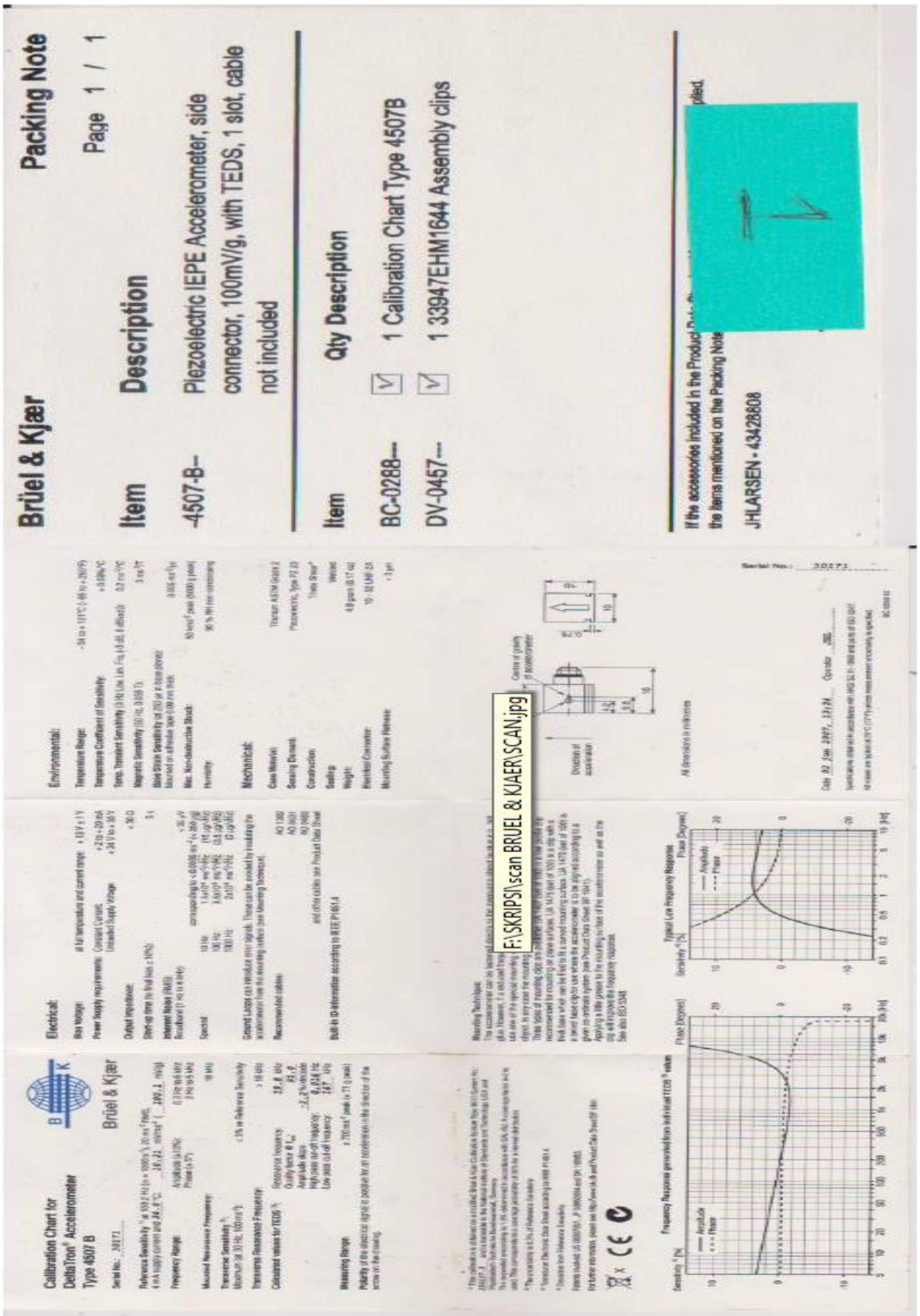
LAMPIRAN

Lampiran 1. Spesifikasi Sensor Accelerometer



Lampiran 2. Script MATLAB Data Akuisisi

```
%Script to run data acquisition using National Instrument NI 9234
%Created: Oct 2016, Berli Kamiel

clear all;
clc;
close all;

tic;
s = daq.createSession('ni');
s.DurationInSeconds = 10;
Dur = s.DurationInSeconds;
s.Rate = 17066;
s.addAnalogInputChannel('cDAQ1Mod1', 'ai0', 'Accelerometer');
%s.addAnalogInputChannel('cDAQ1Mod1', 'ai1', 'Accelerometer');
%s.addAnalogInputChannel('cDAQ1Mod1', 'ai2', 'Accelerometer');
%s.addAnalogInputChannel('cDAQ1Mod1', 'ai3', 'Voltage'); %
Tachometer
%s.addAnalogInputChannel('cDAQ1Mod2', 'ai0', 'Microphone');
%s.addAnalogInputChannel('cDAQ1Mod2', 'ai1', 'Microphone');

s.Channels(1).Sensitivity = 100.10E-3; %mV/g Type 4507B
serial:11165
%s.Channels(2).Sensitivity = 95.83E-3; %mV/g Type 4507B
serial:11026
%s.Channels(3).Sensitivity = 99.56E-3; %mV/g Type 4507B
serial:10984
%s.Channels(4).Sensitivity = 94.50E-3;
%s.Channels(5).Sensitivity = 9.40E-3;  %mV/Pa Model 130B40
serial:41741
%s.Channels(6).Sensitivity = 8.60E-3;  %mV/Pa Model 130B40
serial:41842

for i=1:50
data = s.startForeground();           % start recording vibration
data
data_ch1 = data(:,1);
%data_ch2 = data(:,2);
%data_ch3 = data(:,3);
%data_ch4 = data(:,4);
%data_ch5 = data(:,5);
%data_ch6 = data(:,6);
rootname = 'E:\Tugas_Akhir';         % drive tujuan dan nama file
extension = '.mat';                  % ekstension utk nama file
namafile = [rootname,'Level1_',num2str(i),extension];
data_all = [data_ch1];
eval(['save ', namafile ,' data_all']);
pause(2)
pesan = ['Acquiring and saving data at loop number: ',num2str(i)];
disp(pesan)
end

toc
```

Lampiran 3. Script MATLAB Plot Data Domain Waktu

```
% Plotting Time Domain
% Updated: July, 15th 2018, Muhammad Taufiq Akbar (20140130069)

clear; close all; clc;

%% Initialization
load('D:\Campus\KULIAH\Tugas Akhir\Data Tugas Akhir',...
