LAMPIRAN

Lampiran 1: Spesifikasi Akselerometer tipe 4507B seri 30171

Calibration Chart for DeltaTron® Accelerometer Type 4507 B

Brüel & Kjær

Serial No.: 30171

Reference Sensitivity [1] at 159.2 Hz (ω = 1000 s^{-1}), 20 ms^{-2} RMS, 4 mA supply current and 24.0 °C: 10.21 mV/ms^{-2} (100.1 mV/g)

Frequency Range:	Amplitude (±10%):	0.3 Hz to 6 kHz
	Phase (± 5°):	2 Hz to 5 kHz
Mounted Resonance Frequency:		18 kHz
Transverse Sensitivity [2]: Maximum (at 30 Hz, 100 ms^{-2}):		< 5% re Reference Sensitivity
Transverse Resonance Frequency:		> 18 kHz
Calculated values for TEDS [3]:	Resonance frequency:	19.8 kHz
	Quality factor @ f_{res}:	91.0
	Amplitude slope:	-2.2%/decade
	High pass cut-off frequency:	0.016 Hz
	Low pass cut-off frequency:	147 kHz
Measuring Range:		± 700 ms^{-2} peak (± 71 g peak)

Polarity of the electrical signal is positive for an acceleration in the direction of the arrow on the drawing.

Electrical:

Bias Voltage:	at full temperature and current range:	+ 13 V ± 1 V
Power Supply requirements:	Constant Current:	+ 2 to + 20 mA
	Unloaded Supply Voltage:	+ 24 V to + 30 V
Output Impedance:		< 30 Ω
Start-up time (to final bias ± 10%):		5 s
Inherent Noise (RMS): Broadband (1 Hz to 6 kHz):		< 35 μV corresponding to < 0.0035 ms^{-2} (< 350 μg)
Spectral:	10 Hz:	1.5x10^{-4} ms^{-2}/√Hz (15 μg/√Hz)
	100 Hz:	3.5x10^{-5} ms^{-2}/√Hz (3.5 μg/√Hz)
	1000 Hz:	2x10^{-5} ms^{-2}/√Hz (2 μg/√Hz)

Ground Loops can introduce error signals. These can be avoided by insulating the accelerometer from the mounting surface (see Mounting Technique).

Recommended cables: AO 1382, AO 0531, AO 0463 and other cables see Product Data Sheet

Built-in ID-information according to IEEE P1451.4

Environmental:

Temperature Range:	- 54 to + 121°C (- 65 to + 250°F)
Temperature Coefficient of Sensitivity:	+ 0.09%/°C
Temp. Transient Sensitivity (3 Hz Low. Lim. Frq. (-3 dB, 6 dB/oct)):	0.2 ms^{-2}/°C
Magnetic Sensitivity (50 Hz, 0.038 T):	3 ms^{-2}/T
Base Strain Sensitivity (at 250 με in base plane): Mounted on adhesive tape 0.09 mm thick:	0.005 ms^{-2}/με
Max. Non-destructive Shock:	50 kms^{-2} peak (5000 g peak)
Humidity:	90 % RH non-condensing

Mechanical:

Case Material:	Titanium ASTM Grade 2
Sensing Element:	Piezoelectric, Type PZ 23
Construction:	Theta Shear®
Sealing:	Welded
Weight:	4.8 gram (0.17 oz)
Electrical Connector:	10 - 32 UNF-2A
Mounting Surface Flatness:	< 3 μm

[1] This calibration is obtained on a modified Brüel & Kjær Calibration System Type 9610 System No.: 150157.3 and is traceable to the National Institute of Standards and Technology, USA and Physikalisch-Technische Bundesanstalt, Germany.
The expanded uncertainty is 1.0% determined in accordance with EAL-R2. A coverage factor k=2 is used. This corresponds to a coverage probability of 95% for a normal distribution.

[2] The uncertainty is 0.3% of Reference Sensitivity.

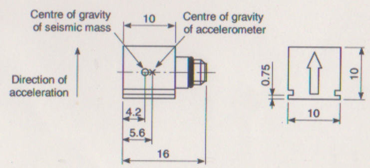
[3] Transducer Electronic Data Sheet according to IEEE P1451.4.

