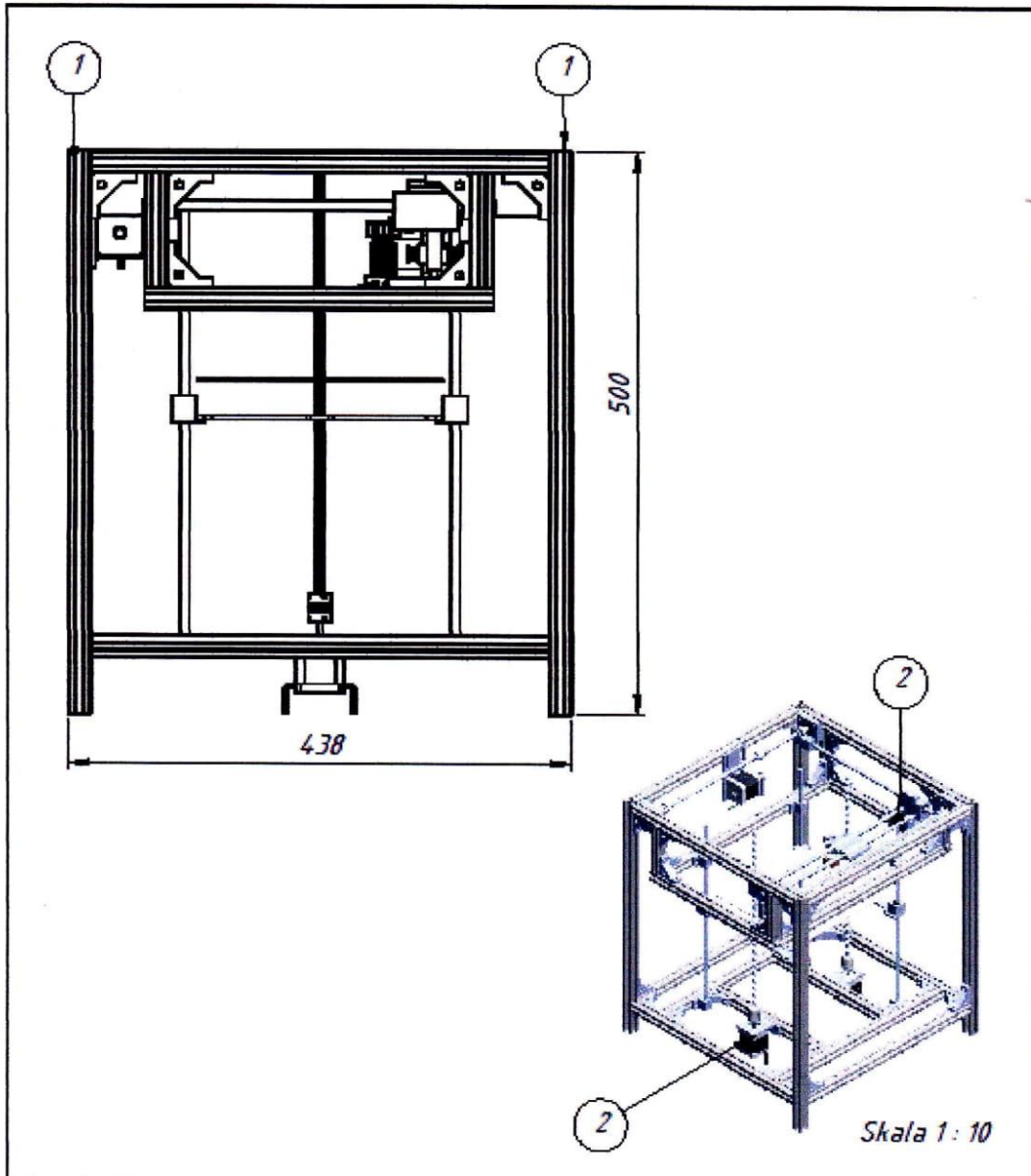


LAMPIRAN



		4	Motor Stepper	2	Asembling	50x20x20	Nema 23	
		4	Tiang Rangka	1	Aluminium	500x20x20	Profil	
Jumlah			Nama Bagian	No.Bag	Bahan	Ukuran	Keterangan	
III	II	I	Perubahan					
PRINTER 3-D FDM CORE XY						Skala	Digambar	Tim
						1:5	Diperiksa	IM/Az
POLITEKNIK NEGERI SRIWIJAYA						DRAW. ME/01/2019		