    '\2. Data Pakai\1.Normal\Tugas_AkhirNormal1_209')
y1=data_all(:,1);

load('D:\Campus\KULIAH\Tugas Akhir\Data Tugas Akhir',...
    '\2. Data Pakai\2. Kavitasi Level
1\Tugas_AkhirKavitasiLevel1_209')
y2=data_all(:,1);

load('D:\Campus\KULIAH\Tugas Akhir\Data Tugas Akhir',...
    '\2. Data Pakai\3. Kavitasi Level
2\Tugas_AkhirKavitasiLevel2_209')
y3=data_all(:,1);

load('D:\Campus\KULIAH\Tugas Akhir\Data Tugas Akhir',...
    '\2. Data Pakai\4. Kavitasi Level
3\Tugas_AkhirKavitasiLevel3_209')
y4=data_all(:,1);

%% Plot Amplitudo of each Data
figure
subplot(9,1,1)
plot(y1(1:170660))
axis([0 9000 -3 3])
legend('Normal')

subplot(9,1,2)
plot(y2(1:170660),'b')
axis([0 9000 -3 3])
legend('Kavitasi1')

subplot(9,1,3)
plot(y3(1:170660),'g')
axis([0 9000 -3 3])
legend('Kavitasi2')

subplot(9,1,4)
plot(y4(1:170660),'b')
axis([0 9000 -3 3])
legend('Kavitasi3')

xlabel('Sampel')
```

Lampiran 4. Script MATLAB Esktraksi Parameter Statistik Domain Waktu

```
% Data Extraction Using Statistical Feature in Time Domain
% Created: Juli 15th 2018, Muhammad Taufiq Akbar (20140130069)

clear; close all; clc;

%% Initialization (1)
% Normal Datasets
signal_in=['D:\Campus\KULIAH\Tugas Akhir\Data Tugas Akhir',...
    '\2. Data Pakai\1.
Normal\Tugas_AkhirNormal1_',int2str(d),'.mat'];
load (signal_in);

for d=(1:500)                             % Number Of Iteration

% Data Extraction
a=data_all(:,1);

RMS(d)=rms(a);                               % RMS
SD(d)=std(a);                                % Standar Deviation
Peak_Value(d)=((max(abs(a))-min(abs(a)))/2); % Peak Value
Kurtosis(d)=kurtosis(a)-3;                   % Kurtosis
Variance(d)=var(a);                          % Variance
Crest_Factor(d)=peak2rms(a);                 % Crest Faktor
Mean(d) = mean(a) ;                          % Mean
Entropy (d) = entropy(a);                    % Entropy
Min(d)= min(a);                              % Minimum Value
SEM(d) = std(a)/sqrt(length(a));             % Standar Error

rootname = 'F:';
extension = '.mat';
namafile = [rootname,'Hasil_Ekstraksi_Data_Normal',extension];
save (namafile)

RMS=RMS';                             % Transpose RMS
SD=SD';                               % Transpose Standar Deviation
Peak_Value=Peak_Value';               % Transpose Peak Value
Kurtosis=Kurtosis';                   % Transpose Kurtosis
Variance=Variance';                   % Transpose Variance
Crest_Factor=Crest_Factor';           % Transpose Crest Factor
Mean=Mean';                           % Transpose Mean
Entropy = Entropy';                   % Transpose Entropy
Min = Min';                           % Transpose Minimum Value
SEM = SEM';                           % Transpose Standar Error
end



% Combine All Result Into One File

Data_Normal_Gabungan=zeros(500,10);            % Create New Cell

Data_Normal_Gabungan(:,1) = RMS;
```

```
Data_Normal_Gabungan(:,2) = SD;
Data_Normal_Gabungan(:,3) = Peak_Value;
Data_Normal_Gabungan(:,4) = Kurtosis;
Data_Normal_Gabungan(:,5) = Variance;
Data_Normal_Gabungan(:,6) = Crest_Factor;
Data_Normal_Gabungan(:,7) = Mean;
Data_Normal_Gabungan(:,8) = Entropy;
Data_Normal_Gabungan(:,9) = Min;
Data_Normal_Gabungan(:,10) = SEM;

Gabungan_N = zscore(Data_Normal_Gabungan);      % Standardize All

%% Initialization (2)
% Kavitasi Level 1 Datasets
signal_in=['D:\Campus\KULIAH\Tugas Akhir\Data Tugas Akhir\2. Data
Pakai',...
