DAFTAR PUSTAKA

[1] D. R. Tisna, B. J. Martin Putra, T. Maharani, and H. Hasnira, "Metode Peningkatan Akurasi pada Sensor TDS Berbasis Arduino untuk Nutrisi Air Menggunakan Regresi Linier," *J. Integr.*, vol. 14, no. 1, pp. 61–68, 2022, doi: 10.30871/ji.v14i1.3906.

[2] A. Rosyidi, R. Alfita, and K. Joni, "Rancang Bangun Smart River System Untuk Menentukan Kualitas Air Sungai," *Indones. J. Eng. Technol.*, vol. 2, no. 1, pp. 11–17, 2019, doi: 10.26740/inajet.v2n1.p11-17.

[3] F. Ariska, I. Hadi, and Lindawati, "Perancangan Alat Pendeteksi Kualitas Air Berbasis Android," *Semin. Nas. Inov. dan Apl. Teknol. di Ind. 2019*, pp. 173–176, 2019.

[4] S. A. Akbar, D. B. Kalbuadi, and A. Yudhana, "Online Monitoring Kualitas Air Waduk Berbasis Thingspeak," *Transmisi*, vol. 21, no. 4, pp. 109–115, 2019, doi: 10.14710/transmisi.21.4.109-115.

[5] "Data BPS Pencemaran Air 2018 sampai 2021." https://www.bps.go.id/indicator/168/959/1/banyaknya-desa-kelurahan-menurut-jenis-pencemaran-lingkungan-hidup.html

[6] R. P. Wirman, I. Wardhana, and V. A. Isnaini, "Kajian Tingkat Akurasi Sensor pada Rancang Bangun Alat Ukur Total Dissolved Solids (TDS) dan Tingkat Kekeruhan Air," *J. Fis.*, vol. 9, no. 1, pp. 37–46, 2019, doi: 10.15294/jf.v9i1.17056.

[7] T. M. Raihan, "Sistem pemantauan kualitas air menggunakan esp32 dengan fuzzy logic sugeno berbasis android," 2022.

[8] Unicef, "No TIndonesia: Hampir 70 persen sumber air minum rumah tangga tercemar limbah tinjaitle", [Online]. Available: https://www.unicef.org/indonesia/id/siaran-pers/indonesia-hampir-70-persen-sumber-air-minum-rumah-tangga-tercemar-limbah-tinja

[9] R. Rahman and F. B. Gultom, "DESAIN DAN PERANCANGAN INSTRUMEN MONITORING KEKERUHAN AIR DENGAN SISTEM REAL TIME CLOCK (RTC)," vol. 5, no. 1, 2022.

[10] T. Lembaran, T. Lembaran, T. Lembaran, and T. Lembaran, "BERITA NEGARA," no. 55, 2023.

[11] "Harga Turbiditymeter", [Online]. Available: https://www.tokopedia.com/rekomendasi/10641019121?srsltid=

AfmBOooFLKCSjdpfB-W4XP0Ai-f937gVFROtxr-Z6nl6AwT3vC25ZBfm3Lk

[12] "Harga Alat Ukur Kualitas Air 5in1", [Online]. Available: https://www.tokopedia.com/rekomendasi/10641019121?srsltid=AfmBOooFLKCSjdpfB-W4XP0Ai-f937gVFROtxr-Z6nl6AwT3vC25ZBfm3Lk

[13] P. Sistem *et al.*, "PERANCANGAN SISTEM MONITORING KELAYAKAN KUALITAS AIR BERSIH METODE FUZZY LOGIC DESIGN OF CLEAN WATER QUALITY APPROPRIATENESS MONITORING SYSTEM WITH MULTISENSOR FOR HYGIENE SANITATION WATER USING," vol. 8, no. 2, pp. 992–1000, 2021.

[14] U. T. Khasanah, E. Puspita, B. N. Iman, and F. Arifin, "Rancang Bangun Alat Ukur Kualitas Air Menggunakan Metode Storet Dengan Parameter Suhu , pH , Kekeruhan dan TDS," pp. 62–71, 2021.

