LAMPIRAN A
PROGRAM Node MCU

```
// Anisa099696@gmail.com
// PW : wtpglc2023#

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
#include <SoftwareSerial.h>
SoftwareSerial mySerial(D7, D3);
#include <ArduinoJson.h>
int sensorTDS;
//Sensor PH
int pinoutPH = A0;
float PH_Step;
float pHvalue = 0;
float voltagePH;

//kalibrasi PH
float PH4 = 1.16;
float PH7 = 1.86;
//float PH7 = 2.52;
//pin GPIO
```

```
#define relay1    16
#define relay2    2
#define relay3    14
#define relay4    12
int pin_flow = D8;

// pin GPIO ultrasonik
//const int pin_echo = 15;
//const int pin_trig = 0;

float konst = 7.2; // faktor kalibrasi
float debit_air;
volatile byte count;
unsigned int flow_mlt;
unsigned long total_volume;
float total_volume_liter;
unsigned long oldTime;

float jarak;

// token blynk
#define BLYNK_TEMPLATE_ID "TMPL6TCepXuBy"
```

```
#define BLYNK_TEMPLATE_NAME "watertreatment"
#define BLYNK_AUTH_TOKEN "QQwnp9IIDFG8qw2AbSadj-_gv4DtcZca"

#define BLYNK_PRINT Serial // tampilakan informasi blynk di serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// input wifi dan password
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "meee";  // type your wifi name
char pass[] = "12345678";  // type your wifi password
BlynkTimer timers;


// membaca input dari blnyk
BLYNK_WRITE(V0) {
  int pinValue = param.asInt();
  if (pinValue == 1) digitalWrite(relay1, 0);
  if (pinValue == 0) digitalWrite(relay1, 1);
  Serial.print("in1: ");
  Serial.println(pinValue);
}
```

```
BLYNK_WRITE(V1) {
  int pinValue = param.asInt();
  if (pinValue == 1) digitalWrite(relay2, 0);
  if (pinValue == 0) digitalWrite(relay2, 1);
  Serial.print("in2: ");
  Serial.println(pinValue);
}

BLYNK_WRITE(V2) {
  int pinValue = param.asInt();
  if (pinValue == 1) digitalWrite(relay3, 0);
  if (pinValue == 0) digitalWrite(relay3, 1);
  Serial.print("in3: ");
  Serial.println(pinValue);
}

BLYNK_WRITE(V3) {
  int pinValue = param.asInt();
  if (pinValue == 1) digitalWrite(relay4, 0);
  if (pinValue == 0) digitalWrite(relay4, 1);
  Serial.print("in4: ");
```

```
  Serial.println(pinValue);
}

// external interupsi
ICACHE_RAM_ATTR void  countPulse() {
  count++;
}

// baca ultrasonik
// https://www.instructables.com/id/Distance-Measurement-Using-HC-SR04-Via-NodeMCU/
// https://www.makerguides.com/jsn-sr04t-arduino-tutorial/
//float read_srf() {
//  unsigned int duration;
//  float distance;
//  // trigger off
//  digitalWrite(pin_trig, LOW);
//  delayMicroseconds(5);
//  // trigger on
//  digitalWrite(pin_trig, HIGH);
//  delayMicroseconds(10);
//  // trigger off
//  digitalWrite(pin_trig, LOW);
```

```
//  // baca pulsa
//  duration = pulseIn(pin_echo, HIGH);
//  // konversi ke cm
//  distance = duration * 0.034 / 2;
//  return distance;
//}

// update ke blnyk dan tampilan blynk
void tasktimer() {

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(jarak, 1);
  lcd.print("cm");
  lcd.setCursor(9, 0);
  lcd.print(int(debit_air));
  lcd.print("L/min");

  lcd.setCursor(0, 1);
  lcd.print(total_volume_liter);
  lcd.print("L");
  lcd.setCursor(12, 1);
```

```
  lcd.print(digitalRead(relay1));
  lcd.print(digitalRead(relay2));
  lcd.print(digitalRead(relay3));
  lcd.print(digitalRead(relay4));
  //menampilkan di blynk
  Blynk.virtualWrite(V4, jarak);
  Blynk.virtualWrite(V5, debit_air);
  Blynk.virtualWrite(V6, total_volume_liter);
  Blynk.virtualWrite(V7, pHvalue);
  Blynk.virtualWrite(V8, sensorTDS);

  //Blynk.logEvent("notif","text");

}

// fungsi setup hanya di run sekali saat esp pertama on
// gunanya untuk inisalisasi / konfigurari input output
void setup() {
  Serial.begin(9600);
  lcd.begin();
  lcd.backlight();
  mySerial.begin(9600);
```

```
//  // konfigurasi I/O
//  pinMode(pin_trig, OUTPUT);
//  pinMode(pin_echo, INPUT);

pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
pinMode(relay3, OUTPUT);
pinMode(relay4, OUTPUT);

digitalWrite(relay1, 1); // off
digitalWrite(relay2, 1); // off
digitalWrite(relay3, 1); // off
digitalWrite(relay4, 1); // off

lcd.begin();
lcd.backlight();

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Connect To Wifi");
lcd.setCursor(0, 1);
```

```
  lcd.print(ssid);


  // inisialisasi blynk
  Blynk.begin(auth, ssid, pass);
  timers.setInterval(1000L, tasktimer);


  count        = 0;
  debit_air    = 0.0;
  flow_mlt     = 0;
  total_volume = 0;
  oldTime      = 0;


  attachInterrupt(digitalPinToInterrupt(pin_flow), countPulse,
FALLING);
  Serial.println("READY");


}


String in = "";
void loop()
{
  Blynk.run(); // jalankan blynk
  timers.run(); // jalankan timer
```

```
if ((millis() - oldTime) > 1000) {
  detachInterrupt(0);
  debit_air = ((1000.0 / (millis() - oldTime)) * count) / konst;
  oldTime = millis();
  flow_mlt = (debit_air / 60) * 1000;
  total_volume += flow_mlt;

  total_volume_liter = (float)total_volume / 1000;

  Serial.print("R1:");
  Serial.print(digitalRead(relay1));
  Serial.print(" R2:");
  Serial.print(digitalRead(relay2));
  Serial.print(" R3:");
  Serial.print(digitalRead(relay3));
  Serial.print(" R4:");
  Serial.print(digitalRead(relay4));
  Serial.print(" Jarak:");
  Serial.print(jarak, 1);
  Serial.print(" DEBIT AIR: ");
  Serial.print(int(debit_air));
```

```
  Serial.print("L/m");
  Serial.print("  Vol: ");
  Serial.print(total_volume);
  Serial.print("mL");
  Serial.print("  Vol: ");
  Serial.print(total_volume_liter);
  Serial.println("L");

  count = 0;

  attachInterrupt(0, countPulse, FALLING);
}
StaticJsonBuffer<200> jsonBuffer;
JsonObject& data = jsonBuffer.parseObject(mySerial);

if (data == JsonObject::invalid()) {
  Serial.println("Invalid Json Object");
  jsonBuffer.clear();
  return;
}
int tds = data["data1"];
Serial.print("NILAI PPM:");
```

```
  Serial.println(tds);
  int jk = data["data2"];
  Serial.print("Jarak:");
  Serial.println(jk);

  int adcPH = analogRead(pinoutPH);
  voltagePH = adcPH * 3.3 / 1023;
  Serial.print("Tegangan PH: ");
  Serial.println(voltagePH);

  PH_Step = (PH4 - PH7) / 3;
  pHvalue = 7.00 + ((PH7 - voltagePH) / PH_Step);
  Serial.print("Nilai pH : ");
  Serial.println(pHvalue);
  sensorTDS =  tds;
  jarak = jk;
}
```