    '\2. Kavitasi Level
1\Tugas_AkhirKavitasiLevel1_',int2str(d),'.mat'];
load (signal_in);

for dd=(1:500)                                % Number Of Iteration

% Data Extraction
b=data_all(:,1);

RMS1(dd)=rms(b);                                 % RMS
SD1(dd)=std(b);                                  % Standar Deviation
Peak_Value1(dd)=((max(abs(b))-min(abs(b)))/2); % Peak Value
Kurtosis1(dd)=kurtosis(b)-3;                     % Kurtosis
Variance1(dd)=var(b);                            % Variance
Crest_Factor1(dd)=peak2rms(b);                   % Crest Faktor
Mean1(dd) = mean(b) ;                            % Mean
Entropy1 (dd) = entropy(b);                      % Entropy
Min1(dd)= min(b);                                % Minimum Value
SEM1(dd) = std(b)/sqrt(length(b));               % Standar Error

rootname = 'F:';
extension = '.mat';
namafile = [rootname,'Hasil_Ekstraksi_Data_Awal',extension];
save (namafile)

RMS1 = RMS1';                        % Transpose RMS
SD1 = SD1';                          % Transpose Standar Deviation
Peak_Value1 = Peak_Value1';          % Transpose Peak Value
Kurtosis1 = Kurtosis1';              % Transpose Kurtosis
Variance1 = Variance1';              % Transpose Variance
Crest_Factor1 = Crest_Factor1';      % Transpose Crest Factor
Mean1 = Mean1';                      % Transpose Mean
Entropy1 = Entropy1';                % Transpose Entropy
Min1 = Min1';                        % Transpose Minimum Value
SEM1 = SEM1';                        % Transpose Standar Error
end

% Combine All Result Into One File
Data_Awal_Gabungan=zeros(500,10);             % Create New Cell
```

```
Data_Awal_Gabungan(:,1) = RMS1;
Data_Awal_Gabungan(:,2) = SD1;
Data_Awal_Gabungan(:,3) = Peak_Value1;
Data_Awal_Gabungan(:,4) = Kurtosis1;
Data_Awal_Gabungan(:,5) = Variance1;
Data_Awal_Gabungan(:,6) = Crest_Factor1;
Data_Awal_Gabungan(:,7) = Mean1;
Data_Awal_Gabungan(:,8) = Entropy1;
Data_Awal_Gabungan(:,9) = Min1;
Data_Awal_Gabungan(:,10) = SEM1;

Gabungan_K1 = zscore(Data_Awal_Gabungan);     % Standardize All

%% Initialization (3)
% Kavitasi Level 2 Datasets
signal_in=['D:\Campus\KULIAH\Tugas Akhir\Data Tugas Akhir\2. Data 
Pakai',...
    '\3. Kavitasi Level 
2\Tugas_AkhirKavitasiLevel2_',int2str(d),'.mat'];
load (signal_in);

for ddd=(1:500)                               % Number Of Iteration

% Data Extraction
c = data_all(:,1);

RMS2(ddd)=rms(c);                               % RMS
SD2(ddd)=std(c);                                % Standar Deviation
Peak_Value2(ddd)=((max(abs(c))-min(abs(c)))/2);% Peak Value
Kurtosis2(ddd)=kurtosis(c)-3;                   % Kurtosis
Variance2(ddd)=var(c);                          % Variance
Crest_Factor2(ddd)=peak2rms(c);                 % Crest Faktor
Mean2(ddd)  = mean(c) ;                         % Mean
Entropy2 (ddd)  = entropy(c);                   % Entropy
Min2(ddd)= min(c);                              % Minimum Value
SEM2(ddd)  = std(c)/sqrt(length(c));            % Standar Error

rootname = 'F:';
extension = '.mat';
namafile = [rootname,'Hasil_Ekstraksi_Data_Tengah',extension];
save (namafile)

RMS2 = RMS2';                         % Transpose RMS
SD2 = SD2';                           % Transpose Standar Deviation
Peak_Value2 = Peak_Value2';           % Transpose Peak Value
Kurtosis2 = Kurtosis2';               % Transpose Kurtosis
Variance2 = Variance2';               % Transpose Variance
Crest_Factor2 = Crest_Factor2';       % Transpose Crest Factor
Mean2 = Mean2';                       % Transpose Mean
Entropy2 = Entropy2';                 % Transpose Entropy
Min2 = Min2';                         % Transpose Minimum Value
SEM2 = SEM2';                         % Transpose Standar Error
end
```

```
% Combine All Result Into One File
Data_Tengah_Gabungan=zeros(500,10);              % Create New Cell

Data_Tengah_Gabungan(:,1) = RMS2;
Data_Tengah_Gabungan(:,2) = SD2;
Data_Tengah_Gabungan(:,3) = Peak_Value2;
Data_Tengah_Gabungan(:,4) = Kurtosis2;
Data_Tengah_Gabungan(:,5) = Variance2;
Data_Tengah_Gabungan(:,6) = Crest_Factor2;
Data_Tengah_Gabungan(:,7) = Mean2;
Data_Tengah_Gabungan(:,8) = Entropy2;
Data_Tengah_Gabungan(:,9) = Min2;
Data_Tengah_Gabungan(:,10) = SEM2;

Gabungan_K2 = zscore(Data_Tengah_Gabungan);      % Standardize All

%% Initialization (4)
% Kavitasi Level 3 Datasets
signal_in=['D:\Campus\KULIAH\Tugas Akhir\Data Tugas Akhir\2. Data
Pakai',...
