Lampiran 1
Lembar Bimbingan

LEMBAR BIMBINGAN SKRIPSI

Nama : GITA JULIWANTI
NIM : 2017694900030
Jurusan : Teknik Informatika
Konsentrasi : Jaringan
Judul : KLASIFIKASI PENYAKIT PADA DAUN TOMAT MENGGUNAKAN LS-SVM BERDASARKAN EKSTRAKSI FITUR WARNA *COLOR MOMENT*

No	Hari	Tgl	BAB	MateriBimbingan	TTD Pembimbing
1.	Rabu	17 Feb 2021	-	Pemaparanusulantopik	
2.	Rabu	24 Feb 2021	-	Pemantapantopik&Judul	
3.	Rabu	03 Mar 202	I	Revisi BAB I	

		1			
4.	Rabu	10 Mar 2021	I & II	Revisi BAB I & Revisi BAB II	
5.	Kamis	25 Mar 2021	III	Check plagiasi &Revisi BAB III	
6.	Rabu	31 Maret 2021	III	Revisi BAB III	
7.	Rabu	07 April 2021	III	Revisi format penulisan	
8.	Kamis	15 April 202	I-III	Revisi BAB I, II & III	

		1			
9.	Kamis	20 Mei 2021	-	Pembahasanhasil seminar proposal	
10.	Kamis	27 Mei 2021	-	Pembahasanhasil seminar proposal	
11.	Rabu	02 Juni 2021	I, II, III	RevisiKeseluruan	
12.	Kamis	10 Juni 2021	III	Revisi BAB III	
13.	Selasa	15 Juni 2021	IV	Progrespembangunanprojek dan BAB VI	
14.	Rabu	23 Juni	IV	Progrespembangunanprojek dan BAB VI	

		2021			
15.	Rabu	30 Juni 2021	IV	Progrespembangunanprojek	
16.	Sabtu	03 Juli 2021	IV	Progrespembangunanprojek	
17.	Kamis	08 Juli 2021	IV	Pengujian	
18.	Kamis	15 Juli 2021	IV	Pengujian	
19.	Kamis	22 Juli 2021	IV	Pengujian& BAB IV	
20.	Rabu	28	V	Kesimpulan & Saran	

		Juli 2021			

Pasuruan, 4 Agustus 2021
Pembimbing

M.Imron Rosadi, M.Kom
NIP.Y0690213121

Lampiran 2

Coding Aplikasi

```
function [BW,maskedRGBImage] = maskTrain(RGB)
%maskTrain Threshold gambar RGB menggunakankode
yang dibuatsecaraotomatisdaricolorThresholder
app.
% [BW,MASKEDRGBIMAGE] = maskTrain(RGB)
ambangbatas(thresholds) gambar RGB menggunakan
% kode yang
dibuatsecaraotomatisdaricolorThresholder app.
Ruang warna dan
% rentang(range) untuksetiap channel
ruangwarnaditetapkandalam app. Itu
% segmentasi mask dikembalikandalam BW, dan
gabungan mask dan
% gambar RGB
aslidikembalikandalammaskedRGBImage.
%-----------------------------------------------
-------

% Konversigambar RGB ke HSV
I = rgb2hsv(RGB);

% Tentukan thresholds untuk channel 1
berdasarkanpengaturan histogram
channel1Min = 0.127;
channel1Max = 0.375;

% Tentukan thresholds untuk channel 2
berdasarkanpengaturan histogram
channel2Min = 0.075;
channel2Max = 0.694;

% Tentukan thresholds untuk channel 3
berdasarkanpengaturan histogram
channel3Min = 0.108;
channel3Max = 0.855;
```

```
% Buat mask berdasarkan thresholds histogram
yang dipilih
sliderBW = (I(:,:,1) >= channel1Min ) &
(I(:,:,1) <= channel1Max) &...
    (I(:,:,2) >= channel2Min ) & (I(:,:,2) <=
channel2Max) &...
    (I(:,:,3) >= channel3Min ) & (I(:,:,3) <=
channel3Max);
BW = sliderBW;

% Inisialisasi output masked image berdasarkan
input gambar
maskedRGBImage = RGB;

% Atur background pixels dimana BW false kenol
maskedRGBImage(repmat(~BW,[1 1 3])) = 0;

end
```

