

DAFTAR PUSTAKA

Adriani Noor. 2000. Pengelolaan Kualitas Air Tambak Bandeng, Samarinda. Loka Pengkajian Teknologi Pertanian. Samarinda

Anak Kendali. Arduino, Tutorial Mengakses Sensor TDS dan Kalibrasi. <https://www.anakkendali.com/arduino-cara-mengakses-sensor-tds-dan-kalibrasi/> (diakses tanggal 9 Januari 2021)

Anton. 2018. SIM800L & SIM800L EVB v2.0 - Variations and wiring. <https://forum.arduino.cc/index.php?topic=525769.0> (diakses tanggal 4 Januari 2021)

Arduino. Arduino Uno Rev3 Schematic. http://cse.unl.edu/~carrick/courses/2012/236/Arduino_Uno_Rev3-schematic.pdf (diakses tanggal 13 Desember 2020)

Arduino. Arduino Uno SMD. <http://arduino.cc/en/Main/ArduinoBoardUnoSMD> (diakses tanggal 13 Desember 2020)

Banzi Massimo. 2009. Getting Started With Arduino. Published by Make: Books, an imprint of Maker Media, a division of O'Reilly Media, Inc.

Effendi, Hefni. 2003. Telaah Kualitas Air : Bagi Pengelolaan Sumber Daya dan Lingkungan Perairan. Penerbit : Kanisius. Yogyakarta

ElectronicWings. LM35 Temperature Sensor. <https://www.electronicwings.com/sensors-modules/lm35-temperature-sensor> (diakses tanggal 4 Januari 2021)

Fahkrudin, Teuku. 2011. Analisis Perbandingan Kualitas Jaringan 2G GSM Frekuensi 900 Mhz Dan 1800 Mhz Berdasarkan Data Drive Test. Tesis: USU

Henhen Suherman, Iskandar, dan Sri Astuy. 2002. Studi Kualitas Air Pada Petakan Pendederan Benih Udang Windu (*Penaeus Monodon Fab.*) Di Kabupaten Indramayu. Fakultas Pertanian Universitas Padjajaran. Bandung.

Hobby King. Zippy 5000mAh 2S1P 30C Hardcase. https://hobbyking.com/en_us/zippy-5000mah-2s1p-30c-hardcase-%20pack.html?__store=en_us (diakses tanggal 9 Januari 2021)



an, Bagas. 2012. Prinsip Kerja Relay. <s://bagaskawarasan.wordpress.com/tag/prinsip-kerja-relay/>. (diakses gal 3 Maret 2021)

- Klinkmann. GSM-Control SMS Gateway (GSM-Control standalone).
<https://www.klinkmann.com/products/klinkmann/wireless-automation-software/gsm-control-sms-gateway-gsm-control-standalone/> (diakses tanggal 4 Januari 2021)
- Mappong, Suriani. 2020. Produksi Udang Windu Kabupaten Pinrang Pasok Pasar Jepang. <https://makassar.antaranews.com/berita/223624/produksi-udang-windu-kabupaten-pinrang-pasok-pasar-jepang> (diakses tanggal 11 Desember 2020)
- Shato Media. Sensor Suhu DS18B20. <http://shatomedia.com/2008/12/sensor-suhu-DS18B20> (diakses tanggal 3 Maret 2021)
- Sidik, Nurcahyo. 2012. Aplikasi dan Teknik Pemrograman AVR Atmel. Yogyakarta: Penerbit Andi.
- Texas Instruments. 1999. LM35 Precision Centigrade Temperature Sensors. Texas.
- Valery, Vodovozov. 2010. "Introduction to Power Electronic". Ventus Publishing ApS.
- Vernier. pH Sensor. <https://www.vernier.com/product/ph-sensor> (diakses tanggal 4 Januari 2021)
- Wikipedia. pH Meter. https://en.wikipedia.org/wiki/PH_meter (diakses tanggal 4 Januari 2021)



