LAMPIRAN

1. List Program

```
#include <math.h>

#include <TimerOne.h>

#include <SD.h>

#include <SPI.h>

#include <LiquidCrystal.h>

#include <Wire.h>

#include "RTClib.h"

RTC_DS1307 RTC;

File myFile;

int pinCS = 10;

LiquidCrystal lcd(8, 7, 3, 4, 5, 6);

#define FIRST_ADC_INPUT 1

#define LAST_ADC_INPUT 2

unsigned int adc_data[LAST_ADC_INPUT-FIRST_ADC_INPUT+1];

const int save = A3;

int buttonState = LOW;

int lastButtonState = LOW;

int lastDebounceTime = 0;

int debounceDelay = 50,dat_filt=0;

unsigned int ADC1_temp=0,ADC2_temp=0,ADC2_hsl=0,ADC1_hsl=0;

#define ADC_VREF_TYPE ((0<<REFS1) | (1<<REFS0) | (0<<ADLAR))
```

```
int last_adc1=0,last_adc2=0,ADC1 = 0,ADC2 = 0;

int lop1=0,lop2=0,msec=0,sec=0,out=0,inpt=0;

long period=0,tot_period=0;

float dat_des=0,dat_temp=0,hasil,hasil_filter=0;

const byte interpt_Pin0 = 2;

String dataString = "";

int thn = 0, bln = 0, hri = 0, jam = 0, mnt = 0, dtk = 0;

void setup() {

  lcd.begin(16, 2);

  pinMode(save, INPUT_PULLUP);

  Serial.begin(9600);

  pinMode(interpt_Pin0, INPUT_PULLUP);

  attachInterrupt(digitalPinToInterrupt(interpt_Pin0), Ex_Int0,
FALLING);

    Serial.println("contuine");

  cli();//diable interrupts

  ADCSRA = 0;

  ADCSRB = 0;

  DIDR0=(0<<ADC5D) | (0<<ADC4D) | (0<<ADC3D) | (0<<ADC2D) |
(0<<ADC1D) | (0<<ADC0D);

  ADMUX=FIRST_ADC_INPUT | ADC_VREF_TYPE;

  ADCSRA=(1<<ADEN) | (1<<ADSC) | (1<<ADATE) | (0<<ADIF) |
(1<<ADIE) | (1<<ADPS2) | (0<<ADPS1) | (1<<ADPS0);

  //ADCSRA=(1<<ADEN) | (1<<ADSC) | (1<<ADATE) | (0<<ADIF) |
(1<<ADIE) | (1<<ADPS2) | (1<<ADPS1) | (0<<ADPS0);
```

```
  //ADCSRA=(1<<ADEN) | (1<<ADSC) | (1<<ADATE) | (0<<ADIF) |
(1<<ADIE) | (1<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);

  ADCSRB=(0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);

  sei();//enable interrupts

  Timer1.initialize(5000);

  Timer1.attachInterrupt( timerIsr ); // attach the service
routine here

  lcd.print("Initializing SD card...");

  // see if the card is present and can be initialized:

  if (!SD.begin(pinCS)) {

    lcd.print("Card failed, or not present");

    // don't do anything more:

    return;

  }

  else{  lcd.print("card initialized.");}

  delay(1000);

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("  TUGAS AKHIR");

  lcd.setCursor(0,1);

  lcd.print("   Audio Meter");

  delay(2000);

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("MIFTAH IBRAHIM");
```

```
  lcd.setCursor(0,1);
  lcd.print("20153010091");
  delay(2000);
  lcd.clear();
  Wire.begin();
  RTC.begin();
  RTC.adjust(DateTime(thn,bln,hri,jam,mnt,dtk));
  RTC.adjust(DateTime(2018,8,8,17,13,0));
  DateTime now = RTC.now();
  //int thn=0,bln=0,hri=0,jam=0,mnt=0,dtk=0;
  //thn = now.year(); bln = now.month(); hri = now.day(); jam =
now.hour(); mnt = now.minute(); dtk = now.second();
}
void loop() {
  lcd.setCursor(0,0);
  inpt = (inpt + ADC2_hsl)/2;out = (out + ADC1_hsl)/2;
  hasil = (float)out/(inpt);
  dat_des = 20 * log(hasil);
  hasil_filter = (6.8356*dat_des) + 34.99;
  hasil_filter = hasil_filter - 16;
  lcd.print("Intens:");lcd.print((int)hasil_filter); lcd.print("
dB ");
  lcd.setCursor(0,1);
  lcd.print("Freq:");lcd.print(tot_period);lcd.print(" Hz  ");
```

```
  int set_mode = digitalRead(save);
  if(set_mode != lastButtonState){lastDebounceTime = millis();}
  if((millis() - lastDebounceTime) > debounceDelay)
  {if(set_mode != buttonState)
    {buttonState = set_mode;
     if(buttonState == LOW)
      {
         DateTime now = RTC.now();
         thn = now.year();
         bln = now.month(); hri = now.day(); jam = now.hour(); mnt
= now.minute(); dtk = now.second();
       lcd.clear();dataString="";
       lcd.setCursor(0,0);
       dataString += String("Freq: ");dataString +=
String(tot_period);dataString += String(" Hz");dataString +=
String("\r");
       dataString += String(" Intensitas suara: ");dataString +=
String((int)hasil_filter);dataString += String("dB ");dataString
+= String("jam ");
       dataString += String(jam);dataString +=
String(":");dataString += String(mnt);dataString +=
String(":");dataString += String(dtk);
       dataString += String(" tgl: ");dataString +=
String(hri);dataString += String("/");dataString +=
String(bln);dataString += String("/");
       dataString += String(thn);dataString += String("  ");
       myFile = SD.open("data.txt", FILE_WRITE);
```

```
        // if the file opened okay, write to it:
        if (myFile)
        {
         Serial.print("Writing to test.txt...");
         myFile.println(dataString);
         lcd.setCursor(0,0);
         lcd.print("Writing file  ");lcd.print(" ");
         lcd.setCursor(0,1);
         lcd.print("txt....save..");lcd.print(" ");
         myFile.close();
         lcd.clear();
         Serial.println("done.");
         lcd.setCursor(0,0);
         //lcd.print("Done...  ");lcd.print(" ");
         lcd.print("Jam:
");lcd.print(jam);lcd.print(":");lcd.print(mnt);lcd.print(":");lcd
.print(dtk);lcd.print("  ");
         lcd.setCursor(0,1);
         lcd.print("tgl:
");lcd.print(hri);lcd.print("/");lcd.print(bln);lcd.print("/");lcd
.print(thn);lcd.print("  ");
         }
        else
        {
         // if the file didn't open, print an error:
```

```
      Serial.println("error opening test.txt");
      lcd.print("error opening test.txt");lcd.print(" ");
     }
     // re-open the file for reading:
     myFile = SD.open("data.txt");
     if (myFile)
     {
      Serial.println("open.txt:");
      // read from the file until there's nothing else in it:
      while (myFile.available()) {
        Serial.write(myFile.read());
        }
        // close the file:
     myFile.close();
    }
    else {
      // if the file didn't open, print an error:
      Serial.println("error opening test.txt");
      }
     delay(1000);
     lcd.clear();
    }
  }
}
```

```
lastButtonState = set_mode;

}

ISR(ADC_vect){

static unsigned char input_index=0;

// Read the AD conversion result

adc_data[input_index]=ADCW;

input

if (++input_index > (LAST_ADC_INPUT-FIRST_ADC_INPUT))

   input_index=0;

ADMUX=(FIRST_ADC_INPUT | ADC_VREF_TYPE)+input_index;

// Delay needed for the stabilization of the ADC input voltage

delayMicroseconds(10);

// Start the AD conversion

ADCSRA|=(1<<ADSC);

if(input_index==0)

{if(last_adc1>adc_data[input_index])

  {if(lop1==0){ADC1=adc_data[input_index];lop1=1;}

  }

 if(last_adc1<adc_data[input_index]){lop1=0;}

 last_adc1=adc_data[input_index];

}

if(input_index==1)

{if(last_adc2>adc_data[input_index])

  {if(lop2==0)
```

```
   {//period = period + 1;//
    ADC2=adc_data[input_index];lop2=1;
   }
  }
 if(last_adc2<adc_data[input_index]){lop2=0;}
 last_adc2=adc_data[input_index];
}
  if(dat_filt<=500)
  {
    if(ADC2_temp<ADC2){ADC2_temp=ADC2;}
    if(ADC1_temp<ADC1){ADC1_temp=ADC1;}
   }
  if(++dat_filt>500)
  {
    ADC2_hsl = (ADC2_hsl + ADC2_temp)/2;ADC2_temp=0;
    ADC1_hsl = (ADC1_hsl + ADC1_temp)/2;ADC1_temp=0;
    dat_filt=0;
  }}
void timerIsr()
{  if(++msec>200)
  {if(++sec>0){tot_period = period;period=0;sec=0;}
   msec=0;  }}
void Ex_Int0() {period = period + 1;}
```