[4] Deviation from Reference Sensitivity.

Patents involved: US 08387851, JP 50952694 and DK 169653.

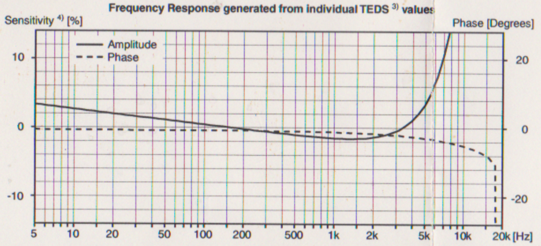
For further information, please see http://www.bk.dk and Product Data Sheet BP 1841.

Mounting Technique:
The accelerometer can be fastened directly to the measuring object by glue e.g., hot glue. However, if a reduced frequency range can be accepted, it is recommended to use one of the special mounting clips (see below) which is glued to the measuring object. In any case the mounting surface must be clean and smooth.
Three types of mounting clips are available: UA 1407 (set of 100) is a low profile clip recommended for mounting on plane surfaces. UA 1475 (set of 100) is a clip with a thick base which can be filed to fit a curved mounting surface. UA 1478 (set of 100) is a swivel base clip for use where the accelerometer is to be aligned according to a given co-ordinate system (see Product Data Sheet BP 1841).
Applying a little grease to the mounting surface of the accelerometer as well as the clip will improve the frequency response.
See also ISO 5348.

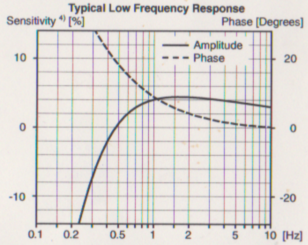


All dimensions in millimetres

Frequency Response generated from individual TEDS [3] values



Typical Low Frequency Response



Date 02 jun 2007, 13:24 Operator JHL

Specifications obtained in accordance with ANSI S2.11-1969 and parts of ISO 5347.

All values are typical at 25°C (77°F) unless measurement uncertainty is specified.

BC 0288-12

Serial No.: 30171

Lampiran 2: Spesifikasi modul NI 9234

Lampiran 3: *Script* Matlab pengambilan data pada akusisi.

```
%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;        %durasi perekaman
Dur = s.DurationInSeconds;
s.Rate = 17066;                  %sampling rate Hz
s.addAnalogInputChannel('cDAQ1Mod1', 'ai0', 'Accelerometer');
s.addAnalogInputChannel('cDAQ1Mod1', 'ai1', 'Accelerometer');

s.Channels(1).Sensitivity = 100.1E-3; %V/g Type 4507B serial:30171
s.Channels(2).Sensitivity = 97.60E-3; %V/g Type 4507B serial:11026
```

```
for i=1:30                      %  jumlah file yang diinginkan

data = s.startForeground();           % start recording vibration
data
data_ch1 = data(:,1);
data_ch2 = data(:,2);

rootname = 'E:\MATLAB\impellercavition\rpm_1000\fullv\';      %
drive tujuan dan nama file disesuaikan dengan variasi kecepatan
dan bukaan katup
extension = '.mat';                                           %
ekstension utk nama file
namafile = [rootname,'pump',num2str(i),extension];
data_all = [data_ch1 data_ch2];
eval(['save ', namafile ,' data_all']);

pause(5)
pesan = ['Acquiring and saving data at loop number: ',num2str(i)];
disp(pesan)
end

toc
```