LAMPIRAN

Lampiran 1. Source Code Monitoring dan Controlling Kualitas Air

```
#include <SoftwareSerial.h>
SoftwareSerial gprsSerial(2,3);

#include <String.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <EEPROM.h>
#include "GravityTDS.h"
#include <DS3231.h>
#include <Wire.h>

const int analogInPin = A0;
int sensorValue = 0;
unsigned long int avgValue;
float b;
int buf[10],temp;

#define TdsSensorPin A1
GravityTDS gravityTds;
DS3231 clock_;
bool century = false;
bool hl2Flag;
bool pmFlag;

byte f1 = 0;
byte f2 = 0;

float temperature = 25,tdsValue = 0;

#define ONE_WIRE_BUS A2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

void setup()
{
  gprsSerial.begin(9600); // the GPRS baud rate
  Serial.begin(9600);

  sensors.begin(); // Start sensor
  Wire.begin();
  gravityTds.setPin(TdsSensorPin);
  gravityTds.setAref(5.0); //reference voltage on ADC, default
5.0V on Arduino UNO
  tyTds.setAdcRange(1024); //1024 for 10bit ADC;4096 for
DC
  tyTds.begin(); //initialization

  de(8, OUTPUT);
  alWrite(8, LOW);
```



```

    delay(1000);
}

void loop()
{
    //ph
    for(int i=0;i<10;i++)
    {
        buf[i]=analogRead(analogInPin);
        delay(10);
    }
    for(int i=0;i<9;i++)
    {
        for(int j=i+1;j<10;j++)
        {
            if(buf[i]>buf[j])
            {
                temp=buf[i];
                buf[i]=buf[j];
                buf[j]=temp;
            }
        }
    }
    avgValue=0;
    for(int i=2;i<8;i++)
    avgValue+=buf[i];
    float pHVol=(float)avgValue*5.0/1024/6;
    float pHValue = -5.70 * pHVol + 21.34;

    if(pHValue <=6.5){
        f1 = 1;
    }
    else if(pHValue >= 8.5){
        f1 = 1;
    }
    else{
        f1 = 0;
    }

    //suhu
    sensors.requestTemperatures();

    //tds
    gravityTds.setTemperature(temperature); // set the
    temperature and execute temperature compensation
    gravityTds.update(); //sample and calculate
    tdsValue = gravityTds.getTdsValue(); // then get the value
    if(tdsValue <= 15){
        f2 = 1;
    }
    else if(tdsValue >= 30){
        f2 = 1;
    }
    else{

```



```

        f2 = 0;
    }

    if((f1+f2) > 0) digitalWrite(8, HIGH);
    else digitalWrite(8, LOW);

    float ph = pHValue;
    float suhu = sensors.getTempCByIndex(0);
    float tds = tdsValue * 0.001;
    delay(100);

    Serial.print("pH = ");
    Serial.println(ph);
    Serial.print("Suhu = ");
    Serial.print(suhu);
    Serial.println(" °C");
    Serial.print("TDS = ");
    Serial.print(tds, 3);
    Serial.println(" ppt");

    if (gprsSerial.available())
        Serial.write(gprsSerial.read());

    gprsSerial.println("AT");
    delay(1000);

    gprsSerial.println("AT+CPIN?");
    delay(1000);

    gprsSerial.println("AT+CREG?");
    delay(1000);

    gprsSerial.println("AT+CGATT?");
    delay(1000);

    gprsSerial.println("AT+CIPSHUT");
    delay(1000);

    gprsSerial.println("AT+CIPSTATUS");
    delay(2000);

    gprsSerial.println("AT+CIPMUX=0");
    delay(2000);

    ShowSerialData();

    gprsSerial.println("AT+CSTT=\"indosatgprs\"");
    delay(1000);

```



```
SerialData();
```

```
Serial.println("AT+CIICR");//bring up wireless connection
delay(3000);
```

```

ShowSerialData();

gprsSerial.println("AT+CIFSR");//get local IP adress
delay(2000);

ShowSerialData();

gprsSerial.println("AT+CIPSPRT=0");
delay(3000);

ShowSerialData();

gprsSerial.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\",
\"80\");//start up the connection
delay(6000);

ShowSerialData();

gprsSerial.println("AT+CIPSEND");//begin send data to remote
server
delay(4000);
ShowSerialData();

String str="GET
https://api.thingspeak.com/update?api_key=QJQ9WYZMYHDQDHI0&field1
=" + String(ph) + "&field2="+String(suhu) + "&field3="+String(tds);
Serial.println(str);
gprsSerial.println(str);//begin send data to remote server

delay(4000);
ShowSerialData();

gprsSerial.println((char)26);//sending
delay(5000);//waitting for reply, important! the time is base
on the condition of internet
gprsSerial.println();

ShowSerialData();

gprsSerial.println("AT+CIPSHUT");//close the connection
delay(100);
ShowSerialData();
}
void ShowSerialData()
{
while(gprsSerial.available()!=0)
Serial.write(gprsSerial.read());
delay(5000);
}
}

```

