DAFTAR PUSTAKA

A.A. Miranda, Y.A. Le Borgne, and G. Bontempi. New Routes From Minimal Approximation Error to Principal Component,Volume 27,Number 3/ june 2008, Neural Processing Letter,Springer.

Al Hashmi, S.A. (2009). Statistical Analysis of Vibration Signals for Cavitation Detection ISIEA. 2009 IEEE Symposium on Industrial Electronics and Application, ISIEA 2009 – Proceedings 1 (ISIEA), 78 – 82.

Al Tobi, M.A.S., Al Sabari, M.H.J. (2016). *Cavitation Detection of Centrifugal Pump Using Time – Domain Method*, 4 (5), 161 – 167.

Anta, Felly., (2013). Tugas Akhir *"Deteksi Kavitasi Pompa Sentrifugal dengan Analisis Sinyal Getaran".* Semarang: Jurusan Teknik Mesin Fakultas Teknik Universitas Diponegoro.

Aydin, Ihan., Karakose, Mehmet. Akin, Erhan. 2011. *A Multi Objective Artificial Immune Algorithm for Paramater Optimization in SVM.* Computer Engineering Department, Firat University, 23119, Elazig, Turkey.

Delly, Jenny. (2009). Tugas Akhir "*Pengaruh Temperatur Terhadap Terjadinya Kavitasi Pada Sudu Pompa Sentrifugal".* Kendari : Jurusan Teknik Mesin, Fakultas Teknik Universitas Haluoleo.

Kamiel, Berli Paripurna. 2015. *Vibration-based multi-fault diagnosis for centrifugal pumps*. Ph.D. Curtin University, Department of Mechanical Engineering.

Karassik, I.J., Cooper, P., Heald, C.C. (2008). *Pump Handbook.*

Luo, Y., Sun, H., Yuan, S., Yuan, J. (2015). *Research on Statistical Characteristics of Vibration in Centrifugal Pump,* 38, 49 – 61.

Pratama, M. Sofyan Bagus. 2017. Tugas Akhir "*Metode Deteksi Fenomena Kavitasi Pompa Sentrifugal Berbasis Domain Waktu dan Domain Frekuensi Sinyal Getaran*"**.** Jurusan Teknik Mesin Fakultas Teknik Universitas Muhammadiyah Yogyakarta.

Saberi, A.Azadeh, A.Nourmuhammadzadeh, and P.Pazhoheshfar. 2011. *Comparing Performance and Robustness of SVM and ANN for Fault Diagnosis in a Centrifugal Pump. 19th International Congress on Modelling and Simulation, Perth, Australia, 12–16 Desember.* http://mssanz.org.au/modsim2011

Santosa B., 2007, *Data Mining (Teori dan Aplikasi)*, Graha Ilmu, Yogyakarta.

Sakthivel, Sugumaran, and Nair, Binoy B. 2010. *Application of Support Vector Machine (SVM) and Proximal Support Vector Machine (PSVM) for Fault Classification of Monoblock Centrifugal Pump.* Int. J. Data Analysis Techniques and Strategies, Vol 2, No 1. India

Scheffer, Cornelius., and Girdhar, Paresh. 2004. "*Practical Machinery Vibration Analysis and Predictive Maintenance*". Newnes An imprint of Elsevier Linacre l-louse, Jordan Hill, Oxford OX2 BDP 200 Wheeler Road, Burlington, MA01 803.

Sitorus, Asril., (2010). Tugas Akhir "*Studi Eksperimental Fenomena Kavitasi pada Pompa Sentrifugal Melalui Pengamatan Pola Aliran yang Diinterprestasikan Terhadap Perilaku Sinyal Vibrasi*". Medan: Departemen Teknik Mesin Fakultas Teknik Univesitas Sumatera Utara.

Sularso, (1993). "Pompa dan kompresor". Jakarta: Pradnya Paramita.

Syafutra, Kurniady. (2017). Tugas Akhir "*Metode Deteksi Fenomena Kavitasi pada Pompa Sentrifugal Berbasis Domain Frekuensi Sinyal Getaran*".Jurusan Teknik Mesin Fakultas Teknik Universitas Muhammadiyah Yogyakarta.

Taghi, Mohammed., Ettefagh, Mir Mohammed., Lotfan, Saeed., and Safarzadeh, Hamed. 2017. *Cavitation Intensity Monitoring in an Axial Flow Pump Based on Vibration Signals Using Multiclass Support Vector Machine.* Research Laboratory of Vibration and Modal Analysis, Department of Mechanical Engineering, University of Tabriz, Tabriz, Iran.

