LAMPIRAN

Lampiran 1. Daftar Riwayat Hidup

Anggy Jovano Vano

Jl. Raya Wonosari, Ds. Wonosari RT/RW: 03/03
Kec. Gondangwetan
Kab. Pasuruan

089530420544

anggyjovano11@gmail.com

Data Pribadi

- Tempat/Tanggal Lahir : Pasuruan, 21 November 1997
- Jenis Kelamin : Laki-laki
- Agama : Islam
- Kewarganegaraan : Indonesia
- Status : Belum Menikah

Riwayat Pendidikan

- SDN Wonosari, 2003-2009
- SMPN 2 Gondangwetan, 2009-2012
- SMK Syamsul Arifin, 2012-2015
- Universitas Yudharta Pasuruan, 2017-2021

Pengalaman Organisasi

- Anggota Organisasi Siswa Intra Sekolah (OSIS), 2012-2015
- Teater Semut Hitam, 2012-2015
- Anggota Himpunan Mahasiswa Teknik Informatika, 2017-2019

Kemampuan

- Mengoperasikan Microsoft Office
- Mendesain dengan CorelDraw
- Photography

Lampiran 2 Lembar Bimbingan Skripsi

LEMBAR BIMBINGAN SKRIPSI

Nama : ANGGY JOVANO
NIM : 2017694900056
Jurusan : Teknik Informatika
Konsentrasi : Jaringan
Judul : KLASIFIKASI JENIS PENYAKIT DAUN ANGGUR MENGGUNAKAN METODE EKSTRAKSI FITUR GLCM DAN *NEURAL NETWORK*

No	Hari	Tgl	BAB	Materi Bimbingan	TTD Pembimbing
1.	Rabu	17 Feb 2021	-	Pemaparan usulan topik	
2.	Rabu	24 Feb 2021	-	Pemantapan topik & Judul	
3.	Rabu	03 Mar 2021	I	Revisi BAB I	
4.	Rabu	10 Mar 2021	I & II	Revisi BAB I & Revisi BAB II	
5.	Kamis	25	III	Check	

		Mar 2021		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 2021	I-III	Revisi BAB I, II & III	
9.	Kamis	20 Mei 2021	-	Pembahasan hasil seminar proposal	
10.	Kamis	27 Mei 2021	-	Pembahasan hasil seminar proposal	
11.	Rabu	02 Juni 2021	I, II, III	Revisi Keseluruan	

12.	Kamis	10 Juni 2021	III	Revisi BAB III	
13.	Selasa	15 Juni 2021	IV	Progres pembangunan projek dan BAB VI	
14.	Rabu	23 Juni 2021	IV	Progres pembangunan projek dan BAB VI	
15.	Rabu	30 Juni 2021	IV	Progres pembangunan projek	
16.	Sabtu	03 Juli 2021	IV	Progres pembangunan projek	
17.	Kamis	08 Juli 2021	IV	Pengujian	
18.	Kamis	15	IV	Pengujian	

		Juli 2021			
19.	Kamis	22 Juli 2021	IV	Pengujian & BAB IV	
20.	Rabu	28 Juli 2021	V	Kesimpulan & Saran	

Pasuruan, Agustus 2021
Pembimbing

M. Imron Rosadi, M.Kom
NIP.Y 0690213121

Lampiran 3 Coding

Coding pemisahan objek dengan background

```
image_folder = 'D:\img\';
image_target = 'D:\img\';
filenames = dir(fullfile(image_folder, '*.jpg'));
total_images = numel(filenames);
for i=1 : total_images
   full_name= fullfile(image_folder, filenames(i).name);
   Img = imread(full_name);

   % Color-Based Segmentation Using K-Means Clustering
   cform = makecform('srgb2lab');
   lab = applycform(Img,cform);
   ab = double(lab(:,:,2:3));
   nrows = size(ab,1);
   ncols = size(ab,2);
   ab = reshape(ab,nrows*ncols,2);

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

   pixel_labels = reshape(cluster_idx,nrows,ncols);
   RGB = label2rgb(pixel_labels);

   segmented_images = cell(1,3);
   rgb_label = repmat(pixel_labels,[1 1 3]);

   for k = 1:nColors
      color = Img;
      color(rgb_label ~= k) = 0;
      segmented_images{k} = color;
   end
   % daun segmentation
   area_cluster1 = sum(find(pixel_labels==1));
```

```
    area_cluster2 = sum(find(pixel_labels==2));

    [~,cluster_daun] = min([area_cluster1,area_cluster2]);
    daun_bw = (pixel_labels==cluster_daun);
    daun_bw = imfill(daun_bw,'holes');
    daun_bw = bwareaopen(daun_bw,1000);

    daun = Img;
    R = daun(:,:,1);
    G = daun(:,:,2);
    B = daun(:,:,3);
    R(~daun_bw) = 255;
    G(~daun_bw) = 255;
    B(~daun_bw) = 255;
    daun_rgb = cat(3,R,G,B);

    % daun croping
    bw = im2bw(daun_rgb,.9);
    bw = imcomplement(bw);
    bw = imfill(bw,'holes');

    blobMeasurements = regionprops(bw, 'all');
    %numberOfBlobs = size(blobMeasurements, 1);
    % Loop through all blobs.
    % Find the bounding box of each blob.
    thisBlobsBoundingBox = blobMeasurements.BoundingBox;  % Get
list of pixels in current blob.

    % Extract out this coin into it's own image.
    subImage = imcrop(daun_rgb, thisBlobsBoundingBox);

    imwrite(subImage,strcat(image_target,filenames(i).name));
    disp(fprintf('Telah Selesai : %s', filenames(i).name));
end
```

