LAMPIRAN

Lampiran 1 Program Arduino pada Board ESP32-CAM

```
const char* ssid     = "Huawei";    //your network SSID

const char* password = "koyokbiyen";    //your network password

String token = "[redacted]";

String chat_id = "[redacted]";

#include "DHT.h"

#define DHTPIN 13

#include <WiFi.h>

#include <WiFiClientSecure.h>

#include "soc/soc.h"

#include "soc/rtc_cntl_reg.h"

#include <UniversalTelegramBot.h>

#define DHTTYPE DHT22

#include "esp_camera.h"

#define PWDN_GPIO_NUM     32

#define RESET_GPIO_NUM    -1
```

```
#define XCLK_GPIO_NUM      0

#define SIOD_GPIO_NUM     26

#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35

#define Y8_GPIO_NUM       34

#define Y7_GPIO_NUM       39

#define Y6_GPIO_NUM       36

#define Y5_GPIO_NUM       21

#define Y4_GPIO_NUM       19

#define Y3_GPIO_NUM       18

#define Y2_GPIO_NUM        5

#define VSYNC_GPIO_NUM    25

#define HREF_GPIO_NUM     23

#define PCLK_GPIO_NUM     22

#include "time.h"

#include <Wire.h>

DHT dht(DHTPIN, DHTTYPE);
```

```
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;

int Bot_mtbs = 1000; //mean time between scan messages

long Bot_lasttime;   //last time messages' scan has been done

bool Start = false;

String SuhuC;

String SuhuF;

String Lembab;
```

```
int lampu = 2; //pompa

int kipas = 14;

int motor = 15;

const int ledPin = 12;//lampu yang dikontrol

int ledStatus = 0;



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;

    if (from_name == "") from_name = "Guest";

    if (text == "/now") {
```

```
   bot.sendMessage(chat_id, currentNow, "");}
 if (text == "/pompaon") {
   digitalWrite(lampu,HIGH);
   bot.sendMessage(chat_id,"pompa on", "");}
 if (text == "/motoron") {
   digitalWrite(motor,HIGH);
   bot.sendMessage(chat_id,"motor On", "");}
 if (text == "/pompaoff") {
   digitalWrite(lampu,LOW);
   bot.sendMessage(chat_id,"pompa off", "");}
 if (text == "/motoroff") {
   digitalWrite(motor,LOW);
   bot.sendMessage(chat_id,"motor off", "");}
if (text == "/KeadaanTelur") {
   bot.sendMessage(chat_id, "Kamera Memotret", "");
   sendCapturedImage2Telegram(token, chat_id);
   delay(10000);
```

```
}
else if (text == "/nyalakanlampu") {
  digitalWrite(ledPin, HIGH);   // turn the LED on (HIGH is the voltage level)
  ledStatus = 1;
  bot.sendMessage(chat_id, "lampu pendukung On", "");
  delay(10000);
}
else if (text == "/matikanlampu") {
  digitalWrite(ledPin, LOW);   // turn the LED on (HIGH is the voltage level)
  //ledStatus = 1;
  bot.sendMessage(chat_id, "lampu pendukung off", "");
}
else if (text == "/suhu") {
String replay = "Intensitas suhu : ";
replay += SuhuC;
replay += " *C\n";
replay += "fahreheit : ";
```

```
    replay += SuhuF;

    replay += " *F\n";

    replay += "Kelembapan : ";

    replay += Lembab;

    replay += "  %\n";

    replay += "Suhu Keadaan maksimal\n";

    bot.sendMessage(chat_id, replay, "");

    }

    else if (text == "/MulaiGan") {

     String welcome = "selamat datang di mesin penetetas telur by MNCholis, " +
from_name + ".\n";

     welcome += "silahkan pilih sesuai kebutuhan.\n\n";

    //welcome += "/now : cek waktu \n";

     welcome += "/suhu : cek suhu \n";

     welcome += "/KeadaanTelur : proses mengambil gambar \n";

     welcome += "/nyalakanlampu : menyalakan lampu pendukung\n";

     welcome += "/matikanlampu : mematikan lampu pendukung\n";

     welcome += "/motoron : menyalakan motor\n";
```

```
      welcome += "/motoroff : mematikan motor\n";
      welcome += "/pompaon : menyalakan pompa air\n";
      welcome += "/pompaoff : mematikan pompa air\n";
      bot.sendMessage(chat_id, welcome, "Markdown");
    }
    }}
void cek_time(){
  if(timeAl_=="06:00" || timeAl_=="12:00"||timeAl_=="18:00"||
  timeAl_=="00:00"){
    digitalWrite(motor,HIGH);}
    else{digitalWrite(motor,LOW);}


}


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 setup()
{
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);

  Serial.begin(115200);
  delay(10);
  dht.begin();
  WiFi.mode(WIFI_STA);

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

  long int StartTime=millis();
  while (WiFi.status() != WL_CONNECTED)

  {

    delay(500);

    if ((StartTime+10000) < millis()) break;

  }



  Serial.println("");

  Serial.println("STAIP address: ");

  Serial.println(WiFi.localIP());



  Serial.println("");

configTime(gmtOffset_sec*(my_GMT - 1), daylightOffset_sec, ntpServer);

  printLocalTime();

  if (WiFi.status() != WL_CONNECTED) {

    Serial.println("Reset");



    ledcAttachPin(4, 3);
    ledcSetup(3, 5000, 8);

    ledcWrite(3,10);

    delay(200);

    ledcWrite(3,0);

    delay(200);

    ledcDetachPin(3);



    delay(1000);

    ESP.restart();

  }

  else {

    ledcAttachPin(4, 3);

    ledcSetup(3, 5000, 8);

    for (int i=0;i<5;i++) {

      ledcWrite(3,10);

      delay(200);

      ledcWrite(3,0);
      delay(200);
    }
    ledcDetachPin(3);
  }


  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;

  config.pin_vsync = VSYNC_GPIO_NUM;

  config.pin_href = HREF_GPIO_NUM;

  config.pin_sscb_sda = SIOD_GPIO_NUM;

  config.pin_sscb_scl = SIOC_GPIO_NUM;

  config.pin_pwdn = PWDN_GPIO_NUM;

  config.pin_reset = RESET_GPIO_NUM;

  config.xclk_freq_hz = 20000000;

  config.pixel_format = PIXFORMAT_JPEG;

  //init with high specs to pre-allocate larger buffers

  if(psramFound()){

    config.frame_size = FRAMESIZE_UXGA;

    config.jpeg_quality = 2; //0-63 lower number means higher quality

    config.fb_count = 2;

  } else {

    config.frame_size = FRAMESIZE_SVGA;

    config.jpeg_quality = 4; //0-63 lower number means higher quality
```

```
    config.fb_count = 1;
  }

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    delay(1000);
    ESP.restart();
  }

  //drop down frame size for higher initial frame rate
  sensor_t * s = esp_camera_sensor_get();
  s->set_framesize(s, FRAMESIZE_VGA); //
UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
  pinMode(kipas, OUTPUT);
  pinMode(lampu, OUTPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(motor, OUTPUT);

  pinMode(kipas, OUTPUT); // initialize digital ledPin as an output.

  delay(10);

  digitalWrite(kipas, LOW); // initialize pin as off

  digitalWrite(lampu, HIGH);

  digitalWrite(ledPin, LOW);

  digitalWrite(motor, LOW);

  delay(10);



}



void loop(){

  printLocalTime();

  cek_time();

  float humidity_1 = dht.readHumidity();

  float fahrenheit_1 = dht.readTemperature(true);

  float celcius_1 = dht.readTemperature();
```