Wen, Y. and Henry, M. (2002). *Time Frequency Characteristics of the Vibroacoustic Signal of Hydrodynamic Cavitations. Journal of Vibration and Acoustics*. Vol 124:469-475.

LAMPIRAN

Lampiran 1.1: *Script Matlab Pengambilan Akuisisi Data*

```
%Script to run data acquisition using National Instrument NI 9234
%Created: July 2018,


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,'Normal_',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
```

Lampiran 1.2: Script Matlab Pengolahan Data Mentah Menjadi Plot Domain Waktu

```
clear
clc
%Direktori tempat data mentah getaran berada
load('E:\yudha\Data Tugas Akhir\2. Data Pakai\1. Training Data\1.
Normal\Tugas_AkhirNormal1_15.mat')
y1=data_all(:,1);  %diberi nama dengan variabel baru dan berbeda
untuk masing-masing variasi bukaan katup
```

```
load('E:\yudha\Data Tugas Akhir\2. Data Pakai\2. Testing Data\2.
Kavitasi Level 1\Tugas_AkhirKavitasiLevel1_15
.mat')

y2=data_all(:,1);

load(E:\yudha\Data Tugas Akhir\2. Data Pakai\1. Training Data\3.
Kavitasi Level 2\Tugas_AkhirKavitasiLevel2_
15.mat')

y3=data_all(:,1);

load('E:\yudha\Data Tugas Akhir\2. Data Pakai\2. Testing Data\4.
Kavitasi Level 3\Tugas_AkhirKavitasiLevel3
_15.mat')

y4=data_all(:,1);

% plot amplitude time domain

figure

subplot(5,1,1)

plot(y1(1:170660))

axis([0 5000 -5 5])

legend('Normal')

xlabel('Sampel')

ylabel('Amplitudo')


subplot(5,1,2)

plot(y2(1:170660),'r')

axis([0 5000 -5 5])

legend('Level 1')

xlabel('Sampel')

ylabel('Amplitudo')


subplot(5,1,3)

plot(y3(1:170660),'g')

axis([0 5000 -5 5])

legend('Level 2')

xlabel('Sampel')
```

```
ylabel('Amplitudo')


subplot(5,1,4)

plot(y4(1:170660),'b')

axis([0 5000 -5 5])

legend('Level 3')

xlabel('Sampel')

ylabel('Amplitudo')
```