Lampiran 4: Contoh *script* pengolahan data mentah menjadi plot domain waktu

```
clear
clc

%Direktori tempat data mentah getaran berada
load('E:\MATLAB\impellercavition\rpm_1000\quarterv\pump15.mat')
y1=data_all(:,1); %diberi nama dengan variabel baru dan berbeda
untuk masing-masing variasi bukaan katup
load('E:\MATLAB\impellercavition\rpm_1200\quarterv\pump15.mat')
y2=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_1400\quarterv\pump15.mat')
y3=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_1600\quarterv\pump15.mat')
y4=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_1800\quarterv\pump15.mat')
y5=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_2000\quarterv\pump15.mat')
y6=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_2200\quarterv\pump15.mat')
y7=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_2400\quarterv\pump15.mat')
y8=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_2600B\quarterv\pump15.mat')
y9=data_all(:,1);
```

```
% plot amplitude time domain
figure
subplot(9,1,1)
plot(y1(1:170660))
axis([0 9000 -3 3])
legend('1000rpm')
subplot(9,1,2)
plot(y2(1:170660),'r')
axis([0 9000 -3 3])
legend('1200rpm')
subplot(9,1,3)
plot(y3(1:170660),'g')
axis([0 9000 -3 3])
legend('1400rpm')
subplot(9,1,4)
plot(y4(1:170660),'b')
axis([0 9000 -3 3])
legend('1600rpm')
subplot(9,1,5)
plot(y5(1:170660),'c')
axis([0 9000 -3 3])
legend('1800rpm')
ylabel('Amplitudo Getaran (mV)')
subplot(9,1,6)
plot(y6(1:170660),'m')
axis([0 9000 -3 3])
legend('2000rpm')
subplot(9,1,7)
plot(y7(1:170660),'y')
axis([0 9000 -3 3])
legend('2200rpm')
subplot(9,1,8)
plot(y8(1:170660),'k')
axis([0 9000 -3 3])
legend('2400rpm')
subplot(9,1,9)
plot(y9(1:170660),'Color',[0,0.4,0.6])
axis([0 9000 -3.5 3.5])
legend('2600rpm')
xlabel('Sampel')
```

