

LAMPIRAN

LAMPIRAN 1 : Program Arduino pada board NodeMCU ESP8266

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
// Atur SSID Dan Password Wifi Router / Wifi Hotspot / Wifi Tethring Dari HP
char ssid[] = "bismillah"; // Masukan Nama SSID (Besar Kecil Pengaruh)
char password[] = "arirehan123"; // Masukan Password Yang Di Pakai Router
// inputoutput
const int ledPin = D1;
const int mcPin = D2;
const int buzPin = D0;
// kondisi
int kondisimc = 0;
int kondisiauto1 = 0;
// Untuk Setting Perintah Pada Telegram Ganti Nama Dalam Tanda Kutip Sesuai Keinginan Anda
const char* Menu_1 = "/Start"; //sudah set
const char* Menu_2 = "/Masuk"; //sudah set
const char* Menu_3 = "/about" //sudah set
const char* Menu_4 = "/Manual" //sudah set
const char* Menu_5 = "/AutoON"; //belum set
const char* Menu_6 = "/AutoOFF"; //belum set
//perintah manual & ON
const char* Perintah_1 = "/Memotret"; //sudah set
const char* Perintah_2 = "/MatikanBuzzer"; //sudah set
const char* Perintah_ON_2 = "/NyalakanLEDbantuan"; //sudah set
//perintah OFF
```

```

const char* Perintah_OFF_2 = "/MatikanLEDbantuan"; //sudah set
//perintah status
const char* Status_MC38 = "/StatusMC38"; //sudah
const char* Status_LED = "/StatusLED"; //sudah
const char* chat_id = "1041995790"; //diubah
// Masukan Token Telegram BOT Kalian (Jika Lupa Token Bisa Lihat Pada
BotFather Ketikan /token)
#define BOTtoken "125562967:AAFbwj9A-E6IL2cS7_LAKKbPq82TPCFJX1I"
// Masukan Token Kalian (Silakan Copy Dari Botfather)
// #define chat_id "1041995790"
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);
int Bot_mtbs = 1000; //mean time between Untuk Lamanya Scan Pesan Baru
long Bot_lasttime; // Pesan Terakhir
bool Start = false;
int statusmc38 = 0;
int statusled = 0;
int statusbuz = 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 Pesan = bot.messages[i].text;
String from_name = bot.messages[i].from_name;
if (from_name == "") from_name = "Guest";
if (Pesan == (Menu_1))
{
String start1 = "Selamat datang di Cam-bing " + from_name + ".\n";
start1 += "Sistem Informasi Keamanan Kandang Kambing Berbasis IOT\n";
}
}
}

```

```

start1 += "Cam-bing by Electro Ndeso\n\n";
start1 += "Sistem keamanan ini dilengkapi\n";
start1 += "  -Sensor PIR\n";
start1 += "  -Sensor MC-38\n";
start1 += "  -ESP32-CAM\n";
start1 += "  -NODEMCU ESP8266\n";
start1 += "  -Buzzer\n\n";
start1 += "Menggunakan media aplikasi Telegram Messenger\n";
start1 += "\n";
start1 += "Manfaat bagi para peternak kambing\n";
start1 += "1. Untuk membantu para peternak kambing menjaga hewan ternaknya dari";
start1 += "  orang yang berniat jahat\n";
start1 += "2. Agar mengetahui siapa yang berada disekitar area kandang\n";
start1 += "3. Melalui alarm yang menandakan sinyal peringatan bahaya, pemilik";
start1 += "  kandang akan terjaga dari tidurnya.\n\n";
start1 += "Untuk melanjutkan ke Pengendalian alat anda dapat mengklik /Masuk)\n";
bot.sendMessage(chat_id, start1, "Markdown");
}
else if (Pesan == (Menu_2))
{
String Masuk = "Selamat Datang di sistem keamanan cam-bing " + from_name + ".\n";
Masuk += "Gunakan Kode Berikut Untuk Mengontrol\n\n";
Masuk += "- /AutoON   : mengaktifkan Mode pintar (disini proses pengambilan";
Masuk += "gambar dan pengiriman data akan berlangsung otomatis \n";
Masuk += "- /AutoOFF  : menonaktifkan Mode pintar \n";
Masuk += "- /Manual   : menuju ke pengendalian manual \n";
Masuk += "- /about    : Penjelasan mengenai alat keamanan Cam-bing ini \n\n";

```

