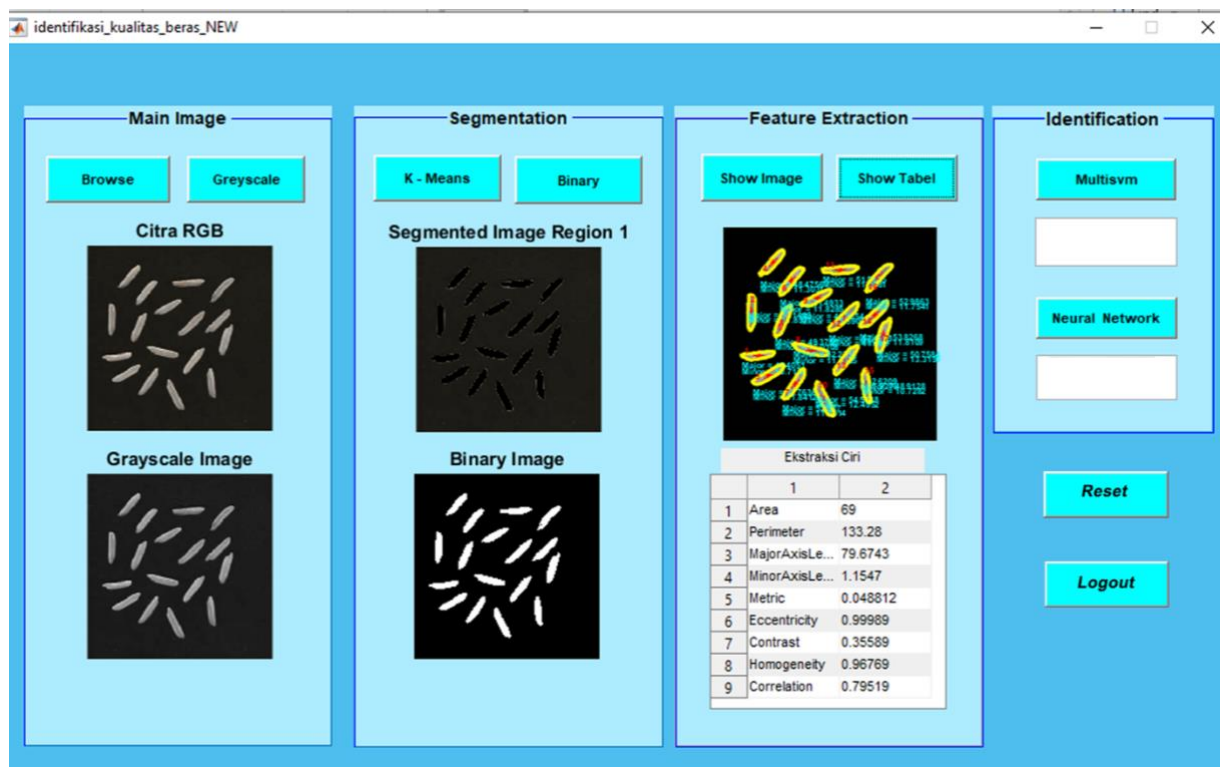
- b. Segmentasi dengan mengubahnya menjadi binary image, setelah dipilih citra hasil segmentasi dengan K-Means pada kluster 2, maka dilakukan konversi citra dari RGB menjadi citra biner. Berikut pada Gambar 5 tampilan hasil segmentasi dengan *binary image* pada aplikasi SI-RICE:



Gambar 5. *Binary Segmentation Section*

2.1.5. Feature Extraction

Bagian ketiga dari aplikasi SI-RICE ini adalah *Feature Extraction Section* atau ekstraksi fitur dari citra beras. Data citra beras yang telah dilakukan segmentasi diturunkan atau ditampilkan nilai-nilai yang menjadi bahan pengujian untuk identifikasi. Pada aplikasi ini metode fitur morfologi dengan *Regionprop* dan *Gray Level Co-Occurance Method (GLCM)* dipilih sebagai metode ekstraksi citra untuk identifikasi jenis citra butir beras. Terdapat 9 fitur yang dihasilkan dari kombinasi metode *Regionprops* dan GLCM yang meliputi *eccentricity*, *contrast*, *homogeneity*, *correlation*, *Area*, *Perimeter*, *Metric*, *Major Axis Length*, dan *Minor Axis Length*.



Gambar 6. *Feature Extraction Section*

Terdapat dua button pada *feature extraction section* aplikasi SI-RICE ini dengan masing-masing fungsi, yang terdiri dari:

- Show Image, untuk memproses citra beras yang telah disegmentasi dengan K-Means dan binary image lalu ditampilkan pada axes 1
- Show Tabel, berfungsi untuk menampilkan nilai ekstraksi fitur dari citra yang diinputkan dalam sebuah tabel

Selain 2 button yang memiliki fungsi masing-masing, terdapat juga 2 axes pada feature extraction section ini, yaitu:

- a. Axes 1 berfungsi untuk menampilkan figure atau hasil pengolahan dari feature extraction. Pada axes ini ditampilkan nilai masing-masing fitur pada setiap biji berasnya dengan hasil citra yang telah tersegmentasi. Axes ini akan ditampilkan ketika user klik buton “Show Image”.
- b. Axes 2 berfungsi untuk menampilkan figure atau hasil pengolahan dari feature extraction dari citra yang diinputkan. Pada axes 2 ini ditampilkan nilai fitur dari citra yang diinputkan yang selanjutnya akan diproses pada identifikasi jenis citra beras. Axes 2 ini diberi title “Ekstraksi Ciri”, akan tampil Ketika user klik button “Show Table”

