

**Sistem keamanan Rumah berbasis Esp 32 CAM dan  
Sensor PIR Menggunakan Aplikasi Telegram untuk  
Pengawasan *Real-Time***



Oleh:

**Muhammad Rafli Saputra (FIB 022 083)**

**Ilham Ramdanil (F1B 022 099)**

**Abianda Aimar Rahman (F1B 022 101)**

Mata Kuliah: Teknologi IoT

Pembimbing: Dr. Misbahuddin, ST., MT., IPU.

**JURUSAN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS MATARAM**

**2024**

## KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan book chapter yang berjudul *Sistem Keamanan Rumah Berbasis ESP32-CAM dan Sensor PIR Menggunakan Aplikasi Telegram untuk Pengawasan Real-Time*. Book chapter ini disusun sebagai salah satu tugas dalam mata kuliah Teknologi IoT yang dibimbing oleh Dr. Misbahuddin, ST., MT., IPU.

Perkembangan teknologi Internet of Things (IoT) telah membawa berbagai inovasi dalam berbagai aspek kehidupan, salah satunya pada sistem keamanan rumah. Dalam book chapter ini, kami membahas implementasi perangkat berbasis ESP32-CAM dan sensor PIR sebagai solusi pengawasan keamanan rumah yang efisien dan terintegrasi dengan aplikasi Telegram untuk memberikan notifikasi real-time. Pembahasan meliputi prinsip kerja perangkat, metode pengolahan data, hingga pengintegrasian dengan platform komunikasi modern.

Kami menyadari bahwa dalam proses penyusunan ini masih terdapat kekurangan baik dalam penulisan maupun isi materi. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari berbagai pihak untuk penyempurnaan karya ini di masa mendatang. Kami juga mengucapkan terima kasih yang sebesar-besarnya kepada Dr. Misbahuddin, ST., MT., IPU., yang telah memberikan bimbingan, arahan, dan inspirasi selama proses penyusunan ini. Ucapan terima kasih juga kami sampaikan kepada semua pihak yang telah memberikan dukungan, baik secara langsung maupun tidak langsung.

Akhir kata, kami berharap book chapter ini dapat memberikan manfaat bagi pembaca, khususnya dalam memahami dan mengembangkan teknologi IoT untuk kebutuhan sistem keamanan rumah. Semoga karya ini dapat menjadi referensi yang bermanfaat bagi perkembangan ilmu pengetahuan dan teknologi di bidang IoT.

Mataram, 12 Desember 2024

Penulis

## PENGANTAR

### A. Latar Belakang

*Internet of Things* (IoT) telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk sistem keamanan rumah. IoT memungkinkan perangkat pintar saling terhubung dan berkomunikasi untuk memberikan solusi yang lebih efisien dan terintegrasi. Dalam konteks keamanan rumah, teknologi ini sangat relevan karena mampu memberikan pengawasan yang real-time dengan biaya yang relatif terjangkau.

Keamanan rumah menjadi perhatian penting bagi banyak individu di tengah meningkatnya risiko keamanan. Sistem tradisional seperti kamera CCTV seringkali memerlukan perangkat tambahan dan tidak selalu memberikan notifikasi instan. Oleh karena itu, muncul kebutuhan untuk mengembangkan sistem yang tidak hanya mendeteksi potensi ancaman tetapi juga memberi peringatan secara langsung kepada pengguna melalui platform komunikasi modern.

Proyek ini mengintegrasikan perangkat ESP32-CAM dan sensor PIR untuk menciptakan sistem keamanan rumah berbasis IoT. Dengan menggunakan aplikasi Telegram sebagai media komunikasi, sistem ini mampu mengirimkan notifikasi real-time saat sensor mendeteksi aktivitas mencurigakan. Implementasi ini diharapkan dapat menjadi solusi praktis yang mudah diterapkan oleh masyarakat luas.

### B. Rumusan Masalah

Dalam era digital, keamanan rumah menjadi tantangan yang semakin kompleks. Beberapa masalah yang menjadi fokus utama dalam penelitian ini antara lain:

- Bagaimana mendesain sistem keamanan yang dapat mendeteksi ancaman secara otomatis?
- Bagaimana mengintegrasikan perangkat IoT dalam sistem keamanan rumah yang efektif dan terjangkau?
- Bagaimana cara mengirimkan notifikasi secara real-time kepada pengguna saat terdeteksi adanya ancaman?

