DAFTAR PUSTAKA

Agus, K. 2014. *Sekrap. https://kunaugust.wordpress.com/2014/05/24/sekrap/*. (diakses tanggal 30 Agustus 2020), 10.39 WIB

ASM Handbook vol.1:148 (1993), *Properties and Selection: Irons, Steels, and High Performance Alloys, ASM Handbook Committee, United State*.

Budi, S. (tanpa tahun). *Mesin Shaping. https://independent.academia.edu/jdlsetiabudi*. (diakses tanggal 29 Agustus 2020), 23.33 WIB

Daryus, Asyari, 2005. Proses Produksi II Universitas Darma Persada. Jakarta.

Hermawan. 1990. *Faktor yang Mempengaruhi Getaran Sebuah Mesin sekrap: Media Teknik Edisi No 1*. Yogyakarta: FT – UGM.

Imran S.A, 2014. *Pengembangan Sistem Monitoring Vibrasi Pada Kipas Pendingin Menggunakan Accelerometer Adxl345 Dengan Metode Fft Berbasis Labview.* Skripsi. Fakultas sains dan teknologi, Universitas Islam Syarif Hidayatullah

Kalpakjian, Serope. 1995. *Manufacturing Process For Engineering Material 3th Edition*. MC Graw-Hill. USA.

Mulyadi, S. 2012. "*Pengaruh kecepatan potong, gerak makan dan ketebalan pemotongan terhadap getaran benda kerja pada proses sekrap*". Jurnal ROTOR, Volume 5 Nomor 1.

Rochim, T. 1993. *Teori dan Teknologi Proses Permesinan*. Bandung: Teknik Mesin FTI- ITB.

Rohman, A.Z. 2015. "*Rancang Bangun Alat Ukur Getaran Mesinberbasis Arduino".* Skripsi. Teknik Elektro, Universitas Negeri Semarang.

Sukma, JA. 2012. Bab II Teori Dasar Baja Karbon. *http://eprints.undip.ac.id/41534/4/BAB_II.pdf* (diakses pada 2 September 2020), 10.16 WIB

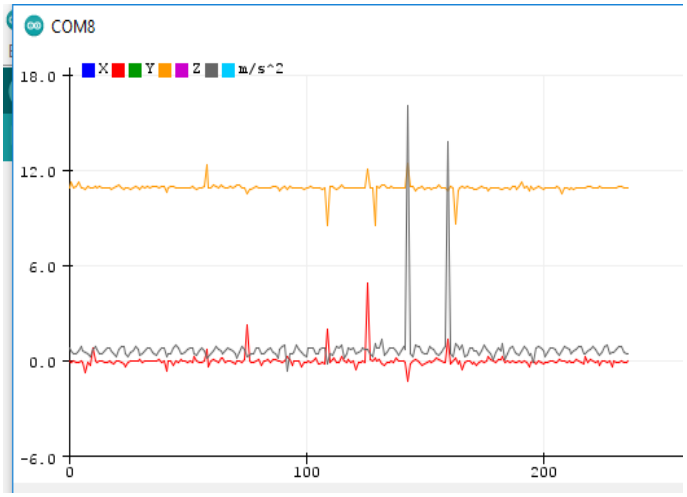
Widarto. 2008. *Teknik Pemesinan Buku Jilid 2 Untuk SMK*. Jakarta: Direktorat Pembinaan Sekolah Menengah Kejuruan.

