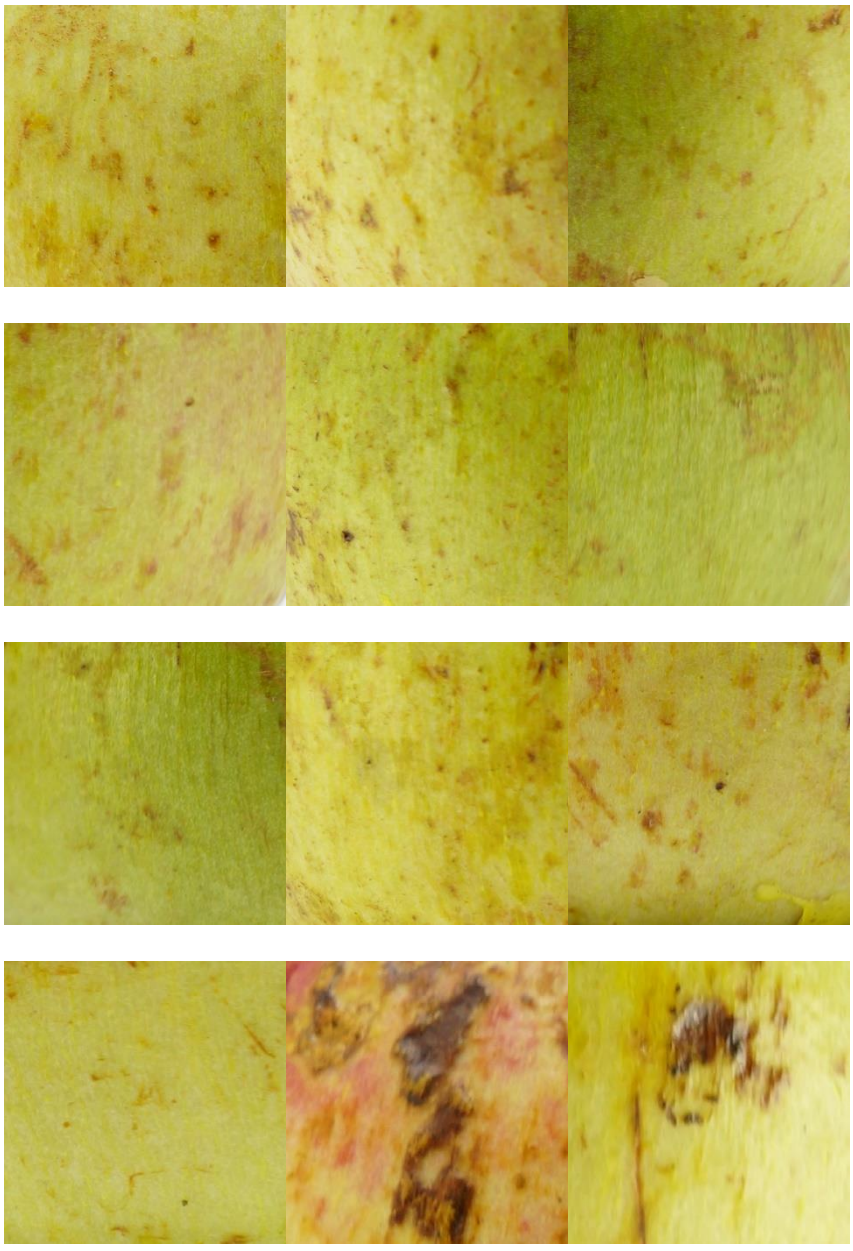
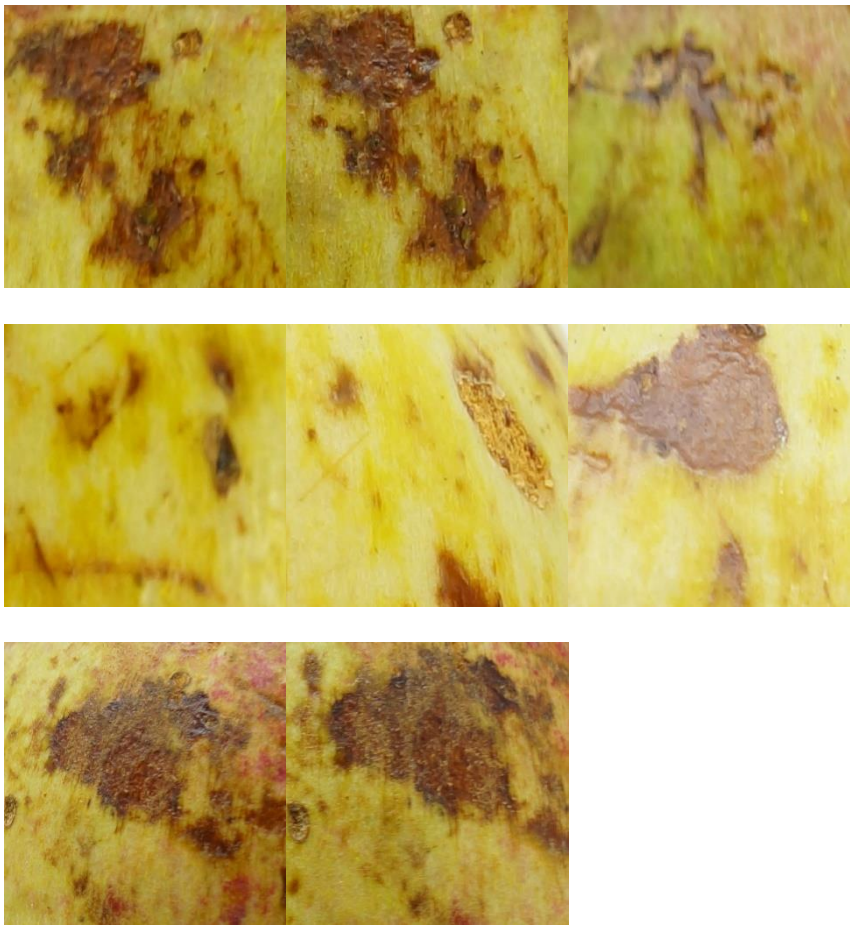
LAMPIRAN