Lampiran 5: Contoh *script* fungsi pengolahan data dalam parameter PDF

```
function [pd1,pd2,pd3,pd4,pd5,pd6,pd7,pd8,pd9] =
createFit(y1,y2,y3,y4,y5,y6,y7,y8,y9)
%CREATEFIT    Create plot of datasets and fits
%   [PD1,PD2,PD3,PD4,PD5,PD6,PD7,PD8,PD9] =
CREATEFIT(Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9)
%   Creates a plot, similar to the plot in the main distribution
fitting
%   window, using the data that you provide as input.  You can
%   apply this function to the same data you used with dfittool
```

```
%   or with different data.  You may want to edit the function to
%   customize the code and this help message.
%
%   Number of datasets:  9
%   Number of fits:  9
%
%   See also FITDIST.

% This function was automatically generated on 19-Jul-2017
18:43:38

% Output fitted probablility distributions:
PD1,PD2,PD3,PD4,PD5,PD6,PD7,PD8,PD9

% Data from dataset "y1 data":
%    Y = y1

% Data from dataset "y2 data":
%    Y = y2

% Data from dataset "y3 data":
%    Y = y3

% Data from dataset "y4 data":
%    Y = y4

% Data from dataset "y5 data":
%    Y = y5

% Data from dataset "y6 data":
%    Y = y6

% Data from dataset "y7 data":
%    Y = y7

% Data from dataset "y8 data":
%    Y = y8

% Data from dataset "y9 data":
%    Y = y9

% input variabel berupa data mentah dari domain waktu
load('E:\MATLAB\impellercavition\rpm_1000\halfv\pump15.mat')
y1=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_1200\halfv\pump15.mat')
y2=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_1400\halfv\pump15.mat')
y3=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_1600\halfv\pump15.mat')
y4=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_1800\halfv\pump15.mat')
y5=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_2000\halfv\pump15.mat')
y6=data_all(:,1);
```

```
load('E:\MATLAB\impellercavition\rpm_2200\halfv\pump15.mat')
y7=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_2400\halfv\pump15.mat')
y8=data_all(:,1);
load('E:\MATLAB\impellercavition\rpm_2600B\halfv\pump15.mat')
y9=data_all(:,1);

% Prepare figure
clf;
hold on;


% --- Plot data originally in dataset "y1 data"
% This dataset does not appear on the plot

% --- Plot data originally in dataset "y2 data"
% This dataset does not appear on the plot

% --- Plot data originally in dataset "y3 data"
% This dataset does not appear on the plot

% --- Plot data originally in dataset "y4 data"
% This dataset does not appear on the plot

% --- Plot data originally in dataset "y5 data"
% This dataset does not appear on the plot

% --- Plot data originally in dataset "y6 data"
% This dataset does not appear on the plot

% --- Plot data originally in dataset "y7 data"
% This dataset does not appear on the plot

% --- Plot data originally in dataset "y8 data"
% This dataset does not appear on the plot

% --- Plot data originally in dataset "y9 data"
% This dataset does not appear on the plot

% Get data limits to determine plotting range
XLim = [min(y1), max(y1)];
XLim = [min(y1), max(y1)];
XLim(1) = min(XLim(1), min(y2));
XLim(2) = max(XLim(2), max(y2));
XLim(1) = min(XLim(1), min(y3));
XLim(2) = max(XLim(2), max(y3));
XLim(1) = min(XLim(1), min(y4));
XLim(2) = max(XLim(2), max(y4));
XLim(1) = min(XLim(1), min(y5));
XLim(2) = max(XLim(2), max(y5));
XLim(1) = min(XLim(1), min(y6));
XLim(2) = max(XLim(2), max(y6));
XLim(1) = min(XLim(1), min(y7));
XLim(2) = max(XLim(2), max(y7));
```

```
XLim(1) = min(XLim(1), min(y8));
XLim(2) = max(XLim(2), max(y8));
XLim(1) = min(XLim(1), min(y9));
XLim(2) = max(XLim(2), max(y9));

% Create grid where function will be computed
XLim = XLim + [-1 1] * 0.01 * diff(XLim);
XGrid = linspace(XLim(1),XLim(2),1000);


% --- Create fit "1000rpm"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%     pd1 = ProbDistUnivParam('normal',[ 0.0004661917746911,
0.0785099953887])
pd1 = fitdist(y1, 'normal');
YPlot = pdf(pd1,XGrid);
hLine = plot(XGrid,YPlot,'Color',[1 0 0],...
    'LineStyle','-', 'LineWidth',1,...
    'Marker','none', 'MarkerSize',6);

% --- Create fit "1200rpm"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%     pd2 = ProbDistUnivParam('normal',[ 0.000145416383953,
0.1015119626129])
pd2 = fitdist(y2, 'normal');
YPlot = pdf(pd2,XGrid);
hLine = plot(XGrid,YPlot,'Color',[0 0 1],...
    'LineStyle','-', 'LineWidth',1,...
    'Marker','none', 'MarkerSize',6);

% --- Create fit "1400rpm"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%     pd3 = ProbDistUnivParam('normal',[ 0.0001104482801489,
0.1325186411885])
pd3 = fitdist(y3, 'normal');
YPlot = pdf(pd3,XGrid);
hLine = plot(XGrid,YPlot,'Color',[0.666667 0.333333 0],...
    'LineStyle','-', 'LineWidth',1,...
    'Marker','none', 'MarkerSize',6);

% --- Create fit "1600rpm"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%     pd4 = ProbDistUnivParam('normal',[ 0.0001804526655009,
0.1571012041355])
pd4 = fitdist(y4, 'normal');
YPlot = pdf(pd4,XGrid);
```

```
hLine = plot(XGrid,YPlot,'Color',[0.333333 0.333333 0.333333],...
    'LineStyle','-', 'LineWidth',1,...
    'Marker','none', 'MarkerSize',6);

% --- Create fit "1800rpm"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%     pd5 = ProbDistUnivParam('normal',[ 0.0001591450291158,
0.2289871180555])
pd5 = fitdist(y5, 'normal');
YPlot = pdf(pd5,XGrid);
hLine = plot(XGrid,YPlot,'Color',[1 0 1],...
    'LineStyle','-', 'LineWidth',1,...
    'Marker','none', 'MarkerSize',6);

% --- Create fit "2000rpm"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%     pd6 = ProbDistUnivParam('normal',[ 0.0002510872034025,
0.3939529028838])
pd6 = fitdist(y6, 'normal');
YPlot = pdf(pd6,XGrid);
hLine = plot(XGrid,YPlot,'Color',[1 1 0],...
    'LineStyle','-', 'LineWidth',1,...
    'Marker','none', 'MarkerSize',6);

% --- Create fit "2200rpm"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%     pd7 = ProbDistUnivParam('normal',[ -6.941889467918e-06,
0.5780293513971])
pd7 = fitdist(y7, 'normal');
YPlot = pdf(pd7,XGrid);
hLine = plot(XGrid,YPlot,'Color',[1 0.666667 0.333333],...
    'LineStyle','-', 'LineWidth',1,...
    'Marker','none', 'MarkerSize',6);

% --- Create fit "2400rpm"

% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%     pd8 = ProbDistUnivParam('normal',[ 0.0001502678006569,
0.7806753657395])
pd8 = fitdist(y8, 'normal');
YPlot = pdf(pd8,XGrid);
hLine = plot(XGrid,YPlot,'Color',[0.666667 0.666667 0.666667],...
    'LineStyle','-', 'LineWidth',1,...
    'Marker','none', 'MarkerSize',6);

% --- Create fit "2600rpm"
```

```
% Fit this distribution to get parameter values
% To use parameter estimates from the original fit:
%     pd9 = ProbDistUnivParam('normal',[ -0.0003399840186455,
0.936135285761])
pd9 = fitdist(y9, 'normal');
YPlot = pdf(pd9,XGrid);
hLine = plot(XGrid,YPlot,'Color',[0.666667 0.333333 1],...
    'LineStyle','-', 'LineWidth',1,...
    'Marker','none', 'MarkerSize',6);

% Adjust figure
box on;
grid on;
axis([-4 4 0 5.5])
legend('\fontsize{18} 1000rpm','\fontsize{18} 1200rpm',...
    '\fontsize{18} 1400rpm','\fontsize{18} 1600rpm','\fontsize{18}
1800rpm',...
    '\fontsize{18} 2000rpm','\fontsize{18} 2200rpm','\fontsize{18}
2400rpm',...
    '\fontsize{18} 2600rpm')
xlabel('Amplitudo Getaran (mV)','fontsize',20);
ylabel('Nilai dari PDF','fontsize',20)
set(gca,'fontsize',18)
hold off;
end
```