### **C. Tujuan Penulisan**

Tujuan utama dari penulisan ini adalah untuk mengembangkan sistem keamanan rumah berbasis teknologi IoT yang efisien dan dapat diakses melalui aplikasi Telegram. Secara spesifik, tujuan penulisan ini adalah :

- Merancang dan mengimplementasikan sistem keamanan rumah menggunakan ESP32-CAM dan sensor PIR.
- Mengintegrasikan sistem tersebut dengan aplikasi Telegram untuk memberikan pemberitahuan secara real-time.
- Menguji dan menganalisis kinerja sistem dalam mendeteksi ancaman dan memberikan notifikasi kepada pengguna.

### **D. Manfaat Penulisan**

Manfaat dari penulisan ini adalah sebagai referensi dalam pengembangan sistem keamanan berbasis IoT, terutama untuk aplikasi yang berfokus pada keamanan rumah. Penelitian ini juga diharapkan dapat memberikan kontribusi dalam pemahaman dan penerapan teknologi IoT untuk kebutuhan praktis sehari-hari. Selain itu, sistem yang dikembangkan dapat memberikan solusi keamanan yang lebih terjangkau dan mudah diakses oleh masyarakat umum.

## PUSTAKA PENDUKUNG

### 1. Library Yang Diperlukan

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <WebServer.h>
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
```

#### Analisa :

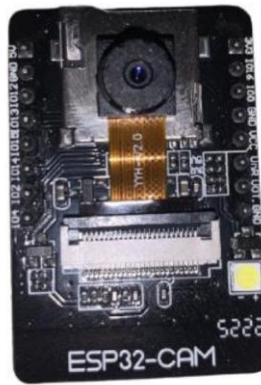
- `#include <Arduino.h>` adalah bagian utama dari semua program Arduino. Ini menyediakan fungsi-fungsi dasar seperti `setup()` dan `loop()`, serta akses ke pustaka standar Arduino.
- `#include <WiFi.h>` Pustaka ini digunakan untuk menghubungkan ESP32 ke jaringan Wi-Fi. Pustaka ini menyediakan fungsi untuk mengelola koneksi Wi-Fi, seperti `WiFi.begin(ssid, password)` untuk menghubungkan ke Wi-Fi.
- `#include <WiFiClientSecure.h>` Pustaka ini digunakan untuk membuat koneksi aman melalui protokol HTTPS menggunakan SSL/TLS. Ini biasanya diperlukan saat berkomunikasi dengan server yang membutuhkan keamanan data, seperti API yang mendukung HTTPS.
- `#include <WebServer.h>` Pustaka ini digunakan untuk membuat server web lokal di ESP32. Dengan pustaka ini, Anda bisa membuat aplikasi berbasis web yang berjalan di ESP32, seperti mengontrol perangkat atau menampilkan data.
- `#include "esp_camera.h"` Header ini digunakan untuk mengakses modul kamera pada ESP32 (seperti ESP32-CAM). Ini menyediakan fungsi-fungsi untuk mengonfigurasi kamera, menangkap gambar, dan mengolah hasilnya.