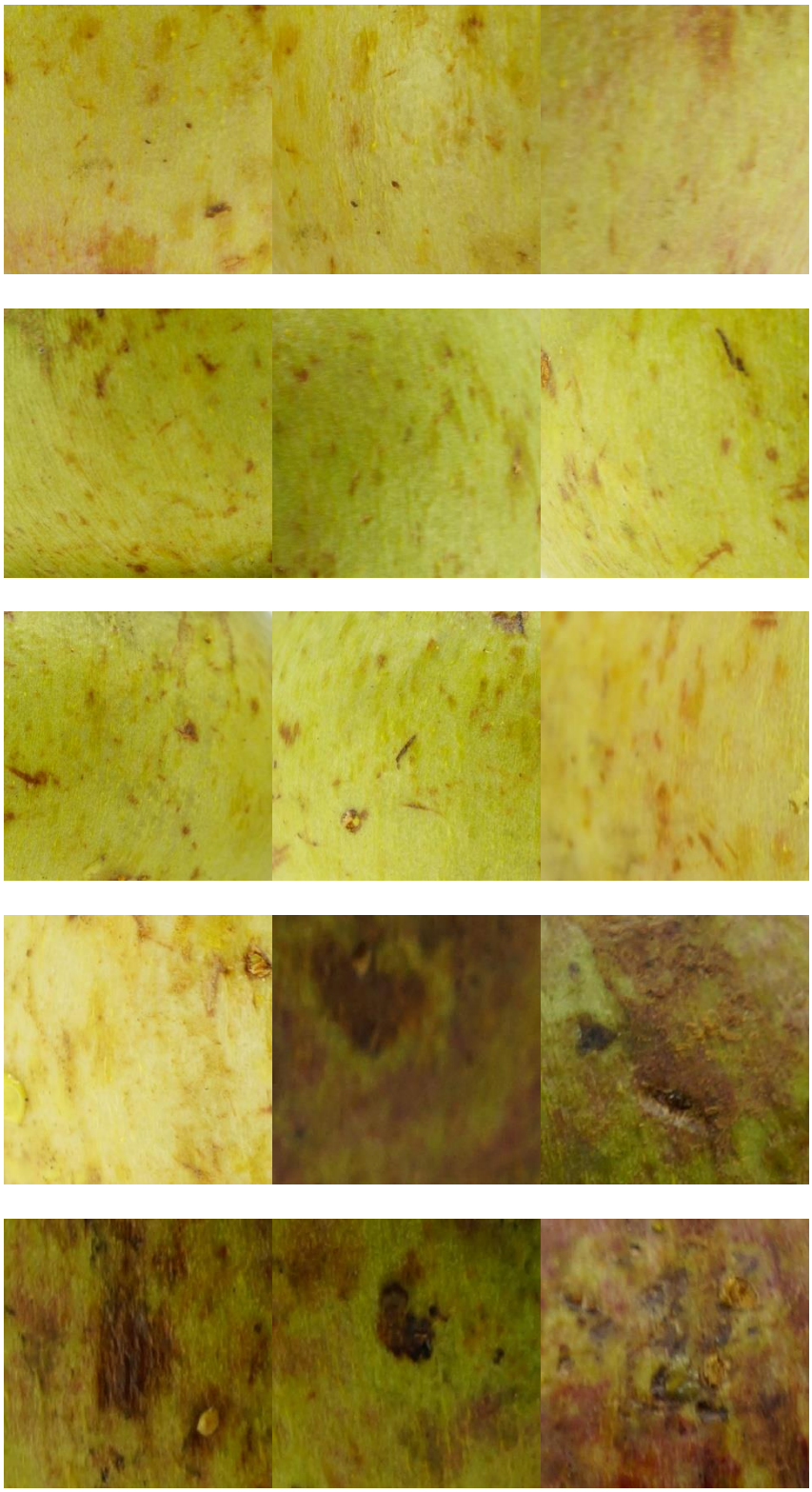
Lampiran 1 – Citra

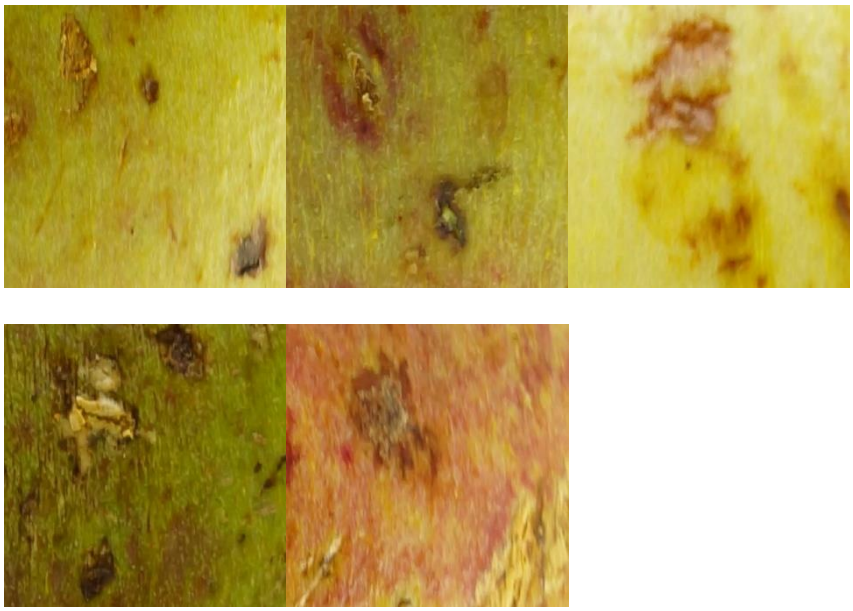
a. Fold 1



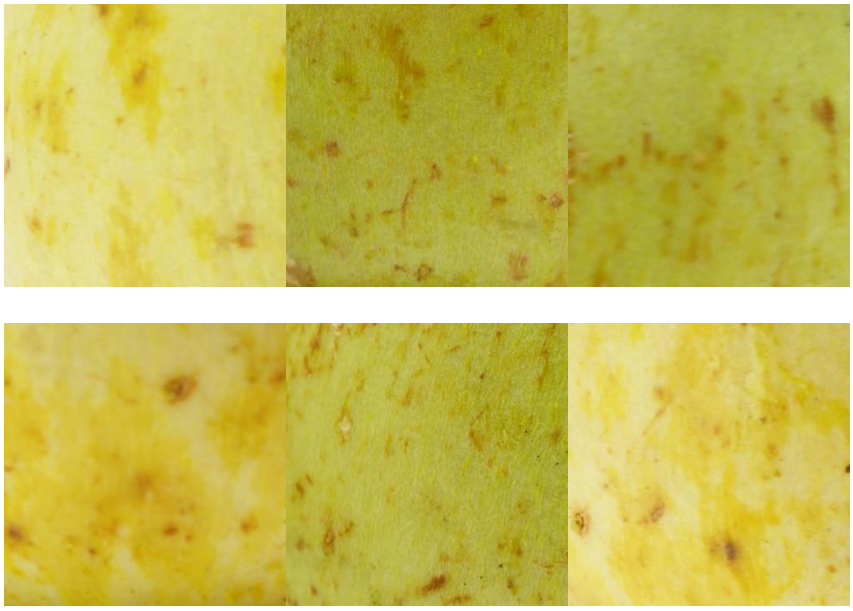


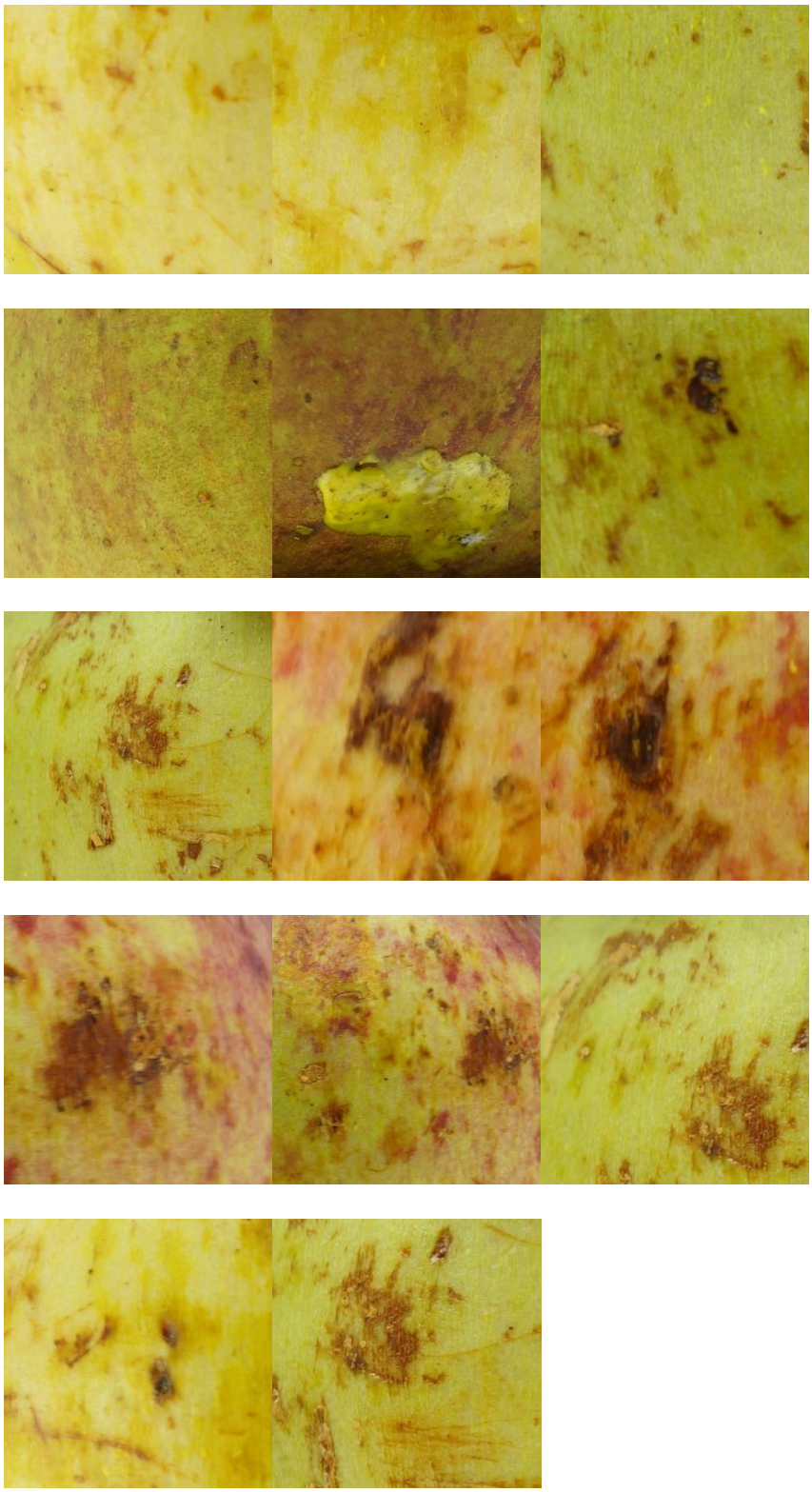
b. Fold 2



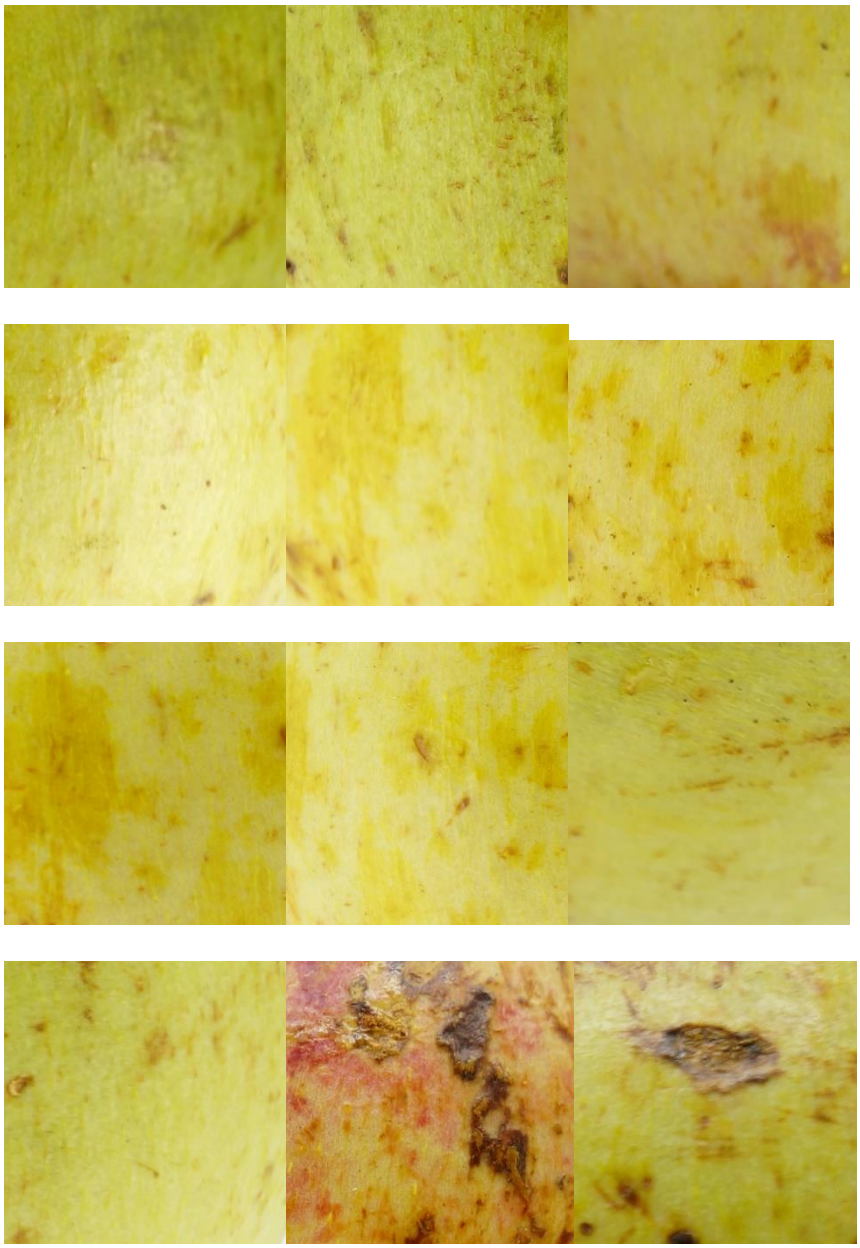


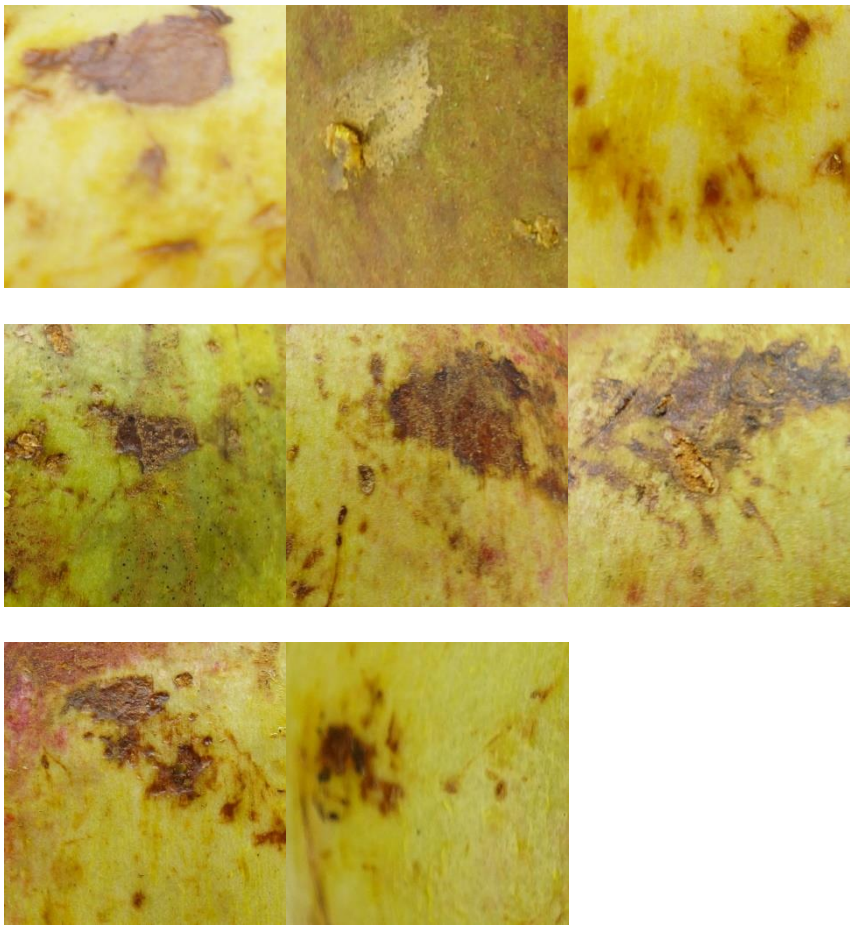
c. Fold 3





d. Fold 4





Lampiran 2 – Fungsi

a. get_images_fold

```
function images = get_images_fold(fold, fileName)
    currfolder=pwd;

    if length(fold)==1
        dataFold{1}=fold{1};
    elseif length(fold)==2
        dataFold{1}=fold{1};
        dataFold{2}=fold{2};
    elseif length(fold)==3
        dataFold{1}=fold{1};
        dataFold{2}=fold{2};
        dataFold{3}=fold{3};
    elseif length(fold)==4
```

```
        dataFold{1}=fold{1};
        dataFold{2}=fold{2};
        dataFold{3}=fold{3};
        dataFold{4}=fold{4};
    end
    count=1;
    for i=1:length(dataFold)