LAMPIRAN B

PROGRAM
ARDUINO UNO

```
#include <SoftwareSerial.h>

#define TdsSensorPin A1

#define VREF 3.3              // analog reference voltage(Volt) of the ADC

#define SCOUNT  30            // sum of sample point


int analogBuffer[SCOUNT];     // store the analog value in the array,
read from ADC

int analogBufferTemp[SCOUNT];

int analogBufferIndex = 0;

int copyIndex = 0;


float averageVoltage = 0;

float tdsValue = 0;

float temperature = 23;       // current temperature for compensation

// pin GPIO ultrasonik

const int pin_echo = 5; //GPIO15 D8

const int pin_trig = 6;//GPIO0 D3


float jarak;

float read_srf() {
```

```
  unsigned int duration;
  float distance;
  // trigger off
  digitalWrite(pin_trig, LOW);
  delayMicroseconds(5);
  // trigger on
  digitalWrite(pin_trig, HIGH);
  delayMicroseconds(10);
  // trigger off
  digitalWrite(pin_trig, LOW);
  // baca pulsa
  duration = pulseIn(pin_echo, HIGH);
  // konversi ke cm
  distance = duration * 0.034 / 2;
  return distance;
}

// median filtering algorithm
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;
}
SoftwareSerial mySerial(2, 3);
#include <ArduinoJson.h>
void setup() {
  Serial.begin(9600);
```

```
  mySerial.begin(9600);
  pinMode(TdsSensorPin, INPUT);
  pinMode(pin_trig, OUTPUT);
  pinMode(pin_echo, INPUT);
}

void loop() {
  static unsigned long analogSampleTimepoint = millis();
  if (millis() - analogSampleTimepoint > 40U) { //every 40
milliseconds,read the analog value from the ADC
    analogSampleTimepoint = millis();
    analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin);
//read the analog value and store into the buffer
    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];


    // read the analog value more stable by the median filtering
algorithm, and convert to voltage value
    averageVoltage = getMedianNum(analogBufferTemp, SCOUNT) *
(float)VREF / 1024.0;


    //temperature compensation formula: fFinalResult(25^C) =
fFinalResult(current)/(1.0+0.02*(fTP-25.0));
    float compensationCoefficient = 1.0 + 0.02 * (temperature - 25.0);
    //temperature compensation
    float compensationVoltage = averageVoltage /
compensationCoefficient;


    //convert voltage value to tds value
    tdsValue = (133.42 * compensationVoltage * compensationVoltage *
compensationVoltage - 255.86 * compensationVoltage *
compensationVoltage + 857.39 * compensationVoltage ) * 0.5;


    Serial.print("voltage:");
    Serial.print(averageVoltage, 2);
    Serial.print("V   ");
    Serial.print("TDS Value:");
    Serial.print(tdsValue, 0);
    Serial.println("ppm");
```