[15] F. Chuzaini, D. Wedi, S. Mata, A. Grogolan, D. Ngunut, and S. Tirta, "IoT MONITORING KUALITAS AIR DENGAN MENGGUNAKAN SENSOR SUHU , pH , DAN TOTAL DISSOLVED SOLIDS (TDS)," vol. 11, pp. 46–56, 2022.

[16] M. P. Fisika, "Metode pengukuran fisika," 2020.

[17] M. F. Zarkashie, *Rancang Bangun Sistem Pengukuran Kualitas Air Untuk Keperluan Higiene Sanitasi Berbasis Arduino Uno*, vol. 2, no. 12. 2021.

[18] P. Wibisono, "Rancang Bangun Sistem Monitoring Pencemaran Air Berdasarkan Parameter Total Dissolved Solids (Tds) Dan Kekeruhan," 2022, [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/66974

[19] K. Mlk and Y. F. Lopez, "Konsentrasi Larutan dalam Satuan PPM," pp. 1–3.

[20] R. Y. Endra and U. B. Lampung, "Analisis Cara Kerja Sensor Ph-E4502c Menggunakan Mikrokontroler Arduino Uno Untuk Merancang Alat Pengendalian Ph Air Pada Tambak Disusun Oleh :," no. December, 2020, doi: 10.13140/RG.2.2.32110.84809.

[21] A. Maulana, N. Ratama, F. Teknik, T. Informatika, and U. Pamulang, "SISTEM MONITORING DAN CONTROLING TINGKAT KEKERUHAN AIR PADA AQUARIUM MENGGUNAKAN METODE," vol. 1, no. 2, pp. 167–171, 2023.

[22] "sensor TDS DfRobot", [Online]. Available: https://www.dfrobot.com/product-1662.html

[23] "Sensor pH DFrobot", [Online]. Available: https://www.dfrobot.com/product-1782.html

[24] T. M. Kawinda *et al.*, "Penerapan Teknologi Internet Of Things Pada Hidroponik Cabai Rawit Dengan Sistem Dutch Bucket Menggunakan ESP32 Dan Blynk Application Of Internet Of Things Technology On Hydroponic Of Chillies With Dutch Bucket System Using ESP32 And Blynk," vol. 8, no. 6, pp. 3377–3385, 2022.

[25] I. Naria, "Sistem Monitoring Smart Fish Farm dengan Menggunakan Interface Website," p. 18, 2017.