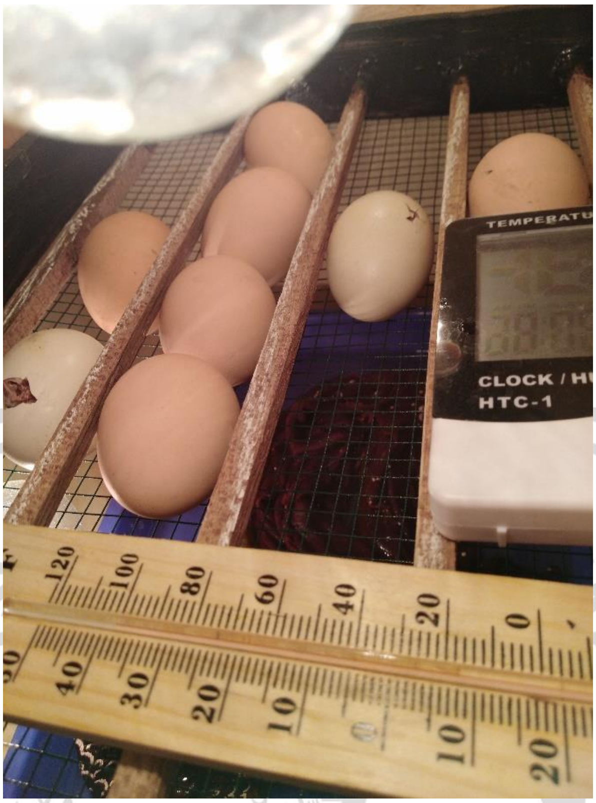
```
SuhuF = String(fahrenheit_1);

SuhuC = String(celcius_1);;

Lembab = String(humidity_1);

if(celcius_1<=39){digitalWrite(kipas,LOW);}

else{digitalWrite(kipas,HIGH);}

//sendCapturedImage2Telegram(token, chat_id);

//delay(72000);

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();

}
```

```
  }


String sendCapturedImage2Telegram(String token, String chat_id) {
  const char* myDomain = "api.telegram.org";
  String getAll="", getBody = "";

  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    delay(1000);
    ESP.restart();
    return "Camera capture failed";
  }

  Serial.println("Connect to " + String(myDomain));
  WiFiClientSecure client_tcp;
  if (client_tcp.connect(myDomain, 443)) {
    Serial.println("Connection successful");

    String head = "--Taiwan\r\nContent-Disposition: form-data; name=\"chat_id\";
\r\n\r\n" + chat_id + "\r\n--Taiwan\r\nContent-Disposition: form-data;
name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type:
image/jpeg\r\n\r\n";
    String tail = "\r\n--Taiwan--\r\n";

    uint16_t imageLen = fb->len;
    uint16_t extraLen = head.length() + tail.length();
    uint16_t totalLen = imageLen + extraLen;

    client_tcp.println("POST /bot"+token+"/sendPhoto HTTP/1.1");
    client_tcp.println("Host: " + String(myDomain));
    client_tcp.println("Content-Length: " + String(totalLen));
    client_tcp.println("Content-Type: multipart/form-data; boundary=Taiwan");
    client_tcp.println();
    client_tcp.print(head);


    uint8_t *fbBuf = fb->buf;
    size_t fbLen = fb->len;
    for (size_t n=0;n<fbLen;n=n+1024) {
      if (n+1024<fbLen) {
        client_tcp.write(fbBuf, 1024);
        fbBuf += 1024;
      }
      else if (fbLen%1024>0) {
        size_t remainder = fbLen%1024;
        client_tcp.write(fbBuf, remainder);
      }
    }
    client_tcp.print(tail);
```

```
esp_camera_fb_return(fb);
int waitTime = 10000;   // timeout 10 seconds
long startTime = millis();
boolean state = false;

while ((startTime + waitTime) > millis())
{
  Serial.print(".");
  delay(100);
  while (client_tcp.available())
  {
      char c = client_tcp.read();
      if (c == '\n')
      {
        if (getAll.length()==0) state=true;
        getAll = "";
      }
```