    '\4. Kavitasi Level
3\Tugas_AkhirKavitasiLevel3_',int2str(d),'.mat'];
load (signal_in);

for dddd=(1:500)                              % Number Of Iteration
% Data Extraction
e = data_all(:,1);

RMS3(dddd)=rms(e);                              % RMS
SD3(dddd)=std(e);                               % Standar Deviation
Peak_Value3(dddd)=((max(abs(e))-min(abs(e)))/2);% Peak Value
Kurtosis3(dddd)=kurtosis(e)-3;                   % Kurtosis
Variance3(dddd)=var(e);                          % Variance
Crest_Factor3(dddd)=peak2rms(e);                 % Crest Faktor
Mean3(dddd) = mean(e) ;                          % Mean
Entropy3 (dddd) = entropy(e);                    % Entropy
Min3(dddd)= min(e);                              % Minimum Value
SEM3(dddd) = std(e)/sqrt(length(e));             % Standar Error

rootname = 'F:';
extension = '.mat';
namafile = [rootname,'Hasil_Ekstraksi_Data_Akhir',extension];
save (namafile)

RMS3 = RMS3';                          % Transpose RMS
SD3 = SD3';                            % Transpose Standar Deviation
Peak_Value3 = Peak_Value3';            % Transpose Peak Value
Kurtosis3 = Kurtosis3';                % Transpose Kurtosis
Variance3 = Variance3';                % Transpose Variance
Crest_Factor3 = Crest_Factor3';        % Transpose Crest Factor
Mean3 = Mean3';                        % Transpose Mean
Entropy3 = Entropy3';                  % Transpose Entropy
Min3 = Min3';                          % Transpose Minimum Value
SEM3 = SEM3';                          % Transpose Standar Error
end
```

```
% Combine All Result Into One File
Data_Akhir_Gabungan=zeros(500,10);          % Create New Cell

Data_Akhir_Gabungan(:,1) = RMS3;
Data_Akhir_Gabungan(:,2) = SD3;
Data_Akhir_Gabungan(:,3) = Peak_Value3;
Data_Akhir_Gabungan(:,4) = Kurtosis3;
Data_Akhir_Gabungan(:,5) = Variance3;
Data_Akhir_Gabungan(:,6) = Crest_Factor3;
Data_Akhir_Gabungan(:,7) = Mean3;
Data_Akhir_Gabungan(:,8) = Entropy3;
Data_Akhir_Gabungan(:,9) = Min3;
Data_Akhir_Gabungan(:,10) = SEM3;

Gabungan_K3 = zscore(Data_Akhir_Gabungan);      % Standardize All
%% Plotting

% Root Mean Square (RMS)
figure;
s = 14;                                         % Size (Normal)
c ='r';                                         % Colour (Normal)
scatter(x,(RMS),s,c,'v');                       % Plot RMS (Normal)
hold on
s=14;                                           % Size (Level 1)
c='b';                                          % Colour (Level 1)
scatter(x1,(RMS1),s,c,'x');                     % Plot RMS Level 1
hold on
s=14;                                           % Size (Level 2)
c='g';                                          % Colour (Level 2)
scatter(x2,(RMS2),s,c,'o');                     % Plot RMS Level 2
hold on
s=14;                                           % Size (Level 3)
c='c';                                          % Colour (Level 3)
scatter(x3,(RMS3),s,c,'+');                     % Plot RMS Level 3

axis([0 500 0 1.5])
title('Grafik RMS')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

% Standar Deviation (SD)
figure
s=14;                                            % Size (Normal)
c='r';                                           % Colour (Normal)
scatter(x,(SD),s,c,'v');                         % Plot SD Normal
hold on
s=14;                                            % Size (Level 1)
c='b';                                           % Colour (Level 1)
scatter(x1,(SD1),s,c,'x');                       % Plot SD Level 1
hold on
s=14;                                            % Size (Level 2)
c='g';                                           % Colour (Level 2)
scatter(x2,(SD2),s,c,'o');                       % Plot SD Level 2
hold on
```

```
s=14;                                        % Size (Level 3)
c='c';                                       % Colour (Level 3)
scatter(x3,(SD3),s,c,'+');                   % Plot SD Level 3

axis([0 500 0 1.5])
title('Grafik Standar Deviasi')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

% Peak Value
figure
s=14;                                        % Size (Normal)
c='r';                                       % Colour (Normal)
scatter(x,(Peak_Value),s,c,'v');             % Plot Peak Value
Normal
hold on
s=14;                                        % Size (Level 1)
c='b';                                       % Colour (Level 1)
scatter(x1,(Peak_Value1),s,c,'x');           % Plot Peak Value
Level 1
hold on
s=14;                                        % Size (Level 2)
c='g';                                       % Colour (Level 2)
scatter(x2,(Peak_Value2),s,c,'o');           % Plot Peak Value
Level 2
hold on
s=14;                                        % Size (Level 3)
c='c';                                       % Colour (Level 3)
scatter(x3,(Peak_Value3),s,c,'+');           % Plot Peak Value