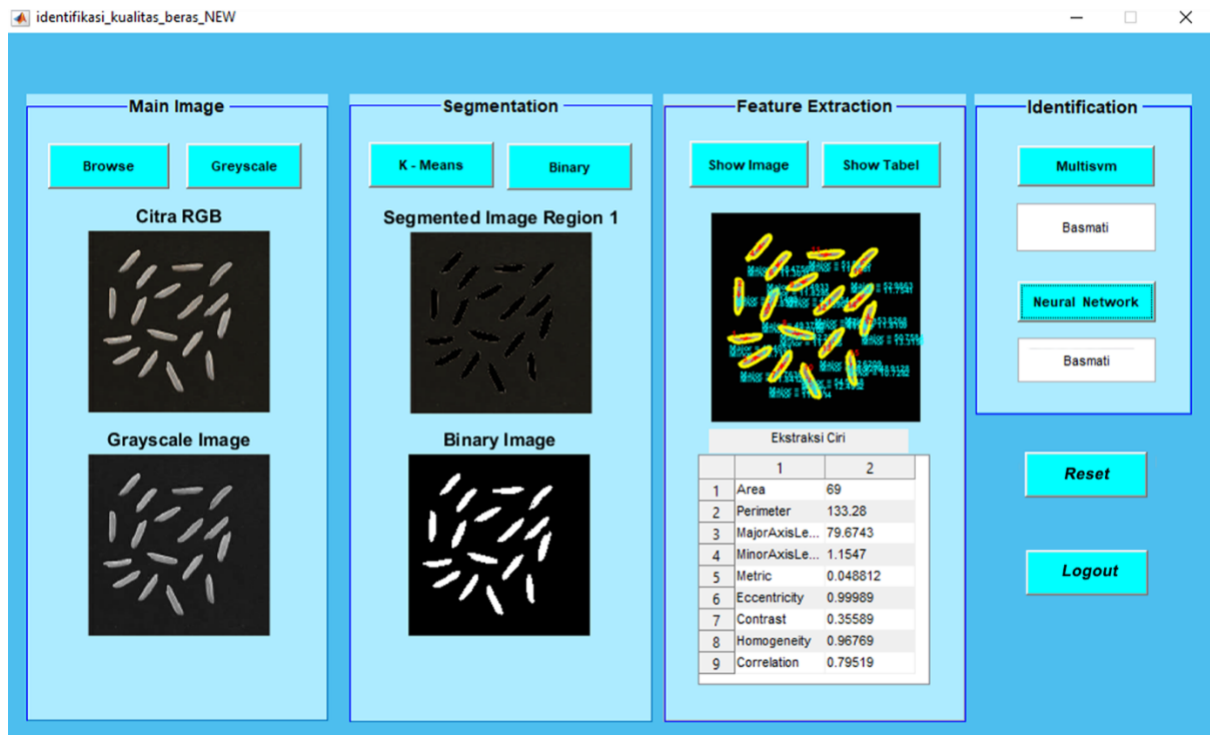
2.1.6. Identification

Tahap identifikasi ini merupakan tahap akhir dalam penentuan jenis citra beras. Pada identification section akan ditampilkan jenis citra beras berdasarkan hasil pre-processing dengan mengubah format citra asli beras dari .JPG, .PNG atau jenis format lainnya yang masih berupa RGB (Red Green Blue) menjadi citra greyscale atau skala abu-abu. Setelah itu dilakukan segmentasi dengan K-Means dan Binari dan muncul nilai fitur yang menjadi penentu hasil identifikasi ini. Pada langkah identifikasi ini, aplikasi menyediakan 2 algoritma yang digunakan yaitu Multi-SVM dan Neural Network dengan hasil akurasi yang sama-sama tinggi dari hasil penelitian yang dilakukan oleh Ridan dkk dengan judul penelitian “Identifikasi Citra Beras Menggunakan Algoritma Multi-SVM dan Neural Network Pada Segmentasi K-Means” yang diterbitkan di Jurnal RESTI pada tahun 2021 bulan Februari.

Identifikasi ini dilakukan dengan 2 algoritma pada 3 jenis citra beras berikut adalah hasil masing-masing pada jenis citra beras:

