LAMPIRAN

```
#define windPin 2 // Receive the data from sensor

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <LiquidCrystal_I2C.h>

#include <BH1750.h>

#include <DHT.h>

#include <SPI.h>

#include <SD.h>

#define DHTPIN 8

#define DHTTYPE DHT22

LiquidCrystal_I2C lcd(0x27,20,4);

// anemometer parameters


volatile byte rpmcount; // count signals

volatile unsigned long last_micros;

unsigned long timeold;

unsigned long timemeasure = 1.00; // seconds

int timetoSleep = 1;              // minutes
```

```
unsigned long sleepTime = 15;      // minutes
unsigned long timeNow;
int countThing = 0;
int GPIO_pulse = 2; // Arduino = D2
float rpm, rps;     // frequencies
float radius = 0.1; // meters - measure of the lenght of each the anemometer wing
float velocity_kmh; // km/h
float velocity_ms;  //m/s
float omega = 0;    // rad/s
float calibration_value = 0.433;
float kalibrasi_suhu = -2.7;
float kalibrasi_hum = 7.8
float hasil_suhu;
float hasil_hum;
float hasil_lux;
float kalibrasi_lux =0.65;
const int CS = 10;
BH1750 lightMeter;
File dataku;
```

```
DHT dht(DHTPIN, DHTTYPE);

float hum;

float temp;

float lux;

void setup()

{

  pinMode(GPIO_pulse, INPUT_PULLUP);

  digitalWrite(GPIO_pulse, LOW);


  Serial.begin(9600);


  detachInterrupt(digitalPinToInterrupt(GPIO_pulse));
// force to initiate Interrupt on zero

  attachInterrupt(digitalPinToInterrupt(GPIO_pulse),
rpm_anemometer, RISING); //Initialize the intterrupt pin

  rpmcount = 0;

  rpm = 0;

  timeold = 0;

  timeNow = 0;
  Wire.begin();
lightMeter.begin();
dht.begin();
lcd.init();
lcd.backlight();
lcd.setCursor(1,0);
lcd.print("Alat ukur otomatis");
lcd.setCursor(1,1);
lcd.print("Suhu,Kelembaban");
lcd.setCursor(1,2);
lcd.print("Angin dan Cahaya");
delay(5000);
lcd.clear();

//Status pembacaan micro sd
  //Serial.print("Membaca SD Card...");
  if (!SD.begin(CS))
  {
    //Serial.println("Empty/SD Card Rusak!");
    //lcd.setCursor(0,0);
    //lcd.print("Gagal/SD Card Rusak");
    //lcd.clear();
    //while (1);
   }
  //Serial.println("Berhasil");
  //lcd.setCursor(2,1);
  //]lcd.print("SD Card detected");
  //lcd.clear();

} // end of setup

void loop()
{
hum = dht.readHumidity();
 temp = dht.readTemperature();
 lux = lightMeter.readLightLevel();
hasil _hum = hum + kalibrasi_hum;
hasil_suhu= temp + kalibrasi_suhu;
hasil_lux= lux – (lux * kalibrasi_lux);
lcd.init();
lcd.backlight();

lcd.setCursor(0,0);
lcd.print("Suhu      =");
lcd.setCursor(12,0);
lcd.print(hasil_suhu);
lcd.setCursor(17,0);
lcd.print(" *C ");
lcd.setCursor(0,1);
lcd.print("Kelembaban =");
lcd.setCursor(13,1);
lcd.print(hasil_hum);
lcd.setCursor(18,1);
lcd.print("%");
lcd.setCursor(0,2);
lcd.print("Cahaya =");
lcd.setCursor(9,2);
lcd.print(hasil_lux);
lcd.setCursor(16,2);
lcd.print(" lux");
lcd.setCursor(0,3);
lcd.print("Angin =");
lcd.setCursor(8,3);
lcd.print(velocity_ms);
lcd.setCursor(14,3);
lcd.print("m/s");
delay(2000);

  //Measure RPM
  if ((millis() - timeold) >= timemeasure * 1000)
  {
    countThing++;
    detachInterrupt(digitalPinToInterrupt(GPIO_pulse)); //
Disable interrupt when calculating
    rps = float(rpmcount) / float(timemeasure);         //
rotations per second
    rpm = 60 * rps;                                     // rotations per
minute
    omega = 2 * PI * rps;                               // rad/s
    velocity_ms = omega * radius * calibration_value;   // m/s
    velocity_kmh = velocity_ms * 3.6;                   // km/h
    Serial.print("rps=");
    Serial.print(rps);
    Serial.print("   rpm=");
    Serial.print(rpm);
    Serial.print("   velocity_ms=");
    Serial.print(velocity_ms);
    Serial.print("   velocity_kmh=");
    Serial.print(velocity_kmh);
    Serial.println("   ");
    if (countThing == 1) // Send data per 25 seconds
    {
      Serial.println("Send data to server");
      countThing = 0;
    }
    timeold = millis();
    rpmcount = 0;
    attachInterrupt(digitalPinToInterrupt(GPIO_pulse), 
rpm_anemometer, RISING); // enable interrupt
  }dataku = SD.open("data1.txt", FILE_WRITE);
  if (dataku)
```

```
{ dataku.println(" ");
  dataku.print(temp);
  dataku.print(" *C");
  dataku.print(" ");
  dataku.print(temp);
  dataku.print(" %");
  dataku.print(" ");
  dataku.print(lux);
  dataku.print(" Lux");
  dataku.print(" ");
  dataku.print(velocity_ms);
  dataku.print("  m/s");
  dataku.print(" ");
  dataku.close();
 lcd.setCursor(0,1);
 // lcd.print("Berhasil Tersimpan!");
  //delay (500);
  //lcd.clear() ;

  //Serial.println("Berhasil Tersimpan! ");
```

```
    //delay(5000);
   }

  else
  {
    //Serial.println(" Gagal Tersimpan! ");
    lcd.setCursor(3,1);
    lcd.print("SD Card Empty");
    delay (500);
  lcd.clear();
}

} // end of loop

void rpm_anemometer()
{
  if (long(micros() - last_micros) >= 5000)
  { // time to debounce measures
    rpmcount++;
    last_micros =}
```