Kode Program Rancang Bangun Sistem *Monitoring* Kualitas Air Pada Akuarium Berbasis *Internet of Things* (IOT)

- **blynk.ino**

```
#define BLYNK_TEMPLATE_ID "TMPL6EDtWaxxF"
#define BLYNK_TEMPLATE_NAME "Smart Akuarium"
#define BLYNK_AUTH_TOKEN "XmYuPb-dVtbK2uqOArpP3HPrwrmtu9Lu"

#include <BlynkSimpleEsp32.h>
#include <Wire.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <LiquidCrystal_I2C.h>

const int ledwifi = 13;
const int ledphatas = 19;
const int ledphbawah = 18;
const int ledindikator = 23;
const int ledakuarium = 14;
const int filterudara = 4;
const int filterair = 2;

const int sensorturbidity = 34;
float voltageturbidity;
int sensorValue;
int ntu; // Nephelometric Turbidity Units

const int PhSensor = 36;
int buf[10];
float ph (float voltage) {
  return 7 + (2.70 - voltage) -0.18;
}

const unsigned long delayph = 1000;
const unsigned long delayturbidity = 1300;
unsigned long lasttimeturbidity;
unsigned long lasttimeph;

//wifi setup
char ssid[] = "smartakuarium"; //Enter Your SSID
char passwordwifi[] = "smartakuarium"; //Enter Your Password
//end wifi setup

LiquidCrystal_I2C lcd (0x27, 16, 2);
BlynkTimer timer;

unsigned long timerwifi;
int counter ;
int tombolfilterudara = 0;
int tombolfilterair = 0;
int tombollampu = 0;
int modefilter = 0;


void setup()
{
  Serial.begin(115200); // for debugging purposes
  Wire.begin();

  pinMode(sensorturbidity, INPUT);
  pinMode(PhSensor, INPUT);
  pinMode(ledwifi, OUTPUT);
  pinMode(ledphatas, OUTPUT);
  pinMode(ledphbawah, OUTPUT);
  pinMode(ledindikator, OUTPUT);
  pinMode(ledakuarium, OUTPUT);
  pinMode(filterudara, OUTPUT);
  pinMode(filterair, OUTPUT);

  lcd.init();                  //Init the LCD
  lcd.backlight();             //Activate backlight

  lcd.setCursor (0, 0);
  lcd.print(F("INISIALISASI I/O"));
  Serial.println("INISIALISASI I/O");

  digitalWrite(ledwifi, HIGH);
  digitalWrite(ledphatas, LOW);
  digitalWrite(ledphbawah, LOW);
  digitalWrite(ledindikator, HIGH);
  digitalWrite(ledakuarium, HIGH);
  digitalWrite(filterudara, HIGH);
  digitalWrite(filterair, HIGH);

//connecting wifi
  Serial.print("Connecting wifi");
  lcd.setCursor (0, 0);
  lcd.print(F("Connecting Wi-Fi"));
  lcd.setCursor (0, 1);

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, passwordwifi);
  Serial.println("INISIALISASI I/O");

//connecting wifi
  Serial.print("Connecting Wifi");

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(200);
    if (++counter > 30)
```

```
      ESP.restart();
    Serial.print( "." );
  }
//end connecting wifi

  lcd.setCursor (0, 1);
  lcd.print(" --CONNECTED-- ");
  digitalWrite(ledwifi, LOW);
  delay (2000);
  lcd.clear();

  //connecting blynk
  Serial.print("Connecting blynk");
  lcd.setCursor (0, 0);
  lcd.print(F("Connecting BLYNK"));
  lcd.setCursor (0, 1);
  delay(1000);

  lcd.setCursor (0, 1);
  lcd.print(" --CONNECTED-- ");
  delay (2000);
  lcd.clear();

  digitalWrite(ledphatas, HIGH);
  digitalWrite(ledphbawah, HIGH);
  digitalWrite(ledakuarium, HIGH);
  digitalWrite(ledindikator, HIGH);
  digitalWrite(filterudara, HIGH);
  digitalWrite(filterair, HIGH);
  delay(1500);

  digitalWrite(ledphatas, LOW);
  digitalWrite(ledphbawah, LOW);
  digitalWrite(ledindikator, LOW);
  digitalWrite(filterudara, HIGH);
  digitalWrite(filterair, HIGH);

}

void loop() {
  Blynk.run();
    lcd.setCursor(0,0);
    lcd.print("   pH     NTU  ");

  turbiditysens2();
  phsens2();
  mode();
    // delay(10);
}
```

```
BLYNK_WRITE(V2) {
  tombolfilterudara = param.asInt();
}

BLYNK_WRITE(V3) {
  tombolfilterair = param.asInt();
}

BLYNK_WRITE(V6) {
  tombollampu =param.asInt();
  if (tombollampu == 1) {
    digitalWrite(ledakuarium, LOW);
  }
  else {
    digitalWrite(ledakuarium, HIGH);
  }
}

BLYNK_WRITE(V7) {
  modefilter = param.asInt();
}
```

