



LAMPIRAN

LAMPIRAN A

SOURCE CODE

PROGRAM DI NODEMCU ESP32

```
//KOMUNIKASI SERIAL//
#define RXD2 16
#define TXD2 17

//DHT//
#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
//dht variable global
float h;
float t;
unsigned long previousMillis =
0;
const long interval = 2000;

//RELAY//
#define KPS2pin 14
#define KPS1pin 27
#define AIRpin 26
#define SRKLSpin 25

//HC-SR04//
int trigPin = 19;
int echoPin = 21;
long duration;
float distance;
float tinggiair;

//tds variable global//
int TDS;

void setup() {
    // put your setup code here, to
    run once:
    Serial.begin(115200);
    Serial2.begin(115200,
SERIAL_8N1, RXD2, TXD2);

    pinMode (trigPin, OUTPUT);
    pinMode (echoPin, INPUT);
    pinMode (KPS2pin, OUTPUT);
    pinMode (KPS1pin, OUTPUT);
    pinMode (AIRpin, OUTPUT);
    pinMode (SRKLSpin,
OUTPUT);

    dht.begin();
}

void loop() {
    serial_terima();
    Dht();
    ultrasoniksensor();

    Serial.print ("TDS Value:");
    Serial.print (TDS,0);
    Serial.print ("ppm\t");
    Serial.print (tinggiair);
    Serial.print ("cm\t");
    Serial.print ("Humidity: ");
    Serial.print (h);
    Serial.print ("% \t");
    Serial.print ("Temperature: ");
    Serial.print (t);
    Serial.print ("°C");
```

```

Serial.println();
delay(2000);
}

PROGRAM UNTUK
KOMUNIKASI SERIAL

String dataMasuk = "";
void serial_terima() {
  if (Serial2.available() > 0) {
    dataMasuk =
Serial2.readStringUntil('\n');
    dataMasuk.trim();

    byte buka =
dataMasuk.indexOf('{');
    byte tutup =
dataMasuk.indexOf('}');

    String firstVal =
dataMasuk.substring(buka + 1,
tutup);

    TDS = firstVal.toInt();
  }
  if (TDS < 550){

digitalWrite(SRKLSpin,LOW);
  }
  else {

digitalWrite(SRKLSpin,HIGH);
  }
}

```

PROGRAM UNTUK SENSOR DHT

```
void Dht() {
```

```

// delay(2000);
  unsigned long currentMillis =
millis();
  if(currentMillis - previousMillis
> interval)
  {
    previousMillis = currentMillis;

    h = dht.readHumidity();
    t = dht.readTemperature();
    if (isnan(h) || isnan(t)) {
      Serial.println(F("Failed to read
from DHT sensor!"));
      return;
    }
    if (t > 29.00) {
      digitalWrite(KPS1pin, LOW);
      digitalWrite(KPS2pin, LOW);
    }
    else {
      digitalWrite(KPS1pin, HIGH);
      digitalWrite(KPS2pin, HIGH);
    }
  }
}

```

PROGRAM UNTUK SENSOR ULTRASONIK

```

void ultrasoniksensor(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin,
HIGH);
  distance = (duration / 2) / 29.1;
}

```

```

tinggiair = (27 - distance);

if(tinggiair < 2.50) {
    digitalWrite(AIRpin,LOW);
}
else {
    digitalWrite(AIRpin,HIGH);
}
}

```

PROGRAM DI ARDUINO UNO

```

#include <SoftwareSerial.h>
//komunikasi serial
SoftwareSerial serial_slave(2, 3);
//(RX,TX);
#include <Wire.h>
#include <RtcDS3231.h>
RtcDS3231<TwoWire>
Rtc(Wire);

#define PMPpin 4
#define NTRSpin 5
#define TdsSensorPin A3
#define VREF 5.0
#define SCOUNT 30

//tds variable global
int analogBuffer[SCOUNT]; //
menyimpan nilai analog dalam
array, baca dari ADC
int
analogBufferTemp[SCOUNT];
int analogBufferIndex = 0;
int copyIndex = 0;
float averageVoltage = 0;
float tdsValue = 0;
float temperature = 27; //
suhu saat ini untuk kompensasi