Lampiran 6: Contoh *script* pengolahan data mentah menjadi data statistik domain waktu.

```
clc
close all
clear
for d=1:30

signal_in=['E:\MATLAB\impellercavition\rpm_1000\fullv\pump',int2st
r(d),'.mat'];
load (signal_in)
a=data_all(:,1);
Rs1(d)=rms(a);                          %RMS
Sd1(d)=std(a);                          %Standar Deviasi
Pv1(d)=((max(abs(a))-min(abs(a)))/2);   %Peak Value
K1(d)=kurtosis(a)-3;                    %kurtosis
V1(d)=var(a);                           %varians
Cf1(d)=peak2rms(a);                     %crest faktor
end

%direktori tempat hasil pengolahan data disimpan
save('E:\MATLAB\STATIS\full\Rs1.mat')
save('E:\MATLAB\STATIS\full\Sd1.mat')
save('E:\MATLAB\STATIS\full\Pv1.mat')
save('E:\MATLAB\STATIS\full\K1.mat')
```

```
save('E:\MATLAB\STATIS\full\V1.mat')
save('E:\MATLAB\STATIS\full\Cf1.mat')
```

Lampiran 7: Contoh *script plotting* data statistik bentuk distribusi terhadap set data

```
%x sama dengan jumlah set data
x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30];
figure
load('E:\MATLAB\STATIS\Full\Cf1.mat')
load('E:\MATLAB\STATIS\Full\Cf2.mat')
load('E:\MATLAB\STATIS\Full\Cf3.mat')
load('E:\MATLAB\STATIS\Full\Cf4.mat')
load('E:\MATLAB\STATIS\Full\Cf5.mat')
load('E:\MATLAB\STATIS\Full\Cf6.mat')
load('E:\MATLAB\STATIS\Full\Cf7.mat')
load('E:\MATLAB\STATIS\Full\Cf8.mat')
load('E:\MATLAB\STATIS\Full\Cf9.mat')
hold on
scatter(x,Cf1,'+b');
hold on
scatter(x,Cf2,'ob')
hold on
scatter(x,Cf3,'*b')
hold on
scatter(x,Cf4,'sb')
hold on
scatter(x,Cf5,'db')
hold on
scatter(x,Cf6,'^b')
hold on
scatter(x,Cf7,'pb')
hold on
scatter(x,Cf8,'hb')
hold on
scatter(x,Cf9,'>b')
%script kemudian diulang setiap variasi bukaan katup
```