- `#include <UniversalTelegramBot.h>` Pustaka ini digunakan untuk mengintegrasikan bot Telegram dengan ESP32. Anda dapat membuat bot Telegram untuk menerima dan mengirim pesan, serta menjalankan perintah tertentu dari pengguna melalui Telegram.
- `#include <ArduinoJson.h>` Pustaka ini digunakan untuk menangani data JSON. JSON adalah format data yang sering digunakan untuk pertukaran informasi, terutama dengan API. Pustaka ini memungkinkan parsing, pembuatan, dan manipulasi data JSON di Arduino.

## 2. Modul Yang Dibutuhkan & Wiring Diagram

### a. ESP32-CAM

ESP32-CAM fungsinya sebagai mikrokontroler yang dapat terhubung ke WiFi sebagai mikrokontroler ini akan bersistem *Internet Of Things*.

Modul kamera pada ESP32 Cam berfungsi untuk mengambil foto secara real time dari Sistem keamanan Rumah. Modul kamera ini memungkinkan pengambilan foto dari lingkungan rumah dan kemudian mengirimkan data tersebut ke mikrokontroler ESP32 Cam. Data foto yang dihasilkan oleh modul kamera akan digunakan untuk memantau keadaan keamanan Rumah dengan lebih detail. Melalui koneksi WiFi, data foto ini akan dikirimkan ke aplikasi telegram di smartphone, sehingga pengguna dapat melihat kondisi keamanan Rumah secara langsung (Ine Agustine Cahyaningtyas, 2023).



Gambar 1. ESP32-CAM dan Modul Kamera

### b. Sensor PIR (*Passive Infrared Receiver*)

Sensor PIR (*Passive Infrared Receiver*) adalah sebuah sensor yang biasa digunakan untuk mendeteksi keberadaan manusia. Aplikasi ini biasa digunakan untuk system alarm pada rumah-rumah atau perkantoran. Keadaan ruangan dengan perubahan temperatur pada manusia dalam suatu ruangan menjadi nilai awal (*set point*) yang menjadi acuan dalam sistem pengontrolan. Perubahan temperatur pada manusia dalam ruangan akan terdeteksi oleh Sensor PIR. Dikatakan PIR (*Passive Infrared Receiver*) karena sensor ini hanya mengenali lingkungan tanpa adanya energi yang harus dipancarkan. PIR merupakan kombinasi sebuah kristal *pyroelectric*, filter dan lensa Fresnel (Alfin Fandriana, 2019).



Gambar 2. Sensor PIR

c. TTL (*Transistor-Transistor Logic*)

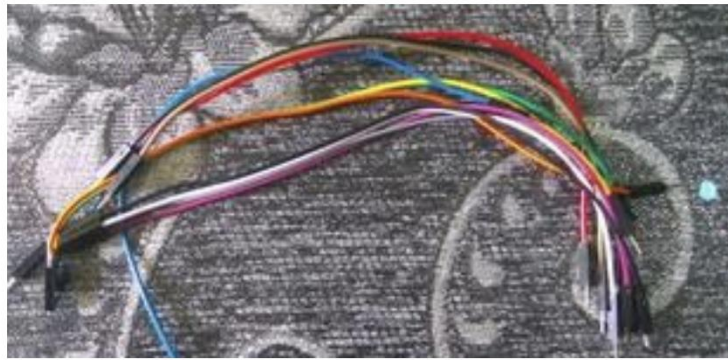
USB to TTL Converter adalah modul elektronik yang digunakan untuk menghubungkan perangkat berbasis logika digital (TTL: *Transistor-Transistor Logic*) dengan komputer melalui port USB. Modul ini berguna dalam komunikasi serial, terutama dengan mikrokontroler atau modul komunikasi yang tidak memiliki port USB langsung.