Lampiran 1.3: Script Matlab Ekstrasi Parameter Statistik

```
clc
close all
clear

%Normal
for d=(1:250)
signal_in=['E:\yudha\Data Tugas Akhir\2. Data Pakai\1. Training
Data\1. Normal\Tugas_AkhirNormal1_',int2str(d),'.mat'];
load (signal_in)

a=data_all(:,1);
R(d)=rms(a);
S(d)=std(a);
P(d)=((max(abs(a))-min(abs(a)))/2);
K(d)=kurtosis(a);
V(d)=var(a);
C(d)=peak2rms(a);
M(d)=mean(a);
SK(d)=skewness(a);
x=1:250;

Normaltrai=zeros(250,7);
R=R';                                          %Transpose RMS
S=S';                                            %Transpose Standar
Deviation
P=P';                              %Transpose Peak Value
K=K';                                  %Transpose Kurtosis
V=V';                                  %Transpose Variance
C=C';                          %Transpose Crest Factor
M=M';                           %Transpose Mean
SK=SK';                      %Transpose Skewness
end
%Kavitasi1
for di=(1:250)
```

```
signal_in=['E:\yudha\Data Tugas Akhir\2. Data Pakai\1. Training
Data\2. Kavitasi Level
1\Tugas_AkhirKavitasiLevel1_',int2str(di),'.mat'];
load (signal_in)

b=data_all(:,1);
R1(di)=rms(b);
S1(di)=std(b);
P1(di)=((max(abs(b))-min(abs(b)))/2);
K1(di)=kurtosis(b);
V1(di)=var(b);
C1(di)=peak2rms(b);
M1(di)=mean(b);
SK1(di)=skewness(b);
x1=1:250;

Kavitasi1test=zeros(250,8);
R1=R1';                                          %Transpose RMS
S1=S1';                                            %Transpose Standar
Deviation
P1=P1';                              %Transpose Peak Value
K1=K1';                                  %Transpose Kurtosis
V1=V1';                                  %Transpose Variance
C1=C1';                           %Transpose Crest Factor
M1=M1';                             %Transpose Mean
SK1=SK1';                                        %Transpose Skewness
end

%kavitasi2
for dii=(1:250)
signal_in=['E:\yudha\Data Tugas Akhir\2. Data Pakai\1. Training
Data\3. Kavitasi Level
2\Tugas_AkhirKavitasiLevel2_',int2str(dii),'.mat'];
load (signal_in)

f=data_all(:,1);
R2(dii)=rms(f);
S2(dii)=std(f)
P2(dii)=((max(abs(f))-min(abs(f)))/2);
K2(dii)=kurtosis(f);
V2(dii)=var(f);
C2(dii)=peak2rms(f);
M2(dii)=mean(f);
SK2(dii)=skewness(f);
x2=1:250;

Kavitasi2test=zeros(250,8);
R2=R2';                                          %Transpose RMS
S2=S2';                                            %Transpose Standar
Deviation
P2=P2';                              %Transpose Peak Value
```

```
K2=K2';                                 %Transpose Kurtosis
V2=V2';                                 %Transpose Variance
C2=C2';                          %Transpose Crest Factor
M2=M2';                                          %Transpose Mean
SK2=SK2';                       %Transpose Skewness
end

%kavitasi3
for diii=(1:250)
signal_in=['E:\yudha\Data Tugas Akhir\2. Data Pakai\1. Training
Data\4. Kavitasi Level
3\Tugas_AkhirKavitasiLevel3_',int2str(diii),'.mat'];
load (signal_in)

e=data_all(:,1);
R3(diii)=rms(e);
S3(diii)=std(e);
P3(diii)=((max(abs(e))-min(abs(e)))/2);
K3(diii)=kurtosis(e);
V3(diii)=var(e);
C3(diii)=peak2rms(e);
M3(diii)=mean(e);
SK3(diii)=skewness(e);
x3=1:250;

Kavitasi3trai=zeros(250,8);
R3=R3';                                             %Transpose RMS
S3=S3';                                               %Transpose Standar
Deviation
P3=P3';                              %Transpose Peak Value
K3=K3';                                 %Transpose Kurtosis
V3=V3';                                 %Transpose Variance
C3=C3';                          %Transpose Crest Factor
M3=M3';                                          %Transpose Mean
SK3=SK3';                          %Transpose Skewness
end

%R
figure
s=14;
c='r';
scatter(x,(R),s,c,'v');
hold on
s=14;
c='b';
scatter(x1,(R1),s,c,'x');
hold on
s=14;
c='g';
scatter(x2,(R2),s,c,'o');
hold on
```

```
s=14;
c='c';
scatter(x3,(R3),s,c,'+');

axis([0 250 0 1.5])
title('Grafik RMS')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

%S
figure
s=14;
c='r';
scatter(x,(S),s,c,'v');
hold on
s=14;
c='b';
scatter(x1,(S1),s,c,'x');
hold on
s=14;
c='g';
scatter(x2,(S2),s,c,'o');
hold on
s=14;
c='c';
scatter(x3,(S3),s,c,'+');

axis([0 250 0 1.5])
title('Grafik Standar Deviasi')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

%P
figure
s=14;
c='r';
scatter(x,(P),s,c,'v');
hold on
s=14;
c='b';
scatter(x1,(P1),s,c,'x');
hold on
s=14;
c='g';
scatter(x2,(P2),s,c,'o');
hold on
s=14;
c='c';
scatter(x3,(P3),s,c,'+');

axis([0 250 0 3.5])
```

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

%K
figure
s=14;
c='r';
scatter(x,(K),s,c,'v');
hold on
s=14;
c='b';
scatter(x1,(K1),s,c,'x');
hold on
s=14;
c='g';
scatter(x2,(K2),s,c,'o');
hold on
s=14;
c='c';
scatter(x3,(K3),s,c,'+');

axis([0 250 2.5 4])
title('Grafik Kurtosis')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

%V
figure
s=14;
c='r';
scatter(x,(V),s,c,'v');
hold on
s=14;
c='b';
scatter(x1,(V1),s,c,'x');
hold on
s=14;
c='g';
scatter(x2,(V2),s,c,'o');
hold on
s=14;
c='c';
scatter(x3,(V3),s,c,'+');

axis([0 250 0 2])
title('Grafik Varians')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

%C
```

```
figure
s=14;
c='r';
scatter(x,(C),s,c,'v');
hold on
s=14;
c='b';
scatter(x1,(C1),s,c,'x');
hold on
s=14;
c='g';
scatter(x2,(C2),s,c,'o');
hold on
s=14;
c='c';
scatter(x3,(C3),s,c,'+');

axis([0 250 3 6])
title('Grafik Crest Factor')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