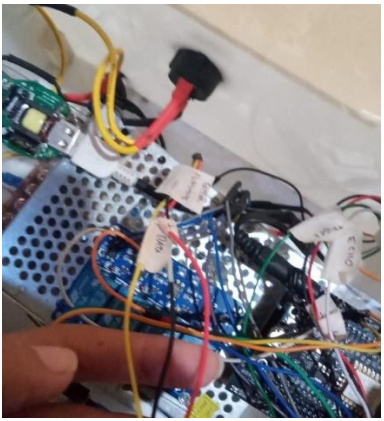
```
    delay(30);
   }
  }
  StaticJsonBuffer<1000> jsonBuffer;
  JsonObject& data = jsonBuffer.createObject();
  data["data1"] = tdsValue;
  data["data2"] = read_srf();
  data.printTo(mySerial);
  jsonBuffer.clear();

}
```

LAMPIRAN C
DOKUMENTASI
Hasil Alat dan Pengujian

Alat tampak depan

Alat tampak belakang

Gambar tampak depan box panel

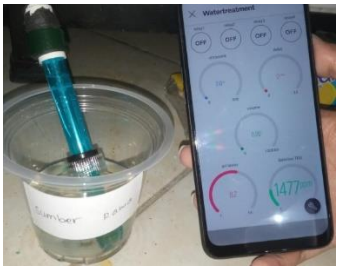
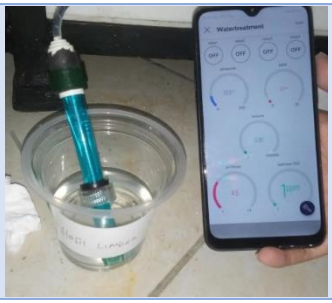


Proses perarakitan sistem

Proses pembentukan flok pada uji coba

Pengujian sensor pH

Pengukuran Dengan Alat	Pengukuran Dengan Sensor
	
	
	

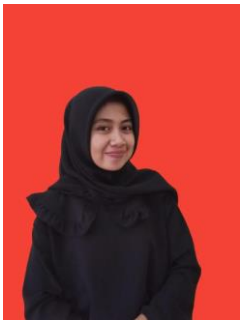
	 pH 8.1 ATC	 Watertreatment OFF OFF OFF OFF 1477
	 pH 3.1 ATC	 Watertreatment
	 pH 3.1 ATC	 Watertreatment OFF OFF OFF OFF 1493

Pengujian sensor TDS

Pengukuran Dengan Alat	Pengukuran Dengan Sensor
	
	
	


TDS&EC
Hasil Sungai
139
TDS&EC
Sumber
341
TDS&EC
Sumber Sungai
477

BIODATA PENULIS



Nama	: Anisa Tyas Ambarini
Tempat/Tanggal Lahir	: Cilacap, 30 Desember 2002
Alamat	: Jl. Rejasari RT 03 RW 03, Kel. Gunung Telu, Kec. Karang Pucung, Kab. Cilacap, Jawa Tengah
Email	: ambaranisa07@gmail.com
Telepon/HP	: 081334164461
Hobi	: Traveling
Moto	: Jalani , Syukuri , Nikmati

Riwayat Pendidikan

- SD Negeri Gunung Telu 01 — Tahun 2008 - 2014
- SMP Negeri 3 Karang Pucung — Tahun 2014 - 2017
- SMK Negeri 1 Karang Pucung — Tahun 2017 - 2020
  Teknik Instalasi Tenaga Listrik
- Politeknik Negeri Cilacap — Tahun 2020 - 2023
  Prodi D3 Teknik Listrik