Level 3

axis([0 500 0 3.5])
title('Grafik Peak Value')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

% Kurtosis
figure
s=14;                                    % Size (Normal)
c='r';                                   % Colour (Normal)
scatter(x,(Kurtosis),s,c,'v');           % Plot Kurtosis Normal
hold on
s=14;                                    % Size (Level 1)
c='b';                                   % Colour (Level 1)
scatter(x1,(Kurtosis1),s,c,'x');         % Plot Kurtosis Level
1
hold on
s=14;                                    % Size (Level 2)
c='g';                                   % Colour (Level 2)
scatter(x2,(Kurtosis2),s,c,'o');         % Plot Kurtosis Level
2
hold on
s=14;                                    % Size (Level 3)
c='c';                                   % Colour (Level 3)
scatter(x3,(Kurtosis3),s,c,'+');         % Plot Kurtosis Level
3
```

```
axis([0 500 2.5 4])
title('Grafik Kurtosis')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

% Variance
figure
s=14;                                       % Size (Normal)
c='r';                                      % Colour (Normal)
scatter(x,(Variance),s,c,'v');              % Plot Variance Normal
hold on
s=14;                                       % Size (Level 1)
c='b';                                      % Colour (Level 1)
scatter(x1,(Variance1),s,c,'x');            % Plot Variance Level
1
hold on
s=14;                                       % Size (Level 2)
c='g';                                      % Colour (Level 2)
scatter(x2,(Variance2),s,c,'o');            % Plot Variance Level
2
hold on
s=14;                                       % Size (Level 3)
c='c';                                      % Colour (Level 3)
scatter(x3,(Variance3),s,c,'+');            % Plot Variance Level
3

axis([0 500 0 2])
title('Grafik Variance')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

% Crest Factor
figure
s=14;                                       % Size (Normal)
c='r';                                      % Colour (Normal)
scatter(x,(Crest_Factor),s,c,'v');          % Plot Crest Factor
Normal
hold on
s=14;                                       % Size (Level 1)
c='b';                                      % Colour (Level 1)
scatter(x1,(Crest_Factor1),s,c,'x');        % Plot Crest Factor
Level 1
hold on
s=14;                                       % Size (Level 2)
c='g';                                      % Colour (Level 2)
scatter(x2,(Crest_Factor2),s,c,'o');        % Plot Crest Factor
Level 2
hold on
s=14;                                       % Size (Level 3)
c='c';                                      % Colour (Level 3)
scatter(x3,(Crest_Factor3),s,c,'+');        % Plot Crest Factor
Level 3

axis([0 500 3 6])
```

```
title('Grafik Crest Factor')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

% Mean
figure
s=9;                                        % Size (Normal)
c='r';                                      % Colour (Normal)
scatter(x,(Mean),s,c,'v');                  % Plot Mean Normal
hold on
s=9;                                        % Size (Level 1)
c='b';                                      % Colour (Level 1)
scatter(x1,(Mean1),s,c,'x');                % Plot Mean Level
1
hold on
s=9;                                        % Size (Level 2)
c='g';                                      % Colour (Level 2)
scatter(x2,(Mean2),s,c,'o');                % Plot Mean Level
2
hold on
s=9;                                        % Size (Level 3)
c='c';                                      % Colour (Level 3)
scatter(x3,(Mean3),s,c,'+');                % Plot Mean Level
3

axis([0 500 -0.01 0.01])
title('Grafik Mean')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

% Entropy
figure
s=9;                                    % Size (Normal)
c='r';                                  % Colour (Normal)
scatter(x,(Entropy),s,c,'v');           % Plot Entropy Normal
hold on
s=9;                                    % Size (Level 1)
c='b';                                  % Colour (Level 1)
scatter(x1,(Entropy1),s,c,'x');         % Plot Entropy Level 1
hold on
s=9;                                    % Size (Level 2)
c='g';                                  % Colour (Level 2)
scatter(x2,(Entropy2),s,c,'o');         % Plot Entropy Level 2
hold on
s=9;                                    % Size (Level 3)
c='c';                                  % Colour (Level 3)
scatter(x3,(Entropy3),s,c,'+');         % Plot Entropy Level 3

axis([0 500 16 17.5])
title('Grafik Entropy')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

% Minimum Value
figure
```

```
s=9;                                              % Size (Normal)
c='r';                                            % Colour (Normal)
scatter(x,(Min),s,c,'v');                         % Plot Minimum Normal
