LAMPIRAN

Lampiran 1 Kode Program Ekstraksi Fitur

```
nama_folder = 'JPG/robusta';

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

jumlah_file = numel(nama_file);

data = cell(jumlah_file,10);


for n = 1:jumlah_file

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

    HSV = rgb2hsv(Img);

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

    bw = imbinarize(V);

    bw2 = imcomplement(bw);

    bw3 = imfill(bw2,'holes');

    bw4 = medfilt2(bw3,[5 5]);

    bw5 = bwareaopen(bw4, 100000);

    H(~bw5)      = 0;
    S(~bw5)      = 0;

    GLCM = graycomatrix(V,'offset',[0 1;...
    -1 1; -1 0; -1 -1]);

    stats =
graycoprops(GLCM,{'Contrast','Correlation','Energy','Ho
mogeneity'});


    Contrast     = stats.Contrast;
    Correlation = stats.Correlation;
```

```
 Energy      = stats.Energy;
 Homogeneity = stats.Homogeneity;

 s = regionprops(bw5,'Area','Perimeter');

 Contrast    = (mean(Contrast));
 Correlation = (mean(Correlation));
 Energy      = (mean(Energy));
 Homogeneity = (mean(Homogeneity));
 Hue         = sum(sum(H))/sum(sum(bw5));
 Saturation  = sum(sum(S))/sum(sum(bw5));
 Luas        = cat(1,s.Area);
 Keliling    = cat(1,s.Perimeter);
 Roundness   = 4*pi*Luas./(Keliling.^2);

data{n,1} = nama_file(n).name;
data{n,2} = Contrast;
data{n,3} = Correlation;
data{n,4} = Energy;
data{n,5} = Homogeneity;
data{n,6} = Hue;
data{n,7} = Saturation;
data{n,8} = Luas;
data{n,9} = Keliling;
data{n,10} = Roundness;
```

Lampiran 2 Kode Program Klasifikasi Kualitas Biji Kopi Robusta

```
import pandas as pd

kopi_df = pd.read_csv('Tesis_600data.csv')

from sklearn.model_selection import train_test_split

from sklearn.metrics import
accuracy_score,confusion_matrix, classification_report


X = kopi_df.drop(columns='Kualitas')


# Normalisasi Data

X_min = X.min()

X_max = X.max()

X_range = (X_max - X_min)

X_scaled = (X - X_min)/(X_range)


y = kopi_df['Kualitas']


X_train, X_test, y_train, y_test =
train_test_split(X_scaled, y, test_size=0.2,
random_state=2)


from sklearn.naive_bayes import GaussianNB

#Naive Bayes Classifier

model_nb = GaussianNB()

model_nb.fit(X_train, y_train)

y_pred = model_nb.predict(X_test)


print(accuracy_score(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

print(classification_report(y_test, y_pred))

# Decission Tree Classifier
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree

model = DecisionTreeClassifier(max_depth=4)

model_dt = DecisionTreeClassifier()
model_dt.fit(X_train, y_train)
y_pred = model_dt.predict(X_test)

print(accuracy_score(y_test, y_pred))

print(confusion_matrix(y_test, y_pred))

print(classification_report(y_test, y_pred)
```