```

bot.sendMessage(chat_id, Masuk, "Markdown");
}
else if (Pesan == (Menu_4))
{
String welcome = "Selamat Datang di sistem keamanan cam-bing, " + from_name
+ ".\n";

welcome += "Gunakan Kode Berikut Untuk Mengontrol Cam-bing :\n\n";
elcome += (Perintah_1);
welcome += " <- Tekan untuk memotret\n\n";
welcome += "Kontrol buzzer & LED :\n\n";
elcome += (Perintah_ON_2);
welcome += " <- Tekan ON\n";
welcome += (Perintah_OFF_2);
welcome += " <- Tekan OFF \n\n";
welcome += (Perintah_2);
welcome += " <- Tekan OFF \n\n";
welcome += "Cek Status Sensor & LED :\n\n";
welcome += (Status_MC38);
welcome += " <- Cek Status\n";
welcome += (Status_LED);
welcome += " <- Cek Status \n\n";
bot.sendMessage(chat_id, welcome, "Markdown");
}
//-----settingan keseluruhan perintah-----
else if (Pesan == (Menu_5)) //autoon //perintah mengambil gambar
{
bot.sendMessage(chat_id, "proses mengaktifkan auto....");
kondisimc = 1;
kondisiauto1 = 1;
delay(100);
}

```

```

}

//-----settingan memotret-----
else if (Pesan == (Menu_6)) //autooff //perintah mengambil gambar
{
digitalWrite (buzPin, LOW); //tambah iki barusan
bot.sendMessage(chat_id, "proses menonaktifkan auto....");
kondisimc = 0;
kondisiauto1 = 0;
delay(100);
}

//-----settingan mematikan buzzer-----
else if (Pesan == (Perintah_2)) //perintah mematikan buzzer
{
digitalWrite (buzPin, LOW);
statusbuz = 0;
kondisimc = 0; //tambah iki
bot.sendMessage(chat_id, "Buzzer dimatikan");
}

/-----LED BANTUAN-----
else if (Pesan == (Perintah_OFF_2)) //menonaktifkan LED
{
statusled = 1;
digitalWrite (ledPin, LOW);
bot.sendMessage(chat_id, "LED bantuan dimatikan");
}

else if (Pesan == (Perintah_ON_2)) //mengaktifkan LED
{
statusled = 0;
digitalWrite (ledPin, HIGH);
}

```

```

        bot.sendMessage(chat_id, "LED bantuan dinyalakan");
    }
//-----STATUS-----
    else if (Pesan == (Status_MC38))
    {
        if(statusmc38){
            bot.sendMessage(chat_id, "Lapor Mas Ari, MC38 dalam keadaan nonaktif",
            "");
        }
        else
        {
            bot.sendMessage(chat_id, "Lapor Mas Ari, MC38 dalam keadaan aktif", "");
        }
    }
    else if (Pesan == (Status_LED))
    {
        if(statusled){
            bot.sendMessage(chat_id, "Lapor Mas Ari, LED Bantuan dalam keadaan
            nonaktif", "");
        }
        else
        {
            bot.sendMessage(chat_id, "Lapor Mas Ari, LED Bantuan dalam keadaan
            aktif", "");
        }
    }
//-----
}

void setup() {
    Serial.begin(115200);

```

```

// Set WiFi to station mode and disconnect from an AP if it was Previously
// connected
WiFi.mode(WIFI_STA);
WiFi.disconnect();
delay(100);
// attempt to connect to Wifi network:
Serial.print("Connecting Wifi: ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}
client.setInsecure(); // FIX NODEMCU Wajib Board esp8266 version
2.5.0
Serial.println("");
Serial.println("WiFi connected");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
bot.longPoll = 60; // Panjang Polling
pinMode(mcPin, INPUT_PULLUP); //mc38
pinMode(buzPin, OUTPUT); //buzzer
pinMode(ledPin, OUTPUT);
digitalWrite(buzPin, LOW);
digitalWrite(ledPin, LOW);
}