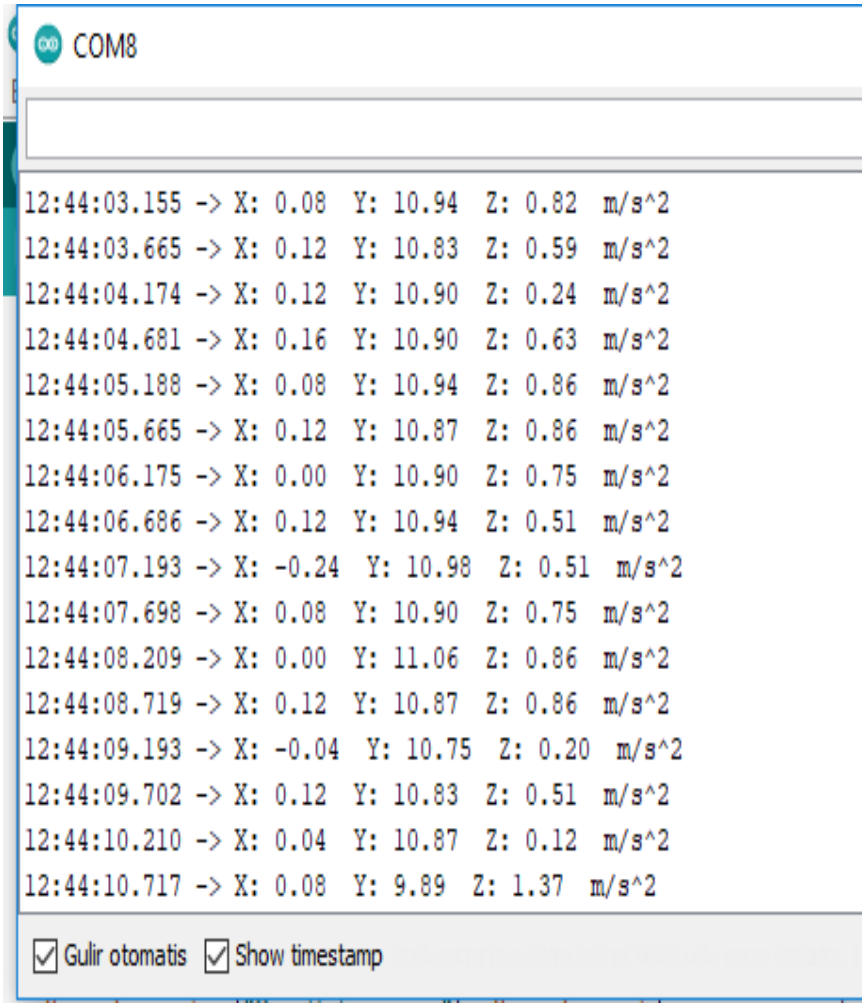
LAMPIRAN 1

Data Hasil Pengujian

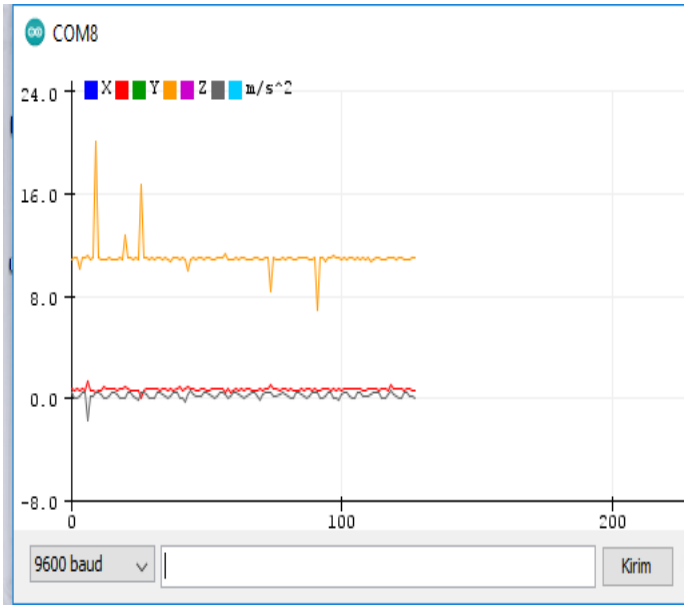
1. Pengujian dengan Kecepatan 1

a. Pengukuran Kedalaman 0,2 mm dengan gerak makan 5





b. Pengukuran Kedalaman 0,4 mm dengan gerak makan 8



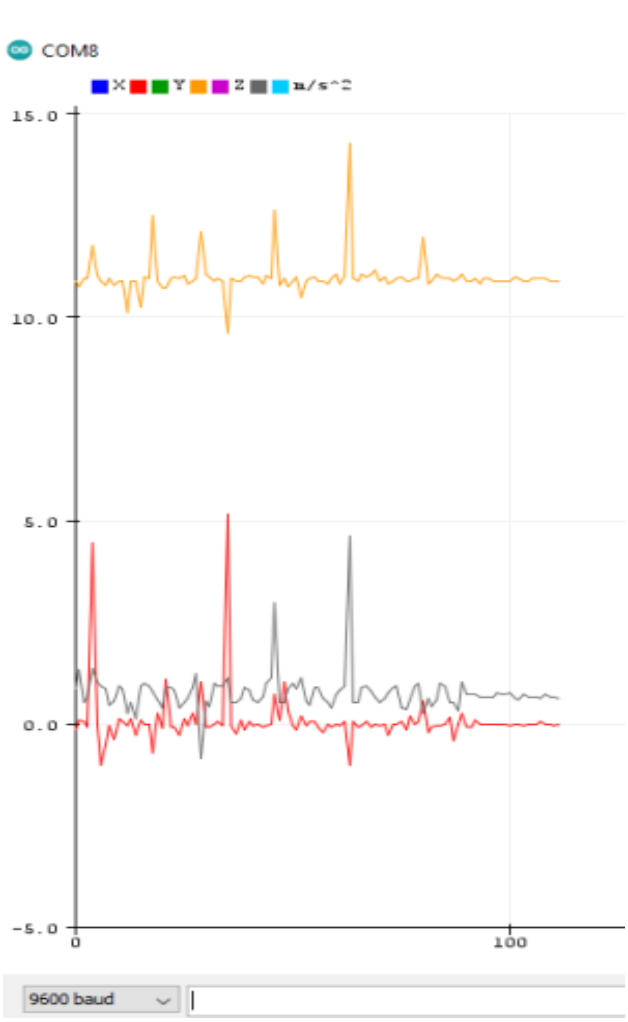
COM8

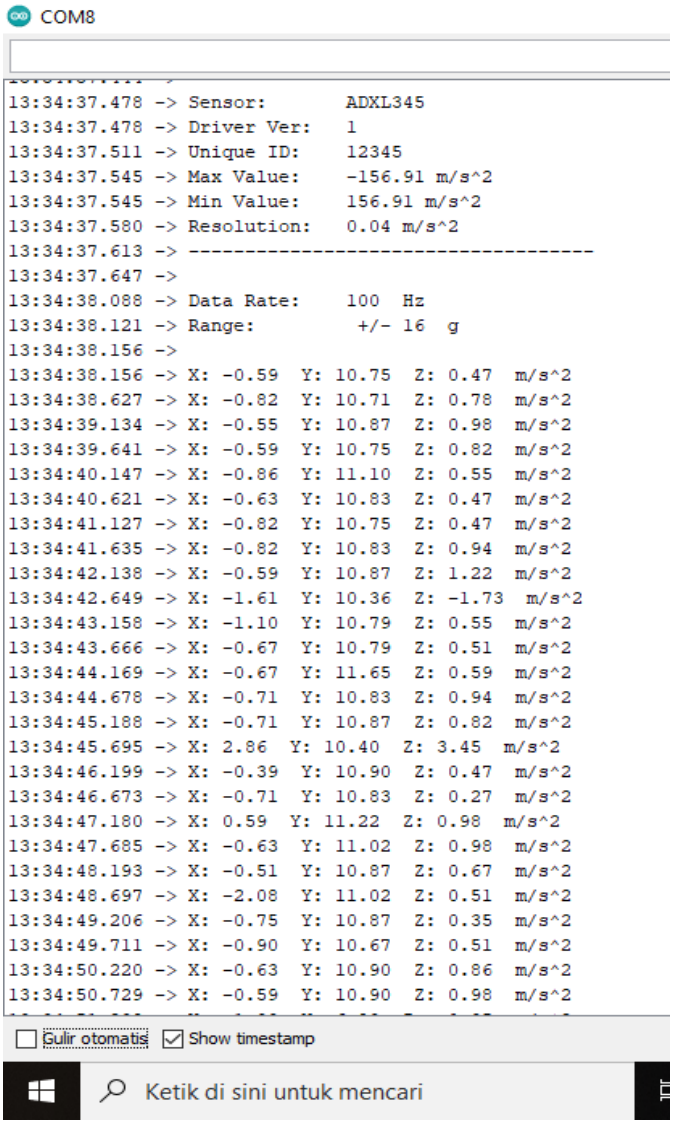
```
13:14:12.266 -> X: 0.20  Y: 10.90  Z: 0.86  m/s^2
13:14:12.777 -> X: -0.43  Y: 10.90  Z: -0.51  m/s^2
13:14:13.282 -> X: 0.12  Y: 11.02  Z: 0.59  m/s^2
13:14:13.791 -> X: -0.24  Y: 10.87  Z: 0.39  m/s^2
13:14:14.300 -> X: 0.20  Y: 11.41  Z: 0.63  m/s^2
13:14:14.775 -> X: 0.04  Y: 10.94  Z: 1.02  m/s^2
13:14:15.284 -> X: -0.04  Y: 10.94  Z: 1.02  m/s^2
13:14:15.793 -> X: -25.69  Y: 9.02  Z: 1.33  m/s^2
13:14:16.300 -> X: -0.16  Y: 10.98  Z: 0.59  m/s^2
13:14:16.807 -> X: 0.12  Y: 10.94  Z: 0.47  m/s^2
13:14:17.315 -> X: 0.04  Y: 10.98  Z: 0.86  m/s^2
13:14:17.823 -> X: 0.00  Y: 11.02  Z: 0.90  m/s^2
13:14:18.330 -> X: 0.12  Y: 10.98  Z: 0.86  m/s^2
13:14:18.838 -> X: 0.12  Y: 11.06  Z: 0.51  m/s^2
13:14:19.314 -> X: -0.04  Y: 10.90  Z: 0.39  m/s^2
13:14:19.826 -> X: 0.04  Y: 10.87  Z: 0.55  m/s^2