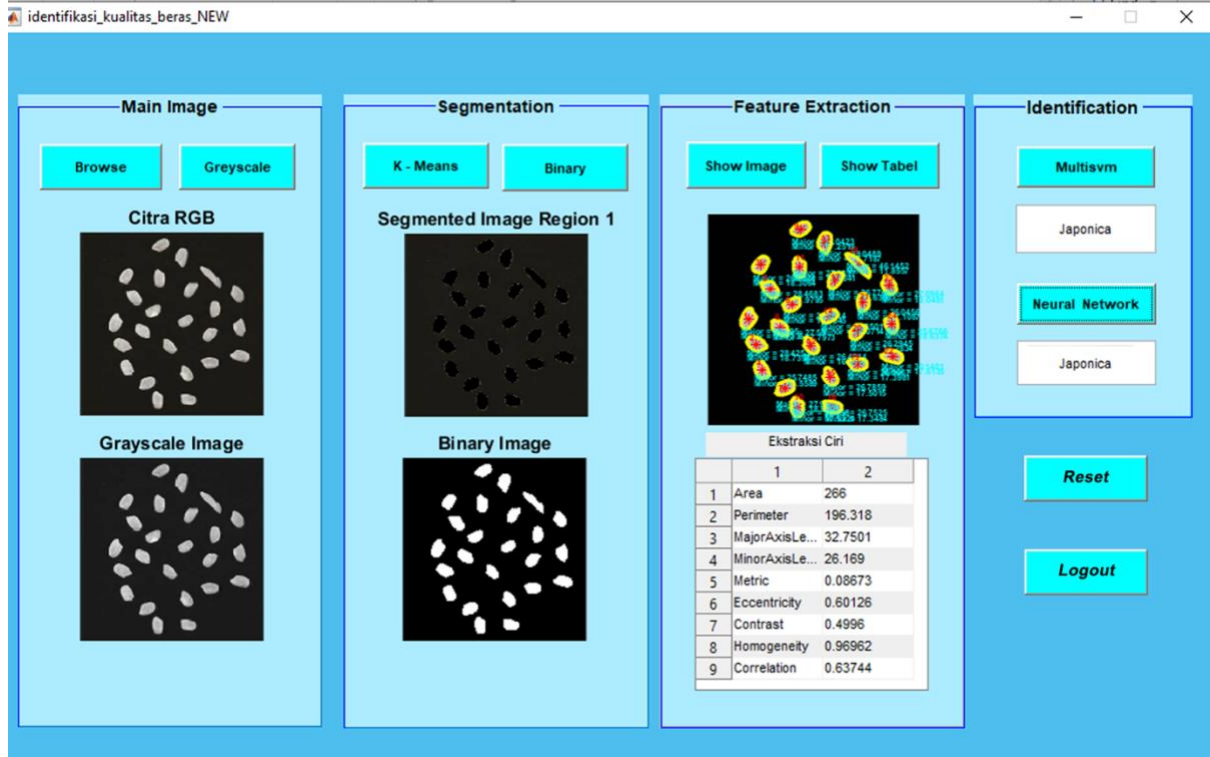
- a. Identifikasi Jenis Citra Beras Basmati

Seperti ditampilkan pada Gambar 7. Adalah tampilan halaman identifikasi pada jenis citra beras Basmati yang diujicobakan pada aplikasi SI-RICE dengan kedua algoritma. Algoritma Multi-SVM menampilkan hasil yang sesuai yaitu jenis Basmati dan algoritma Neural Network pun menampilkan hasil yang sesuai yaitu jenis citra beras Basmati.



