

## **LAMPIRAN**

## LAMPIRAN A

### LISTING PROGRAM RANCANG BANGUN PENJERNIH AIR OTOMATIS DENGAN MONITORING TINGKAT KEKERUHAN AIR MENGGUNAKAN SMARTPHONE BERBASIS INTERNET OF THINGS

```
#define BLYNK_AUTH_TOKEN "cvv_37eNGiEX-  
3jddxiefxIUWRwZ1qF7"
```

```
#define BLYNK_PRINT Serial  
#include <BlynkSimpleEsp32.h>  
#include <WiFi.h>  
#include <WiFiClient.h>  
#include <HCSR04.h>
```

```
WidgetLED LED_PompaSebelum (V4);  
WidgetLED LED_PompaSetelah (V5);  
WidgetLED LED_ReadyReset (V9);
```

```
char auth[] = BLYNK_AUTH_TOKEN;  
char ssid[] = "@id.wifi";  
char pass[] = "12345678";
```

```
BlynkTimer timer;
```

```
// ----- LCD -----  
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>  
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
// ----- Relay Pompa 1 dan 2 -----  
#define pumpSebelum_PIN 25  
#define pumpSetelah_PIN 26  
int modePump;
```

```

int manualPump1, manualPump2;
int statePompa1, statePompa2;
int statePenampung2;

// ----- TURBIDITY Sebelum -----
int TurSebelum_PIN = 36;
float Volt1;
float NTU1;
float round_to_dp1( float in_value1, int decimal_place1 )
{
    float multiplier1 = powf( 10.0f, decimal_place1 );
    in_value1 = roundf( in_value1 * multiplier1 ) / multiplier1;
    return in_value1;
}

// ----- TURBIDITY Setelah -----
int TurSetelah_PIN = 39;
float Volt2;
float NTU2;
float round_to_dp2( float in_value2, int decimal_place2 )
{
    float multiplier2 = powf( 10.0f, decimal_place2 );
    in_value2 = roundf( in_value2 * multiplier2 ) / multiplier2;
    return in_value2;
}

// ----- UT -----

#define echoPin 17
#define trigPin 16
UltrasonicDistanceSensor distanceSensor(trigPin, echoPin);

int durasi, jarak;
int Volume;
int batas_atas = 5;
int batas_bawah = 40; // dalam cm

// -----

```

```

//          MANUAL PUMP
// -----

BLYNK_WRITE(V3) {
  modePump = param.asInt();

  Serial.println(modePump);
}

// -----

BLYNK_WRITE(V6) {
  manualPump1 = param.asInt();
  if (modePump == 1) {
    if (manualPump1 == HIGH && Volume < 90) {
      statePompa1 = HIGH;
    }
    if (manualPump1 == LOW || Volume >= 90) {
      statePompa1 = LOW;
    }
  }
}

  Serial.println(manualPump1);
}

// -----

BLYNK_WRITE(V7) {
  manualPump2 = param.asInt();
  if (modePump == 1) {
    if (manualPump2 == HIGH ) {
      digitalWrite(pumpSetelah_PIN, LOW);
      statePompa2 = HIGH;
    }
    else {
      digitalWrite(pumpSetelah_PIN, HIGH);
      statePompa2 = LOW;
    }
  }
}

```

```

Serial.println(manualPump2);
}

// -----

BLYNK_WRITE(V8) {
  int resetSistem = param.asInt();
  if (resetSistem == HIGH && Volume < 10) {
    statePenampung2 = LOW;
  }
  Serial.println(resetSistem);
}

void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.clear();

  lcd.setCursor(0, 0);
  lcd.print("SSID: ");
  lcd.print(ssid);
  lcd.setCursor(0, 1);
  lcd.print("PASS: ");
  lcd.print(pass);

  pinMode(pumpSetelah_PIN, OUTPUT);
  pinMode(pumpSebelum_PIN, OUTPUT);

  //-----
  statePenampung2 = LOW;
  //-----
  Blynk.begin(auth, ssid, pass);
  lcd.clear();
}