13:14:20.333 -> X: 0.04  Y: 10.94  Z: 0.82  m/s^2
13:14:20.845 -> X: -0.16  Y: 10.90  Z: 0.90  m/s^2
13:14:21.353 -> X: 0.82  Y: 7.34  Z: 0.59  m/s^2
13:14:21.860 -> X: 0.16  Y: 10.90  Z: 0.51  m/s^2
13:14:22.369 -> X: 0.31  Y: 10.90  Z: 0.39  m/s^2
13:14:22.848 -> X: 0.31  Y: 10.43  Z: 0.90  m/s^2
13:14:23.355 -> X: 0.08  Y: 10.90  Z: 0.94  m/s^2
13:14:23.862 -> X: 0.63  Y: 10.98  Z: 0.94  m/s^2
13:14:24.371 -> X: 0.08  Y: 10.43  Z: 0.20  m/s^2
13:14:24.878 -> X: 0.08  Y: 10.90  Z: 0.59  m/s^2
13:14:25.386 -> X: 0.16  Y: 10.79  Z: 0.47  m/s^2
13:14:25.892 -> X: 0.27  Y: 10.90  Z: 0.86  m/s^2
13:14:26.401 -> X: 0.16  Y: 10.98  Z: 0.98  m/s^2
13:14:26.877 -> X: 0.08  Y: 10.83  Z: 0.82  m/s^2
13:14:27.383 -> X: -0.35  Y: 11.02  Z: 0.47  m/s^2
13:14:27.892 -> X: 0.04  Y: 10.83  Z: 0.43  m/s^2
13:14:28.404 -> X: -0.31  Y: 10.87  Z: 0.67  m/s^2
13:14:28.907 -> X: -0.16  Y: 11.02  Z: 0.86  m/s^2
13:14:29.415 -> X: 0.04  Y: 10.94  Z: 0.82  m/s^2
13:14:29.921 -> X: 0.00  Y: 14.00  Z: 0.43  m/s^2
13:14:30.427 -> X: 0.08  Y: 10.94  Z: 0.39  m/s^2
```

Gulir otomatis Show timestamp

Ketik di sini untuk mencari

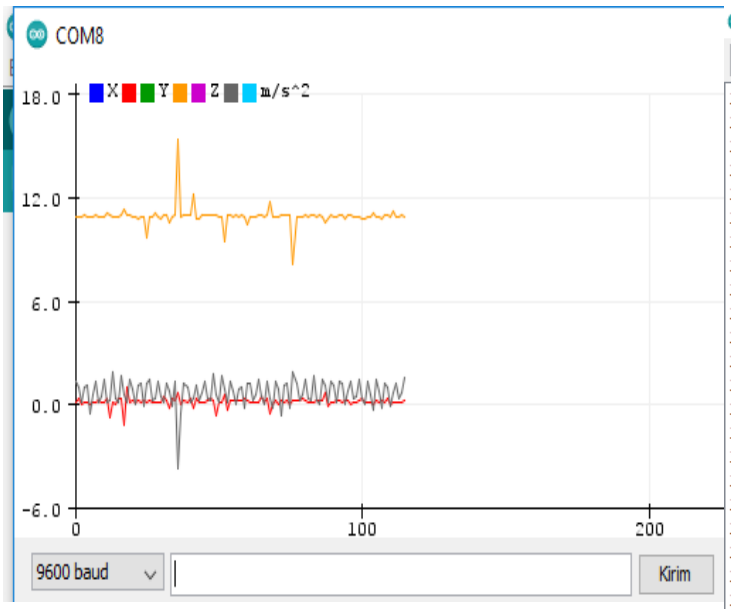
c. Pengukuran kedalaman 0,6 dengan gerak makan 10