Gambar 7. Identifikasi Citra Beras Basmati

b. Identifikasi Jenis Citra Beras Japonica

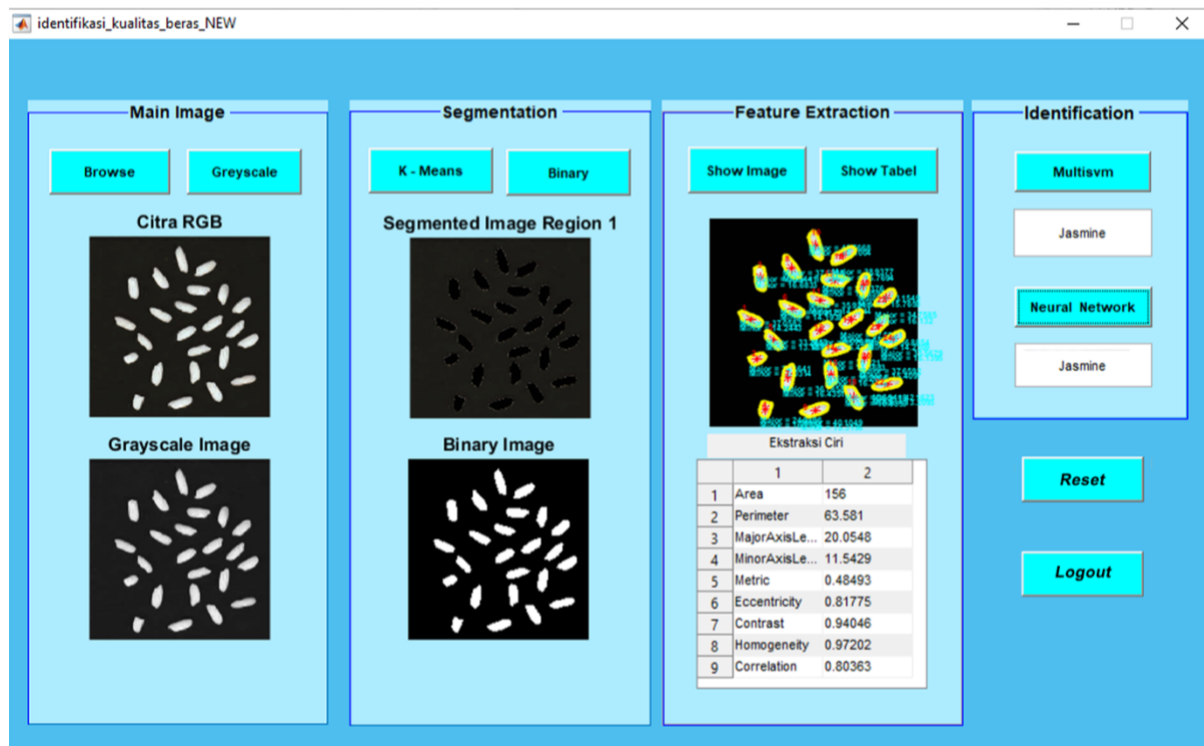


Gambar 8. Identifikasi Citra Beras Japonica

Gambar 8 diatas merupakan tampilan dari hasil identifikasi citra beras jenis Japonica pada aplikasi SI-RICE. Dapat dilihat bahwa aplikasi ini menunjukkan hasil yang akurat dengan

kedua algoritmanya. Algoritma Multi-SVM yang ditampilkan pada identification section baris pertama menunjukkan hasil yang benar yaitu “Japonica”, begitupun dengan algoritma Neural Network pada aplikasi ini berjalan dengan baik menunjukkan hasil yang sesuai yaitu menampilkan hasil citra beras jenis Japonica.

c. Identifikasi Jenis Citra Beras Jasmine



Gambar 9. Identifikasi Citra Beras Jasmine

Pada Gambar 9. diatas adalah tampilan dari hasil identifikasi beras jenis jasmine. Kedua algoritma menunjukkan hasil yang sama yaitu jenis beras Jasmine pada Algoritma Multi-SVM maupun algoritma Neural Network.

2.2. Algoritma Identifikasi

Seperti dijelaskan pada tampilan aplikasi di atas, aplikasi SI-RICE ini menggunakan dua algoritma dalam pemrosesan identifikasi jenis berasnya, yaitu algoritma Multi-SVM dan algoritma Neural Network. Berikut adalah kedua algoritma yang digunakan pada aplikasi SI-RICE:

2.2.1. Algoritma Multi-SVM

Algoritma Multi-SVM yang digunakan pada aplikasi SI-RICE ini menghasilkan nilai akurasi yang tinggi yaitu 90% dengan citra latih yang digunakan, seperti yang ditampilkan pada Gambar 10 berikut:

```
75
76 % penentuan nilai target untuk masing2 jenis beras
77 - target_latih = zeros(1,jumlah_data);
78 - target_latih(1:20) = 1; % basmati
79 - target_latih(21:40) = 2; % japonica
80 - target_latih(41:60) = 3; % jasmine
81
82 % pelatihan menggunakan algoritma multisvm
83 - output = multisvm(latih_database2,target_latih,latih_database2);
84
85 % menghitung nilai akurasi pelatihan
86 - [n,~] = find(target_latih==output');
87 - jumlah_benar = sum(n);
88 - akurasi = jumlah_benar/jumlah_data*100
89
90 % menyimpan variabel data_latih dan target_latih
91 - save latih_database2 latih_database2
92 - save target_latih target_latih
```

```
Command Window

akurasi =

    90

fx >>
```

Gambar 10. Hasil Akurasi Algoritma Multi-SVM

Dengan hasil akurasi yang tinggi pada aplikasi SI-RICE ini, berikut adalah *source code* algoritma Multi-SVM yang digunakan:

```
function [svmlrh] = multisvm( T,C,test )
svmlrhsize=size(test,1);
svmlrh=[];
Cb=C;
Tb=T;
for rlhtemp=1:svmlrhsize
    tst=test(rlhtemp,:);
    C=Cb;
    T=Tb;
    u=unique(C);
```

```

RLH_R=length(u);
c4=[];
c3=[];
y=1;
k=1;
if(RLH_R>2)
    ulang=1;
    classes=0;
    cond=max(C)-min(C);
    while((classes~=1)&&(ulang<=length(u))&& size(C,2)>1 && cond>0)
        c1=(C==u(ulang));
        newClass=c1;
        svmStruct = svmtrain(T,newClass,'kernel_function','rbf');
        classes = svmclassify(svmStruct,tst);
        for i=1:size(newClass,2)
            if newClass(1,i)==0;
                c3(k,:)=T(i,:);
                k=k+1;
            end
        end
        T=c3;
        c3=[];
        k=1;
        for i=1:size(newClass,2)
            if newClass(1,i)==0;
                c4(1,y)=C(1,i);
                y=y+1;
            end
        end
        C=c4;
        c4=[];
        y=1;
        cond=max(C)-min(C);
        if classes~=1
            ulang=ulang+1;
        end
    end
end
val=Cb==u(ulang);