        imagefiles = dir(fullfile(strcat(currfolder, '\',
dataFold{i}), fileName)); %membaca seluruh file
        nfiles = length(imagefiles);    % menghitung jumlah file
        for ii=1:nfiles %looping memasukan data gambar variable
array
           namaFile{count} = imagefiles(ii). name;%membaca nama file
           img{count} = imread(fullfile(imagefiles(ii). folder,
namaFile{count}));%membaca file gambar
           count=count+1;
        end


    end

    images. img =img;
    images. nama =namaFile;

end
```

b. get_CT

```
function CT = get_CT(imgsize, imagefiles)
    nfiles = length(imagefiles);    % menghitung jumlah file
    for ii=1:nfiles
        img = imresize(imagefiles{ii}, imgsize);%ubah ukuran gambar
        imageGray=rgb2gray(img);%konvert setiap file gambar di array
ke grayscale
        tic; curva{ii} = fdct_usfft(imageGray, 0); toc;
        coef{ii} = fdct_usfft_dispcoef(curva{ii});
        tic; spat{ii} = afdct_usfft(curva{ii}, 0); toc;
        freq{ii} = ifftshift(fft2(fftshift(spat{ii})));
    end
    CT. curva=curva;
    CT. coef = coef;
    CT. frequency = freq;
    CT. spatial = spat;
end
```

c. get_FE

```
function FE =get_FE(curva, level, methode)
    nfiles = length(curva);
```

```
    for ii=1:nfiles
        if strcmp(methode, 'curva') || strcmp(methode, 'param')
            if level==1 || level==length(curva{1, 1})
                get_Mean(ii) = mean(mean(curva{1, ii}{1, level}{1,
1}));
                get_Energy(ii) =calEnergy(curva{1, ii}{1, level}{1,
1});
                get_str_dv(ii) = mean(std(curva{1, ii}{1, level}{1,
1}));
                get_entropy(ii) = EntropyManual(curva{1, ii}{1,
level}{1, 1});

            else
                count1 = length(curva{1, 1}{1, level});
                for iii=1:count1
                    g_Mean(iii) = mean(mean(curva{1, ii}{1,
level}{1, iii}));
                    g_Energy(iii) =calEnergy(curva{1, ii}{1,
level}{1, iii});
                    g_str_dv(iii) = mean(std(curva{1, ii}{1,
level}{1, iii}));
                    g_entropy(iii) = EntropyManual(curva{1, ii}{1,
level}{1, iii});
                end
                get_Mean(ii) = mean(g_Mean);
                get_Energy(ii) = mean(g_Energy(:));
                get_str_dv(ii) = mean(g_str_dv);
                get_entropy(ii) = mean(g_entropy);
            end
        else
            get_Mean(ii) = mean(mean(curva{1, ii}));
            get_Energy(ii) =calEnergy(curva{1, ii});
            get_str_dv(ii) = mean(std(curva{1, ii}));
            get_entropy(ii) = EntropyManual(curva{1, ii});
        end
    end
    FE. mean =get_Mean;
    FE. energy =get_Energy;
    FE. entropy = get_entropy;
    FE. std =get_str_dv;

 %----------------------------



end
```