LAMPIRAN A

Listing Program Arduino IDE

```
#include "GravityTDS.h"
#include <Wire.h>
#include <RtcDS3231.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#define ph_pin A0
#define turbi_pin A1
#define TdsSensorPin A2
#define t1 4
#define t2 5
#define t3 6
#define t4 7

float temperature = 25;
unsigned long pre   fSD;
unsigned long pref;
unsigned long proveus;
char daysOfTheWeek[7][12] = {"minggu", "senin", "selasa", "rabu",
"kamis", "jumat", "sabtu"};
int detik, menit, jam, hari;
int data_ph, data_turbi;
float data_tds;
float  temp;
float  temp2;
int buton1, buton2, buton3, buton4;
//variable sensor ph
float phEquals;
double Vph;
double ph_stp;
double vph7 = 1.55;
double vph4 = 2.14;
//variable turbi
double voltTurbi;
double hasilKeruh;
float pengurangan;
String indikatorsuhu, indikatorph, indikatorturbi, indikatortds;

//for logic
int count_logic = 0;
int countTemp, countPh, countTurbi, countTds;
String idkTemp = " NO", idkPh = " NO", idkTurbi = " NO", idkTds = "
NO";
int timeSD = 1;
const int chipSelect = 10;
String updateTime;

float tempVal, tempVal2, phVal, turbiVal, tdsVal;

GravityTDS gravityTds;
RtcDS3231<TwoWire> Rtc(Wire);
OneWire pin_DS18B20(8);
DallasTemperature DS18B20(&pin_DS18B20);
LiquidCrystal_I2C lcd(0x27, 20, 4);
SoftwareSerial ss(2, 3);
void setup() {
  Serial.begin(9600);
  ss.begin(9600);//kecepatan komunikasi serial esp
  while (!Serial) {
    ;
  }
  lcd.init();
  lcd.backlight();
  rtcSet();
  DS18B20.begin();
  pinMode(ph_pin, INPUT);
  pinMode(turbi_pin, INPUT);
  gravityTds.setPin(TdsSensorPin);
  gravityTds.setAref(5.0);
  gravityTds.setAdcRange(1024);
  gravityTds.begin();  //initialization
  pinMode(t1, INPUT_PULLUP);
  pinMode(t2, INPUT_PULLUP);
  pinMode(t3, INPUT_PULLUP);
  pinMode(t4, INPUT_PULLUP);
}

void loop() {
  readingSensor();
  logicAll();
  printing();
  Serial.println("cek");
}

#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() );
  updateTime = datestring;
}
```

```
void printing() {
  ss.print("{");
  ss.print(String(temp));
  ss.print(",");
  ss.print(String(phEquals));
  ss.print(",");
  ss.print(String(hasilKeruh));
  ss.print(",");
  ss.print(String(data_tds));
  ss.print(",");
  ss.print(String(updateTime));
  ss.print(",");
  ss.print(String(timeSD));
  ss.println("}");
}

void readingSensor() {
  DS18B20.requestTemperatures();
  temp = DS18B20.getTempCByIndex(0);
  temp2 = DS18B20.getTempCByIndex(1);
  RtcDateTime now = Rtc.GetDateTime();
  printDateTime(now);
  buton1 = digitalRead(t1);
  buton2 = digitalRead(t2);
  buton3 = digitalRead(t3);
  buton4 = digitalRead(t4);

  //  printDateTime(now);
  jam = now.Hour();//jam saat ini
  menit = now.Minute();//menit saat ini
  detik = now.Second();//detik saat ini

  data_ph = analogRead(ph_pin);
  Vph = 5 / 1023.0 * data_ph;
  ph_stp = (vph4 - vph7) / (7 - 4);
  phEquals = 7.00 + ((vph7 - Vph) / ph_stp);

  data_turbi = analogRead(turbi_pin);
  voltTurbi = data_turbi * (5.0 / 1023);
  hasilKeruh = (110.00 - (voltTurbi / 4.21) * 100.00);

  gravityTds.setTemperature(temperature);
  gravityTds.update();
  data_tds = gravityTds.getTdsValue();
}

void logicAll() {
  if (buton4 == LOW) {
    count_logic = count_logic + 1;
    lcd.clear();
  }
  if (buton1 == LOW) {
    count_logic = count_logic - 1;
    lcd.clear();
  }

  switch (count_logic ) {
    case 0:
      lcd.setCursor(0, 1);
      lcd.print("     ALAT UKUR  ");
      lcd.setCursor(0, 2);
      lcd.print("    KUALITAS AIR  ");
      break;
    case 1:
      lcd.setCursor(0, 1);