hold on
s=9;                                             % Size (Level 1)
c='b';                                           % Colour (Level 1)
scatter(x1,(Min1),s,c,'x');                      % Plot Minimum Level 1
hold on
s=9;                                             % Size (Level 2)
c='g';                                           % Colour (Level 2)
scatter(x2,(Min2),s,c,'o');                      % Plot Minimum Level 2
hold on
s=9;                                             % Size (Level 3)
c='c';                                           % Colour (Level 3)
scatter(x3,(Min3),s,c,'+');                      % Plot Minimum Level 3

axis([0 500 -9.5 0])
title('Grafik Minimum')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

% Standar Error (SE)
figure
s=9;                                                 % Size (Normal)
c='r';                                               % Colour (Normal)
scatter(x,(SEM),s,c,'v');                            % Plot SE Normal
hold on
s=9;                                                 % Size (Level 1)
c='b';                                               % Colour (Level 1)
scatter(x1,(SEM1),s,c,'x');                          % Plot SE Level 1
hold on
s=9;                                                 % Size (Level 2)
c='g';                                               % Colour (Level 2)
scatter(x2,(SEM2),s,c,'o');                          % Plot SE Level 2
hold on
s=9;                                                 % Size (Level 3)
c='c';                                               % Colour (Level 3)
scatter(x3,(SEM3),s,c,'+');                          % Plot SE Level 3

axis([0 500 0 0.1])
title('Grafik SE')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')
```

Lampiran 5. Script MATLAB Seleksi Parameter Statistik

```
% ====================Relief Feature Selection====================
% =Updated: July, 21st 2018, Muhammad Taufiq Akbar (20140130069)=

clear; close all; clc;
%% Datasets Preparation
load Classification_SVM;

X = Input_Baru;                                            % Predictor
Y = categories;                                            % Response
%% Selection Process
[ranked,weights] = relieff(X,Y,2000);

ClassName = ranked.predictorsName;
%% Plotting
bar(weights(ranked));
xlabel('Predictor rank');
ylabel('Predictor importance weight');
h = gca;
h.XTickLabel = ({'ClassName'});
h.XTickLabelRotation = 45;
h.TickLabelInterpreter = 'none';
title('Relieff Feature Selection');
```

Lampiran 6. Script MATLAB Klasifikasi *Binary SVM*

```
% Binary Support Vector Machine
% Updated: July, 21st 2018, Muhammad Taufiq Akbar (20140130069)

clear; close all; clc;

%% Datasets Preparation
load Classification_SVM;

%Change Data Type
jenis = grp2idx(categories(1:1000,:));                  % N & L1
jenis = grp2idx(categories([(1:500);(1001:1500)],:)); % N & L2
jenis = grp2idx(categories([(1:500);(1501:2000)],:)); % N & L3

%% Binary Classification Problem
X = Input(1:1000,[5 2 1]);                              % N & L1
X = Input([(1:500);(1001:1500)],[5 2 1]);               % N & L2
X = Input([(1:500);(1501:2000)],[5 2 1]);               % N & L3

y = jenis;                                              % Response

%% Cross Validation Partition
c = cvpartition(y,'k',10);          % 10Fold Cross-Validation
c = cvpartition(y,'HoldOut',0.10);  % Separating Training and Test

%% Training Datasets
X_train = X(training(c),:);                   % Observation
y_train = y(training(c),:);                   % Categories

% Class 1
x1 = X_train((1:450),1);
y1 = X_train((1:450),2);
z1 = X_train((1:450),3);

% Class 2
x2 = X_train((451:900),1);
y2 = X_train((451:900),2);
z2 = X_train((451:900),3);

%% Test Datasets
X_test = X(test(c),:);                        % Observation
y_test = y(test(c),:);                        % Categories

% Class 1
x3 = X_test((1:50),1);
y3 = X_test((1:50),2);
z3 = X_test((1:50),3);

% Class 2
x4 = X_test((51:100),1);
y4 = X_test((51:100),2);
z4 = X_test((51:100),3);
```

```
%% Train Optimal Hyperparameter
Mdl = fitcsvm(X_train,y_train,'KernelFunction','rbf',...

'OptimizeHyperparameters','auto','HyperparameterOptimizationOption
',...
    struct('AcquisitionFunctionName','expected-improvement-
plus',...
    'ShowPlots',true));

ClassOrder = Mdl.ClassNames;

%% Training Model Evaluation Process
[Label,Score] = predict(Mdl,X_test);
Accuracy = sum(predict(Mdl,X_test)== y_test)/length(y_test)*100;

%% Create Support Vector
sv = Mdl.SupportVectors;

%% Plot Training Result
figure;
h = scatter3(X_train(:,1),X_train(:,2),X_train(:,3),y_train);
title('Training Klasifikasi Normal dan Kavitasi Level 3');
hold on
scatter3(x1,y1,z1,'ro','filled');
scatter3(x2,y2,z2,'bo','filled');
hold on
plot3(X_train(Mdl.IsSupportVector,1),X_train(Mdl.IsSupportVector,2
),...