```

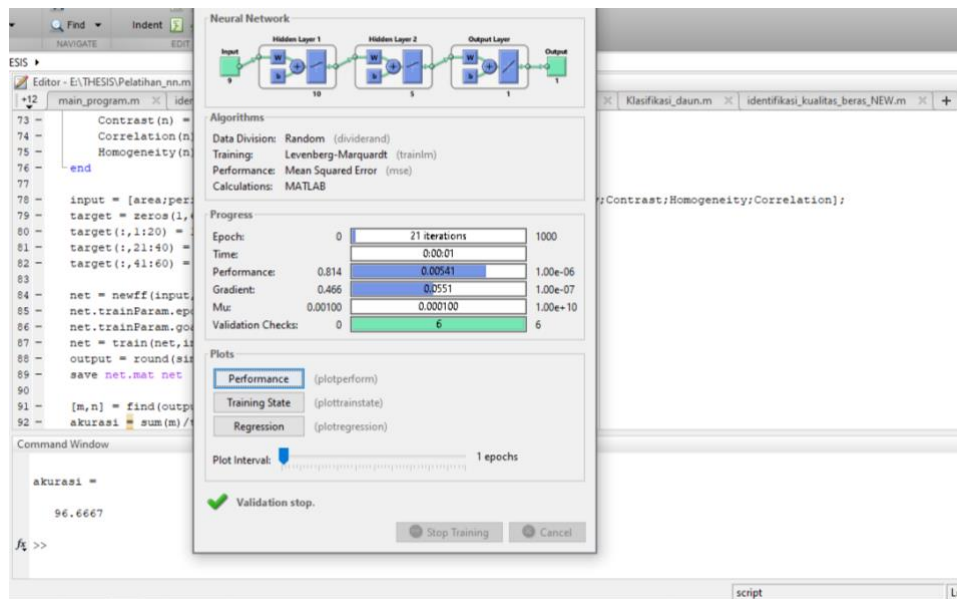
```

val=Cb(valt==1);
val=unique(val);
svmlh(rlhtemp,:)=val;
end
end

```

2.2.2. Algoritma Neural Network

Algoritma kedua yang digunakan pada aplikasi SI-RICE ini adalah Neural Network yang menghasilkan nilai akurasi sebesar 96,67% yang ditampilkan pada Gambar 11 berikut:



Gambar 11. Hasil Akurasi Algoritma Neural Network

Berikut adalah source code yang digunakan pada aplikasi SI-RICE untuk algoritma Neural Network:

```

folder_sirice = 'citra_latih70';
filenames = dir(fullfile(folder_sirice, '*.png'));
sirice_total_citra = numel(filenames);

area = zeros(1,sirice_total_citra);
perimeter = zeros(1,sirice_total_citra);
metric = zeros(1,sirice_total_citra);
eccentricity = zeros(1,sirice_total_citra);
majoraxislength = zeros(1,sirice_total_citra);
minoraxislength = zeros(1,sirice_total_citra);
Contrast = zeros(1,sirice_total_citra);
Homogeneity = zeros(1,sirice_total_citra);
Correlation = zeros(1,sirice_total_citra);

```

```

for rlh_r = 1:sirice_total_citra
    full_name= fullfile(folder_sirice, filenames(rlh_r).name);
    sirice_citra = imread(full_name);

    sirice_cform = makecform('srgb2lab');
    siricelab = applycform(sirice_citra,sirice_cform);

    ab = double(siricelab(:,:,2:3));
    nrows = size(ab,1);
    ncols = size(ab,2);
    ab = reshape(ab,nrows*ncols,2);

    nColors = 2;
    [cluster_idx, ~] = kmeans(ab,nColors,'distance','sqEuclidean', ...
        'Replicates',3);

    siricepix_labels = reshape(cluster_idx,nrows,ncols);

    citra_sirice_segmented = cell(1,3);
    rgb_label = repmat(siricepix_labels,[1 1 3]);

    for k = 1:nColors
        color = sirice_citra;
        color(rgb_label ~= k) = 0;
        citra_sirice_segmented{k} = color;
    end

    sirice_cluster1 = sum(find(siricepix_labels==1));
    sirice_cluster2 = sum(find(siricepix_labels==2));

    [~,cluster_min] = min([sirice_cluster1,sirice_cluster2]);

    Citra_sirice_bw = (siricepix_labels==cluster_min);
    Citra_sirice_bw = imfill(Citra_sirice_bw,'holes');
    Citra_sirice_bw = bwareaopen(Citra_sirice_bw,50);

    stats = regionprops(Citra_sirice_bw,'Area','Perimeter','MajorAxisLength','MinorAxisLength','Eccentricity');
    area(rlh_r) = stats.Area;
    perimeter(rlh_r) = stats.Perimeter;

```

```

majoraxislength(rlh_r) = stats.MajorAxisLength;
minoraxislength(rlh_r) = stats.MinorAxisLength;
metric(rlh_r) = 4*pi*area(rlh_r)/(perimeter(rlh_r)^2);
eccentricity(rlh_r) = stats.Eccentricity;

Citra_sirice_abu = rgb2gray(sirice_citra);
Citra_sirice_abu(~Citra_sirice_bw) = 0;

pixel_dist = 1;
GLCM = graycomatrix(Citra_sirice_abu,'Offset',[0 pixel_dist; -pixel_dist pixel_dist; -pixel_dist 0; -pixel_dist -
pixel_dist]);
stats = graycoprops(GLCM,{'contrast','correlation','homogeneity'});
Contrast(rlh_r) = mean(stats.Contrast);
Correlation(rlh_r) = mean(stats.Correlation);
Homogeneity(rlh_r) = mean(stats.Homogeneity);
end

input = [area;perimeter;majoraxislength;minoraxislength;metric;eccentricity;Contrast;Homogeneity;Correlation];
target = zeros(1,45);
target(:,1:21) = 1;
target(:,22:42) = 2;
target(:,43:63) = 3;

net = newff(input,target,[10 5],{'logsig','logsig'},'trainlm');
net.trainParam.epochs = 1000;
net.trainParam.goal = 1e-6;
net = train(net,input,target);
output = round(sim(net,input));
save net.mat net

[m,rlh_r] = find(output==target);
akurasi = sum(m)/sirice_total_citra*100