```

```

timer.setInterval(1000L, Control_Monitoring);
timer.setInterval(1000L, TampilanLCD);
//-----

}

void loop() {
  Blynk.run();
  Turbidity();
  timer.run();
}

void Control_Monitoring() {

  //----- VOLUME
  jarak = distanceSensor.measureDistanceCm();

  if (jarak > batas_bawah) {
    jarak = batas_bawah;
    Volume = 0;
  }
  if (jarak < batas_atas) {
    jarak = batas_atas;
    Volume = 100;
  }
  Volume = map (jarak, batas_atas, batas_bawah, 100, 0);
  Blynk.virtualWrite(V2, Volume);
  Serial.println (jarak);

  //----- PUMP
  MANUAL
  (modePump == 1) ?
  Serial.println("MODE MANUAL") :
  Serial.println("MODE OTOMATIS");

  // ----- PUMP 1
  if (statePompal == HIGH)
  {
    Serial.println("PUMP 1 ON MANUAL");
  }
}

```

```

digitalWrite(pumpSebelum_PIN, LOW); // PUMP ON
LED_PompaSebelum.on();
}
else
{
  Serial.println("PUMP 1 OFF MANUAL");
  digitalWrite(pumpSebelum_PIN, HIGH); // PUMP OFF
  LED_PompaSebelum.off();
}

// ----- PUMP 2
if (statePompa2 == HIGH)
{
  Serial.println("PUMP 2 ON MANUAL");
  digitalWrite(pumpSetelah_PIN, LOW); // PUMP ON
  LED_PompaSetelah.on();
}
else
{
  Serial.println("PUMP 2 OFF MANUAL");
  digitalWrite(pumpSetelah_PIN, HIGH); // PUMP OFF
  LED_PompaSetelah.off();
}

//----- PUMP
OTOMATIS

(Volume < 10) ? LED_ReadyReset.on() : LED_ReadyReset.off();

if (modePump == 2) { // OTOMASI ON
  //----- PUMP 1
  if (Volume < 100 && statePenampung2 == LOW) {
    digitalWrite(pumpSebelum_PIN, LOW); // PUMP ON
    Serial.println("PUMP 1 ON");
  }
  if (Volume = 100 || statePenampung2 == HIGH) {
    digitalWrite(pumpSebelum_PIN, HIGH); // PUMP OFF
    Serial.println("PUMP 1 OFF");
  }
}

```

```

}

//----- PUMP 2
if (Volume >= 100) {
  statePenampung2 = HIGH;
}

if (NTU2 > 25) { // keruh
  digitalWrite(pumpSetelah_PIN, LOW); // PUMP ON
  Serial.println("PUMP 2 ON");
}

if (NTU2 <= 25) {
  digitalWrite(pumpSetelah_PIN, HIGH); // PUMP OFF
  Serial.println("PUMP 2 OFF");
}
}

Serial.println("-----");
}

```

```

void TampilanLCD() {
  // ----- Turbidity
  Blynk.virtualWrite(V0, NTU1);
  Serial.print("\t\t\t\t\tVolt 1 : ");
  Serial.print(Volt1, 3);
  Serial.println(" V <<<<<<---");
  Serial.print("NTU Sebelum Filtrasi : ");
  Serial.println(NTU1);

  Blynk.virtualWrite(V1, NTU2);
  Serial.print("\t\t\t\t\tVolt 2 : ");
  Serial.print(Volt2, 3);
  Serial.println(" V <<<<<<---");
  Serial.print("NTU Setelah Filtrasi: ");
  Serial.println(NTU2);
}

```



```
lcd.setCursor(0, 0);  
lcd.print("NTU 1 : ");  
lcd.print(NTU1);
```

```
lcd.setCursor(0, 1);  
lcd.print("NTU 2 : ");  
lcd.print(NTU2);  
}
```

```
float mapfloat(float x, float in_min, float in_max, float out_min,  
float out_max)  
{  
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) +  
  out_min;  
}
```