Ekstraksi Fitur Color Moment

```
clc; clear; close all;

fiturTrain = []; % array menyimpannilaifitur
Train sementara
labelTrain = [];
folderTrain =
'C:\Users\VAIO\Documents\MATLAB\bismillah\dataTr
ain'; %folder data latih

dirFolder = dir(folderTrain);

dirFolder = dirFolder(3:end);
fori = 1:length(dirFolder)
F_name = dirFolder(i).name;
dirFile = dir([folderTrain,
'\',F_name,'\*.jpg']);
   hm = length(dirFile);
```

```
   data(i).file_name = F_name;
   data(i).file_location = [folderTrain,
'\',F_name];
   source = imread(data(i).file_location); %
Read All Image PNG

   layer = rgb2hsv(source);
layerBW = im2bw(layer,graythresh(layer))
%      [BW,maskedRGBImage] = maskTrain(img);
%      img_hsv = rgb2hsv(maskedRGBImage);

   group = cell(120, 1); % TOTAL 120 data
% label gesturangka pada tangan ==>group(Xawal,
Xakhir) = {'X'};
group(1:30) = {'A'};  % EB
group(31:60) = {'B'}; % LM
group(61:90) = {'C'}; % Sehat
group(91:120) = {'D'}; % TMV

% Klasifikasi Data Train (Proses
pengenalankekomputer)
if(i>= 1 &&i<= 30)
       data(i).class = 1; %
elseif(i>=31 &&i<= 60)
       data(i).class = 2; %
elseif(i>=61 &&i<= 90)
       data(i).class = 3; %
elseif(i>=91 &&i<= 120)
       data(i).class = 4; %
end

    data(i).rata = mean(mean(layer(:,:,1)));
    data(i).standar =
std(std(double(layer(:,:,1))));
    data(i).skewn =
skewness(skewness(double(layer(:,:,1))));
```

```
   train =
[data(i).class;data(i).rata;data(i).standar;data
(i).skewn]'; % Save Hasil EkstraksiCiritiap File
Gambar
fiturTrain = [fiturTrain; train];

   label = data(i).class'; % Save Hasil
EkstraksiCiritiap File Gambar
labelTrain = [labelTrain; label];
%     disp(data(i));
end

disp(fiturTrain);

% x = struct2table(data); % export data
menjadiTabel
% writetable(x, 'EkstraksiTeksturUji.xlsx'); %
Export ke Excel

% xlswrite('train_set.xls', fiturTrain);
xlswrite('train.xls', fiturTrain);
% savedbFiturTrain.mat group fiturTrain; % Save
Nilai Fitur Train kedalamdbFitur.mat
% savedbLabelTrain.mat group labelTrain;

disp('EkstraksiBerhasil')
```

Klasifikasi LS-SVM berdasarkanfitur CM

```
addpath(genpath('C:\Users\VAIO\Documents\aplikas
i\LSSVMlab'));

load ('datatraining.mat');
load ('datatesting.mat');
X=training(:,1:3);
Y=training(:,4);
Xt=testing(:,1:3);
Yt=testing(:,4);
%
```

```
% train LS-SVM classifier with linear kernel
%
type='c';
gam = 1;
disp('Linear kernel'),

[alpha,b] =
trainlssvm({X,Y,type,gam,[],'lin_kernel'});

figure;
plotlssvm({X,Y,type,gam,[],'lin_kernel','preproc
ess'},{alpha,b});

[Yht, Zt] =
simlssvm({X,Y,type,gam,[],'lin_kernel'},
{alpha,b}, Xt);

err = sum(Yht~=Yt);
fprintf('\n on test: #misclass = %d, error rate
= %.2f%%\n', err, err/length(Yt)*100)

acc = sum(( Yht== Yt))/length(Yt)*100;
fprintf('\n on test: #acc = %d', acc,
acc/length(Yt)*100)
disp('Press any key to continue...'), pause,




%
% Train the LS-SVM classifier using polynomial
kernel
%
type='c';
gam = 1;
t = 1;
degree = 5;
```

```
[alpha,b] = trainlssvm({X,Y,type,gam,[t;
degree],'poly_kernel'});

figure; plotlssvm({X,Y,type,gam,[t;
degree],'poly_kernel','preprocess'},{alpha,b});

[Yht, Zt] = simlssvm({X,Y,type,gam,[t;
degree],'poly_kernel'}, {alpha,b}, Xt);

err = sum(Yht~=Yt);
fprintf('\n on test: #misclass = %d, error rate
= %.2f%%\n', err, err/length(Yt)*100)
acc = sum(( Yht== Yt))/length(Yt)*100;
fprintf('\n on test: #acc = %d', acc,
acc/length(Yt)*100)
disp('Press any key to continue...'), pause,


%
% use RBF kernel
%

% tune the sig2 while fix gam
%
disp('RBF kernel')
gam = 1; sig2list=[0.01, 0.1, 1, 5, 10, 25];

errlist=[];

for sig2=sig2list,
disp(['gam : ', num2str(gam), '   sig2 : ',
num2str(sig2)]),
    [alpha,b] =
trainlssvm({X,Y,type,gam,sig2,'RBF_kernel'});

% Plot the decision boundary of a 2-d LS-SVM
classifier
```

```
plotlssvm({X,Y,type,gam,sig2,'RBF_kernel','prepr
ocess'},{alpha,b});

% Obtain the output of the trained classifier
    [Yht, Zt] =
simlssvm({X,Y,type,gam,sig2,'RBF_kernel'},
{alpha,b}, Xt);
    err = sum(Yht~=Yt); errlist=[errlist; err];
fprintf('\n on test: #misclass = %d, error rate
= %.2f%% \n', err, err/length(Yt)*100)
acc = sum(( Yht== Yt))/length(Yt)*100;
fprintf('\n on test: #acc = %d', acc,
acc/length(Yt)*100)
disp('Press any key to continue...'), pause,
end



%
% make a plot of the misclassification rate wrt.
sig2
%
figure;
plot(log(sig2list), errlist, '*-'),
xlabel('log(sig2)'), ylabel('number of
misclass'),
```