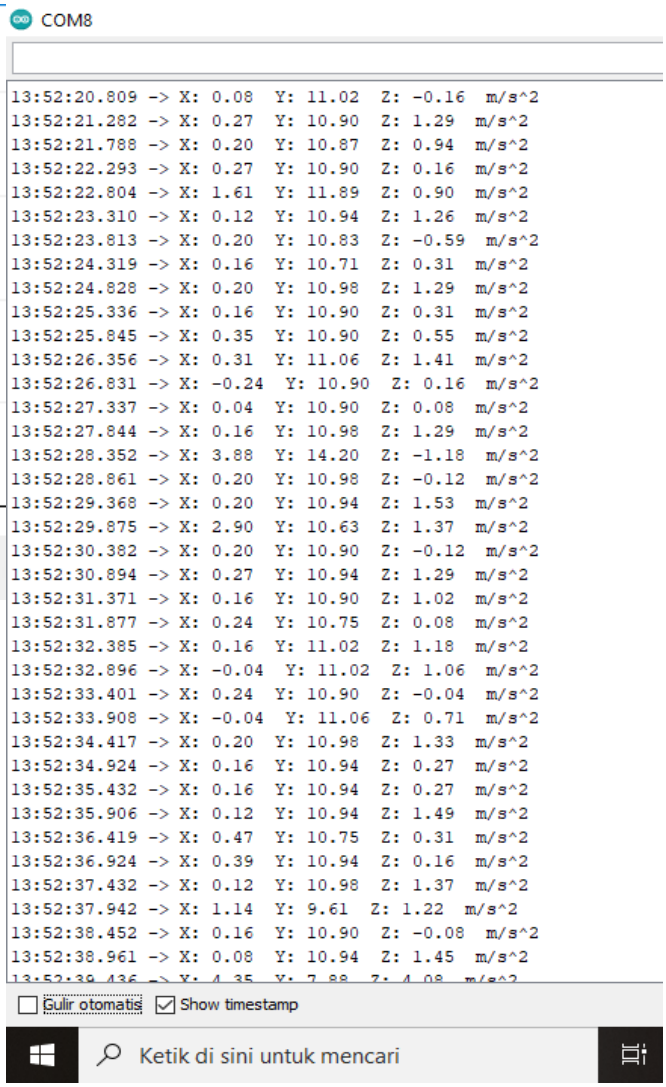




2. **Pengujian dengan Step Kecepatan 2**

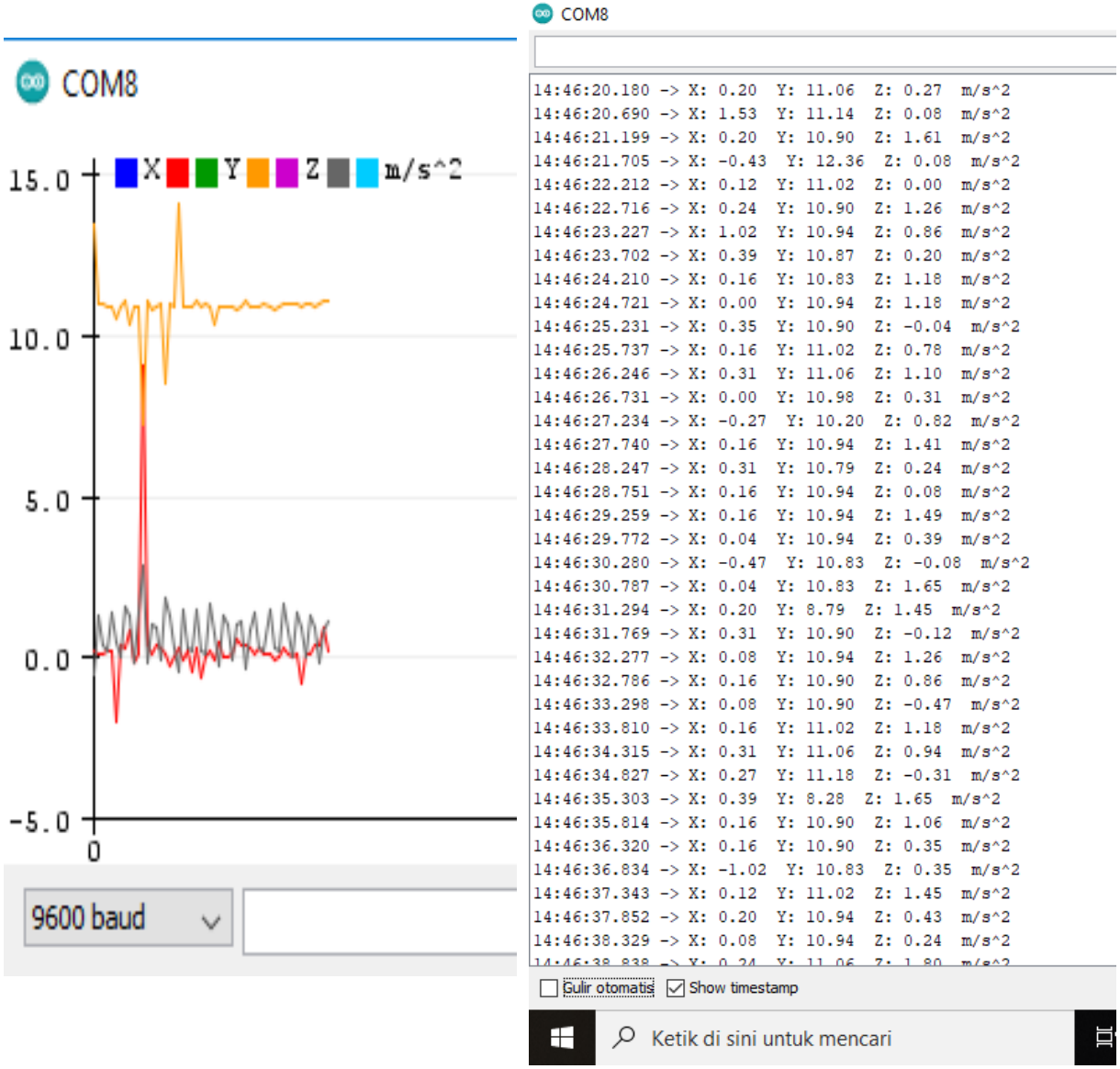
   a. **Pengukuran kedalaman 0,2 mm dengan gerak makan 5**





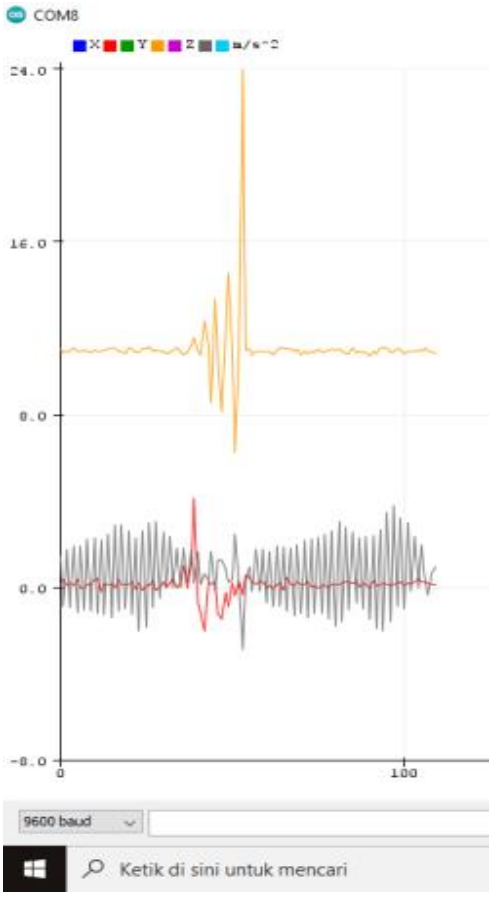
b. Pengukuran Kedalaman 0,4 mm dengan gerak makan 8