    X_train(Mdl.IsSupportVector,3), 'ko','MarkerSize',5);
legend('data','Training Normal','Training Kavitasi 3','Support
Vector');

%% Plot Evaluation Result
figure;
scatter3(X_test(:,1),X_test(:,2),X_test(:,3),y_test);
hold on
title('Klasifikasi Normal dan Kavitasi Level 3');
scatter3(x3,y3,z3,'rx');
scatter3(x4,y4,z4,'bx');
legend('data','Test Normal','Test Kavitasi 3');
```

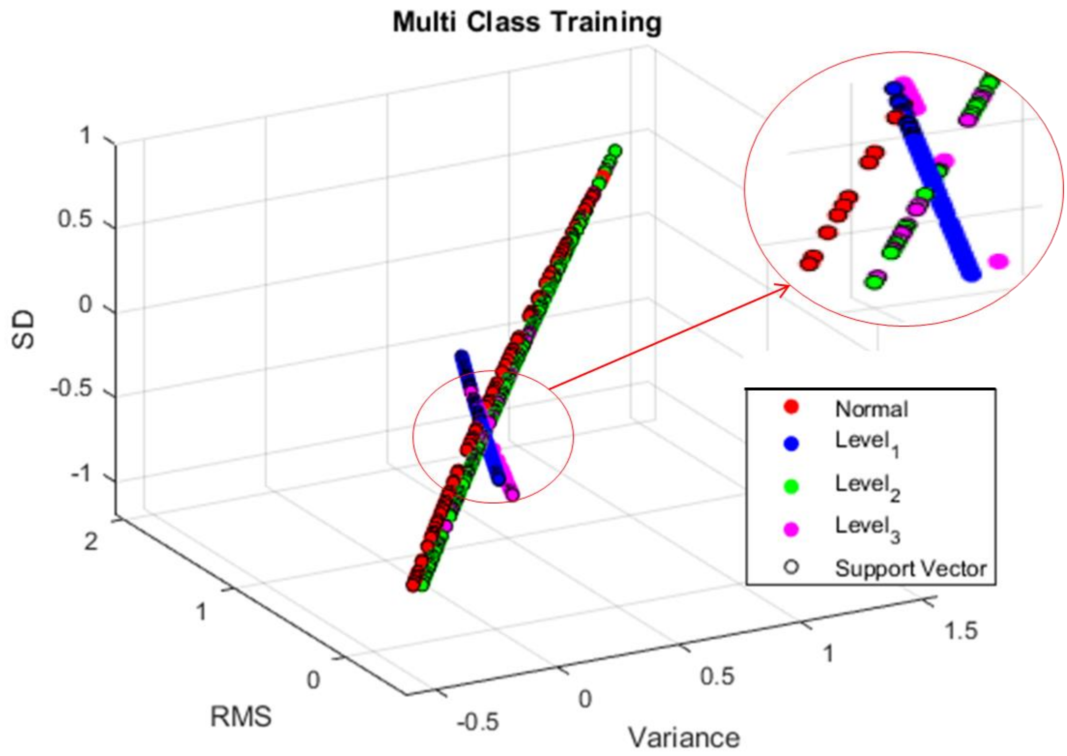
Lampiran 7. Script MATLAB Klasifikasi *Multi-Class SVM*

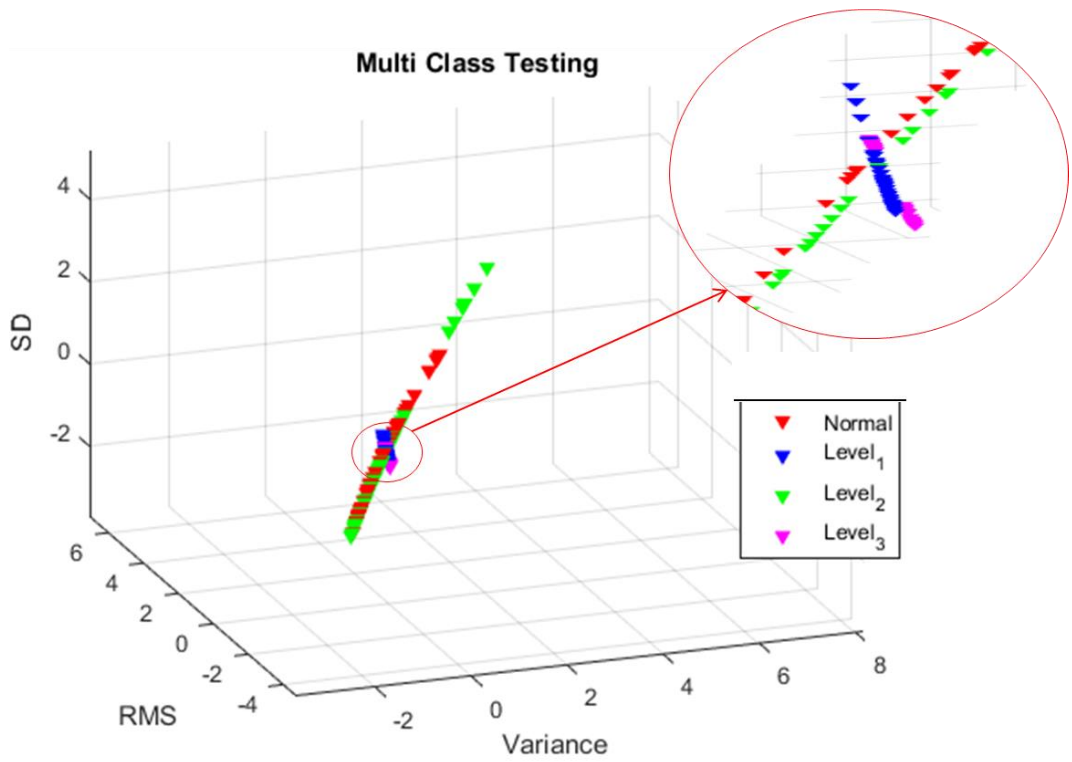
```
% Multi-Class Support Vector Machine
% Updated: July, 21st 2018, Muhammad Taufiq Akbar (20140130069)

clear; close all; clc;

%% Datasets Preparation
load Classification_SVM;

%% Multi-Class Classification Problem
X = Input_Baru(1:2000,[5 1 2]);             % Input Predictor
y = categories(1:2000,:);                   % Input Response
%% Cross-Validation Partition
c = cvpartition(y,'k',10)           % 10Fold Cross-Validation
c = cvpartition(y,'HoldOut',0.10);  % Separating Training and Test

%% Training Datasets
X_train = X(training(c),:);
y_train = y(training(c),:);

jenis1 = grp2idx(y_train);
% Class 1
x1 = X_train((1:450),1);
y1 = X_train((1:450),2);
z1 = X_train((1:450),3);
% Class 2
x2 = X_train((451:900),1);
y2 = X_train((451:900),2);
z2 = X_train((451:900),3);
% Class 3
x3 = X_train((901:1350),1);
y3 = X_train((901:1350),2);
z3 = X_train((901:1350),3);
% Class 4
x4 = X_train((1301:1800),1);
y4 = X_train((1301:1800),2);
z4 = X_train((1301:1800),3);
%% Test Datasets
X_test= X(test(c),:);
y_test = y(test(c),:);

jenis2 = grp2idx(y_test);
% Class 1
x5 = X_test((1:50),1);
y5 = X_test((1:50),2);
z5 = X_test((1:50),3);
% Class 2
x6 = X_test((51:100),1);
y6 = X_test((51:100),2);
z6 = X_test((51:100),3);
% Class 3
x7 = X_test((101:150),1);
y7 = X_test((101:150),2);
```

```
z7 = X_test((101:150),3);
% Class 4
x8 = X_test((151:200),1);
y8 = X_test((151:200),2);
z8 = X_test((151:200),3);

%% Train Optimal Hyperparameter
t = templateSVM('KernelFunction','rbf','SaveSupportVectors',true);

% Multi-Class SVM Without Optimization (1)
Mdl = fitcecoc(X_train,y_train,'Learners',t,'ClassNames',...