2. Perhitungan Data

a. Frekuensi 125 Hz

1). Rata-rata Osiloskop

$$\text{Rata – Rata } (\overline{X}\text{ n}) = \frac{\Sigma Xi}{n}$$

$$\text{Rata-rata } (\overline{X}\text{ n}) = \frac{124,1+124,2+124,6+124,7+124,8}{5}$$

$$= 124,48$$

2). Rata-rata Modul TA

$$\text{Rata – Rata } (\overline{X}) = \frac{\Sigma Xi}{n}$$

$$\text{Rata-rata } (\overline{X}) = \frac{125+125+125+125+125+125}{5}$$

$$= 125$$

3). Simpangan

$$\text{Simpangan} = (\overline{X}\text{ n} - \overline{X})$$

$$= 124,48 - 125$$

$$= -0,52$$

4). % *Error*

$$\%Error = \frac{\overline{X}_{\text{n}} - \overline{X}}{\overline{X}_{\text{n}}} \text{x } 100\%$$

$$= \frac{124,48-125}{124,48} x\ 100\%$$

$$= -0,42\%$$

b. Frekuensi 250 Hz

1). Rata-rata Osiloskop

Rata – Rata ($\overline{X}$ n) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$ n) = $\frac{249,1+249,4+249,2+249,3+249,4}{5}$