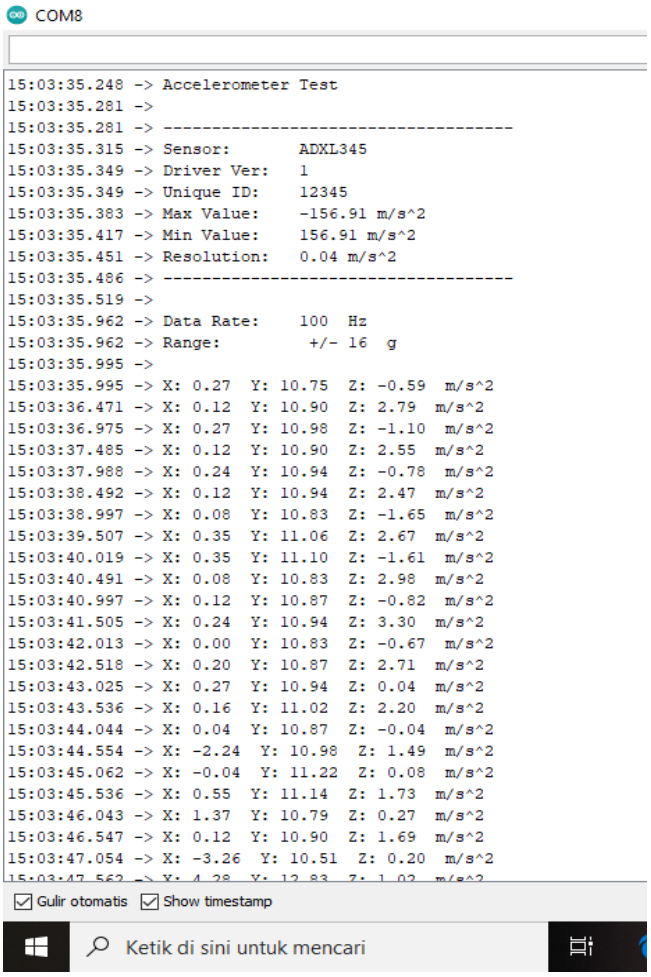
c. Pengukuran kedalaman 0,6 dengan gerak makan 10



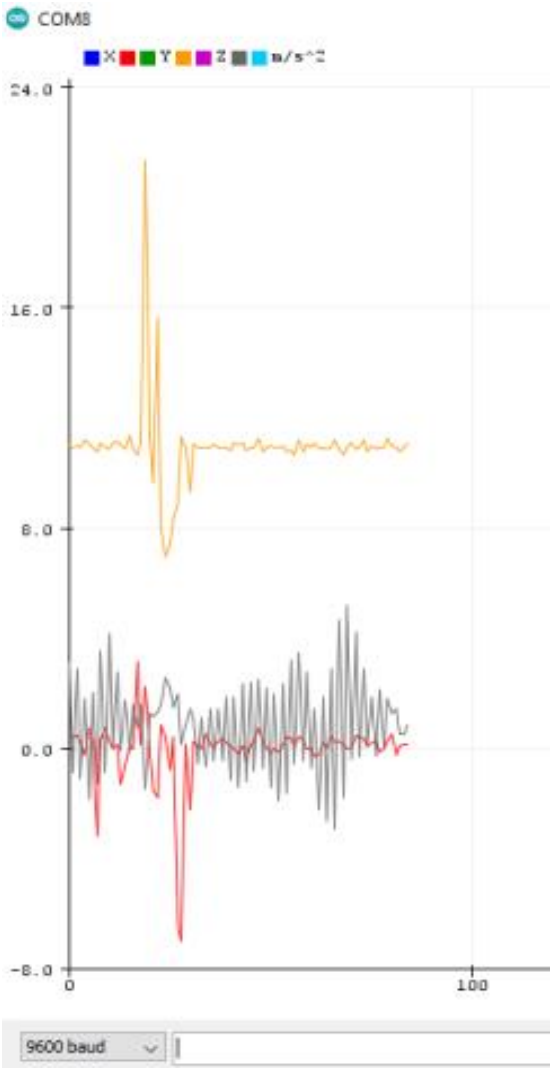
3. Pengujian dengan Step Kecepatan 3

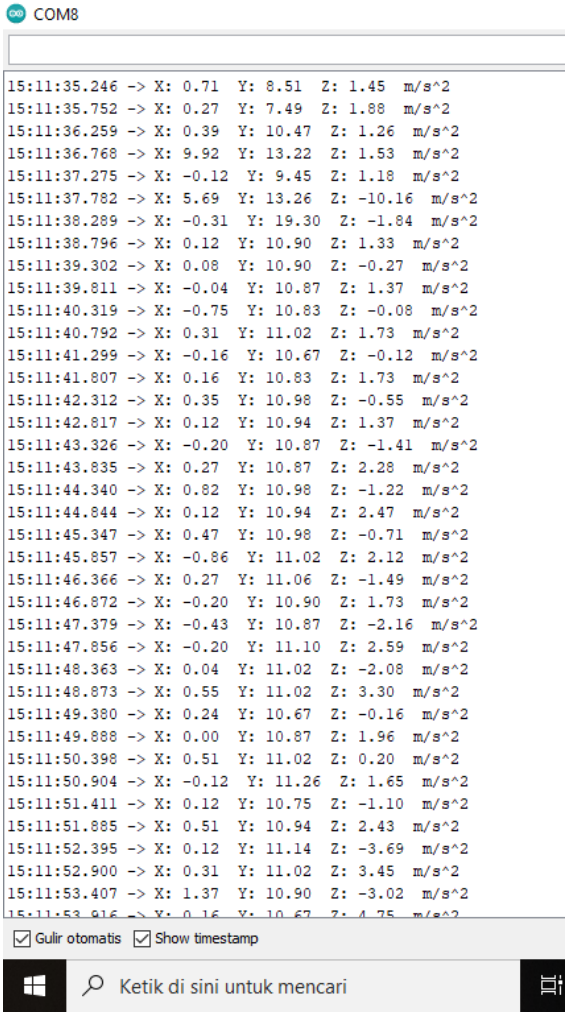
a. Pengukuran kedalaman 0,2 mm dengan gerak makan 5