KlasifikasiberdasarkanfiturGLCM

```
% Project Title: Pomegranate Leaf Disease
Detection
%
addpath(genpath('C:\Users\VAIO\Documents\MATLAB\
Plant
Disease_MutiSVM\Leaf_Disease_Detection_code'));

clc
close all
clear all
```

```
% Train_Label = [];
[filename, pathname] =
uigetfile({'*.*';'*.bmp';'*.jpg';'*.gif'}, 'Pick
a Leaf Image File');
I = imread([pathname,filename]);
I = imresize(I,[256,256]);
%figure, imshow(I); title('Query Leaf Image');

% Enhance Contrast
I = imadjust(I,stretchlim(I));
figure, imshow(I);title('Contrast Enhanced');

% Otsu Segmentation
I_Otsu = im2bw(I,graythresh(I));
% Conversion to HIS
I_HIS = rgb2hsi(I);

%% Extract Features

% Function call to evaluate features
%[feat_diseaseseg_img] =  EvaluateFeatures(I)

% Color Image Segmentation
% Use of K Means clustering for segmentation
% Convert Image from RGB Color Space to L*a*b*
Color Space
% The L*a*b* space consists of a luminosity
layer 'L*', chromaticity-layer 'a*' and 'b*'.
% All of the color information is in the 'a*'
and 'b*' layers.
cform = makecform('srgb2lab');
% Apply the colorform
lab_he = applycform(I,cform);

% Classify the colors in a*b* colorspace using K
means clustering.
```

```
% Since the image has 3 colors create 3
clusters.
% Measure the distance using Euclidean Distance
Metric.
ab = double(lab_he(:,:,2:3));
nrows = size(ab,1);
ncols = size(ab,2);
ab = reshape(ab,nrows*ncols,2);
nColors = 3;
[cluster_idxcluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean', ...
'Replicates',3);
%[cluster_idxcluster_center] =
kmeans(ab,nColors,'distance','sqEuclidean','Repl
icates',3);
% Label every pixel in tha image using results
from K means
pixel_labels = reshape(cluster_idx,nrows,ncols);
%figure,imshow(pixel_labels,[]), title('Image
Labeled by Cluster Index');

% Create a blank cell array to store the results
of clustering
segmented_images = cell(1,3);
% Create RGB label using pixel_labels
rgb_label = repmat(pixel_labels,[1,1,3]);

for k = 1:nColors
colors = I;
colors(rgb_label ~= k) = 0;
segmented_images{k} = colors;
end



figure,
subplot(3,1,1);imshow(segmented_images{1});title
('Cluster 1');
```

