LAMPIRAN

LAMPIRAN A

LISTING PROGRAM RANCANG BANGUN MONITORING AKUARIUM DAN PAKAN IKAN OTOMATIS BERBASIS IOT

```
#Include  <BlynkSimpleEsp32.h>
#define BLYNK_PRINT Serial
BlynkTimer timer;
WidgetLED LED_lebih_30 (V2);
WidgetLED LED_0_30 (V3);
WidgetLED LED_Pompa (V8);
char auth[] = "nH6AiZOmJtNi-H9VeIOSHXzBU2qUiV3N";
char ssid[] = "Wifi T.Elektronika";
char pass[] = "wifipnc2020";
BLYNK_CONNECTED() {
  Blynk.syncAll();
}
// ---------- LCD ----------
#Include  <Wire.h>
#Include  <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
int stateLCD;
// ---------- LED ----------
```

```
#define LED_Habis       25
#define LED_Penuh       26
#define LED_Turbidity   27

// ---------- BUZZER ----------
#define Buzzer  23
int stateBuzzer;
unsigned long Buzz_1;
unsigned long previousMillis_B1;
const long interval_B1 = 5000;
// ---------- TURBIDITY ----------
int Tur_PIN = 36;
float Volt;
float NTU;
float round_to_dp( float in_value, int decimal_place )
{
  float multiplier = powf( 10.0f, decimal_place );
  in_value = roundf( in_value * multiplier ) / multiplier;
  return in_value;
}
// ---------- NOTIF ----------
```

```
int Notif_Volume, Notif_Turbidity;


// ---------- SERVO ----------
#Include <Servo.h>
Servo servo;
#define servo_pin 12
// ---------- RTC ----------
#Include "RTClib.h"
#Include <Wire.h>
RTC_DS3231 rtc;
String currentTime, Jam;
boolean sama_1, sama_2, sama_3;
// ---------- UT ----------
#define echoPin 16
#define trigPin 17
int duration, distance, real_distance;
int Volume;
int State_NotifPakan;
// ---------- RELAY-POMPA ----------
#define Relay_Pompa 5
// ---------- yang bisa diubah
```

```
int batas_atas    = 4;
int batas_bawah   = 8;
int derajat_servo = 45;
int waktu_servo   = 500L;
int Volume_Tengah = 30;
// ------------------------  Set 3 waktu default
int Start_Hour_1 = 9;  int Start_Min_1, Start_Sec_1 = 00;
int Start_Hour_2 = 13; int Start_Min_2, Start_Sec_2 = 00;
int Start_Hour_3 = 17; int Start_Min_3, Start_Sec_3 = 00;
// ------------------------- Tab Lain
#Include  "TampilanLCD.h"
#Include  "Buzzer.h"
#Include  "Notif.h"
#Include  "Koneksi_BLYNK.h"
#Include  "Control_Monitoring.h"
//            TIME INPUT KE 1
BLYNK_WRITE(V4) {
  TimeInputParam t(param);
  Start_Hour_1  = t.getStartHour();
  Start_Min_1   = t.getStartMinute();
  Start_Sec_1   = t.getStartSecond();
```

```
}
//            TIME INPUT KE 2
BLYNK_WRITE(V5) {
  TimeInputParam t(param);
  Start_Hour_2  = t.getStartHour();
  Start_Min_2   = t.getStartMinute();
  Start_Sec_2   = t.getStartSecond();
}
//            TIME INPUT KE 3
BLYNK_WRITE(V6) {
  TimeInputParam t(param);
  Start_Hour_3  = t.getStartHour();
  Start_Min_3   = t.getStartMinute();
  Start_Sec_3   = t.getStartSecond();
}
void setup()
{
  Serial.begin(9600);
  CheckConnection();
  timer.setInterval(5000L, CheckConnection);
  timer.setInterval(1000L, Control_Monitoring);
```

```
  lcd.begin();
  lcd.backlight();
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  servo.attach(servo_pin);
  servo.write(0);
  pinMode(Tur_PIN,          INPUT);
  pinMode(Relay_Pompa,      OUTPUT);
  pinMode(LED_Penuh,        OUTPUT);
  pinMode(LED_Habis,        OUTPUT);
  pinMode(LED_Turbidity,    OUTPUT);
  pinMode(Buzzer,           OUTPUT);
  rtc.begin();
  Wire.begin();
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
   // rtc.adjust(DateTime(2020, 5, 15, 21, 50, 30));
}
void loop() {
  if (Blynk.connected()) {
    Blynk.run();
  }
```

```
  bacaTurbidity();
  timer.run();
}
// PROGRAM BUZZER
void Buzzer1kali() {
  Buzz_1 = millis() ;
  if (Volume > 0 &&  Volume < Volume_Tengah) {
    stateBuzzer = HIGH;
  }
  if (Volume >= Volume_Tengah) {
    stateBuzzer = LOW;
  }
  if (stateBuzzer == LOW) {
    digitalWrite(Buzzer, LOW);
    previousMillis_B1 = Buzz_1;
  }
  if (stateBuzzer == HIGH) {
    if (Buzz_1 - previousMillis_B1 >= interval_B1) {
      digitalWrite(Buzzer, LOW);
    }
    else {
```

```
    digitalWrite(Buzzer, HIGH);
   }
  }
}
```