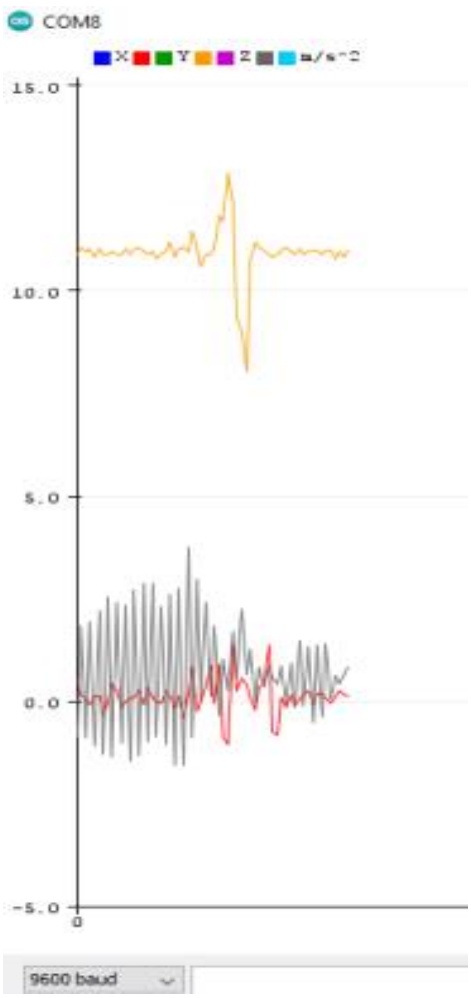


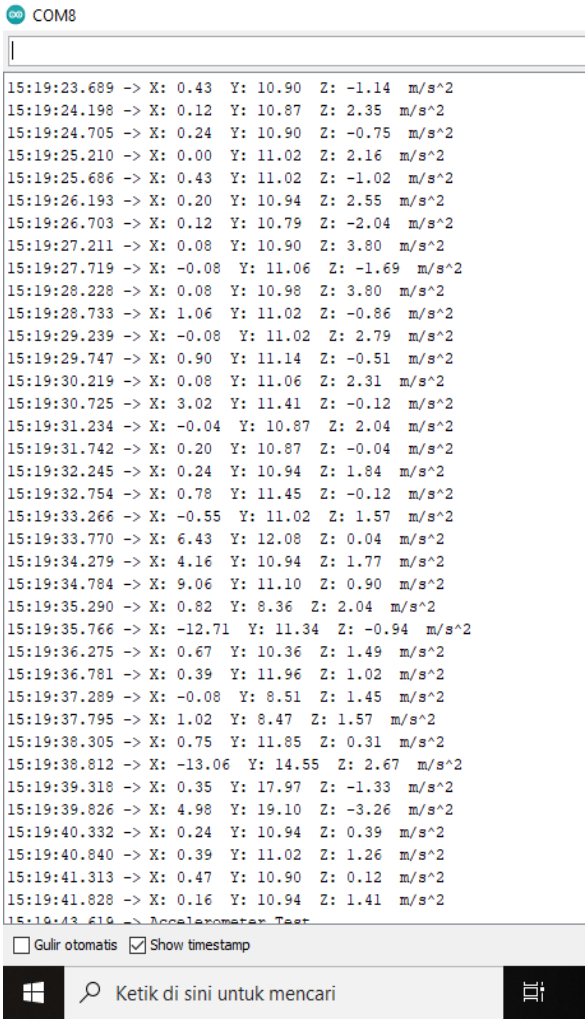
b. Pengukuran Kedalaman 0,4 mm dengan gerak makan 8





c. Pengukuran kedalaman 0,6 dengan gerak makan 10





LAMPIRAN 2

Spesifikasi Sensor

AXES OF ACCELERATION SENSITIVITY

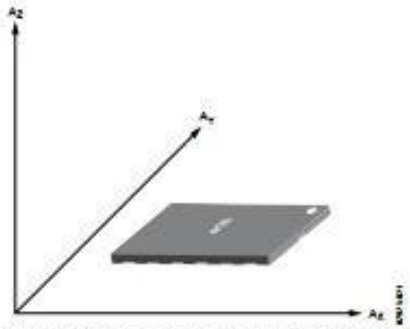


Figure 57. Axes of Acceleration Sensitivity (Corresponding Output Voltage Increases When Accelerated Along the Sensitive Axis)

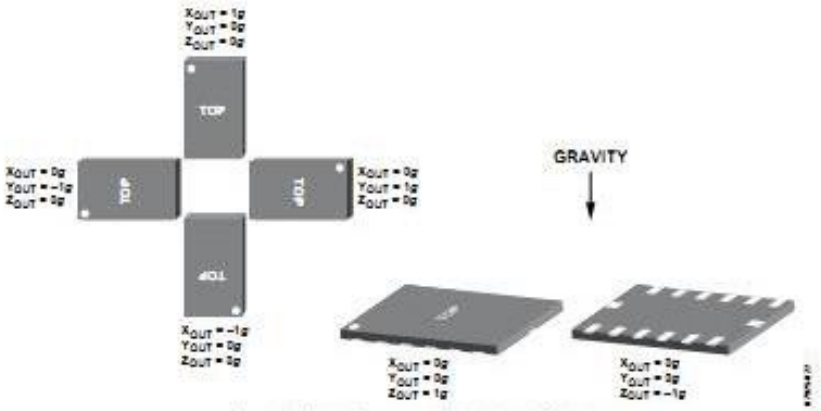


Figure 58. Output Response vs. Orientation to Gravity

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

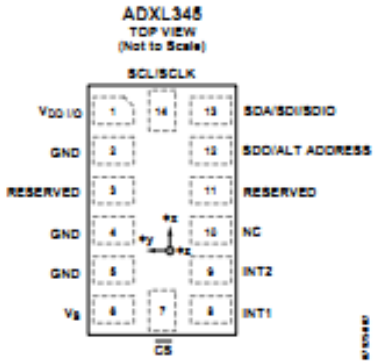


Figure 3. Pin Configuration (Top View)

Table 5. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	$V_{DD I/O}$	Digital Interface Supply Voltage.
2	GND	This pin must be connected to ground.
3	RESERVED	Reserved. This pin must be connected to Vs or left open.
4	GND	This pin must be connected to ground.
5	GND	This pin must be connected to ground.
6	V_S	Supply Voltage.
7	$\overline{CS}$	Chip Select.
8	INT1	Interrupt 1 Output.
9	INT2	Interrupt 2 Output.
10	NC	Not Internally Connected.
11	RESERVED	Reserved. This pin must be connected to ground or left open.
12	SDO/ALT ADDRESS	Serial Data Output (SPI 4-Wire)/Alternate I^2C Address Select (I^2C).
13	SDA/SDI/SDIO	Serial Data (I^2C)/Serial Data Input (SPI 4-Wire)/Serial Data Input and Output (SPI 3-Wire).
14	SCL/SCLK	Serial Communications Clock. SCL is the clock for I^2C, and SCLK is the clock for SPI.