SCRIPT PROGRAM FFT MATLAB

```
% Untuk menganalisis sampling dan menganalisis FFT
clear; clc; clf;
[D,T,R] = xlsread('Data 1 (X).xlsx');
RD = str2double(D);
Signal = D(:,1);
Ts = 500; % Sampling
Interval (milliseconds) %
Ts = Ts*1E-3; %
Sampling Interval (seconds) %
Fs = 1/Ts; %
Sampling Frequency (Hz) %
Fn = Fs/2; %
Nyquist Frequency (Hz) %
N = length(Signal);
meanSignal = mean(Signal); %
?Signal? Mean %
FTSignal = fft(Signal-meanSignal)/N; %
Normalised Fourier Transform Of Baseline-Corrected ?Signal? %
Fv = linspace(0, 1, fix(numel(FTSignal)/2)+1)*Fn; %
Frequency Vector %
Iv = 1:numel(Fv); % Index
Vector
[pks,locs] = findpeaks(abs(FTSignal(Iv))*2, 'MinPeakHeight',0.0);
%

figure
plot(Fv, abs(FTSignal(Iv))*2)
grid
xlabel('Frequency (Hz)')
ylabel('Amplitude')
title('Data Pengujian 12')
plotIdx = 1:Iv(max(locs));
```

PROGRAM ARDUINO IDE

```
#include <Adafruit_ADXL345_U.h>
#include <EEPROM.h>
#include <SoftwareSerial.h>
#include <Wire.h> // I2C Library
// ADXL345
#define ACCELEROMETER (0x53) // Accelerometer Address
#define ACC_READ (6) // 6 Bytes are going to be read for x, y and z axis (2 bytes
per each)
// ITG3200
#define POWER_MNG (0x3E) // *****
#define INTERRUPT_CONF (0x17) // Defined registers based on ITG3200 datasheet
#define SAMPLE_RATE (0x15) // See Page 22
#define FS_DLPF (0x16) // *****
void startACC()
{
    writeTo(ACCELEROMETER, 0x2D, 0); // Accelerometer is turned on, range is -2g
to 2g by default
    writeTo(ACCELEROMETER, 0x2D, 16);
    writeTo(ACCELEROMETER, 0x2D, 8);
}
void getACC(float * data)
{
    int register_1st = (0x32); // First register of all axes
```

```

// float scale_cons = 0.0039; // Constant used for converting data into scaled
float scale_cons = 0.039; // Constant used for converting data into scaled
// Read the accelerometer data 6 bytes starting from the register_1st
readFrom(ACCELEROMETER, register_1st, ACC_READ, buffer);
data[0] = (((buffer[1] << 8) | buffer[0]) + 2) * scale_cons; // Data comes 10 bits
data[1] = (((buffer[3] << 8) | buffer[2]) - 244.999) * scale_cons; // Shifting is
performed
data[2] = (((buffer[5] << 8) | buffer[4]) + 29) * scale_cons; // Conversion from
bytes to integer
data[3] = sqrt(sq(data[0]) + sq(data[1]) + sq(data[2])); // Data is averaged
}
// Writing value to register in ADXL345
void writeTo(int SENSOR, byte REGADD, byte VALUE)
{
Wire.beginTransmission(SENSOR); // Beginning transmission
Wire.write(REGADD); // Sending register address
Wire.write(VALUE); // Sending value to write
Wire.endTransmission(); // Ending the transmission
}
// Reading the bytes starting from the first register of the axes
void readFrom(int SENSOR, byte INITREG, int BYTES, byte BUFFER[])
{
Wire.beginTransmission(SENSOR); // Beginning transmission
Wire.write(INITREG); // Sending the address to be read
Wire.endTransmission(); // Ending the transmission
Wire.beginTransmission(SENSOR); // Beginning transmission
Wire.requestFrom(SENSOR, BYTES); // Requesting bytes