- **mode.ino**

```
void mode() {
  switch(modefilter) {
    case 0:
      if (ntu > 25) {
        digitalWrite(filterair, LOW);
        digitalWrite(ledindikator, LOW);
        Blynk.virtualWrite(V4, 1);

      }
      else {
        digitalWrite(filterair, HIGH);
        digitalWrite(ledindikator, HIGH);
        Blynk.virtualWrite(V4, 0);
      }
      Blynk.virtualWrite(V7, 0);
    break;

    case 1:
      if (tombolfilterudara == 1) {
        digitalWrite(filterudara, LOW);
        Blynk.virtualWrite(V5, 1);
      }
      else {
        digitalWrite(filterudara, HIGH);
        Blynk.virtualWrite(V5, 0);
      }
```

```
   if (tombolfilterair == 1) {
    digitalWrite(filterair, LOW);
    digitalWrite(ledindikator, LOW);
    Blynk.virtualWrite(V4, 1);
   }
   else {
    digitalWrite(filterair, HIGH);
    digitalWrite(ledindikator, HIGH);
    Blynk.virtualWrite(V4, 0);
   }

   Blynk.virtualWrite(V7, 1);
  break;

  default:
  break;
 }
}
```

- **ph2.ino**

```
void phsens2() {
 for (int i=0; i<10; i++) {
  buf[i]=analogRead(36);
  delay(10);
 }

 float avgvalue=0;
 for (int i=0; i<10; i++)
 avgvalue+=buf[i];

 float phvol=(float)avgvalue*3.30/4096.0/10; //10
 float phvalue = 3.5 *phvol;
 // float phvalue= -5.70*phvol + 21.34;

 Serial.println("nilaiph:");
 Serial.println(phvalue, 2);
 Serial.println("nilai volt:");
 Serial.println(phvol);
  lcd.setCursor(0,0);
  lcd.print("   pH     NTU  ");
  lcd.setCursor(2,1);
  lcd.print(phvalue, 2);
  lcd.print(" ");

  Blynk.virtualWrite(V0, phvalue);

  if (phvalue >= 8) {
   digitalWrite(ledphatas, HIGH);
   digitalWrite(ledphbawah, LOW);
  }
```

```
    else if (phvalue <= 6) {
      digitalWrite(ledphatas, LOW);
      digitalWrite(ledphbawah, HIGH);
    }
    else {
      digitalWrite(ledphatas, LOW);
      digitalWrite(ledphbawah, LOW);
    }
}
```

- **sens2.ino**

```
void turbiditysens2() {
  if (millis() - lasttimeturbidity > delayturbidity) {
    sensorValue = analogRead(sensorturbidity);
    voltageturbidity = sensorValue * (5.0 / 4093.0); // Convert the analog reading (which goes from 0 - 4095) to a voltage (0 - 5V):
    ntu = map(voltageturbidity, 0.00, 5.10, 150, 0);
    Serial.println(voltageturbidity);
    Serial.println(ntu);
    // Serial.println(sensorValue);

    lcd.setCursor(0,0);
    lcd.print("   pH     NTU  ");
    lcd.setCursor(8,1);
    lcd.print(ntu);
    lcd.print(" NTU   ");

    lasttimeturbidity = millis();
  }

    Blynk.virtualWrite(V1, ntu);
}
```