SERIAL COMMUNICATIONS

I²C and SPI digital communications are available. In both cases, the ADXL345 operates as a slave. I²C mode is enabled if the $\overline{CS}$ pin is tied high to $V_{DD\ I/O}$. The $\overline{CS}$ pin should always be tied high to $V_{DD\ I/O}$ or be driven by an external controller because there is no default mode if the $\overline{CS}$ pin is left unconnected. Therefore, not taking these precautions may result in an inability to communicate with the part. In SPI mode, the $\overline{CS}$ pin is controlled by the bus master. In both SPI and I²C modes of operation, data transmitted from the ADXL345 to the master device should be ignored during writes to the ADXL345.

SPI

For SPI, either 3- or 4-wire configuration is possible, as shown in the connection diagrams in Figure 34 and Figure 35. Clearing the SPI bit (Bit D6) in the DATA_FORMAT register (Address 0x31) selects 4-wire mode, whereas setting the SPI bit selects 3-wire mode. The maximum SPI clock speed is 5 MHz with 100 pF maximum loading, and the timing scheme follows clock polarity (CPOL) = 1 and clock phase (CPHA) = 1. If power is applied to the ADXL345 before the clock polarity and phase of the host processor are configured, the $\overline{CS}$ pin should be brought high before changing the clock polarity and phase. When using 3-wire SPI, it is recommended that the SDO pin be either pulled up to $V_{DD\ I/O}$ or pulled down to GND via a 10 kΩ resistor.

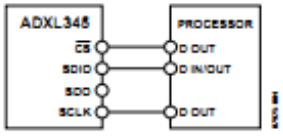


Figure 34. 3-Wire SPI Connection Diagram

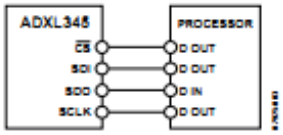


Figure 35. 4-Wire SPI Connection Diagram

$\overline{CS}$ is the serial port enable line and is controlled by the SPI master. This line must go low at the start of a transmission and high at the end of a transmission, as shown in Figure 37. SCLK is the serial port clock and is supplied by the SPI master. SCLK should idle high during a period of no transmission. SDI and SDO are the serial data input and output, respectively. Data is updated on the falling edge of SCLK and should be sampled on the rising edge of SCLK.

To read or write multiple bytes in a single transmission, the multiple-byte bit, located after the R/$\overline{W}$ bit in the first byte transfer (MB in Figure 37 to Figure 39), must be set. After the register addressing and the first byte of data, each subsequent set of clock pulses (eight clock pulses) causes the ADXL345 to point to the next register for a read or write. This shifting continues until the clock pulses cease and $\overline{CS}$ is deasserted. To perform reads or writes on different, nonsequential registers, $\overline{CS}$ must be deasserted between transmissions and the new register must be addressed separately.

The timing diagram for 3-wire SPI reads or writes is shown in Figure 39. The 4-wire equivalents for SPI writes and reads are shown in Figure 37 and Figure 38, respectively. For correct operation of the part, the logic thresholds and timing parameters in Table 9 and Table 10 must be met at all times.

Use of the 3200 Hz and 1600 Hz output data rates is only recommended with SPI communication rates greater than or equal to 2 MHz. The 800 Hz output data rate is recommended only for communication speeds greater than or equal to 400 kHz, and the remaining data rates scale proportionally. For example, the minimum recommended communication speed for a 200 Hz output data rate is 100 kHz. Operation at an output data rate above the recommended maximum may result in undesirable effects on the acceleration data, including missing samples or additional noise.

Preventing Bus Traffic Errors

The ADXL346 $\overline{CS}$ pin is used both for initiating SPI transactions, and for enabling I²C mode. When the ADXL346 is used on a SPI bus with multiple devices, its $\overline{CS}$ pin is held high while the master communicates with the other devices. There may be conditions where a SPI command transmitted to another device looks like a valid I²C command. In this case, the ADXL346 would interpret this as an attempt to communicate in I²C mode, and could interfere with other bus traffic. Unless bus traffic can be adequately controlled to assure such a condition never occurs, it is recommended to add a logic gate in front of the SDI pin as shown in Figure 36. This OR gate will hold the SDA line high when $\overline{CS}$ is high to prevent SPI bus traffic at the ADXL346 from appearing as an I²C start command.

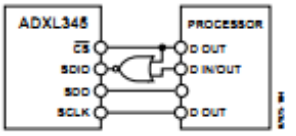


Figure 36. Recommended SPI Connection Diagram when Using Multiple SPI Devices on a Single Bus

LAMPIRAN 3

Coding Program bahasa C++

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>

/* Assign a unique ID to this sensor at the same time */
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);
        void displaySensorDetails(void)
        {
                sensor_t sensor;
                accel.getSensor(&sensor);
                Serial.println("------------------------------------");
                Serial.print  ("Sensor:       "); Serial.println(sensor.name);
                Serial.print  ("Driver Ver:   "); Serial.println(sensor.version);
                Serial.print  ("Unique ID:    "); Serial.println(sensor.sensor_id);
                Serial.print  ("Max Value:    "); Serial.print(sensor.max_value);
                Serial.println(" m/s^2");
                Serial.print  ("Min Value:    "); Serial.print(sensor.min_value);
                Serial.println(" m/s^2");
                Serial.print  ("Resolution:   "); Serial.print(sensor.resolution);
                Serial.println(" m/s^2");
                Serial.println("------------------------------------");
                Serial.println("");
```

```
delay(500);
}
void displayDataRate(void)
{
    Serial.print  ("Data Rate:    ");
    switch(accel.getDataRate())
    {
    case ADXL345_DATARATE_3200_HZ:
        Serial.print  ("3200 ");
        break;
        case ADXL345_DATARATE_1600_HZ:
            Serial.print  ("1600 ");
        break;
        case ADXL345_DATARATE_800_HZ:
            Serial.print  ("800 ");
        break;
        case ADXL345_DATARATE_400_HZ:
            Serial.print  ("400 ");
        break;
        case ADXL345_DATARATE_200_HZ:
            Serial.print  ("200 ");
        break;
        case ADXL345_DATARATE_100_HZ:
            Serial.print  ("100 ");
        break;
```