```

2.3. Source Code

a. Pembuatan Database Citra Latih

```

folder_sirice = 'citra_latih70';
filenames = dir(fullfile(folder_sirice, '*.png'));
sirice_total_citra = numel(filenames);
sirice_ciridb = zeros(sirice_total_citra,9);

```

```

for rlh_r = 1:sirice_total_citra
    full_name= fullfile(folder_sirice, filenames(rlh_r).name);
    sirice_citra = imread(full_name);

    sirice_cform = makecform('srgb2lab');
    siricelab = applycform(sirice_citra,sirice_cform);

    ab = double(siricelab(:,:,2:3));
    nrows = size(ab,1);
    ncols = size(ab,2);
    ab = reshape(ab,nrows*ncols,2);

    nColors = 2;
    [cluster_idx, ~] = kmeans(ab,nColors,'distance','sqEuclidean', ...
        'Replicates',3);

    siricepix_labels = reshape(cluster_idx,nrows,ncols);

    citra_sirice_segmented = cell(1,3);
    rgb_label = repmat(siricepix_labels,[1 1 3]);

    for k = 1:nColors
        color = sirice_citra;
        color(rgb_label ~= k) = 0;
        citra_sirice_segmented{k} = color;
    end

    sirice_cluster1 = sum(find(siricepix_labels==1));
    sirice_cluster2 = sum(find(siricepix_labels==2));
    [~,cluster_min] = min([sirice_cluster1,sirice_cluster2]);

    Citra_sirice_bw = (siricepix_labels==cluster_min);
    Citra_sirice_bw = imfill(Citra_sirice_bw,'holes');
    Citra_sirice_bw = bwareaopen(Citra_sirice_bw,50);

    stats = regionprops(Citra_sirice_bw,'Area','Perimeter','MajorAxisLength','MinorAxisLength','Eccentricity');
    area = stats.Area;
    perimeter = stats.Perimeter;
    majoraxislength = stats.MajorAxisLength;

```

```

minoraxislength = stats.MinorAxisLength;
metric = 4*pi*area/(perimeter^2);
eccentricity = stats.Eccentricity;

Citra_sirice_abu = rgb2gray(sirice_citra);
Citra_sirice_abu(~Citra_sirice_bw) = 0;

pixel_dist = 1;
GLCM = graycomatrix(Citra_sirice_abu,'Offset',[0 pixel_dist; -pixel_dist pixel_dist; -pixel_dist 0; -pixel_dist -
pixel_dist]);
stats = graycoprops(GLCM,{'contrast','correlation','energy','homogeneity'});
Contrast = mean(stats.Contrast);
Correlation = mean(stats.Correlation);
Energy = mean(stats.Energy);
Homogeneity = mean(stats.Homogeneity);
sirice_ciridb(rlh_r,:) =
[area,perimeter,majoraxislength,minoraxislength,metric,eccentricity,Contrast,Homogeneity,Correlation];
end
save latih_database70 sirice_ciridb

```

b. Identifikasi Citra Beras Halaman Utama

```

function varargout = identifikasi_kualitas_beras_NEW(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @identifikasi_kualitas_beras_NEW_OpeningFcn, ...
    'gui_OutputFcn', @identifikasi_kualitas_beras_NEW_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

```

c. GUI Main Image

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

if ~isequal(filename,0)
    sirice_citra = imread(fullfile(pathname,filename));
    axes(handles.axes1)
    imshow(sirice_citra)
    title('Citra RGB')
else
    return
end

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

axes(handles.axes1)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])

axes(handles.axes2)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])

axes(handles.axes3)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])

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

axes(handles.axes8)
cla reset
set(gca,'XTick',[])
set(gca,'YTick',[])
```

```

set(handles.text2,'String',[])
set(handles.uitable1,'Data',[])
set(handles.edit1,'String',[])
set(handles.edit2,'String',[])
set(handles.edit3,'String',[])

thresh = handles.data3;
[name_file_save,path_save] = uiputfile( ...
    {'*.bmp','File Bitmap (*.bmp)';...
    '*.jpg','File jpeg (*.jpg)';
    '*.tif','File Tif (*.tif)';
    '*.*','All Files (*.*)'},...
    'Save Image');
if ~isequal(name_file_save,0)
    imwrite(thresh,fullfile(path_save,name_file_save));
else
    return
end

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

d. Greyscale Image

```

sirice_citra = handles.sirice_citra;
Gray = rgb2gray(sirice_citra);
thresh = graythresh(Gray);
bw = im2bw(Gray,thresh);

axes(handles.axes2)
cla('reset')
imshow(Gray)
title('Grayscale Image')
axes(handles.axes3)
cla('reset')
imshow(bw)
title('Binary Image')

```

```

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

sirice_citra = handles.sirice_citra;
Gray = rgb2gray(sirice_citra)
axes(handles.axes4)
imshow(Gray)
title('Grayscale Image');

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