coding ekstraksi GLCM

```
clear all;
```

```
close all;
clc;

fiturTrain = []; % array menyimpan nilai fitur Train sementara
folderTrain = ['D:\img\dataskripsi\datauji']; %folder data latih

%dirFolder = dir((folderTrain));

%baca gambar
%img = imread('anggur2.jpg');
%figure, imshow(img);

for i = 1:length(dirFolder)
   F_name = dirFolder(i).name;
   dirFile = dir([folderTrain, '\',F_name,'\*.jpg']);
   hm = length(dirFile);
   for j = 1:hm % looping get file inside folder
      data(j).file_name = dirFile(j).name;
      data(j).file_location = [folderTrain, '\',F_name,'\',data(j).file_name];
      source = imread(data(j).file_location); % Read All Image PNG

      % Parameter GLCM Data Train
      layer = rgb2gray(source);
      layerBW = im2bw(layer,graythresh(layer))
      %layer = source(:,:,1)  %layer warna RGB (1 = RED, 2 = GREEN,
3 = BLUE)
      %layerWarna = edge(layer, 'canny'); % edge detection ('canny',
'sobel', 'prewitt')
      %source2 = imresize(layer, [100 100]); % resize (128 to 100)
      glcm = graycomatrix(layer, 'Offset', [0 1]);
      % Jarak Sudut GLCM (D = Jarak tiap Pixel yang dicari)
      % 0'     [0 D]
      % 45'    [-D D]
      % 90'    [-D 0]
      % 135'   [-D -D]
```

```
    [haralick] = haralickTextureFeatures2(glcm); % ekstraksi fitur
GLCM Haralick (14 fitur)
    group = cell(200, 1); % TOTAL 200 data
    % label gestur angka pada tangan ==> group(Xawal, Xakhir) =
{'X'};
    group(1:50) = {'A'};  % Label Citra busuk hitam
    group(51:100) = {'B'}; % Label Citra campak hitam
    group(101:150) = {'C'}; % Label Citra hawar daun
    group(151:200) = {'D'}; % Label Citra sehat

    % Klasifikasi Data Train (Proses pengenalan ke komputer)
    if(j>= 1 && j <= 50)
       data(j).class = 'A'; % Label Citra busuk hitam
    elseif(j>=51 && j <= 100)
       data(j).class = 'B'; % Label Citra campak hitam
    elseif(j>=101 && j <= 150)
       data(j).class = 'C'; % Label Citra hawar daun
    elseif(j>=151 && j <= 200)
       data(j).class = 'D'; % Label Citra sehat
    end
    data(j).class = group;
    data(j).ciri = haralick; % Save Ekstraksi ciri GLCM Haralick
    fiturTrain = [fiturTrain; data(j).ciri]; % Save Hasil Ekstraksi Ciri
tiap File Gambar
  end
end

x = struct2table(data); % export data menjadi Tabel
writetable(x, 'EkstraksiTeksturUji.xlsx'); % Export ke Excel
save dbFitur.mat group fiturTrain; % Save Nilai Fitur Train ke dalam
dbFitur.mat

disp('Ekstraksi Berhasil')
```