d. train_LDA

```
function LDA_t= train_LDA(d, f, fe)
```

```
    %-----------------------------------
    Datamean=d. mean;
    Dataenergy=d. energy;
    Dataentropy=d. entropy;
    Datastd=d. std;
    jumlahFE =length(fe);
%-----------------------------------
    if jumlahFE == 2
            if strcmp(fe{1}, 'mean')
                dataX = [Datamean, f. mean];
            elseif strcmp(fe{1}, 'energy')
                dataX = [Dataenergy, f. energy];
            elseif strcmp(fe{1}, 'entropy')
                dataX = [Dataentropy, f. entropy];
            elseif strcmp(fe{1}, 'std')
                dataX = [Datastd, f. std];
            end
            %-----------------------------------
            if strcmp(fe{2}, 'mean')
                dataY = [Datamean, f. mean];
            elseif strcmp(fe{2}, 'energy')
                dataY = [Dataenergy, f. energy];
            elseif strcmp(fe{2}, 'entropy')
                dataY = [Dataentropy, f. entropy];
            elseif strcmp(fe{2}, 'std')
                dataY = [Datastd, f. std];
            end
            dataY=dataY(:);
            dataX=dataX(:);
            xdata=[dataX, dataY];
%-------------------------------------------
    elseif jumlahFE == 3
            if strcmp(fe{1}, 'mean')
                dataX = [Datamean, f. mean];
            elseif strcmp(fe{1}, 'energy')
                dataX = [Dataenergy, f. energy];
            elseif strcmp(fe{1}, 'entropy')
                dataX = [Dataentropy, f. entropy];
            elseif strcmp(fe{1}, 'std')
                dataX = [Datastd, f. std];
            end
            %-----------------------------------
            if strcmp(fe{2}, 'mean')
                dataY = [Datamean, f. mean];
            elseif strcmp(fe{2}, 'energy')
                dataY = [Dataenergy, f. energy];
            elseif strcmp(fe{2}, 'entropy')
                dataY = [Dataentropy, f. entropy];
            elseif strcmp(fe{2}, 'std')
                dataY = [Datastd, f. std];
            end
            if strcmp(fe{3}, 'mean')
                dataT = [Datamean, f. mean];
```

```
            elseif strcmp(fe{3}, 'energy')
                dataT = [Dataenergy, f. energy];
            elseif strcmp(fe{3}, 'entropy')
                dataT = [Dataentropy, f. entropy];
            elseif strcmp(fe{3}, 'std')
                dataT = [Datastd, f. std];
            end
            dataY=dataY(:);
            dataX=dataX(:);
            dataT=dataT(:);
            xdata=[dataX, dataY, dataT];
%-----------------------------------
     elseif jumlahFE == 4
         dataMean = [Datamean, f. mean];
         dataEnergy = [Dataenergy, f. energy];
         dataEntropy = [Dataentropy, f. entropy];
         DataStd = [Datastd, f. std];
         dataMean=dataMean(:);
         dataEnergy=dataEnergy(:);
         dataEntropy=dataEntropy(:);
         DataStd=DataStd(:);
         xdata=[dataMean, dataEnergy, dataEntropy, DataStd];
     end

     %-----------------------------------
     for i=1:(length(f. std)+length(d. std))
         if i<=length(f. mean)
            group{i}='defect';
         else
            group{i}='fine';
         end
     end
      group=group(:);

      LDA_t =  fitcdiscr(xdata, group, 'DiscrimType', 'linear',
'HyperparameterOptimizationOptions', struct('ShowPlot', true));
```