```

e. Binary Image

```

sirice_citra = handles.sirice_citra;
sirice_citra_reg2 = handles.sirice_citra_reg2
Gray = rgb2gray(sirice_citra_reg2);
thresh = graythresh(Gray);
bw = im2bw(Gray,thresh);

axes(handles.axes3)
cla('reset')
imshow(bw)
title('Binary Image')
set(handles.edit1,'String',thresh)
handles.Citra_sirice_bwg = bw;
guidata(hObject, handles)

```

f. Ekstraksi Feature GLCM & Regionprops

```

sirice_citra = handles.sirice_citra;
sirice_cform = makecform('srgb2lab');
siricelab = applycform(sirice_citra,sirice_cform);

ab = double(siricelab(:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);

```

```

nColors = 2;
[cluster_idx, ~] = kmeans(ab,nColors,'distance','sqEuclidean', ...
    'Replicates',3);

siricepix_labels = reshape(cluster_idx,nrows,ncols);

citra_sirice_segmented = cell(1,3);
rgb_label = repmat(siricepix_labels,[1 1 3]);

%segmentasi k-means
for k = 1:nColors
    color = sirice_citra;
    color(rgb_label ~= k) = 0;
    citra_sirice_segmented{k} = color;
end

sirice_cluster1 = sum(find(siricepix_labels==1));
sirice_cluster2 = sum(find(siricepix_labels==2));

[~,cluster_min] = min([sirice_cluster1,sirice_cluster2]);

Citra_sirice_bw = (siricepix_labels==cluster_min);
Citra_sirice_bw = imfill(Citra_sirice_bw,'holes');
Citra_sirice_bw = bwareaopen(Citra_sirice_bw,50);

stats = regionprops(Citra_sirice_bw,'Area','Perimeter','MajorAxisLength','MinorAxisLength','Eccentricity');
area = stats.Area;
perimeter = stats.Perimeter;
majoraxislength = stats.MajorAxisLength;
minoraxislength = stats.MinorAxisLength;
metric = 4*pi*area/(perimeter^2);
eccentricity = stats.Eccentricity;

Citra_sirice_abu = rgb2gray(sirice_citra);
Citra_sirice_abu(~Citra_sirice_bw) = 0;

pixel_dist = 1;

```

```

GLCM = graycomatrix(Citra_sirice_abu,'Offset',[0 pixel_dist; -pixel_dist pixel_dist; -pixel_dist 0; -pixel_dist -
pixel_dist]);
stats = graycoprops(GLCM,{'contrast','correlation','energy','homogeneity'});
Contrast = mean(stats.Contrast);
Correlation = mean(stats.Correlation);
Energy = mean(stats.Energy);
Homogeneity = mean(stats.Homogeneity);

sirice_ujldb =
[area,perimeter,majoraxislength,minoraxislength,metric,eccentricity,Contrast,Homogeneity,Correlation];

ciri_total = cell(9,2);
ciri_total{1,1} = 'Area';
ciri_total{2,1} = 'Perimeter';
ciri_total{3,1} = 'MajorAxisLength';
ciri_total{4,1} = 'MinorAxisLength';
ciri_total{5,1} = 'Metric';
ciri_total{6,1} = 'Eccentricity';
ciri_total{7,1} = 'Contrast';
ciri_total{8,1} = 'Homogeneity';
ciri_total{9,1} = 'Correlation';
ciri_total{1,2} = num2str(area);
ciri_total{2,2} = num2str(perimeter);
ciri_total{3,2} = num2str(majoraxislength);
ciri_total{4,2} = num2str(minoraxislength);
ciri_total{5,2} = num2str(metric);
ciri_total{6,2} = num2str(eccentricity);
ciri_total{7,2} = num2str(Contrast);
ciri_total{8,2} = num2str(Homogeneity);
ciri_total{9,2} = num2str(Correlation);

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

set(handles.text2,'String','Ekstraksi Ciri')
set(handles.uitable1,'Data',ciri_total,'RowName',1:9)
handles.sirice_ujldb = sirice_ujldb;
guidata(hObject, handles)

```

g. Segmentation Image

```
sirice_citra = handles.sirice_citra;

data = sirice_citra(:,:,2);
seg_I = imquantize(data,[90 300]);
RGB = label2rgb(seg_I);
citra_sirice_segmented = cell(1,3);
rgb_label = repmat(seg_I,[1 1 3]);

color = sirice_citra;
color(rgb_label ~= 1) = 1;
citra_sirice_segmented{1} = color;
sirice_citra_reg1 = citra_sirice_segmented{1};
color = sirice_citra;
color(rgb_label ~= 2) = 1;
citra_sirice_segmented{2} = color;
sirice_citra_reg2 = citra_sirice_segmented{2};
color = sirice_citra;
color(rgb_label ~= 3) = 1;
citra_sirice_segmented{3} = color;

axes(handles.axes2)
cla('reset')
imshow(sirice_citra_reg1), title('Segmented Image Cluster 1');

axes(handles.axes3)
cla('reset')
imshow(sirice_citra_reg2), title('Segmented Image Cluster 2');

handles.sirice_citra_reg2 = sirice_citra_reg2;
guidata(hObject, handles)

switch output
case 1
    jenis_beras = 'Basmati';
case 2
    jenis_beras = 'Japonica';
case 3
    jenis_beras = 'Jasmine';
```

```

otherwise
    jenis_beras = 'tidak dikenali';
end

function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

sirice_citra = handles.sirice_citra;
sirice_cform = makecform('srgb2lab');
siricelab = applycform(sirice_citra,sirice_cform);

ab = double(siricelab(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);

nColors = 2;
[cluster_idx, ~] = kmeans(ab,nColors,'distance','sqEuclidean', ...
    'Replicates',3);

siricepix_labels = reshape(cluster_idx,nrows,ncols);

citra_sirice_segmented = cell(1,3);
rgb_label = repmat(siricepix_labels,[1 1 3]);
for k = 1:nColors
    color = sirice_citra;
    color(rgb_label ~= k) = 0;
    citra_sirice_segmented{k} = color;
end

sirice_cluster1 = sum(find(siricepix_labels==1));
sirice_cluster2 = sum(find(siricepix_labels==2));

[~,cluster_min] = min([sirice_cluster1,sirice_cluster2]);

Citra_sirice_bw = (siricepix_labels==cluster_min);
Citra_sirice_bw = imfill(Citra_sirice_bw,'holes');
Citra_sirice_bw = bwareaopen(Citra_sirice_bw,50);