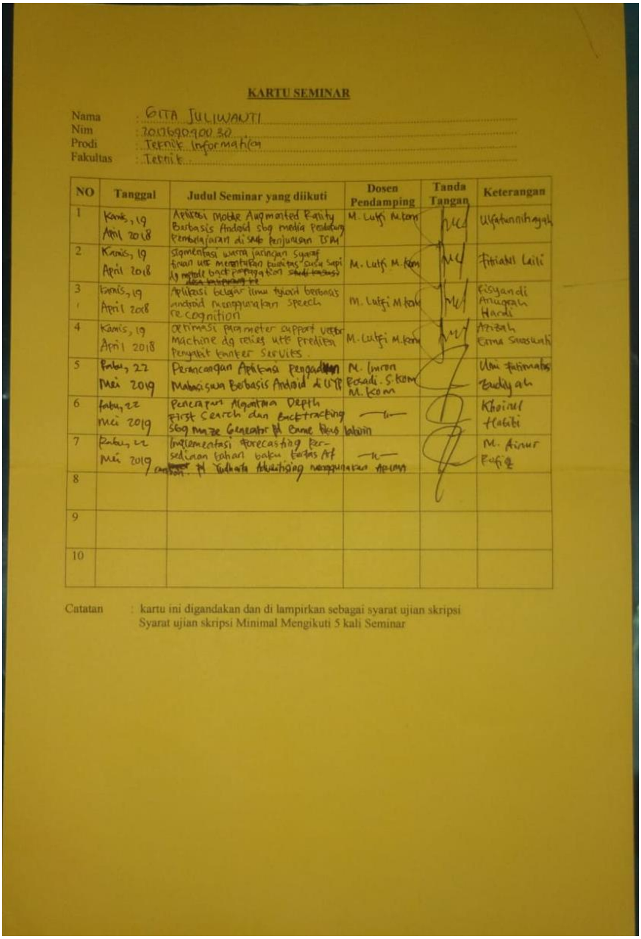
```
subplot(3,1,2);imshow(segmented_images{2});title
('Cluster 2');
subplot(3,1,3);imshow(segmented_images{3});title
('Cluster 3');
set(gcf, 'Position', get(0,'Screensize'));

% Feature Extraction
x = inputdlg('Enter the cluster no. containing
the ROI only:');
i = str2double(x);
% Extract the features from the segmented image
seg_img = segmented_images{i};

% Convert to grayscale if image is RGB
ifndims(seg_img) == 3
img = rgb2gray(seg_img);
end
%figure, imshow(img); title('Gray Scale Image');

% Evaluate the disease affected area
black = im2bw(seg_img,graythresh(seg_img));
%figure, imshow(black);title('Black & White
Image');
m = size(seg_img,1);
n = size(seg_img,2);

zero_image = zeros(m,n);
%G = imoverlay(zero_image,seg_img,[1 0 0]);

cc = bwconncomp(seg_img,6);
diseasedata = regionprops(cc,'basic');
A1 = diseasedata.Area;
sprintf('Area of the disease affected region is
: %g%',A1);

I_black = im2bw(I,graythresh(I));
kk = bwconncomp(I,6);
leafdata = regionprops(kk,'basic');
```

```
A2 = leafdata.Area;
sprintf(' Total leaf area is : %g%',A2);

%Affected_Area = 1-(A1/A2);
Affected_Area = (A1/A2);
ifAffected_Area< 0.1
Affected_Area = Affected_Area+0.15;
end
sprintf('Affected Area is:
%g%%',(Affected_Area*100))

% Create the Gray Level Cooccurance Matrices
(GLCMs)
glcms = graycomatrix(img);

% Derive Statistics from GLCM
stats = graycoprops(glcms,'Contrast Correlation
Energy Homogeneity');
Contrast = stats.Contrast;
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(seg_img);
Standard_Deviation = std2(seg_img);
Entropy = entropy(seg_img);
RMS = mean2(rms(seg_img));
%Skewness = skewness(img)
Variance = mean2(var(double(seg_img)));
a = sum(double(seg_img(:)));
Smoothness = 1-(1/(1+a));
Kurtosis = kurtosis(double(seg_img(:)));
Skewness = skewness(double(seg_img(:)));
% Inverse Difference Movement
m = size(seg_img,1);
n = size(seg_img,2);
in_diff = 0;
fori = 1:m
for j = 1:n
        temp = seg_img(i,j)./(1+(i-j).^2);
```

```
in_diff = in_diff+temp;
end
end
IDM = double(in_diff);

feat_disease =
[Contrast,Correlation,Energy,Homogeneity, Mean,
Standard_Deviation, Entropy, RMS, Variance,
Smoothness, Kurtosis, Skewness, IDM];
% Train_Feat = feat_disease;

% label = helpdlg'; % Save Hasil
EkstraksiCiritiap File Gambar
% Train_Label = [Train_Label; label];
% xlswrite('train.xls', feat_disease);
% saveTraining_Data.matTrain_Feat;
%%
% Load All The Features

load('Training_Data.mat')

% Put the test features into variable 'test'
test = feat_disease;
result = multisvm(Train_Fitur,Train_Label,test);
%disp(result);

% Visualize Results
if result == 0
helpdlg(' early_blight ');
disp(' early_blight ');
elseif result == 1
helpdlg(' leaf_mold ');
disp('leaf_mold');
elseif result == 2
helpdlg(' sehat ');
disp(' sehat ');
elseif result == 3
helpdlg(' tmv ');
disp('tmv');
```

```
end

%% Evaluate Accuracy
load('Accuracy_Data.mat')
Accuracy_Percent= zeros(200,1);
fori = 1:500
data = Train_Fitur;
%groups = ismember(Train_Label,1);
groups = ismember(Train_Label,0);
[train,test] = crossvalind('HoldOut',groups);
cp = classperf(groups);
svmStruct =
svmtrain(data(train,:),groups(train),'showplot',
false,'kernel_function','linear');
classes =
svmclassify(svmStruct,data(test,:),'showplot',fa
lse);
classperf(cp,classes,test);
Accuracy = cp.CorrectRate;
Accuracy_Percent(i) = Accuracy.*100;
end
Max_Accuracy = max(Accuracy_Percent);
sprintf('Accuracy of Linear Kernel with 500
iterations is: %g%%',Max_Accuracy)
```