e. LDA_accuracy

```
function LDA= LDA_accuracy(LDA_T, images, imgsize, level, methode,
fe)
imageTest= get_images(images);
CT = get_CT(imgsize, imageTest. img);
FE = get_FE(CT. spatial, level, methode);
% pca=get_PCA(imageTest, 'citra');
% FE=fe_PCA(pca);
jumlahFE=length(fe);
          %-----------------------------------
     if jumlahFE == 2
             if strcmp(fe{1}, 'mean')
                 dataX = FE. mean;
```

```
            elseif strcmp(fe{1}, 'energy')
                dataX = FE. energy;
            elseif strcmp(fe{1}, 'entropy')
                dataX = FE. entropy;
            elseif strcmp(fe{1}, 'std')
                dataX = FE. std;
            end
            %-----------------------------------
            if strcmp(fe{2}, 'mean')
                dataY = FE. mean;
            elseif strcmp(fe{2}, 'energy')
                dataY = FE. energy;
            elseif strcmp(fe{2}, 'entropy')
                dataY = FE. entropy;
            elseif strcmp(fe{2}, 'std')
                dataY = FE. std;
            end
            dataY=dataY(:);
            dataX=dataX(:);
            xdata=[dataX, dataY];
%-------------------------------------------
    elseif jumlahFE == 3
            if strcmp(fe{1}, 'mean')
                dataX = FE. mean;
            elseif strcmp(fe{1}, 'energy')
                dataX = FE. energy;
            elseif strcmp(fe{1}, 'entropy')
                dataX = FE. entropy;
            elseif strcmp(fe{1}, 'std')
                dataX = FE. std;
            end
            %-----------------------------------
            if strcmp(fe{2}, 'mean')
                dataY = FE. mean;
            elseif strcmp(fe{2}, 'energy')
                dataY = FE. energy;
            elseif strcmp(fe{2}, 'entropy')
                dataY = FE. entropy;
            elseif strcmp(fe{2}, 'std')
                dataY = FE. std;
            end
             %-----------------------------------
            if strcmp(fe{3}, 'mean')
                dataT = FE. mean;
            elseif strcmp(fe{3}, 'energy')
                dataT = FE. energy;
            elseif strcmp(fe{3}, 'entropy')
                dataT = FE. entropy;
            elseif strcmp(fe{3}, 'std')
                dataT = FE. std;
            end
            dataY=dataY(:);
            dataX=dataX(:);
```

```
            dataT=dataT(:);
            xdata=[dataX, dataY, dataT];
%-----------------------------------
     elseif jumlahFE == 4
         dataMean = FE. mean;
         dataEnergy = FE. energy;
         dataEntropy = FE. entropy;
         DataStd = FE. std;
         dataMean=dataMean(:);
         dataEnergy=dataEnergy(:);
         dataEntropy=dataEntropy(:);
         DataStd=DataStd(:);
         xdata=[dataMean, dataEnergy, dataEntropy, DataStd];
     end

    LDA_c = predict(LDA_T, xdata);clc();
    benar=0;salah=0;defect=0;freedefect=0;

    for i=1:length(LDA_c)
        if strcmp(LDA_c{i}(1:3), imageTest. nama{i}(1:3))
            benar=benar+1;
        else
            salah=salah+1;
        end
        if(strcmp(LDA_c{i}, 'defect'))
            defect=defect+1;
        else
            freedefect=freedefect+1;
        end
    end
    accuracy=benar*100/length(LDA_c);
    clc();
    disp(strcat('------------------Detail:---------------'));
    for o=1:length(LDA_c)
        if strcmp(imageTest. nama{o}(1), LDA_c{o}(1))
        disp(strcat(string(o), '. ', imageTest. nama{o}, ' : ',
string(LDA_c{o})));
        else
         disp(strcat(string(o), '. ', '----->>', imageTest. nama{o},
' : ', string(LDA_c{o}), '<<-----'));
        end
    end
        disp(strcat('-------------Hasilnya-------------'));
    disp(strcat('Accuracy:', string(accuracy), '%'));
    disp(strcat('Total Sample:', string(length(LDA_c))));
    disp(strcat('defect:', string(defect)));
    disp(strcat('free defect:', string(freedefect)));
    disp(strcat('Salah klasifikasi:', string(salah)));
    disp(strcat('----------------------------------'));
    LDA. LDA_c=LDA_c;
    LDA. d=defect;
    LDA. f=freedefect;
```

```
end
```

f. start_cv

```
imagesDefect4 = get_images_fold({'fold1', 'fold2', 'fold3'},
'defect*. jpg');
imagesFine4 = get_images_fold({'fold1', 'fold2', 'fold3'}, 'fine*.
jpg');
%-------------
imagesDefect1 = get_images_fold({'fold2', 'fold3', 'fold4'},
'defect*. jpg');
imagesFine1 = get_images_fold({'fold2', 'fold3', 'fold4'}, 'fine*.
jpg');
%-------------
imagesDefect2 = get_images_fold({'fold1', 'fold3', 'fold4'},
'defect*. jpg');
imagesFine2 = get_images_fold({'fold1', 'fold3', 'fold4'}, 'fine*.
jpg');
%-------------
imagesDefect3 = get_images_fold({'fold1', 'fold2', 'fold4'},
'defect*. jpg');
imagesFine3 = get_images_fold({'fold1', 'fold2', 'fold4'}, 'fine*.
jpg');
%-------------------
CT_f_fold4 = get_CT([512 512], imagesFine_fold4. img);
CT_d_fold4 = get_CT([512 512], imagesDefect_fold4. img);
%---------------------------
CT_f_fold1 = get_CT([512 512], imagesFine_fold1. img);
CT_d_fold1 = get_CT([512 512], imagesDefect_fold1. img);
%---------------------------
CT_f_fold2 = get_CT([512 512], imagesFine_fold2. img);
CT_d_fold2 = get_CT([512 512], imagesDefect_fold2. img);
%---------------------------
CT_f_fold3 = get_CT([512 512], imagesFine_fold3. img);
CT_d_fold3 = get_CT([512 512], imagesDefect_fold3. img);
%---------------------------
%-------------------------
FE_f4_lev1 = get_FE(CT_f_fold4. curva, 1, 'curva');
FE_d4_lev1 = get_FE(CT_d_fold4. curva, 1, 'curva');
%---------------------------
FE_f1_lev1 = get_FE(CT_f_fold1. curva, 1, 'curva');
FE_d1_lev1 = get_FE(CT_d_fold1. curva, 1, 'curva');
%-------------------------
FE_f2_lev1 = get_FE(CT_f_fold2. curva, 1, 'curva');
FE_d2_lev1 = get_FE(CT_d_fold2. curva, 1, 'curva');
%-------------------------
FE_f3_lev1 = get_FE(CT_f_fold3. curva, 1, 'curva');
FE_d3_lev1 = get_FE(CT_d_fold3. curva, 1, 'curva');
%-------------------------
 Trainning
```

```
%-------------------------
LDA_t1 = train_LDA(FE_d1_lev1, FE_f1_lev1, {'std', 'energy', 'mean',
'entropy'});
%-----------------
LDA_t2 = train_LDA(FE_d2_lev1, FE_f2_lev1, {'std', 'energy', 'mean',
'entropy'});
%-------------------
LDA_t3 = train_LDA(FE_d3_lev1, FE_f3_lev1, {'std', 'energy', 'mean',
'entropy'});
%---------------------
LDA_t4 = train_LDA(FE_d4_lev1, FE_f4_lev1, {'std', 'energy', 'mean',
'entropy'});

%Testing
lda_c = LDA_accuracy(LDA_t1, '*. jpg', [512 512], 1, 'curva',
{'std', 'energy', 'mean', 'entropy'}, {'fold4'});
%---------------------
lda_c = LDA_accuracy(LDA_t2, '*. jpg', [512 512], 1, 'curva',
{'std', 'energy', 'mean', 'entropy'}, {'fold1'});
%-------------------
lda_c = LDA_accuracy(LDA_t3, '*. jpg', [512 512], 1, 'curva',
{'std', 'energy', 'mean', 'entropy'}, {'fold2'});
%---------------------
lda_c = LDA_accuracy(LDA_t4, '*. jpg', [512 512], 1, 'curva',
{'std', 'energy', 'mean', 'entropy'}, {'fold3'});
```