coding function GLCM

```
function [x] = haralickTextureFeatures2(coOcMat, xFeatures)
%Calculates all Haralick Features.
```

```
%
% Function call:
%   [x] = haralickTextureFeatures(coOcMat) calculates all 14 Haralick
%   Features
%   [x] = haralickTextureFeatures(coOcMat, xFeatures) calculates the
%   Haralick Features specified by xFeatures, the rest will be return as 0.
%   Use this for better legacy
% if you do not need all Haralick Features.
%
%
% Source:          http://haralick.org/journals/TexturalFeatures.pdf
%
%
% input:
%   'coOcMat'       Co-Occurence-Matrix,  which must be a [nxm]
matrix,
%                   see matlab documentation glcm
%   'xFeatures'     (optional) - Feature(s), which should be calculated
%
% output:
%   'x' - [vector with the following feature(s):
%             x(1)  Angular Second Moment (Energy) [checked]
%             x(2)  Contrast [checked]
%             x(3)  Correlation [checked]
%             x(4)  Variance [checked]
%             x(5)  Inverse Difference Moment (Homogeneity) [checked]
%             x(6)  Sum Average [checked]
%             x(7)  Sum Variance [approxemitly (cut out zeros)]
%             x(8)  Sum Entropy [checked]
%             x(9)  Entropy [cut out zeros]
%             x(10) Difference Variance [approxemitly]
%             x(11) Difference Entropy [checked]
%             x(12) Information Measure of Correlation I [checked]
%             x(13) Information Measure of Correlation II [approxemitly]
%             x(14) Maximal Correlation Coefficient [no reference]
%
%
%
%   Example
```

```
%   ---------
%   %Load an image of Matlab
%   I = imread('circuit.tif');
%
%   %Get the co occurence matrix (in Matlab called GLCM: Gray Level
Co
%   %Occurence Matrix)
%   glcm = graycomatrix(I, 'offset', [0 1], 'Symmetric', true);
%
%   %calculate feature 4 (Variance), 5 (Inverse Difference Moment) and
6
%   %(Sum Average)
%   xFeatures = 4:6;
%   x = haralickTextureFeatures(glcm, 4:6);
%
%   %Get only the features you want
%   x = x( xFeatures )
%
%
% Notes:       If x14 Maximal Correlation Coefficient is complex then
the
%              magnitude of MCC will be calculate.
%              See the haralick paper to understand the code.
%
% Info:        ver 1.1
%              - coOcMat will be checked if it is 2-dimensional
%              - Example code added
%              - more documentation
%              - fixed if-end polling (thanks to Ihsan Yassin)
%
% Author:      Rune Monzel, runemonzel(at)gmail.com
%
% See also graycomatrix, graycoprops.


% check input
if nargin == 1
   xFeatures = 1 : 14;
end
```

```
% check coOcMat for dimensions:
if ~(ismatrix(coOcMat))
    error(['\coOcMatInput must be a two dimensional matrix, '...
        'dimensional was %s.',ndims(coOcMat)']);
end

% initialize x
x = zeros(14,1);

% normalize glcm
coOcMat = coOcMat./sum(coOcMat(:));

%% Some pre-calculation:
% columns and rows
if sum(xFeatures == 2) == 1 | ... % Contrast
        sum(xFeatures == 3) == 1 | ... % Correlation
        sum(xFeatures == 4) == 1 | ... % Variance
        sum(xFeatures == 5) == 1 | ... % Inverse Difference Moment
        sum(xFeatures == 6) == 1 | ... % Sum Average
        sum(xFeatures == 7) == 1 | ... % Sum Variance
        sum(xFeatures == 8) == 1 | ... % Sum Entropy
        sum(xFeatures == 10) == 1 | ...% Difference Variance
        sum(xFeatures == 11) == 1 | ...% Difference Entropy
        sum(xFeatures == 14) == 1 % Maximal Correlation Coefficient
    sizecoOcMat = size(coOcMat);
    [col,row] = meshgrid(1:sizecoOcMat(1),1:sizecoOcMat(2));
end

% average and standarddeviation
if sum(xFeatures == 3) == 1 | ... % correlation
        sum(xFeatures == 10) == 1 % difference variance



    rowMean =  sum( row(:).*coOcMat(:) );
    colMean = sum( col(:).*coOcMat(:) );
    rowStd = sqrt( sum( (row(:)-rowMean).^2 .* coOcMat(:) ) );
    colStd = sqrt( sum( (col(:)-colMean).^2 .* coOcMat(:) ) );
```

```
end

% sum of rows p_y(i) und sum of columns p_x(j)
if sum(xFeatures == 12) == 1 |...% Information Measures of Correlation
I
      sum(xFeatures == 13) == 1|... % Information Measures of
Correlation II
      sum(xFeatures == 14) == 1 % Maximal Correlation Coefficient

   rowCoOcMat = sum(coOcMat,2); %sum of rows p_y(i)
   colCoOcMat = sum(coOcMat); %sum of columns p_x(i)
end

% p_x+y
if sum(xFeatures == 6)==1 |... % Sum Average
      sum(xFeatures == 7)==1 |... % Sum Variance
      sum(xFeatures == 8)==1 % Sum Entropy

   start = -(sizecoOcMat(1) -1);
   stop = sizecoOcMat(1) -1;

   % Rotate Matrix 90°
   coOcMat90 = rot90(coOcMat);

   % Initilisiere p_x+y
   p_XplusY = zeros((2*sizecoOcMat(1))-1,1);

   k = 1;
   for index = start : stop
      p_XplusY(k) = sum( diag(coOcMat90,index) );
      k = k + 1;
   end
end

% Initialize  p_x-y
if sum(xFeatures == 10)==1 |... % Difference Variance
      sum(xFeatures == 11)==1 % Difference Entropy
```

```
    start = 1;
    stop = sizecoOcMat(1)-1;

    % Initialize p_XminusY
    p_XminusY = zeros(sizecoOcMat(1),1);
    p_XminusY(1) = sum (diag(coOcMat,0) );

    k = 2;
    for index = start : stop
       p_XminusY(k) = sum( [diag(coOcMat,index);
          diag(coOcMat,-index)] );
       k = k + 1;
    end
end


%% Haralick Feature Calculations
for f = xFeatures
    switch f
       case 1 % Energy (Angular Second Moment)
          x(1) = sum( coOcMat(:).^2 );

       case 2  % Contrast
          matrix = ( abs(row - col).^2 ) .* coOcMat;
          x(2) = sum( matrix(:) );

       case 3  % Correlation
          zaehler = sum ((row(:) - rowMean) .*...
             (col(:) - colMean) .*  coOcMat(:));
          denominator = rowStd * colStd;
          x(3) = zaehler/denominator;

       case 4 % Inverse Difference Moment
          x(4) = sum( coOcMat(:) ./ ( 1+ (row(:)-col(:)).^2 ) );

       case 5 % Entropy
```

```
        x(5) = - sum( coOcMat(coOcMat~=0) .*...
            log2(coOcMat(coOcMat~=0)) );

    end
end

% PELUANG FITUR (3 KOMBINASI = 1680 macam)
x = [x(1) x(2) x(3) x(4) x(5)];
```