      lcd.print("USERNAME :Ta anisatu ");
      lcd.setCursor(0, 2);
      lcd.print("Password : 00000000");
      break;
```

```
case 2:
  if (buton3 == LOW) {
    countTemp = countTemp + 1;
  }
  else if (buton2 == LOW) {
    countTemp = countTemp - 1;
  }
  if (countTemp >> 1) {
    countTemp = 0;
  }

  if (countTemp == 1) {
    idkTemp = " YES";
  }
  else if (countTemp == 0) {
    idkTemp = " NO ";
  }

  lcd.setCursor(0, 0);
  lcd.print(">Suhu Air&Udara" + String (idkTemp));
  lcd.setCursor(0, 1);
  lcd.print(" pH            " + String (idkPh));
  lcd.setCursor(0, 2);
  lcd.print(" Kekeruhan     " + String (idkTurbi));
  lcd.setCursor(0, 3);
  lcd.print(" Zat Padat     " + String (idkTds));
  break;

case 3:
  if (buton3 == LOW) {
    countPh = countPh + 1;
  }
  else if (buton2 == LOW) {
    countPh = countPh - 1;
  }
```

```
    if (countPh >> 1) {
      countPh = 0;
    }

    if (countPh == 1) {
      idkPh = " YES";
    }
    else if (countPh == 0) {
      idkPh = " NO ";
    }

    lcd.setCursor(0, 0);
    lcd.print(" Suhu Air&Udara" + String (idkTemp));
    lcd.setCursor(0, 1);
    lcd.print(">pH            " + String (idkPh));
    lcd.setCursor(0, 2);
    lcd.print(" Kekeruhan     " + String (idkTurbi));
    lcd.setCursor(0, 3);
    lcd.print(" Zat Padat     " + String (idkTds));
    break;

  case 4:
    if (buton3 == LOW) {
      countTurbi = countTurbi + 1;
    }
    else if (buton2 == LOW) {
      countTurbi = countTurbi - 1;
    }
    if (countTurbi >> 1) {
      countTurbi = 0;
    }

    if (countTurbi == 1) {
      idkTurbi = " YES";
    }
```

```
      else if (countTurbi == 0) {
        idkTurbi = " NO ";
      }

      lcd.setCursor(0, 0);
      lcd.print(" Suhu Air&Udara" + String (idkTemp));
      lcd.setCursor(0, 1);
      lcd.print(" pH            " + String (idkPh));
      lcd.setCursor(0, 2);
      lcd.print(">Kekeruhan     " + String (idkTurbi));
      lcd.setCursor(0, 3);
      lcd.print(" Zat Padat     " + String (idkTds));
      break;

    case 5:
      if (buton3 == LOW) {
        countTds = countTds + 1;
      }
      else if (buton2 == LOW) {
        countTds = countTds - 1;
      }
      if (countTds >> 1) {
        countTds = 0;
      }

      if (countTds == 1) {
        idkTds = " YES";
      }
      else if (countTds == 0) {
        idkTds = " NO ";
      }

      lcd.setCursor(0, 0);
      lcd.print(" Suhu Air&Udara" + String (idkTemp));
      lcd.setCursor(0, 1);
```

```
  lcd.print(" pH            " + String (idkPh));
  lcd.setCursor(0, 2);
  lcd.print(" Kekeruhan     " + String (idkTurbi));
  lcd.setCursor(0, 3);
  lcd.print(">Zat Padat     " + String (idkTds));
  break;

case 6:
  if (buton3 == LOW) {
    timeSD = timeSD + 1;
    //delay(250);
  }
  else if (buton2 == LOW) {
    timeSD = timeSD - 1;
    //delay(250);
  }
  lcd.setCursor(0, 0);
  lcd.print("  TIME DATA UPDATE  ");
  lcd.setCursor(0, 2);
  lcd.print("     val = " + String(timeSD) + " Sec ");
  break;

case 7:
  if (countTemp == 1) {
    tempVal = temp;
    tempVal2 = temp2;
  }
  else {
    tempVal = 0;
  }
  if (countPh == 1) {
    phVal = phEquals;
  }
  else {
    phVal = 0;
```

```
      }
      if (countTurbi == 1) {
        turbiVal = hasilKeruh;
      }
      else {
        hasilKeruh = 0;
      }
      if (countTds == 1) {
        tdsVal = data_tds;
      }
      else {
        tdsVal = 0;
      }
      if ((millis() - proveus) > 500) {
        lcd.setCursor(0, 0);
        lcd.print("Suhu Udara: " + String (tempVal) + char(223) + "C");