%M
figure
s=9;
c='r';
scatter(x,(M),s,c,'v');
hold on
s=9;
c='b';
scatter(x1,(M1),s,c,'x');
hold on
s=9;
c='g';
scatter(x2,(M2),s,c,'o');
hold on
s=9;
c='c';
scatter(x3,(M3),s,c,'+');

axis([0 250 -0.01 0.01])
title('Grafik Mean')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')

%SK
figure
s=14;
c='r';
scatter(x,(SK),s,c,'v');
hold on
```

```
s=14;
c='b';
scatter(x1,(SK1),s,c,'x');
hold on
s=14;
c='g';
scatter(x2,(SK2),s,c,'o');
hold on
s=14;
c='c';
scatter(x3,(SK3),s,c,'+');

axis([0 250 -0.1 0])
title('Grafik Skewness')
xlabel('Sampel'),ylabel('Amplitudo')
legend ('Normal','Kavitasi1','Kavitasi2','Kavitasi3')
```

Lampiran 1.4: Script Matlab Principal Component Analysis (PCA)

```
vector_ch1234(:,:,1)=NormalTraining;
vector_ch1234(:,:,2)=KavitasiTraining1;
vector_ch1234(:,:,3)=KavitasiTraining2;
vector_ch1234(:,:,4)=KavitasiTraining3;
for i=1:4
eval (
['[LOADING_ch',int2str(i),',SCORE_ch',int2str(i),',latent_ch',int2st
r(i),',T2_ch',int2str(i),']
=princomp(zscore(vector_ch1234(:,:,',int2str(i),')));']);
eval ( [
'[norm_vector_ch1234(:,:,',int2str(i),'),MU(i,:),SIGMA(i,:)]=
zscore(vector_ch1234(:,:,',int2str(i),'));' ])
eval ( ['loading(:,:,',int2str(i),')=LOADING_ch',int2str(i),';'])
eval ( ['score(:,:,',int2str(i),')=SCORE_ch',int2str(i),';'])
eval ( ['latent(:,',int2str(i),')=latent_ch',int2str(i),';'])

end

figure

pareto(LATENT)

xlabel('Principal Component')

ylabel('Variance Explained (%)')
```

Lampiran 1.5: Script Matlab Klasifikasi Binary SVM

```
%% Classification Process Using Support Vector Machine

clear; close all; clc

%% Prepare Datasets
load SVM_for_Classification
```

```
jenis = grp2idx(categories(1:500,:));
%jenis = grp2idx(categories([(1:250);(501:750)],:));        %Change
Data Type
%jenis = grp2idx(categories([(1:250);(751:1000)],:));

%% Binary Classification Problem
X = Input(1:1000,[1 2 3]);
%X = Input([(1:250);(501:750)],[1 2 3]);
%X = Input([(1:250);(751:1000)],[1 2 3]);

y = jenis;

%% Cross Validation Partition

%c = cvpartition(y,'k',10)
c = cvpartition(y,'HoldOut',0.10);
X_train = X(training(c),:);
y_train = y(training(c),:);
%% Train Optimal Hyperparameter

Mdl =
fitcsvm(X_train,y_train,'KernelFunction','rbf','OptimizeHyperparamet
ers','auto',...

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

ClassOrder = Mdl.ClassNames;

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

[Label,Score] = predict(Mdl,X_test);
Accuracy = sum(predict(Mdl,X_test)== y_test)/length(y_test)*100;
%% Hyperplane
sv = Mdl.SupportVectors;
%% Build Classifier Model/ Train Data
% Class 1
x1 = X_train((1:225),1);
y1 = X_train((1:225),2);
z1 = X_train((1:225),3);

% Class 2
x2 = X_train((226:450),1);
y2 = X_train((226:450),2);
z2 = X_train((226:450),3);
```

```
%%
figure;
h = scatter3(X_train(:,1),X_train(:,2),X_train(:,3),y_train);
%title('Klasifikasi Normal dan Kavitasi Level 3');
hold on
title('Klasifikasi Data Training Normal dan Kavitasi Level 1');
xlabel('PC1'),ylabel('PC2'),zlabel('PC3')
%scatter3(x1,y1,z1,'ro');
%scatter3(x2,y2,z2,'yo,');
scatter3(x1,y1,z1,'ro','filled');
scatter3(x2,y2,z2,'bo','filled');
legend('Train Data Normal','Train Data Normal','Train Data Kavitasi
1');
hold on
%legend([h],{'data','Train Normal','Train Kavitasi1'});
plot3(X_train(Mdl.IsSupportVector,1),X_train(Mdl.IsSupportVector,2),
X_train(Mdl.IsSupportVector,3), 'ko','MarkerSize',5);
%% Classify/Test Data
% Class 1
x3 = X_test((1:25),1);
y3 = X_test((1:25),2);
z3 = X_test((1:25),3);