coding neural network latih

```
% Solve a Pattern Recognition Problem with a Neural Network
% Script generated by Neural Pattern Recognition app
% Created 30-Jul-2021 22:13:37
%
% This script assumes these variables are defined:
%
%   inputTesting - input data.
%   outputTesting - target data.

x = inputTrain';
t = outputTrain';

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm';  % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize, trainFcn);
save('net.mat','net');
% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
```

```
net.input.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivision
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'crossentropy';  % Cross-Entropy

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotconfusion', 'plotroc'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)
```

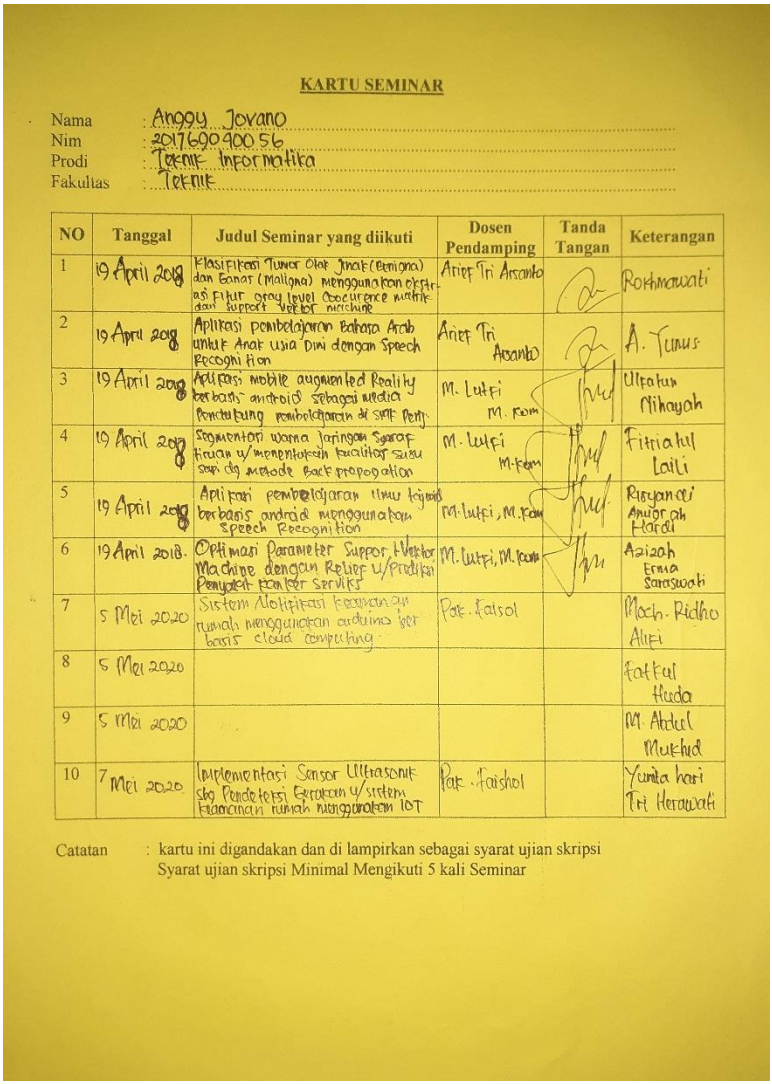
```
% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)

% Deployment
% Change the (false) values to (true) to enable the following code
blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and
Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network
code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```

coding neural network uji