        lcd.setCursor(0, 1);
        lcd.print("Suhu Air  : " + String (tempVal2) + char(223) + "C");
        lcd.setCursor(0, 2);
        lcd.print("Selisih   : " + String (tempVal - tempVal2) + char(223) +
"C");
        lcd.setCursor(0, 3);
        lcd.print("pH        : " + String (phVal));
        proveus = millis();
      }
      //send_sd_card(timeSD, tempVal, phVal, turbiVal, tdsVal);
      break;

    case 8:
      if (countTemp == 1) {
        tempVal = temp;
        tempVal2 = temp2;
      }
      else {
        tempVal = 0;
      }
      if (countPh == 1) {
        phVal = phEquals;
      }
      else {
        phVal = 0;
      }
      if (countTurbi == 1) {
        turbiVal = hasilKeruh;
      }
      else {
        hasilKeruh = 0;
      }
      if (countTds == 1) {
        tdsVal = data_tds;
      }
      else {
        tdsVal = 0;
      }
      if ((millis() - proveus) > 500) {
        lcd.setCursor(0, 0);
        lcd.print("Zat Padat: " + String (tdsVal) + " PPM ");
        lcd.setCursor(0, 1);
        lcd.print("Kekeruhan: " + String (turbiVal) + " NTU ");
        proveus = millis();
      }
      break;

    case 9:
      if (countTemp == 1) {
        tempVal = temp;
        tempVal2 = temp2;
        if (temp - temp2 >= 3 && temp - temp2 <= 4) {
          lcd.clear();
          indikatorsuhu = "Layak";
```

```
  }
  else {
    indikatorsuhu = "Tdk Layak";
  }
}
else {
  tempVal = 0;
  indikatorsuhu = " ----- ";
}
if (countPh == 1) {
  phVal = phEquals;
  if (phEquals >= 6, 5 && phEquals <= 8, 5) {
    lcd.clear();
    indikatorph = "Layak";
  }
  else {
    indikatorph = "Tdk Layak";
  }
}
else {
  phVal = 0;
  indikatorph = " ----- ";
}
if (countTurbi == 1) {
  turbiVal = hasilKeruh;
  if (hasilKeruh <= 3) {
    indikatorturbi = "Layak";
    lcd.clear();
  }
  else {
    indikatorturbi = "Tdk Layak";
  }
}
else {
  hasilKeruh = 0;
```

```
    indikatorturbi = " ----- ";
  }
  if ((millis() - proveus) > 500) {
    lcd.setCursor(0, 0);
    lcd.print("     Keterangan");
    lcd.setCursor(0, 1);
    lcd.print("Suhu     : " + String (indikatorsuhu));
    lcd.setCursor(0, 2);
    lcd.print("pH       : " + String (indikatorph));
    lcd.setCursor(0, 3);
    lcd.print("Kekeruhan: " + String (indikatorturbi));
    proveus = millis();
  }
  break;

case 10:
  if (countTds == 1) {
    tdsVal = data_tds;
    if (data_tds <= 300) {
      indikatortds = "Layak";
      lcd.clear();
    }
    else {
      indikatortds = "Tdk Layak";
    }
  }
  else {
    tdsVal = 0;
    indikatortds = " ----- ";
  }
  if ((millis() - proveus) > 500) {
    lcd.setCursor(0, 0);
    lcd.print("     Keterangan");
    lcd.setCursor(0, 1);
    lcd.print("Zat Padat: " + String (indikatortds));
```

```
        proveus = millis();
      }
      break;
  }
}

void rtcSet() {
  Rtc.Begin();
  RtcDateTime compiled = RtcDateTime(_DATE, __TIME_);
  printDateTime(compiled);
  Serial.println();
  if (!Rtc.IsDateTimeValid())
  {
    if (Rtc.LastError() != 0)
    {
      Serial.print("RTC communications error = ");
      Serial.println(Rtc.LastError());
    }
    else
    {
      Serial.println("RTC lost confidence in the DateTime!");
      Rtc.SetDateTime(compiled);
    }
  }
  if (!Rtc.GetIsRunning())
  {
    Serial.println("RTC was not actively running, starting now");
    Rtc.SetIsRunning(true);
  }

  RtcDateTime now = Rtc.GetDateTime();
  if (now < compiled)
  {
    Serial.println("RTC is older than compile time!  (Updating
DateTime)");
```