    {'Normal','Awal','Tengah','Akhir'},'Verbose',0,...
    'OptimizeHyperparameters','none');

% Multi-Class SVM With Bayesian Optimization (2)
Mdl =
fitcecoc(X_train,y_train,'Coding','onevsall','Learners',t,...

'Prior','uniform','ClassNames',{'Normal','Awal','Tengah','Akhir'},
...
    'Verbose',0,'OptimizeHyperparameters','auto',...
    'HyperparameterOptimizationOptions',...
    struct('Optimizer','bayesopt','ShowPlots',true,...
    'AcquisitionFunctionName','expected-improvement-plus'));

% Multi-Class SVM With Grid Search Method (3)
Mdl =
fitcecoc(X_train,y_train,'Coding','onevsall','Learners',t,...

'Prior','uniform','ClassNames',{'Normal','Awal','Tengah','Akhir'},
...
    'Verbose',0,'OptimizeHyperparameters','auto',...

'HyperparameterOptimizationOptions',struct('Optimizer','gridsearch
',...
    'ShowPlots',true,'AcquisitionFunctionName',...
    'expected-improvement-plus','NumGridDivisions',10));

sortrows(Mdl.ParameterOptimizationResults);
ClassOrder = Mdl.ClassNames;
%% Create Support Vector
Mdl.ClassNames;
Mdl.CodingMatrix ;
L = size(Mdl.CodingMatrix,1);

sv = cell(L,1);           % Preallocate for support vector indices
for j = 1:L;
    SVM = Mdl.BinaryLearners{j};
    sv{j} = SVM.SupportVectors;
end

%% Training Model Evaluation Process
CVMdl = crossval(Mdl);
[Label,Score] = predict(Mdl,X_test);
ConfMat = confusionmat(y_test,Label);
```

```
[n,p] = size(X_test);
isLabels = unique(y_test);
nLabels = numel(isLabels);
tabulate(categorical(y_test));
[~,grpOOF] = ismember(Label,isLabels);
oofLabelMat = zeros(nLabels,n);
idxLinear = sub2ind([nLabels n],grpOOF,(1:n)');
oofLabelMat(idxLinear) = 1;
[~,grpYTS] = ismember(y_test,isLabels);
YMat = zeros(nLabels,n);
idxLinearYTS = sub2ind([nLabels n],grpYTS,(1:n)');
YMat(idxLinearYTS) = 1;

%% Plot Confusion Matrix
figure;
plotconfusion(YMat,oofLabelMat);
h = gca;

%% Plot Training Result
figure;
h = scatter3(X_train(:,1),X_train(:,2),X_train(:,3),jenis1);
hold on
scatter3(x1,y1,z1,'ro','filled');
hold on
scatter3(x2,y2,z2,'bo','filled');
hold on
scatter3(x3,y3,z3,'go','filled');
hold on
scatter3(x4,y4,z4,'mo','filled');
hold on
markers = {'ko','ko','ko','ko'};
for j = 1:L;
    svs = sv{j};
    h(j+3) =
plot3(svs(:,1),svs(:,2),svs(:,3),markers{j},'MarkerSize',5);
end
legend('','Normal','Level_1','Level_2','Level_3','Support
Vector');
title('Training Multi Class SVM with Bayesian Optimization');

%% Plot Evaluation Result
figure;
h2 = scatter3(X_test(:,1),X_test(:,2),X_test(:,3),jenis2);
legend('Normal','Level_1','Level_2','Level_3');
hold on
scatter3(x5,y5,z5,'rv','filled');
scatter3(x6,y6,z6,'bv','filled');
scatter3(x7,y7,z7,'gv','filled');
scatter3(x8,y8,z8,'mv','filled');
title('Testing Bayesian Optimization Model Training');
legend('','Normal','Level_1','Level_2','Level_3');
```

Lampiran 8. Gambar Hasil *Multi-Class SVM* tanpa Optimalisasi





Lampiran 9. Gambar Hasil *Multi-Class SVM* menggunakan *GSM*