% Class 2
x4 = X_test((26:50),1);
y4 = X_test((26:50),2);
z4 = X_test((26:50),3);

figure;
h=scatter3(X_test(:,1),X_test(:,2),X_test(:,3),y_test);
hold on
title('Klasifikasi Data Testing Normal dan Kavitasi Level 1');
xlabel('PC1'),ylabel('PC2'),zlabel('PC3')
scatter3(x3,y3,z3,'rx');
scatter3(x4,y4,z4,'bx');
legend('Test Data Normal','Test Data Normal','Test Data Kavitasi
1');
hold on

%legend([h],{'data','Test Normal','Test Kavitasi1'});
```

Lampiran 1.6: Script Matlab Klasifikasi Multiclass SVM

```
clear; close all; clc
%% Estimate Posterior Probabilities Using ECOC Classifiers

%% Prepare Datasets
load SVM_for_Classification

%% Binary Classification Problem
X = Input(1:1000,[1 2 3]);
```

```
y = categories(1:1000,:);
%% Cross-Validation Partition

%c = cvpartition(y,'k',10)
c = cvpartition(y,'HoldOut',0.10);
X_train = X(training(c),:);
y_train = y(training(c),:);

%%
t = templateSVM('KernelFunction','rbf');

Mdl =
fitcecoc(X_train,y_train,'Coding','onevsall','Learners',t,'Prior','u
niform','ClassNames',{'Normal','Awal','Tengah','Akhir'},...

'Verbose',0,'OptimizeHyperparameters','auto','HyperparameterOptimiza
tionOptions',...
   struct('AcquisitionFunctionName','expected-improvement-plus'));

ClassOrder = Mdl.ClassNames;

%%
Mdl.ClassNames
Mdl.CodingMatrix
L = size(Mdl.CodingMatrix,1);


%% Test Data
X_test= X(test(c),:);
y_test = y(test(c),:);

CVMdl = crossval(Mdl);
[Label,Score] = predict(Mdl,X_test);
ConfMat = confusionmat(y_test,Label);

%% Plot Training Data

jenis1 = grp2idx(y_train);

x1 = X_train((1:225),1);
y1 = X_train((1:225),2);
z1 = X_train((1:225),3);
X1 = meshgrid(X_train((1:225),1));

x2 = X_train((226:450),1);
y2 = X_train((226:450),2);
z2 = X_train((226:450),3);

x3 = X_train((451:675),1);
```

```
y3 = X_train((451:675),2);
z3 = X_train((451:675),3);

x4 = X_train((676:900),1);
y4 = X_train((676:900),2);
z4 = X_train((676:900),3);

%%
figure;
h = scatter3(X_train(:,1),X_train(:,2),X_train(:,3),jenis1);
legend('Normal','Normal','Level_1','Level_2','Level_3');
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
title('Training Data')
xlabel('PC1'),ylabel('PC2'),zlabel('PC3')
legend('Normal','Normal','Level_1','Level_2','Level_3');

%% Plot Test Data

jenis2 = grp2idx(y_test);

x5 = X_test((1:25),1);
y5 = X_test((1:25),2);
z5 = X_test((1:25),3);

x6 = X_test((26:50),1);
y6 = X_test((26:50),2);
z6 = X_test((26:50),3);

x7 = X_test((51:75),1);
y7 = X_test((51:75),2);
z7 = X_test((51:75),3);

x8 = X_test((76:100),1);
y8 = X_test((76:100),2);
z8 = X_test((76:100),3);

figure;
h2 = scatter3(X_test(:,1),X_test(:,2),X_test(:,3),jenis2);
legend('Normal','Normal','Level_1','Level_2','Level_3');
hold on
scatter3(x5,y5,z5,'ro','filled');
scatter3(x6,y6,z6,'bo','filled');
```

```
scatter3(x7,y7,z7,'go','filled');
scatter3(x8,y8,z8,'mo','filled');
title('Testing Data')
xlabel('PC1'),ylabel('PC2'),zlabel('PC3')
legend('Normal','Normal','Level_1','Level_2','Level_3');
```