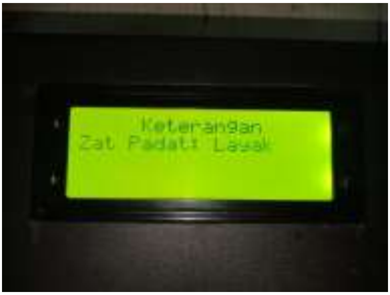
```
    Rtc.SetDateTime(compiled);
  }
  Rtc.Enable32kHzPin(false);
  Rtc.SetSquareWavePin(DS3231SquareWavePin_ModeNone);
}

//program esp32
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <SoftwareSerial.h>
#include "FS.h"
#include "SD.h"
#include <SPI.h>

#define SD_CS 5
String dataMessage;
unsigned long timeShow;
int flagRun;
char auth[] = "v-8idQza3yPOG56Q3MGaVqszAonmF7Wi";//kode
autentikasi blynk
char ssid[] = "cohc";//nama wifi
char pass[] = "00000000";

unsigned long  prefData;
unsigned long pref;
String dataSerial;
String tempStr, phStr, turbiStr, tdsStr, jamStr, timeStr;
float temp, ph , turbi , tds ;
int timeInt ;
SoftwareSerial uno;
BlynkTimer timer;
 WidgetLCD lcdh(V4);
```

```
void myTimerEvent()
{
  Blynk.virtualWrite(V0, temp);
  Blynk.virtualWrite(V1, ph);
  Blynk.virtualWrite(V2, turbi);
  Blynk.virtualWrite(V3, tds);
  show();//fungsi menampilkan lcd pada blynk
}

void setup() {
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  timer.setInterval(1000L, myTimerEvent);
  uno.begin(9600, SWSERIAL_8N1, 17, 16, false);

  SD.begin(SD_CS);
  if (!SD.begin(SD_CS)) {
    Serial.println("Card Mount Failed");
    return;
  }
  uint8_t cardType = SD.cardType();
  if (cardType == CARD_NONE) {
    Serial.println("No SD card attached");
    return;
  }
  Serial.println("Initializing SD card...");
  if (!SD.begin(SD_CS)) {
    Serial.println("ERROR - SD card initialization failed!");
    return;
  }
  File file = SD.open("/data.txt");
  if (!file) {
    Serial.println("File doens't exist");
    Serial.println("Creating file...");
    writeFile(SD, "/data.txt", "ESP32 and SD Card \r\n");
  }
  else {
    Serial.println("File already exists");
  }
  file.close();
}

void loop() {
  Blynk.run();
  timer.run();
  SerialRead();
  logSDCard();
}

void SerialRead() {
  if (uno.available() > 0) {
    dataSerial = uno.readStringUntil('\n');
    int data1 = dataSerial.indexOf('{');
    int data2 = dataSerial.indexOf(',', data1 + 1);
    int data3 = dataSerial.indexOf(',', data2 + 1);
    int data4 = dataSerial.indexOf(',', data3 + 1);
    int data5 = dataSerial.indexOf(',', data4 + 1);
    int data6 = dataSerial.indexOf(",", data5 + 1);
    int data7 = dataSerial.indexOf("}", data6 + 1);

    tempStr = dataSerial.substring(data1 + 1, data2);
    phStr = dataSerial.substring(data2 + 1, data3);
    turbiStr = dataSerial.substring(data3 + 1, data4);
    tdsStr = dataSerial.substring(data4 + 1, data5);
    jamStr = dataSerial.substring(data5 + 1, data6);
    timeStr = dataSerial.substring(data6 + 1, data7);

  }
  temp = tempStr.toFloat();
  ph = phStr.toFloat();
  turbi = turbiStr.toFloat();
  tds = tdsStr.toFloat();
  timeInt = timeStr.toInt();

  if ((millis() - pref) > 1000) {
    //        Serial.println("temp=" + String(temp));
    Serial.println("temp=" + String(temp) + "C");
    Serial.println("ph_out=" + String(phStr));
    Serial.println("turbi=" + String(turbiStr) + "ntu");
    Serial.println("tds=" + String(tds) + "ppm");
    Serial.println("jam=" + String(jamStr));
    Serial.println("jam=" + String(timeInt));

Serial.println("++++++++++++++++++++++++++++++++++++++++++++
+++++++++=");
    pref = millis();
  }
}



void logSDCard() {
  dataMessage = "TIME = " + String(jamStr) + ", " +
                "TEMP = " + String(temp) + " C"  + ", " +
                "phVal = " + String(phStr) + ", " +
                "turbiVal = " + String(turbiStr) + " NTU"  + ", " +
                "tdsVal = " + String(tdsStr) + " PPM" +
                "\n";

  if (timeInt > 0) {
    if ((millis() - prefData) > timeInt * 1000) {
      Serial.print("Save data: ");
      Serial.println(dataMessage);
      appendFile(SD, "/data.txt", dataMessage.c_str());
      prefData = millis();
    }
  }
}
// Write to the SD card (DON'T MODIFY THIS FUNCTION)
void writeFile(fs::FS &fs, const char * path, const char * message) {
  Serial.printf("Writing file: %s\n", path);
  File file = fs.open(path, FILE_WRITE);
  if (!file) {
    Serial.println("Failed to open file for writing");
    return;
  }
  if (file.print(message)) {
    Serial.println("File written");
  } else {
    Serial.println("Write failed");
  }
  file.close();
}
void appendFile(fs::FS &fs, const char * path, const char * message) {
  Serial.printf("Appending to file: %s\n", path);
  File file = fs.open(path, FILE_APPEND);
  if (!file) {
    Serial.println("Failed to open file for appending");
    return;
  }
  if (file.print(message)) {
    Serial.println("Message appended");
  } else {
    Serial.println("Append failed");
  }
  file.close();
}

void show() {
  if (millis() - timeShow > 3000) {
    lcdh.print     (0, 1, " MONITORING DATA");
    flagRun++;
    timeShow = millis();
  }

  switch (flagRun) {

    case 0:
      break;
    case 1:
      lcdh.print     (0, 1, " TEMP=         " + String(char(223)) + "C ");
      lcdh.print(6, 1, String (temp));
      break;
    case 2:
      lcdh.print(0, 1, "   PH=           ");
      lcdh.print(6, 1, String(ph));

      break;
    case 3:
      lcdh.print(0, 1, "TURBI=      NTU ");
      lcdh.print(6, 1, String (turbi));
      break;
    case 4:
      lcdh.print(0, 1, "  TDS=      PPM ");
      lcdh.print(6, 1, String (tds));
      break;
    case 5:
      flagRun = 1;
      break;

  }
}
```