```
case ADXL345_DATARATE_50_HZ:
        Serial.print  ("50 ");
break;
case ADXL345_DATARATE_25_HZ:
        Serial.print  ("25 ");
break;
case ADXL345_DATARATE_12_5_HZ:
        Serial.print  ("12.5 ");
break;
case ADXL345_DATARATE_6_25HZ:
        Serial.print  ("6.25 ");
break;
case ADXL345_DATARATE_3_13_HZ:
        Serial.print  ("3.13 ");
break;
case ADXL345_DATARATE_1_56_HZ:
        Serial.print  ("1.56 ");
break;
case ADXL345_DATARATE_0_78_HZ:
        Serial.print  ("0.78 ");
break;
case ADXL345_DATARATE_0_39_HZ:
        Serial.print  ("0.39 ");
 break;
case ADXL345_DATARATE_0_20_HZ:
```

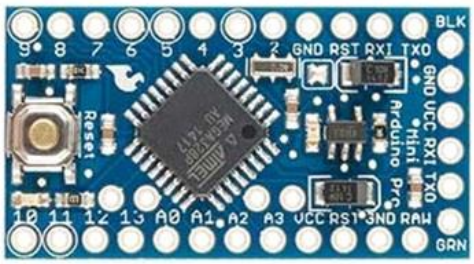
```
                    Serial.print  ("0.20 ");
                    break;
                    case ADXL345_DATARATE_0_10_HZ:
                    Serial.print  ("0.10 ");
                    break;
                    default:
                    Serial.print  ("???? ");
                    break;
              }
              Serial.println(" Hz");
            }
void displayRange(void)
{
            Serial.print  ("Range:         +/- ");
            switch(accel.getRange())
            {
               case ADXL345_RANGE_16_G:
                    Serial.print  ("16 ");
                    break;
               case ADXL345_RANGE_8_G:
                    Serial.print  ("8 ");
                    break;
               case ADXL345_RANGE_4_G:
                    Serial.print  ("4 ");
                    break;
```

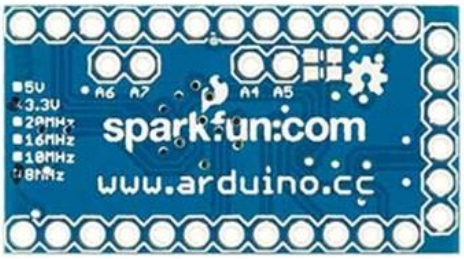
```
                    case ADXL345_RANGE_2_G:
                        Serial.print  ("2 ");
                        break;
                    default:
                        Serial.print  ("?? ");
                        break;
                }
                Serial.println(" g");
                }
void setup(void)
{
#ifndef ESP8266
    while (!Serial); // for Leonardo/Micro/Zero
#endif
    Serial.begin(9600);
    Serial.println("Accelerometer Test"); Serial.println("");

    /* Initialise the sensor */
     if(!accel.begin())
    {
        /* There was a problem detecting the ADXL345 ... check your connections */
        Serial.println("Ooops, no ADXL345 detected ... Check your wiring!");
        while(1);
    }
```

```
    /* Set the range to whatever is appropriate for your project */
    accel.setRange(ADXL345_RANGE_16_G);
    // accel.setRange(ADXL345_RANGE_8_G);
    // accel.setRange(ADXL345_RANGE_4_G);
    // accel.setRange(ADXL345_RANGE_2_G);
     /* Display some basic information on this sensor */
     displaySensorDetails();

    /* Display additional settings (outside the scope of sensor_t) */
    displayDataRate();
    displayRange();
    Serial.println("");
}
void loop(void)
{
        /* Get a new sensor event */
        sensors_event_t event;
        accel.getEvent(&event);

        /* Display the results (acceleration is measured in m/s^2) */
        Serial.print("X: "); Serial.print(event.acceleration.x); Serial.print("  ");
        Serial.print("Y: "); Serial.print(event.acceleration.y); Serial.print("  ");
        Serial.print("Z: "); Serial.print(event.acceleration.z); Serial.print("  ");
        Serial.println("m/s^2 ");
        delay(500);
```

LAMPIRAN 4

Arduino Pro Mini

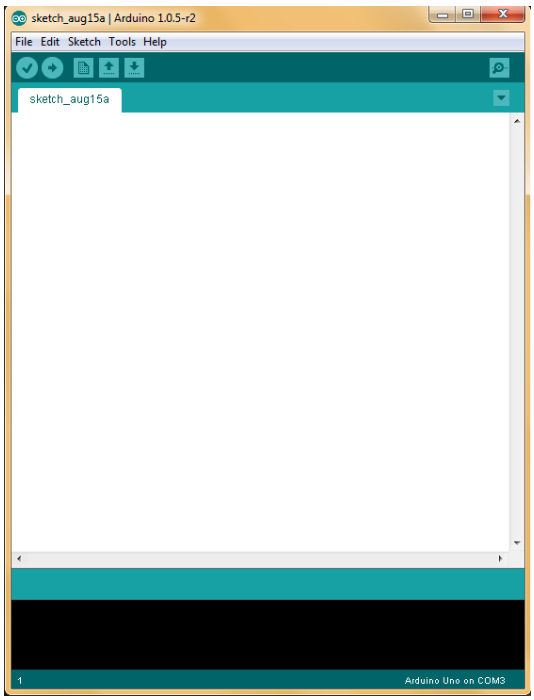




Spesifikasi

Chip mikrokontroller	ATmega328P
Tegangan operasi	5V atau 3.3V (tergantung model)
Digital I/O pin	14 buah, 6 diantaranya menyediakan PWM
Analog Input pin	6 buah
Arus DC per pin I/O	40 mA
Memori Flash	32 KB, 0.5 KB telah digunakan untuk bootloader
SRAM	2 KB
EEPROM	1 KB
Clock speed	8 Mhz (model 3.3V) atau 16 Mhz (Model 5V)
Dimensi	33 mm x 18 mm
Berat	5 g

LAMPIRAN 5



Spesifikasi software arduino

No.	Tombol	Nama	Fungsi
1.		Verify	Menguji apakah ada kesalahan pada program atau sketch. Apabila sketch sudah benar, maka sketch tersebut akan dikompilasi. Kompilasi adalah proses mengubah kode program ke dalam kode mesin
2		Upload	Mengirimkan kode mesin hasil kompilasi ke board arduino
3		New	Membuat sketch yang baru
4		Open	Membuka sketch yang sudah ada
5		Save	Menyimpan sketch
6		Serial Monitor	Menampilkan data yang dikirim dan diterima melalui komunikasi serial

LAMPIRAN 6

Dokumentasi Penelitian