// CONTROL MONITORING

```
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 bacaTurbidity() {
  Serial.print("Analog : ");  Serial.println(analogRead(Tur_PIN));
  for (int i = 0; i < 1000; i++)
  {
   Volt += ((float)analogRead(Tur_PIN) / 4095) * 5; //4095
  }
  Volt = Volt / 800;
  Volt = round_to_dp(Volt, 3);
  float voltBersih = 4.537;
  float voltKeruh  = 4.600;
  float nilaiBersih = 5;
```

```
  float nilaiKeruh  = 25; //250

  NTU = mapfloat(Volt, voltKeruh, voltBersih, nilaiKeruh, nilaiBersih);

  NTU = constrain(NTU, 0, 100); //10000

}

void Control_Monitoring() {

  //---------------------------------------------------------------
KEJERNIHAN AIR

  Serial.print("\t\t\t\tVolt : ");  Serial.print(Volt, 3);  Serial.println(" V <<<<<<---");

  Serial.print("NTU : ");  Serial.println(NTU);

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


  Blynk.virtualWrite(V7, NTU);


  if (NTU <= 25) { //50

    digitalWrite(Relay_Pompa, HIGH) ;

    LED_Pompa.off();

    digitalWrite(LED_Turbidity, LOW);

  }

  else {

    digitalWrite(Relay_Pompa, LOW) ;
```

```
  LED_Pompa.on();
  digitalWrite(LED_Turbidity, HIGH);
 }
 //------------------------------------------------------------- VOLUME
 digitalWrite(trigPin, LOW); delayMicroseconds(2);
 digitalWrite(trigPin, HIGH); delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 duration = pulseIn(echoPin, HIGH);
 distance = duration / 58.2;
 if (distance > batas_bawah ) {
  distance = batas_bawah;
  Volume = 0;
 }
 if (distance < batas_atas ) {
  distance = batas_atas;
  Volume = 100;
 }
Serial.print ("jarak yang terbaca : "); Serial.println (distance);
 Volume = map (distance, batas_atas, batas_bawah, 100, 0);
 Blynk.virtualWrite(V0, Volume);
 if (Volume >= Volume_Tengah) {
```

```
  LED_lebih_30.on();
  LED_0_30.off();
  digitalWrite(LED_Penuh, HIGH);
  digitalWrite(LED_Habis, LOW);
 }
 if ((Volume > 0) &&  (Volume < Volume_Tengah)) {
  LED_lebih_30.off();
  LED_0_30.on();
  digitalWrite(LED_Penuh, LOW);
  digitalWrite(LED_Habis, HIGH);
 }
 //---------------------------------------------------------------------- RTC waktu makan
 DateTime now = rtc.now();
 currentTime = String(now.day()) + "-" + now.month() + "-" + now.year() +
          " " + String(now.hour()) + ":" + now.minute()  + ":" + now.second() ;
 Jam          = String(now.hour()) + ":" + now.minute()  + ":" + now.second() + "    ";
 Serial.println(currentTime);
 Blynk.virtualWrite(V1, currentTime);
```

```
  sama_1 = (now.hour() == Start_Hour_1   && now.minute() == Start_Min_1 && now.second() == Start_Sec_1) ;

  sama_2 = (now.hour() == Start_Hour_2   && now.minute() == Start_Min_2 && now.second() == Start_Sec_2) ;

  sama_3 = (now.hour() == Start_Hour_3   && now.minute() == Start_Min_3 && now.second() == Start_Sec_3) ;

  if ( sama_1 || sama_2 || sama_3) {

    servo.write(derajat_servo);

    stateLCD = HIGH;

    timer.setTimeout(waktu_servo, []() {

      servo.write(0);

      stateLCD = LOW;

    });

  }

  //---------------------------------------------------------------- LCD

  TampilanLCD();

  //---------------------------------------------------------------- BUZZER dan NOTIF

  Buzzer1kali();

  Notif();

}

                              // KONEKSI _BLYNK

void CheckConnection() {
```

```
  if (!Blynk.connected()) {
    yield();
    if (WiFi.status() != WL_CONNECTED)
    {
      Serial.println("Not connected to Wifi! Connect...");
      WiFi.begin(ssid, pass);
      delay(400);
      if (WiFi.status() != WL_CONNECTED)
      {
        Serial.println("Cannot connect to WIFI!");
      }
      else
      {
        Serial.println("Connected to wifi!");
      }
    }
    if ( WiFi.status() == WL_CONNECTED && !Blynk.connected()
)
    {
      Serial.println("Not connected to Blynk Server! Connecting...");
      Blynk.connect();
      Blynk.begin(auth, ssid, pass , "blynk-cloud.com" , 8080);
```

```
      if (!Blynk.connected()) {
        Serial.println("Connection failed!");
      }
    }
  }
  else {
    Serial.println("Connected to Blynk server!");
  }
}
                        // NOTIFIKASI UNTUK BLYNK
void CheckConnection() {
  if (!Blynk.connected()) {
    yield();
    if (WiFi.status() != WL_CONNECTED)
    {
      Serial.println("Not connected to Wifi! Connect...");
      WiFi.begin(ssid, pass);
      delay(400);
      if (WiFi.status() != WL_CONNECTED)
      {
        Serial.println("Cannot connect to WIFI!");
```