Lampiran 8: Contoh *script plotting* grafik hubungan antara nilai statistik terhadap fungsi kecepatan

```
figure
yfK=fK;     %nilai fK, hK, qK didapat dengan merata-ratakan nilai
paramter dari ke 2-29 set data tanpa error
yhK=hK;
yqK=qK;
x=[1000 1200 1400 1600 1800 2000 2200 2400 2600];
fitx=linspace(1000,2600,1000000);
```

```
fityfK=interp1(x,yfK,fitx,'spline');
fityhK=interp1(x,yhK,fitx,'spline');        %spline digunakan agar
sumbu grafik menjadi lebih halus
fityqK=interp1(x,yqK,fitx,'spline');
scatter(x,yfK,100,'db')
hold on
scatter(x,yhK,100,'sr')
hold on
scatter(x,yqK,100,'^c')
hold on
line(fitx,fityfK,'color','b')
hold on
line(fitx,fityhK,'color','r')
hold on
line(fitx,fityqK,'color','c')
grid on
legend('\fontsize{18} Katup bukaan penuh','\fontsize{18} Katup
bukaan setengah','\fontsize{18} Katup bukaan tiga per empat')
ylabel('Nilai Kurtosis Getaran','FontSize',20)
xlabel('Variasi Kecepatan (rpm)','FontSize',20)
set(gca,'fontsize',18)
```

Lampiran 9: Contoh *script* FFT

```
clear
clc
close

sampl_rate=17066; %kecepatan sampling Hz
recording_time=10;   %waktu perekaman data (recording time)
L=sampl_rate*recording_time;  %panjang data (length of signal)
NFFT=2^nextpow2(L);

%%%%%%% BUKAAN KATUP PENUH CODE 11
load('E:\MATLAB\impellercavition\rpm_1000\fullv\pump8.mat')
y11=data_all(:,1);
Y11=fft(y11,NFFT)/L;
f11=sampl_rate/2*linspace(0,1,NFFT/2+1);

load('E:\MATLAB\impellercavition\rpm_1200\fullv\pump8.mat')
y12=data_all(:,1);
Y12=fft(y12,NFFT)/L;
f12=sampl_rate/2*linspace(0,1,NFFT/2+1);

load('E:\MATLAB\impellercavition\rpm_1400\fullv\pump8.mat')
y13=data_all(:,1);
Y13=fft(y13,NFFT)/L;
f13=sampl_rate/2*linspace(0,1,NFFT/2+1);

load('E:\MATLAB\impellercavition\rpm_1600\fullv\pump4.mat')
y14=data_all(:,1);
Y14=fft(y14,NFFT)/L;
```

```
f14=sampl_rate/2*linspace(0,1,NFFT/2+1);

load('E:\MATLAB\impellercavition\rpm_1800\fullv\pump8.mat')
y15=data_all(:,1);
Y15=fft(y15,NFFT)/L;
f15=sampl_rate/2*linspace(0,1,NFFT/2+1);

figure
subplot(4,1,4);
plot (f14,2*abs(Y14(1:NFFT/2+1)))
axis([10 150 0 0.07]);
legend('\fontsize{10} 1600rpm')
set(gca,'fontsize',13)
xlabel('Frekuensi (Hz)')
subplot(4,1,3);
plot (f13,2*abs(Y13(1:NFFT/2+1)))
axis([10 150 0 0.07]);
legend('\fontsize{10} 1400rpm')
set(gca,'fontsize',13)
ylabel('Amplitude (mV)')
subplot(4,1,2);
plot (f12,2*abs(Y12(1:NFFT/2+1)))
axis([10 150 0 0.07]);
legend('\fontsize{10} 1200rpm')
set(gca,'fontsize',13)
subplot(4,1,1);
plot (f11,2*abs(Y11(1:NFFT/2+1)))
axis([10 150 0 0.07]);
legend('\fontsize{10} 1000rpm')
set(gca,'fontsize',13)
```