```

```

stats = regionprops(Citra_sirice_bw,'Area','Perimeter','MajorAxisLength','MinorAxisLength','Eccentricity');
area = stats.Area;
perimeter = stats.Perimeter;
majoraxislength = stats.MajorAxisLength;
minoraxislength = stats.MinorAxisLength;
metric = 4*pi*area/(perimeter^2);
eccentricity = stats.Eccentricity;

Citra_sirice_abu = rgb2gray(sirice_citra);
Citra_sirice_abu(~Citra_sirice_bw) = 0;

```

h. Identification Image

```

pixel_dist = 1;
GLCM = graycomatrix(Citra_sirice_abu,'Offset',[0 pixel_dist; -pixel_dist pixel_dist; -pixel_dist 0; -pixel_dist -
pixel_dist]);
stats = graycoprops(GLCM,{'contrast','correlation','energy','homogeneity'});
Contrast = mean(stats.Contrast);
Correlation = mean(stats.Correlation);
Energy = mean(stats.Energy);
Homogeneity = mean(stats.Homogeneity);

sirice_ujldb =
[area,perimeter,majoraxislength,minoraxislength,metric,eccentricity,Contrast,Homogeneity,Correlation];
input = [area;perimeter;majoraxislength;minoraxislength;metric;eccentricity;Contrast;Homogeneity;Correlation];

load net
output = round(sim(net,input));

if output == 1
    kelas = 'Basmati';
elseif output == 2
    kelas = 'Japonica';
elseif output == 3
    kelas = 'Jasmine';
end

set(handles.edit3,'String',kelas)

```

```

function edit3_Callback(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

sirice_citra_reg2 = handles.sirice_citra_reg2;

axes(handles.axes8)
cla('reset')
imshow(sirice_citra_reg2)

I = sirice_citra_reg2;
bw = im2bw(I, graythresh(I));
bw2 = imfill(bw,'holes');

s = regionprops(bw2, 'centroid','Area','Perimeter','MajorAxisLength','MinorAxisLength','Eccentricity');
centroids = cat(1, s.Centroid);
area = cat(1, s.Area);
perimeter = cat(1, s.Perimeter);
MajorAxisLength = cat(1, s.MajorAxisLength);
MinorAxisLength = cat(1, s.MinorAxisLength);
Eccentricity = cat(1, s.Eccentricity);
hold on
plot(centroids(:,1), centroids(:,2), 'r*')

[B,L] = bwboundaries(bw2,'noholes');
[~,num] = bwlabel(bw2,8);

for k = 1:num
    boundary = B{k};
    plot(boundary(:,2), boundary(:,1), 'y', 'LineWidth', 2)
    text(boundary(1,2),boundary(1,1)-7,strcat([' ',num2str(k)]),'Color','r',...
        'FontSize',5,'FontWeight','bold');
    text(boundary(1,2),boundary(1,1)+14,strcat(['Major = ',num2str(MajorAxisLength(k))]),'Color','c',...
        'FontSize',5,'FontWeight','bold');
    text(boundary(1,2),boundary(1,1)+21,strcat(['Minor = ',num2str(MinorAxisLength(k))]),'Color','c',...
        'FontSize',5,'FontWeight','bold');
    text(boundary(1,2),boundary(1,1)+28,strcat(['Area = ',num2str(area(k))]),'Color','c',...
        'FontSize',5,'FontWeight','bold');
    text(boundary(1,2),boundary(1,1)+35,strcat(['Perimeter = ',num2str(perimeter(k))]),'Color','c',...

```

```
'FontSize',5,'FontWeight','bold');  
end  
  
hold off
```

REFERENSI

- Nurfalah, Ridan, dkk. 2021. Identifikasi Citra Beras Menggunakan Algoritma Multi-SVM dan Neural Network Pada Segmentasi K-Means. Jurnal RESTI Vol 4 No. 6.
- Pamungkas, Adi. Pemrograman MATLAB: Pengolahan Citra Digital, Pengolahan Video, Pengenalan Pola, dan Data Mining. Diunduh 13 Februari 2020.
<https://pemrogramanmatlab.com>