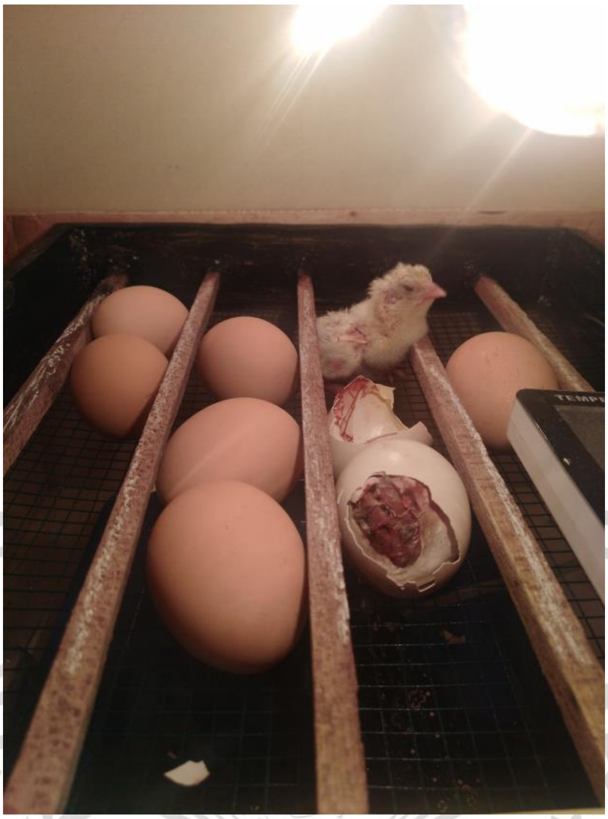
```
        else if (c != '\r')
          getAll += String(c);
        if (state==true) getBody += String(c);
        startTime = millis();
      }
      if (getBody.length()>0) break;
    }
    client_tcp.stop();
    Serial.println(getBody);
  }
  else {
    getBody="Connected to api.telegram.org failed.";
    Serial.println("Connected to api.telegram.org failed.");
  }
  return getBody;
}
```