```

```

while (Wire.available())
{
  BUFFER[i] = Wire.read(); // Receive one byte
  i++; // When i increases
}
Wire.endTransmission(); // Ending transmission
}
void setup()
{
  Wire.begin(); // Initiating I2C
  Serial.begin(9600); // Open the serial monitor
  startACC(); // Calling the function
  Serial.println("CLEARDATA"); //Menggunakan PLX-DAQ
  Serial.println("CLEAR SHEET"); // clears starting at row 1
  Serial.println("LABEL,Time,Smb X,Smb Y, Smb Z, Resultan");
}
void loop()
{
  float acc_data[5];
  // float gyro_data[3];
  getACC(acc_data); // Calling the function
  Serial.println();
  Serial.print("DATA,TIME,"); Serial.print(acc_data[0]); Serial.print(",");
  Serial.print(acc_data[1]); Serial.print(","); Serial.print(acc_data[2]); Serial.print(",");
  Serial.print(acc_data[3]);
  //Serial.println();
  delay(100); // Delay is given

```

HASIL PENGUKURAN VIBRASI PADA FRAME *PRINTER 3D CORE XY*

Smb X	Smb Y	Smb Z	Resultan
-0,08	-0,08	-0,12	-0,09
0,04	0,16	-0,08	0,04
0,27	0,27	-0,51	0,01
-0,23	-0,04	0,66	0,13
0,27	0,2	-0,27	0,07
-0,08	-0,04	0,66	0,18
0,04	-0,08	-0,04	-0,03
-0,04	0,23	0,04	0,08
-0,12	0,31	0,08	0,09
-0,2	0,23	0,04	0,02
-0,35	-0,19	0,7	0,05
-0,31	-0,04	0,62	0,09
0,2	0,35	0,47	0,34
-0,04	-0,04	0,12	0,01
-0,04	0	0	-0,01
-0,04	0,04	0,2	0,07
-0,04	-0,08	0	-0,04
0	0,08	-0,27	-0,06
-0,12	0,27	0,12	0,09
0,12	0,04	0,12	0,09
0	0	-0,35	-0,12
0,27	-0,39	-0,27	-0,13
-0,08	-0,08	-0,12	-0,09
-0,04	0	0,51	0,16
-0,16	-0,27	0,27	-0,05
0,08	0,04	-0,04	0,03
0,31	0,47	-0,74	0,01
0,04	-0,08	-0,59	-0,21
-0,08	0	0,27	0,06
-0,08	0,55	0,31	0,26
-0,08	-0,08	0,23	0,02
0,16	0,04	-0,39	-0,06
-0,08	0,16	0,27	0,12
0,04	0,12	0,04	0,07
0,12	-0,12	-0,27	-0,09

0,04	-0,55	0	-0,17
0,04	-0,31	-0,66	-0,31
-0,16	-0,23	-0,16	-0,18
0,12	-0,19	0,23	0,05
-0,12	0,12	0	0,00
0,2	-0,04	0,04	0,07
-0,55	0,27	-0,39	-0,22
0,31	-0,39	-0,9	-0,33
0,08	0,55	-0,94	-0,10
-0,08	-0,04	0,47	0,12
-0,16	-0,43	0,31	-0,09
-0,04	-0,04	-0,08	-0,05
0,08	0,08	0,2	0,12
0,16	0	0,62	0,26
0,16	-0,08	0,27	0,12
0	0,04	0,82	0,29
-0,27	0,35	0,51	0,20
0,12	-0,23	0,66	0,18
0,31	0,31	-0,08	0,18
-0,16	0	0,31	0,05
0,2	0,12	-0,2	0,04
0,2	0,23	0,27	0,23
-0,16	-0,04	-0,27	-0,16
0	0,23	0,08	0,10
0	0,16	-0,35	-0,06
-0,16	0,12	-0,23	-0,09
0,2	0,2	0	0,13
0,04	-0,31	0,31	0,01
0	-0,12	0,16	0,01
-0,04	0,04	0,2	0,07
-0,16	0,2	-0,2	-0,05
-0,23	-0,27	0,94	0,15
-0,16	0,12	0,12	0,03
0,16	0,08	-0,27	-0,01
-0,04	0,2	-0,35	-0,06
0,27	-0,23	0,7	0,25
0,04	-0,08	0	-0,01

Handwritten signature
FOTO PENYANGKAN SENSOR ADXL345