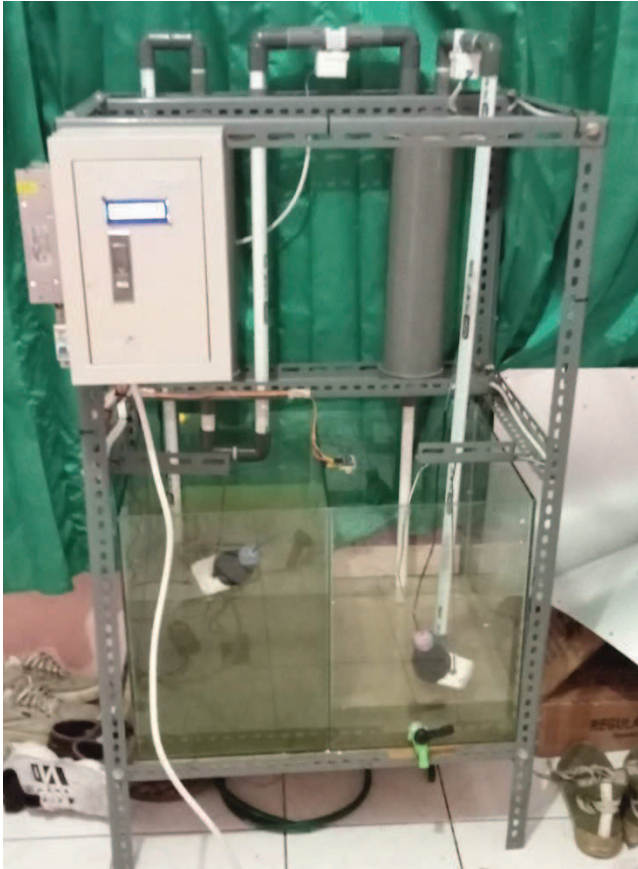
```
void Turbidity () {
```

```
  //----- TURBIDITY  
  1  
  for (int i = 0; i < 1000; i++)  
  {  
    Volt1 += ((float)analogRead(TurSebelum_PIN) / 4095) * 5;  
  }  
  Volt1 = Volt1 / 1000;  
  Volt1 = round_to_dp1(Volt1, 3);  
  
  float volt1Bersih = 3.060;  
  float volt1Keruh = 2.230;  
  float nilai1Bersih = 5;  
  float nilai1Keruh = 25;  
  NTU1 = mapfloat(Volt1, volt1Keruh, volt1Bersih, nilai1Keruh,  
nilai1Bersih);  
  NTU1 = constrain(NTU1, 0, 100);  
  
  //----- TURBIDITY
```

2

```
for (int j = 0; j < 800; j++)  
{  
    Volt2 += ((float)analogRead(TurSetelah_PIN) / 4095) * 5;  
}  
Volt2 = Volt2 / 800;  
Volt2 = round_to_dp2(Volt2, 3);  
  
float volt2Bersih = 0.237;  
float volt2Keruh = 2.900;  
float nilai2Bersih = 5;  
float nilai2Keruh = 25;  
NTU2 = mapfloat(Volt2, volt2Keruh, volt2Bersih, nilai2Keruh,  
nilai2Bersih);  
NTU2 = constrain(NTU2, 0, 100);  
  
}
```

**LAMPIRAN B**  
**LISTING DOKUMENTASI ALAT**





# Filtrasi



NTU SEBELUM FILTRASI

100.000 NTU

NTU SETELAH FILTRASI

32.611 NTU



% VOLUME PENAMPUNG 2 (0 - 100)

1



MANUAL

OTOMATIS

PUMP 1



OFF

PUMP 2



OFF

RESET SISTEM

RESET

READY RESET ?



## BIODATA PENULIS



Nama : Rica Abdullah  
Tempat/Tangah Lahir : Cilacap, 11 Juni 2001  
Alamat : Jl. BANTENG LORENG NO.14  
RT.03/RW.06, Des. Padangjaya,  
Kec. Majenang, Kab. Cilacap, Jawa  
Tengah  
No Hp : (+62)81235801404  
Hobi : Sedang Dicari  
Moto : Apa Yang Kamu Tanam Itu Pula  
Yang Kamu Petik

### **Riwayat Pendidikan :**

SD N 01 Padangjaya	Tahun 2007 – 2013
SMP Diponegoro Majenang	Tahun 2013 – 2016
SMK Diponegoro Majenang	Tahun 2016 – 2019
Politeknik Negeri Cilacap	Tahun 2019 – 2022

Penulis telah mengikuti seminar Tugas Akhir pada tanggal 19 Agustus 2022, sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md).