

LAMPIRAN

CODE ARDUINO IDE

```
/* Mendefinisikan template ID, Nama, dan Token otentikasi pada Blynk */
#define BLYNK_TEMPLATE_ID "TMPL67a1oj8VY"
#define BLYNK_TEMPLATE_NAME "Pengontrol air budidaya kepiting"
#define BLYNK_AUTH_TOKEN "4vK28ri4krdDpyoTvgRVebAp1LBM82dJ"

// Mengimport Library yang dibutuhkan
#include <ESP8266WiFi.h> // Wifi ESP8266
#include <BlynkSimpleEsp8266.h> // Blynk
#include <Adafruit_ADS1X15.h> // Modul ADS1115
#include <Servo.h> // Motor Servo
#include <LiquidCrystal_I2C.h> // LCD I2C 16x2
#include <OneWire.h>
#include <DallasTemperature.h>

/* Mendefinisikan jaringan wifi dan token otentikasi pada Blynk */
char auth[] = "4vK28ri4krdDpyoTvgRVebAp1LBM82dJ";
char ssid[] = "Tara";
char pass[] = "00000000";

#define LCD_ADDRESS 0x27 // alamat LCD I2C 16x2

LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2); // Mendefinisikan LCD I2C
sebagai lcd
Adafruit_ADS1015 ads; // Mendefinisikan ADS1115 sebagai ads
BlynkTimer timer; // Mendefinisikan objek BlynkTimer
```

```

// Deklarasi variabel global
int16_t adc0, adc1, adc2, adc3; // Pin pada modul ADS1115
float volts0, volts1, volts2, volts3; // variabel untuk menampung tegangan
pada pin ADS1115

double Vclear = 2.02;
double tds1 = 0; // Turbidity 1

float pHValue; // pH output
float pHStep; // pH kalibrasi

String tds1Terbilang; // Turbidity 1 terbilang
String pHTerbilang;
String tds1drelay1;
String phdrelay2;
String temp;

// Kalibrasi
float pH4 = 3.054;
float pH7 = 2.682;

const int relay1 = D6; //pin2
const int relay2 = D7; //A0 FOR ARDUINO/ 36 FOR ESP

const int oneWireBus = D4;
OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);

int relayON = LOW; //relay nyala

```

```
int relayOFF = HIGH;

// Memulai program
void setup(void)
{
  Serial.begin(9600);
  sensors.begin();

  // Menghubungkan ke WiFi
  Blynk.begin(auth, ssid, pass);

  // Inisialisasi LCD
  lcd.init();
  lcd.clear();
  lcd.backlight();

  // Menampilkan Hello dan Loading pada LCD
  lcd.setCursor(0, 0);
  lcd.print("HELLO!");
  delay(1000);
  lcd.setCursor(0, 1);
  lcd.print("LOADING...");
  delay(1000);

  /* melakukan pengecekan wiring modul ADS1115
   * Jika modul ADS1115 tidak terhubung, maka tampilkan pesan
   * "Failed to initialize ADS" pada serial monitor
   */
  if (!ads.begin()) {
    Serial.println("Failed to initialize ADS.");
  }
}
```

```

    while (1);
}

/* Jika modul ADS1115 terhubung, maka tampilkan pesan
   "ADS1115 OK" pada LCD
*/
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("ADS1115 OK");
delay(1000);

// Setup 3V pada channel 0 & 1 modul ADS1115
ads.startComparator_SingleEnded(0, 1000);
ads.startComparator_SingleEnded(1, 1000);

pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
}

void loop(void)
{
    Blynk.run(); // Menjalankan Blynk
    timer.run(); // Menjalankan BlynkTimer
    // Send the command to get temperatures
    sensors.requestTemperatures();
    float temperatureC = sensors.getTempCByIndex(0);

    // Membaca data ADC channel 0 & 1 pada modul ADS1115
    adc0 = ads.readADC_SingleEnded(0);