LAMPIRAN B

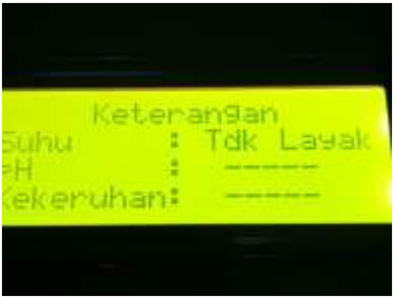
Dokumentasi

Gambar 1. Cairan *buffer* pH



```
    Keterangan
Zat Padat: Layak
```

Gambar 2. Tampilan Indikator Layak



```
    Keterangan
Suhu     : Tdk Layak
pH       : ------
Kekeruhan: ------
```

Gambar 3. Tampilan Indikator Tidak Layak

BIODATA PENULIS

Nama	:	Anisatu Zahroh
Tempat, Tanggal Lahir	:	Cilacap, 16 Desember 2002
Alamat	:	Jl. Anggrek RT 01 RW 07, Tritih Lor, Jeruklegi, Cilacap
Email	:	anisatuzahroh997@gmail.com
No telp/HP	:	088232083896
Hobi	:	Membaca
Motto	:	Sebaik-baiknya manusia adalah manusia yang bermanfaat untuk orang lain

Riwayat Pendidikan

• SD Negeri Tritih Lor 02	Tahun 2008-2014
• SMP Negeri 1 Jeruklegi	Tahun 2014-2017
• SMA Negeri 2 Cilacap Jurusan MIPA	Tahun 2017-2020
• Politeknik Negeri Cilacap Prodi D3 Teknik Elektronika	Tahun 2020-2023

Penulis telah mengikuti sidang Tugas Akhir pada tanggal 9 Agustus 2023 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md.)