LAMPIRAN

LAMPIRAN 1 : Program Arduino pada board Esp 32

```
//=======================================================
======

//              PROGRAM INI SUKSES

//=======================================================
======

const char* ssid     = "Nando";   //ID WiFi

const char* password = "Wakwao16";   //PASSWORD WiFi

String token = "1093513086:AAFP3H9Ugwax7YJpNMASAEa5Po9LzaKWljk";
// BOTFather Telegram

String chat_id = "1088145032";   // Id TELEGRAM


#include <WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

#include "time.h"

#include <Servo_ESP32.h>

#include "DFRobot_ESP_PH.h"


WiFiClientSecure client;

UniversalTelegramBot bot(token, client);

String day_;

String date_;

String time_;

String timeAl_ ;

String currentNow;
```

```
const char* ntpServer = "pool.ntp.org";
const long  my_GMT = 7;
const long  gmtOffset_sec = 3600;
const int   daylightOffset_sec = 3600;

long Bot_lasttime;
int Bot_mtbs = 1000;
#define LED 2
const int Pin26 = 26;
const int Pin25 = 25;

DFRobot_ESP_PH ph;
#define ESPADC 4096.0   //Nilai ADC
#define ESPVOLTAGE 3300 //Tegangan
#define PH_PIN 35       //Data pin 35
float voltage, phValue, temperature = 25;

float adcpin36  = 0;    //Pin Mengikuti Gambar Lama
float voltpin36 = 0;
//float adcpinPH = 0;

//const int PIN_PH = 35;  //Pin Mengikuti Gambar Lama
//DFRobot_ESP_PH ph;

Servo_ESP32 myservo;
int pos = 0;
int close_position = 28 ;
int open_position =  150 ;
```

```
void handleNewMessages(int numNewMessages) {
  Serial.println("handleNewMessages");
  Serial.println(String(numNewMessages));
  for (int i=0; i<numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    String text = bot.messages[i].text;

    //String from_name = bot.messages[i].from_name;
    String from_name = "NANDO ASKHITA VIDENTA";
    if (from_name == "") from_name = "Guest";
    if (text == "/Turbidity") {
      int adcpin36 = analogRead(36);
      float voltpin36 = adcpin36 * (5.0 / 1024.0);
      kirimTurbidity(adcpin36,voltpin36);
    } else if (text == "/PH") {
      int adcpinPH = analogRead(PH_PIN);
      kirimPH(adcpinPH);
    } else if (text == "/Start") {
      String welcome = "Selamat Datang di Mesin Pemberi Pakan Ikan
Otomatis\n";
      welcome += "By  "+ from_name + ".\n";
      welcome += "Silahkan pilih sesuai kebutuhan.\n";
      welcome += "/Turbidity : Lihat data ItensitasAir \n";
      welcome += "/PH : Lihat data PH Air \n";
      bot.sendMessage(chat_id, welcome, "Markdown");
    }
  }
```

```
}

void kirimTurbidity(float adc, float volt){
  bot.sendMessage(chat_id, "Data Turbidity : "+String(adc)+"", "");
  delay(2000);
}

void kirimPH(int adc){
  bot.sendMessage(chat_id, "Data PH : "+String(adc)+"", "");
  delay(2000);
}

void printLocalTime()
{
  struct tm timeinfo;
  if(!getLocalTime(&timeinfo)){
    Serial.println("Failed to obtain time");
    return;
}
  char str_day[10];
  char str_date[12];
  char str_time[12];
  char str_timeAll[12];
  strftime(str_day, sizeof str_day, "%A",&timeinfo);
  strftime(str_date, sizeof str_date, "%d-%m-%Y",&timeinfo);
  strftime(str_time, sizeof str_time, "%H:%M:%S",&timeinfo);
  strftime(str_timeAll, sizeof str_timeAll, "%H:%M",&timeinfo);
  day_ = String(str_day);
```

```
  date_ = String(str_date);
  time_= String(str_time);
  timeAl_= String(str_timeAll) ;
  currentNow=day_ + ", " + date_ +"==" + time_;
}

void closeServo() {
  for (pos = close_position; pos <= open_position; pos += 1) {
    myservo.write(pos);
  }
}

void openServo() {
  for (pos = open_position; pos >= close_position; pos -= 1) {
    myservo.write(pos);
    //delay(30);
  }
}

/*void nyalakan_servo()
{