```

```

adc1 = ads.readADC_SingleEnded(1);

// Mengukur tegangan channel 0 & 1 dan 3
// volts0 = adc0*(3.3/1024);
// volts1 = adc1*(3.3/1024);
volts0 = ads.computeVolts(adc0);
volts1 = ads.computeVolts(adc1);

// Turbidity
tds1 = 100.00 - (volts0 / Vclear)*100.00 ;

// Ph
phStep = (ph4 - ph7) / 3;
phValue = 7.00 + ((ph7 - volts1) / phStep);

tds1relay1 = "NTU1: " + tds1Terbilang + " | " + " SELENOID: " + relay1;
phdrelay2 = "PH : " + phTerbilang + " | " + " SELENOID : " + relay2;

Serial.print("AIN0: "); Serial.print(adc0); Serial.print(" Volt: ");
Serial.print(volts0);
Serial.print(" TDS1: "); Serial.print(tds1); Serial.print(" NTU ");
Serial.println(tds1Terbilang);
Serial.print("phValue: "); Serial.print(phValue); Serial.print(" Voltage: ");
Serial.println(volts1, 3);
Serial.print(temperatureC);
Serial.println("°C");

```

```
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("TDS1: "); lcd.print(tds1); lcd.print(" NTU");  
lcd.setCursor(0, 1);  
lcd.print("PH: "); lcd.print(phValue); lcd.print(" pH");  
delay(1000);
```

```
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("Temp: "); lcd.print(temperatureC); lcd.print("C");  
delay(1000);
```

```
// Mengirim data ke Blynk  
Blynk.virtualWrite(V0, temperatureC);  
Blynk.virtualWrite(V1, phValue);  
Blynk.virtualWrite(V2, tds1);  
Blynk.virtualWrite(V3, tds1drelay1);  
Blynk.virtualWrite(V4, phdrelay2);
```

```
if (phValue < 6) {  
  lcd.setCursor(0, 1);  
  lcd.print("SolenoidOff , ");  
  phTerbilang = "ASAM";  
  Serial.print(" Asam ");  
  digitalWrite(relay1, relayON);  
  digitalWrite(relay2, relayON);  
  Serial.print(" RelayOn , ");  
}  
if ((phValue > 6) && (phValue < 8)) {
```

```
lcd.setCursor(0, 1);  
lcd.print("SolenoidOff");  
phTerbilang = "SEDANG";  
Serial.print(" Netral ");  
digitalWrite(relay1, relayOFF);  
digitalWrite(relay2, relayOFF);  
Serial.print(" RelayOff ,");  
}  
if (phValue > 8) {  
  lcd.setCursor(0, 1);  
  lcd.print("SolenoidOn");  
  phTerbilang = "BASA";  
  Serial.print(" Basa ");  
  digitalWrite(relay1, relayOFF);  
  digitalWrite(relay2, relayOFF);  
  Serial.print(" RelayOff ,");  
}
```

```
// TDS 2 SELENOID
```

```
if (tds1 < 24.70) {  
  tds1Terbilang = "Jernih";  
  digitalWrite(relay1, relayON);  
  digitalWrite(relay2, relayON);  
} else if (24.70 < tds1 < 71.48) {  
  tds1Terbilang = "Sedang";  
  digitalWrite(relay1, relayOFF);  
  digitalWrite(relay2, relayOFF);  
} else if (tds1 > 71.48) {  
  tds1Terbilang = "Keruh";  
  digitalWrite(relay1, relayOFF);
```

```

    digitalWrite(relay2, relayOFF);
}

if (temperatureC < 30) {
    temp = "Dingin";
    digitalWrite(relay1, relayON);
    digitalWrite(relay2, relayON);
} else if (30 < temperatureC < 39) {
    temp = "Normal";
    digitalWrite(relay1, relayOFF);
    digitalWrite(relay2, relayOFF);
} else if (temperatureC > 39) {
    temp = "Panas";
    digitalWrite(relay1, relayON);
    digitalWrite(relay2, relayON);
}
delay(1000);
}

// Fungsi untuk memeriksa koneksi ke server Blynk
void checkConnection()
{
    if (!Blynk.connected()) {
        Serial.println("Reconnecting...");
        if (Blynk.connect()) {
            Serial.println("Reconnected!");
        } else {
            Serial.println("Reconnect failed!");
        }
    }
}
}

```