= 249,28

2). Rata-rata Modul TA

Rata – Rata ($\overline{X}$) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$) = $\frac{250+250+250+250+250+250}{5}$

= 250

3). Simpangan

Simpangan = ($\overline{X}$ n - $\overline{X}$)

= 249,28 – 250

= -0,72

4). % *Error*

%*Error* = $\frac{\overline{X}_n - \overline{X}}{\overline{X}_n}$ x 100%

= $\frac{249,28-250}{249,28} x$ 100%

= -0,29%

c. Frekuensi 500 Hz

1). Rata-rata Osiloskop

Rata – Rata ($\overline{X}$ n) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$ n) = $\frac{500,3+500,1+500,5+500+501,2}{5}$

= 500,4

2). Rata-rata Modul TA

Rata – Rata ($\overline{X}$) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$) = $\frac{503+502+503+503+504}{5}$

= 503

3). Simpangan

Simpangan = ($\overline{X}$ n - $\overline{X}$)

= 500,4 – 503

= -2,6

4). % *Error*

%*Error* = $\frac{\overline{X}_n - \overline{X}}{\overline{X}_n}$ x 100%

= $\frac{497,4+500}{497,4} x$ 100%

= -0,52%

d. Frekuensi 1000 Hz

1). Rata-rata Osiloskop

Rata – Rata ($\overline{X}$ n) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$ n) = $\frac{997,1+1002+997,8+999,9+1001}{5}$

= 999,6

2). Rata-rata Modul TA

Rata – Rata ($\overline{X}$) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$) = $\frac{1001+1007+1002+1004+1006}{5}$

= 1004

3). Simpangan

Simpangan = ($\overline{X}$ n - $\overline{X}$)

= 999,6 – 1004

= -4,4

4). % *Error*

%*Error* = $\frac{\overline{X}_n - \overline{X}}{\overline{X}_n}$ x 100%

= $\frac{999,6-1004}{999,6} x$ 100%

= 0,44%

e. Frekuensi 2000 Hz

1). Rata-rata Osiloskop

Rata – Rata ($\overline{X}$ n) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$ n) = $\frac{1993,5+2002+2006+2020+1996}{5}$

= 2003,5

2). Rata-rata Modul TA

Rata – Rata ($\overline{X}$) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$) = $\frac{2007+2011+2015+2029+2003}{5}$

= 2013

3). Simpangan

Simpangan = ($\overline{X}$ n - $\overline{X}$)

= 2003,5– 2013

= -9,5

4). % *Error*

%*Error* = $\frac{\overline{X}_n - \overline{X}}{\overline{X}_n}$ x 100%

= $\frac{2003,5-2013}{2003,5} x\ 100\%$

= 0,47%

b. Frekuensi 4000 Hz

1). Rata-rata Osiloskop

Rata – Rata ($\overline{X}$ n) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$ n) = $\frac{4071+4008+3994+4018+4033}{5}$

= 4024,8

2). Rata-rata Modul TA

Rata – Rata ($\overline{X}$) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$) = $\frac{4088+4024+4008+4030+4051}{5}$

= 4040,2

3). Simpangan

Simpangan = ($\overline{X}$ n - $\overline{X}$)

= 4024,8 – 4040,2

= -15,4

4). % *Error*

%*Error* = $\frac{\overline{X}_n - \overline{X}}{\overline{X}_n}$ x 100%

= $\frac{4024,8-4040,2}{4024,8} x\ 100\%$

= 0,38%

c. Frekuensi 8000 Hz

1). Rata-rata Osiloskop

Rata – Rata ($\overline{X}$ n) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$ n) = $\frac{8033+7992+7977+7964+7970}{5}$

= 7987,2

2). Rata-rata Modul TA

Rata – Rata ($\overline{X}$) = $\frac{\Sigma Xi}{n}$

Rata-rata ($\overline{X}$) = $\frac{8065 + 8024 + 8012 + 8000 + 8004}{5}$

= 8021

3). Simpangan

Simpangan = ($\overline{X}$ n - $\overline{X}$)

= 7987,2– 8021

= -33.8

4). % *Error*

%*Error* = $\frac{\overline{X}_n - \overline{X}}{\overline{X}_n}$ x 100%

= $\frac{7987,2 - 8021}{7987,2} x\ 100\%$

= -0,42%

3. Rangkaian Keseluruh