Lampiran 3
Bukti kartu seminar



KARTU SEMINAR

Nama : GITA JULIWANTI
Nim : 2017690900 30
Prodi : Teknik Informatika
Fakultas : Teknik

NO	Tanggal	Judul Seminar yang diikuti	Dosen Pendamping	Tanda Tangan	Keterangan
1	Kamis, 19 April 2018	Aplikasi Mobile Augmented Reality Berbasis Android sbg media Pendukung Pembelajaran di SMK Penjurusan TSM	M. Lutfi M.Kom		Ulfatunnihayah
2	Kamis, 19 April 2018	Segmentasi warna jaringan syaraf tiruan utk menentukan kualitas susu sapi dg metode backpropagation	M. Lutfi M.Kom		Fitriatul Laili
3	Kamis, 19 April 2018	Aplikasi belajar ilmu tajwid berbasis android menggunakan speech recognition	M. Lutfi M.Kom		Fisyandi Anugrah Hardi
4	Kamis, 19 April 2018	Optimasi parameter support vector machine dg teliti utk prediksi penyakit kanker serviks.	M. Lutfi M.Kom		Azizah Erma Susanti
5	Rabu, 22 Mei 2019	Perancangan Aplikasi pengaduan Mahasiswa Berbasis Android di UYP	M. Imron Rosadi. S.Kom M.Kom		Umi Fatimatus Zudiyah
6	Rabu, 22 Mei 2019	Penerapan Algoritma Depth First Search dan Backtracking sbg maze Generator pd Game Rescue labirin	—"—		Khoirul Habibi
7	Rabu, 22 Mei 2019	Implementasi forecasting persediaan bahan baku kertas Art carton pd Yudharta Advertising menggunakan Arima	—"—		M. Ainur Rofiq
8					
9					
10					

Catatan : kartu ini digandakan dan di lampirkan sebagai syarat ujian skripsi
Syarat ujian skripsi Minimal Mengikuti 5 kali Seminar

Lampiran 4
Bukti hasil plagiasi

Lampiran 6
Curriculum Vitae

DATA PRIBADI

Nama	: Gita Juliwanti
NIM	: 201769040030
Tempat,tanggal lahir	: Pasuruan, 31 Agustus 1999
Jenis Kelamin	: Perempuan
Agama	: Islam
Perguruan Tinggi	:Universitas yudharta Pasuruan
Fakultas	: Teknik
Program Studi	: Teknik Informatika
Email	: gitajuliwanti31@gmail.com
Alamat	: Kesiman – Lecari – Sukorejo
Riwayat Pendidikan	: 1. MI Nurul Huda tahun 2005 - 2011 2. SMPN 2 Sukorejotahun 2011 - 2014 3. SMKN2 Sukorejotahun 2014 - 2017
Organisasi	: 1. HimpunanMahasiswaInformatika tahun 2017 – 2020 2. PKPT tahun 2017 - 2020