```
load net.mat
x = inputTesting';
t = outputTesting';

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)

% Deployment
% Change the (false) values to (true) to enable the following code
blocks.
% See the help for each generation function for more information.
```

```
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and
Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network
code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```

Lampiran 4. Kartu Seminar



KARTU SEMINAR

Nama : Anggy Jovano
Nim : 2017690900 56
Prodi : Teknik Informatika
Fakultas : Teknik

NO	Tanggal	Judul Seminar yang diikuti	Dosen Pendamping	Tanda Tangan	Keterangan
1	19 April 2018	Klasifikasi Tumor Otak Jinak (Benigna) dan Ganas (Maligna) menggunakan ekstraksi Fitur gray level Coocurence matrik dan support vektor machine	Ariep Tri Arsanto		Rokhmawati
2	19 April 2018	Aplikasi pembelajaran Bahasa Arab untuk Anak usia Dini dengan Speech Recognition	Ariep Tri Arsanto		A. Tunus
3	19 April 2018	Aplikasi mobile augmented Reality berbasis android sebagai media pendukung pembelajaran di SMK Perty.	M. Lutfi M. kom		Ulfatun Nihayah
4	19 April 2018	Segmentasi warna jaringan Syaraf tiruan u/ menentukan kualitas susu sapi dg metode Back propogation	M. Lutfi M.kom		Fitriatul Laili
5	19 April 2018	Aplikasi pembelajaran ilmu tajwid berbasis android menggunakan Speech Recognition	M. Lutfi, M.kom		Rizyanci Anugrah Hardi
6	19 April 2018	Optimasi Parameter Support Vektor Machine dengan Relief u/ Prediksi Penyakit Kanker Serviks	M. Lutfi, M.kom		Azizah Erna Saraswati
7	5 Mei 2020	Sistem Notifikasi keamanan rumah menggunakan arduino berbasis cloud computing	Pak. Faisol		Moch. Ridho Alifi
8	5 Mei 2020				Fatful Huda
9	5 Mei 2020				M. Abdul Mukhid
10	7 Mei 2020	Implementasi Sensor Ultrasonik sbg Pendeteksi Ketinggian u/ sistem keamanan rumah menggunakan IOT	Pak. Faishol		Yunita hari Tri Herawati

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

Lampiran 5. Surat Keterangan Bebas Plagiasi

UNIVERSITAS YUDHARTA PASURUAN
FAKULTAS TEKNIK
Kantor Pusat :
Jl. Yudharta No. 07 (Pesantren Ngalah) Sengonagung Purwosari Pasuruan Telp./ Fax. 0343-611186
e-mail: fakultasteknik@yudharta.ac.id

SURAT KETERANGAN BEBAS PLAGIASI
Nomor : 0320/S9/FT.UYP/II/08/2021

Yang bertanda tangan dibawah ini:
Nama : Misbach Munir, ST., MT
NIP.Y : 0690201015
Jabatan : Dekan Fakultas Teknik

Dengan ini menerangkan bahwa skripsi atas nama mahasiswa :
Nama : Anggy Jovano
NIM : 201769040056
Prodi : Teknik Informatika
Judul Skripsi : KLASIFIKASI JENIS PENYAKIT DAUN ANGGUR MENGGUNAKAN METODE EKSTRAKSI FITUR GLCM DAN NEURAL NETWORK
Hasil Plagiasi : 13%

Demikian surat keterangan ini kami buat untuk digunakan sebagaimana mestinya.

Pasuruan, 23 Agustus 2021
Dekan Fakultas Teknik

Misbach Munir, ST., MT.
NIP.Y. 0690201015