Lampiran 2 Gambar Proses Penetasan Telur



Gambar Penetasan 1 Anak Ayam Sudah Mulai Memecahkan Cangkang Telur (Gambar diambil Melalui Kamera Handphone)

Gambar Penetasan 2 Anak Ayam Sudah Mulai Memecahkan Cangkang Telur (Gambar diambil Melalui ESP32-CAM)



Gambar Penetasan 3 Anak Ayam Berhasil Keluar Dari Cangkang Telur (Gambar diambil Melalui Kamera Handphone)

Gambar Penetasan 4 Anak Ayam Berhasil Keluar Dari Cangkang Telur (Gambar diambil Melalui ESP32-CAM)

Gambar Penetasan 5 Anak Ayam Berhasil Keluar Dari Cangkang Telur (Gambar diambil Melalui Kamera Handphone)

Gambar Penetasan 6 Anak Ayam Yang Telah Menetas

Daftar Riwayat Hidup

A. Biodata Pribadi

Pas Foto 3 x 4

1. Nama Lengkap : Moh Nur Cholis
2. Tempat / tanggal lahir : Gresik, 07 Januari
3. Jenis Kelamin : Laki-laki
4. Kewarganegaraan : Indonesia
5. Tinggi, Berat Badan :164 cm, 65 kg
6. Agama : Islam
7. Status : Belum Menikah
8. Alamat : JL. Kapten Dulasim Ds. Kramat Inggil Gg. 11E RT.04 RW.01
9. No. Telp : 085840954761
10. Email : mcholis47@gmail.com

B. Riwayat Pendidikan

1. SD / Sederajat : SDN 1 Sidomoro
2. SMP / Sederajat : SMP Semen Gresik
3. SMA / Sederajat : SMK Semen Gresik
4. Perguruan Tinggi : Universitas Muhammadiyah Gresik,
   Program Studi Teknik Elektro S1 2016-2020