Skripsi Jaenudin

ORIGINALITY REPORT

10%	10%	0%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Brawijaya Student Paper	1%
2	eprints.ums.ac.id Internet Source	1%
3	pt.scribd.com Internet Source	<1%
4	es.scribd.com Internet Source	<1%
5	uad.portalgaruda.org Internet Source	<1%
6	www.karakuri-melynda.blogspot.com Internet Source	<1%
7	pemrogramanmatlab.wordpress.com Internet Source	<1%
8	repository.uinjkt.ac.id Internet Source	<1%
9	etheses.uin-malang.ac.id Internet Source	<1%

10	sir.stikom.edu Internet Source	<1%
11	id.scribd.com Internet Source	<1%
12	dokumen.tips Internet Source	<1%
13	repository.unair.ac.id Internet Source	<1%
14	eprints.uny.ac.id Internet Source	<1%
15	www.slideshare.net Internet Source	<1%
16	Submitted to Universitas Dian Nuswantoro Student Paper	<1%
17	eprints.uns.ac.id Internet Source	<1%
18	documents.mx Internet Source	<1%
19	eprints.dinus.ac.id Internet Source	<1%
20	eprints.undip.ac.id Internet Source	<1%
21	repository.usu.ac.id Internet Source	<1%

22	journal.umpo.ac.id Internet Source	<1%
23	perkebunan.litbang.deptan.go.id Internet Source	<1%
24	statistikian.blogspot.com Internet Source	<1%
25	Submitted to Universitas Muria Kudus Student Paper	<1%
26	digilib.uin-suka.ac.id Internet Source	<1%
27	Submitted to Universitas Pendidikan Indonesia Student Paper	<1%
28	tvschool.alazhar-cibubur.sch.id Internet Source	<1%
29	scholar.unand.ac.id Internet Source	<1%
30	koneksiartikel.blogspot.com Internet Source	<1%
31	repository.telkomuniversity.ac.id Internet Source	<1%
32	Submitted to Universitas Negeri Jakarta Student Paper	<1%
33	lppm.undip.ac.id	

	Internet Source	<1%
34	boldson.staff.gunadarma.ac.id Internet Source	<1%
35	id.m.wikipedia.org Internet Source	<1%
36	agribisnis.faperta.unsoed.ac.id Internet Source	<1%
37	dosen.tf.itb.ac.id Internet Source	<1%
38	openlibrary.telkomuniversity.ac.id Internet Source	<1%
39	catatanyure.blogspot.com Internet Source	<1%
40	blog.ub.ac.id Internet Source	<1%
41	www.docstoc.com Internet Source	<1%
42	www.mitrariset.com Internet Source	<1%
43	Submitted to Universitas Sebelas Maret Student Paper	<1%
44	siu.edu.in Internet Source	<1%

45	Submitted to Politeknik Negeri Bandung Student Paper	<1%
46	blognyaberlian.blogspot.com Internet Source	<1%
47	Chandra Kusuma Dewa. "Javanese vowels sound classification with convolutional neural network", 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA), 2016 Publication	<1%
48	balitbangda.kutaikartanegarakab.go.id Internet Source	<1%
49	Submitted to Padjadjaran University Student Paper	<1%
50	repository.ipb.ac.id Internet Source	<1%
51	statistikceria.blogspot.com Internet Source	<1%
52	tpa.fateta.unand.ac.id Internet Source	<1%
53	www.ahlinyadiabetes.com Internet Source	<1%
54	www.kajianpustaka.com Internet Source	<1%

55	www.trialanderor.tk Internet Source	<1%
56	bappeda.bengkuluutarakab.go.id Internet Source	<1%
57	www.vstore.si Internet Source	<1%
58	repository.uinsu.ac.id Internet Source	<1%
59	booksreadr.org Internet Source	<1%
60	tempointeraktif.com Internet Source	<1%
61	tumoutou.net Internet Source	<1%
62	docslide.us Internet Source	<1%
63	www.dakwahsunnah.com Internet Source	<1%
64	rodja.tv Internet Source	<1%
65	docplayer.info Internet Source	<1%
66	tanamanhutan.blogspot.com Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		