```

```

int lock = 0;
int kondisi = 1;
int kunci = 0;

void setup() {
    // put your setup code here, to
run once:
    Serial.begin(9600);
    serial_slave.begin(115200);
    Serial.print("compiled: ");
    Serial.print(__DATE__);
    Serial.println(__TIME__);

    Rtc.Begin();
    RtcDateTime compiled =
RtcDateTime(__DATE__,
__TIME__);
    printDateTime(compiled);
    Serial.println();

    RtcDateTime now =
Rtc.GetDateTime();
    Rtc.Enable32kHzPin(false);

Rtc.SetSquareWavePin(DS3231S
quareWavePin_ModeNone);

    pinMode (TdsSensorPin,
INPUT);
    pinMode (PMPpin, OUTPUT);
    pinMode (NTRSpin,
OUTPUT);
    digitalWrite(PMPpin, HIGH);
}

void loop() {
    // put your main code here, to
run repeatedly:

```

```

RtcDateTime now =
Rtc.GetDateTime();
int jam = now.Hour();
int menit = now.Minute();
int detik = now.Second();
tds();

```

PROGRAM UNTUK MODUL RTC DS3231

```

if (jam >= 5 && jam <= 20 &&
menit <= 59 && detik <= 59) {
    switch (kondisi) {
        case 1:
            lock = 0;
            kunci = 0;
            if (menit >= 0 && menit <
15 && lock == 0) {
                lock = 1;
                kunci = 0;
                if (detik >= 0 && detik <=
5 && lock == 1 && kunci == 0)
                {
                    digitalWrite(PMPpin,
LOW);
                    Serial.print("Sprayer
On");
                }
                if (detik > 5 && kunci ==
0) {
                    digitalWrite(PMPpin,
HIGH);
                    kunci = 1;
                }
            }
            if (kunci == 1) {
                digitalWrite(PMPpin,
HIGH);
                // Serial.print("Sprayer

```

```

Off");
                kondisi = 2;
            }
            break;
        case 2:
            lock = 0;
            kunci = 0;
            if (menit >= 15 && menit <
30 && lock == 0) {
                lock = 1;
                kunci = 0;
                if (detik >= 0 && detik <=
5 && lock == 1 && kunci == 0)
                {
                    digitalWrite(PMPpin,
LOW);
                    Serial.print("Sprayer
On");
                }
                if (detik > 5 && kunci ==
0) {
                    digitalWrite(PMPpin,
HIGH);
                    kunci = 1;
                }
            }
            if (kunci == 1) {
                digitalWrite(PMPpin,
HIGH);
                // Serial.print("Sprayer
Off");
                kondisi = 3;
            }
            break;
        case 3:
            lock = 0;
            kunci = 0;
            if (menit >= 30 && menit <
45 && lock == 0) {

```

```

        lock = 1;
        kunci = 0;
        if (detik >= 0 && detik <=
5 && lock == 1 && kunci == 0)
    {
        digitalWrite(PMPpin,
LOW);
        Serial.print("Sprayer
On");
    }
    if (detik > 5 && kunci ==
0) {
        digitalWrite(PMPpin,
HIGH);
        kunci = 1;
    }
    if (kunci == 1) {
        digitalWrite(PMPpin,
HIGH);
//        Serial.print("Sprayer
Off");
        kondisi = 4;
    }
    break;
case 4:
    lock = 0;
    kunci = 0;
    if (menit >= 45 && menit <
60 && lock == 0) {
        lock = 1;
        kunci = 0;
        if (detik >= 0 && detik <=
5 && lock == 1 && kunci == 0)
    {
        digitalWrite(PMPpin,
LOW);
        Serial.print("Sprayer
On");