if(timeAl_=="08:00" ||  timeAl_=="08:02" || timeAl_=="08:04" ||
timeAl_=="08:06" || timeAl_=="08:08" || timeAl_=="08:10" || timeAl_=="08:12"
|| timeAl_=="08:14" || timeAl_=="08:16" || timeAl_=="08:18")

     {
      openServo();
      delay(10000);
      closeServo();
```

```
      delay(50000);
     }
 else {
      closeServo();                  //ketika upload data lepas semua sambungan vcc
agar tidak troubel
     }
}
*/
void nyalakan_motor()
{
if(timeAl_=="07:00" || timeAl_=="11:00" || timeAl_=="15:00" ||
timeAl_=="19:53" || timeAl_=="19:55" || timeAl_=="19:57" || timeAl_=="13:09")

    {
      digitalWrite(Pin25,LOW);
      openServo();
      delay(50000);
      digitalWrite(Pin25,HIGH);
      closeServo();
      delay(10000);
    }
 else {
      digitalWrite(Pin25,HIGH);
      closeServo();
    }
}

void setup()
{
```

```
Serial.begin(115200);
delay(10);

WiFi.mode(WIFI_STA);

Serial.println("");
Serial.print("Connecting to Wifi ");
Serial.println(ssid);
WiFi.begin(ssid, password);


long int StartTime=millis();
while (WiFi.status() != WL_CONNECTED)
{
 delay(500);
 if ((StartTime+10000) < millis()) break;
}
configTime(gmtOffset_sec*(my_GMT - 1), daylightOffset_sec, ntpServer);
printLocalTime();

Serial.println("");
Serial.println("IP anda adalah: ");
Serial.println(WiFi.localIP());

myservo.attach(15);                  // Pin Servo
myservo.write(open_position);

pinMode(LED,OUTPUT);
```

```
 pinMode(Pin26,OUTPUT);
 pinMode(Pin25,OUTPUT);

 Serial.println("");
 if (WiFi.status() != WL_CONNECTED) {
  Serial.println("Reset");

  digitalWrite(LED,HIGH);

  delay(1000);
  ESP.restart();
 }
 else {
  digitalWrite(LED,HIGH);
  delay(1000);
  digitalWrite(LED,LOW);
  delay(1000);
  digitalWrite(LED,HIGH);
  delay(1000);
  digitalWrite(LED,LOW);
 }

 digitalWrite(Pin26,LOW);
 digitalWrite(Pin25,LOW);

 ph.begin();
}
```

```
void loop(){
  printLocalTime();
  Serial.print("Waktu : ");
  Serial.println(currentNow);

  nyalakan_motor();

  int adcpin36 = analogRead(36);
  float voltpin36 = adcpin36 * (5.0 / 1024.0);
  Serial.print("Ambil data Turbidity : ");
  Serial.print(adcpin36);
  Serial.print(" : ");
  Serial.print(voltpin36);
  Serial.println(" Volt");

  if(adcpin36 < 200 ){
  bot.sendMessage(chat_id,"AIR KERUH", "");       //Jika nilai ADC turun maka
akan mengirim notif ke pengguna
  }

  if(adcpin36 < 40){
  bot.sendMessage(chat_id,"AIR HARUS SEGERA DIGANTI", "");
  }

 static unsigned long timepoint = millis();
 if (millis() - timepoint > 1000U) //time interval: 1s
 {
  timepoint = millis();
```

```
  voltage = analogRead(PH_PIN) / ESPADC * ESPVOLTAGE; // Membaca
tegangan              //<-3.1374 NETRAL //>7.7962
  Serial.print("voltage:");
  Serial.println(voltage, 4);


                                  // Membaca temperatur
  Serial.print("temperature:");
  Serial.print(temperature, 1);
  Serial.println("^C");


  phValue = ph.readPH(voltage, temperature);  // convert voltage to pH with
temperature compensation
  Serial.print("pH:");
  Serial.println(phValue, 4);
 }
 ph.calibration(voltage, temperature);       // calibration process


 digitalWrite(Pin26,HIGH);
  // periksa periksa pin Adc Pin 35
  if(phValue > -2.0000){     //Lebih dari ... maka pompa aktif    //3.1374 NETRAL
// 2.0000 TDK NETRAL
  digitalWrite(Pin26,LOW);
  }
//float readTemperature()



    if (millis() > Bot_lasttime + Bot_mtbs)  {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    while(numNewMessages) {
```

```
    Serial.println("got response");
    handleNewMessages(numNewMessages);
    numNewMessages = bot.getUpdates(bot.last_message_received + 1);
   }
   Bot_lasttime = millis();
 }

 delay(2000);
}
```