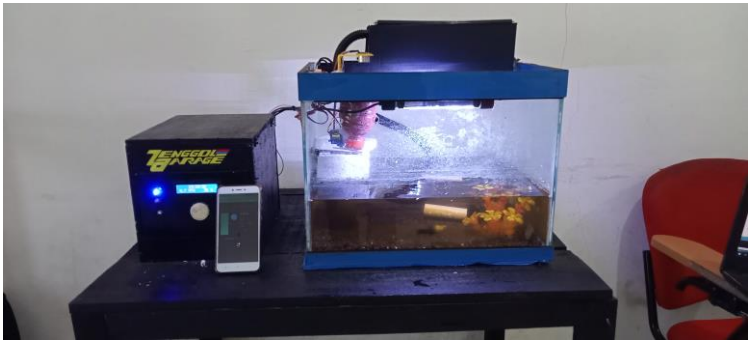
```
      }
      else
      {
        Serial.println("Connected to wifi!");
      }
    }
    if ( WiFi.status() == WL_CONNECTED && !Blynk.connected()
)
    {
      Serial.println("Not connected to Blynk Server! Connecting...");
      Blynk.connect();
      Blynk.begin(auth, ssid, pass , "blynk-cloud.com" , 8080);
      if (!Blynk.connected()) {
        Serial.println("Connection failed!");
      }
    }
  }
  else {
    Serial.println("Connected to Blynk server!");
  }
}
```

//TAMPILAN LCD

```
void TampilanLCD() {
  if (stateLCD == HIGH) {
    lcd.setCursor(0, 0);
    lcd.print("     SELAMAT    ");
    lcd.setCursor(0, 1);
    lcd.print("      MAKAN     ");
    lcd.clear ();
  }
  if (stateLCD == LOW) {
    lcd.setCursor(4, 0);
    lcd.print(Jam);
    lcd.setCursor(0, 1);
    lcd.print(NTU,1);
    lcd.print(" NTU ");
    lcd.setCursor(11, 1);
    lcd.print(Volume);
    lcd.print(" %   ");
  }
}
```

LAMPIRAN B

LISTING DOKUMENTASI ALAT



Gambar Mekanik Alat