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

```

```

Serial.println("got response");
handleNewMessages(numNewMessages);
Serial.println(numNewMessages);
numNewMessages = bot.getUpdates(bot.last_message_received + 1);
}
Bot_lasttime = millis();
}
//mc38 & buzzer Setting
int mcPin = D2; //PIR Motion Sensor
pinMode(mcPin, INPUT_PULLUP);
int v = digitalRead(mcPin);
if( (v==1)&&(kondisimc == 1)) {
  digitalWrite (buzPin,HIGH);
  Serial.println ("Lawang mbukak!");
  bot.sendMessage(chat_id, "pintu dalam keadaan terbuka");
  delay(5000);
}
else if (kondisimc == 0) {
  v = 0;
}
}
}

```


LAMPIRAN 2 : Program Arduino pada board ESP32-CAM

```
// Enter your WiFi ssid and password

const char* ssid = "bismillah"; //your network SSID

const char* password = "arirehan123"; //your network password

String token = "1225562967:AAFbwj9A-E6IL2cS7_LAKKbPq82TPCFJX1I"; //
Create your bot and get the token -> https://telegram.me/fatherbot

String chat_id = "1041995790"; // Get chat_id -> https://telegram.me/userinfobot

#include <WiFi.h>

#include <WiFiClientSecure.h>

#include "soc/soc.h"

#include "soc/rtc_cntl_reg.h"

#include <UniversalTelegramBot.h>

#include "esp_camera.h"

#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

int kondisi = 0;

const int gpioPIR = 14;

int v = digitalRead(gpioPIR);

//int ledStatus = 0;
```

```

WiFiClientSecure client;

UniversalTelegramBot bot(token, client);

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

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

bool Start = false;

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 == "/Memotret") {

bot.sendMessage(chat_id, "Kamera Mengambil Gambar", "");

sendCapturedImage2Telegram(token, chat_id);

delay(10000);

}

if (text == "/AutoON") {

bot.sendMessage(chat_id, "Mode pintar berjalan", "");

kondisi = 1;

elay (100);

}

if (text == "/AutoOFF") {

bot.sendMessage(chat_id, "Mode pintar nonaktif", "");

kondisi = 0;

delay (100);

}

}

```

```

    }
    else if (text == "/Cam_bing") {
        String welcome = "Selamat datang di Cam-bing by Electro-ndeso" +
        from_name + ".\n";
        welcome += "Silahkan pilih sesuai Kebutuhan di ESP32Cam \n\n";
        welcome += "/AutoON      : mengaktifkan Mode keamanan pintar \n";
        welcome += "/AutoOFF     : menonaktifkan Mode keamanan pintar \n";
        welcome += "/Memotret    : perintah mengambil gambar \n";
        bot.sendMessage(chat_id, welcome, "Markdown");
    }
}
}
}
void setup()
{
    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
    Serial.begin(115200);
    delay(10);
    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("");

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;

if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;

    config.jpeg_quality = 10; //0-63 lower number means higher quality
    config.fb_count = 2;
} else {

```

```

config.frame_size = FRAMESIZE_SVGA;
config.jpeg_quality = 12; //0-63 lower number means higher quality
config.fb_count = 1;
}
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();
}
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_SVGA); //
UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
s->set_vflip(s, 1); // vertikal flip
pinMode(gpioPIR, INPUT_PULLUP);
}
void loop()
{
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();
}
}

```

```

// -----sensor PIR-----

int gpioPIR = 15; //PIR Motion Sensor

pinMode(gpioPIR, INPUT_PULLUP);

int v = digitalRead(gpioPIR);

if( (v==1)&&(kondisi == 1)) {

sendCapturedImage2Telegram(token, chat_id);

delay(5000);

}

else if (kondisi == 0) {

v = 0;

}

delay(1000);

}

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;  
}
```