```

```

    }
    if (detik > 5 && kunci ==
0) {
        digitalWrite(PMPpin,
HIGH);
        kunci = 1;
    }
    if (kunci == 1) {
        digitalWrite(PMPpin,
HIGH);
//        Serial.print("Sprayer
Off");
        kondisi = 1;
    }
    break;
case 404:
    Serial.print("waktu
selesai");
    break;
}
}
else {
    kondisi = 404;
}

serial_slave.print ("{}");
serial_slave.print (tdsValue,0);
serial_slave.println ("");
Serial.print("TDS Value:");
Serial.print(tdsValue,0);
Serial.print(" || ");
Serial.print(lock);
Serial.print(" || ");
Serial.print(kunci);
Serial.print(" || ");
Serial.print(kondisi);
Serial.print(" || ");
Serial.print(jam);

```

```

Serial.print(":");
Serial.print(menit);
Serial.print(":");
Serial.println(detik);
// delay(1000);

}

```

PROGRAM UNTUK SENSOR TDS

```

void tds() {
    static unsigned long
    analogSampleTimepoint =
    millis();
    if (millis() -
    analogSampleTimepoint > 40U)
    { //setiap 40 milidetik, baca nilai
    analog dari ADC
        analogSampleTimepoint =
        millis();

    analogBuffer[analogBufferIndex]
    = analogRead(TdsSensorPin);
        analogBufferIndex++;
        if (analogBufferIndex ==
    SCOUNT) {
            analogBufferIndex = 0;
        }
    }
    static unsigned long
    printTimepoint = millis();
    if (millis() - printTimepoint >
    800U) {
        printTimepoint = millis();
        for (copyIndex = 0; copyIndex
    < SCOUNT; copyIndex++) {

    analogBufferTemp[copyIndex] =

```

```

    analogBuffer[copyIndex];
        // baca nilai analog dan ubah
    ke nilai tegangan
        averageVoltage =
    getMedianNum(analogBufferTemp,
    SCOUNT) * (float)VREF /
    1024.0;
        //formula kompensasi suhu:
    fFinalResult(25^C) =
    fFinalResult(current)/(1.0+0.02*(
    fTP-25.0));
        float
    compensationCoefficient = 1.0 +
    0.02 * (temperature - 25.0);
        //kompensasi suhu
        float compensationVoltage =
    averageVoltage /
    compensationCoefficient;
        //mengubah nilai tegangan
    menjadi nilai tds
        tdsValue = (133.42 *
    compensationVoltage *
    compensationVoltage *
    compensationVoltage - 255.86 *
    compensationVoltage *
    compensationVoltage + 857.39 *
    compensationVoltage) * 0.5;
    }
    }
    if (tdsValue < 550){

    digitalWrite(NTRSpin,LOW);
    }
    else {

    digitalWrite(NTRSpin,HIGH);
    }
    }
    //median filtering algoritma

```

```

int getMedianNum(int bArray[],
int iFilterLen) {
    int bTab[iFilterLen];
    for (byte i = 0; i < iFilterLen;
i++)
        bTab[i] = bArray[i];
    int i, j, bTemp;
    for (j = 0; j < iFilterLen - 1;
j++) {
        for (i = 0; i < iFilterLen - j - 1;
i++) {
            if (bTab[i] > bTab[i + 1]) {
                bTemp = bTab[i];
                bTab[i] = bTab[i + 1];
                bTab[i + 1] = bTemp;
            }
        }
    }
    if ((iFilterLen & 1) > 0) {
        bTemp = bTab[(iFilterLen - 1)
/ 2];
    }
    else {
        bTemp = (bTab[iFilterLen / 2]
+ bTab[iFilterLen / 2 - 1]) / 2;
    }
    return bTemp;
}

#define countof(a)(sizeof(a) /
sizeof(a[0]))

void printDateTime(const
RtcDateTime& dt)
{
    char datestring[20];
    snprintf_P(datestring,
countof(datestring),
PSTR("%02u/%02u/%04u
%02u:%02u:%02u"),
dt.Month(),
dt.Day(),
dt.Year(),
dt.Hour(),
dt.Minute(),
dt.Second() );
    Serial.print(datestring);
}

```