Gambar 3. TTL (*Transistor-Transistor Logic*)

d. Kabel Jumper

Kabel jumper adalah kabel yang di pergunakan untuk menghubungkan satu komponen dengan komponen lain ataupun menghubungkan jalur rangkaian yang terputus pada breadboard (Deny Nusyirwan, 2019).

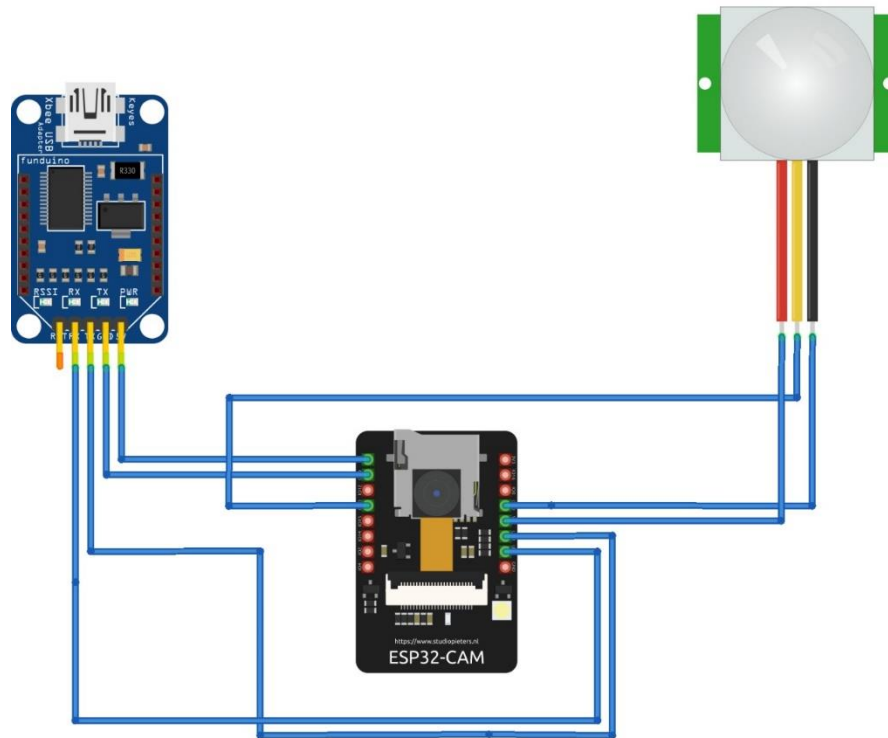


Gambar 4. Kabel Jumper

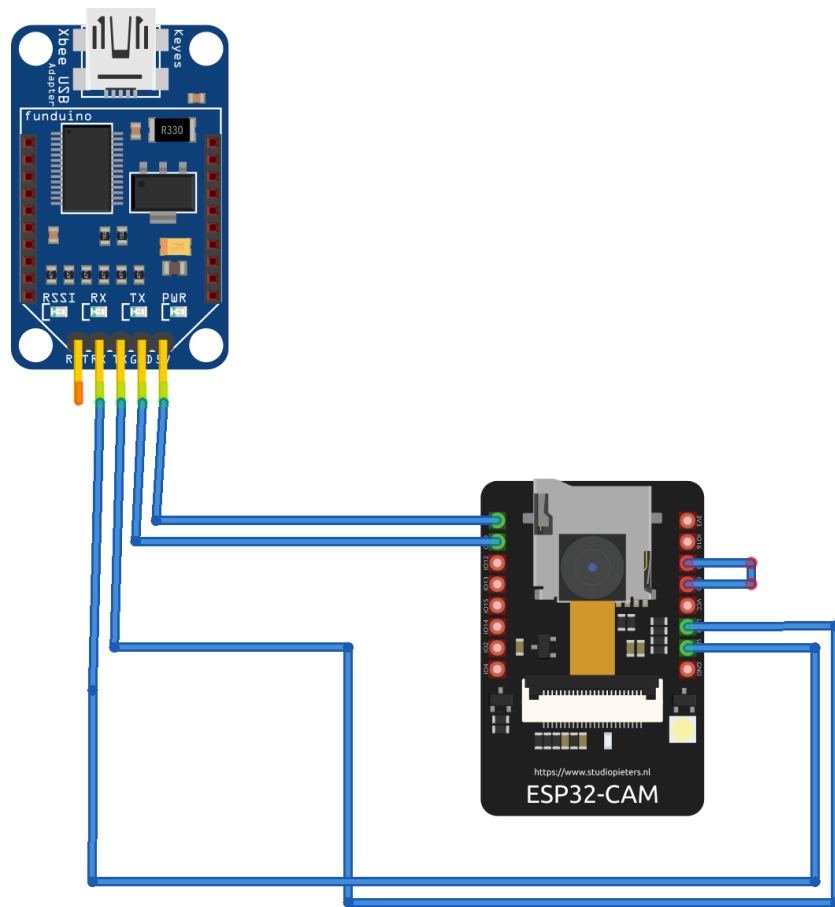


e. Wiring Diagram

- Full



- Sebelum ditambahkan dengan sensor PIR



### **3. Implementasi dibidang yang lain**

- a) Pertanian dan Peternakan
  - Pengawasan Kebun: Mendeteksi hama atau binatang liar.
  - Pemantauan Kandang: Mengawasi ternak dari pencurian atau predator.
- b) Perkantoran dan Gudang
  - Keamanan Aset: Memantau ruang server atau gudang barang berharga.
  - Kontrol Akses: Mengawasi pintu masuk di luar jam kerja.
- c) Kesehatan dan Perawatan
  - Pengawasan Lansia/Anak: Memastikan aktivitas normal di ruangan.
  - Alarm Aktifitas: Deteksi gerakan untuk pengingat aktivitas.

#### 4. Cara Pemrograman di Arduino IDE

a. Instal Arduino IDE :

Pastikan telah mengunduh dan menginstal Arduino IDE dari situs resmi Arduino (<https://www.arduino.cc/en/software>).

b. Siapkan Library :

Sebelum memulai, instal *library* yang diperlukan melalui *Library Manager* di Arduino IDE :

- WiFi untuk konektivitas Wi-Fi.
- WiFiClientSecure untuk koneksi HTTPS.
- UniversalTelegramBot untuk integrasi bot Telegram.
- ArduinoJson untuk parsing JSON.
- Library tambahan untuk kamera ESP32 (dapat diunduh dari ESP32 GitHub).

c. Hubungkan Board ESP32 :

- Hubungkan board ESP32 ke komputer menggunakan kabel USB.
- Pilih board ESP32 yang sesuai di menu: Tools > Board > ESP32 Arduino > AI-Thinker ESP32-CAM.
- Pilih port serial yang sesuai di Tools > Port.

d. Unggah Program :

- Buat Program di editor Arduino IDE sesuai alat yang dibutuhkan.
- Klik tombol Verify untuk memeriksa kesalahan.
- Klik tombol Upload untuk mengunggah kode ke board ESP32.

#### 5. Penjelasan Sketsa

Program di atas adalah proyek ESP32-CAM yang menggunakan sensor gerak (PIR) untuk mendeteksi pergerakan, mengambil foto, dan mengirimkannya melalui bot Telegram. Berikut adalah penjelasan bagian-bagian utama :

a) Deklarasi dan Inisialisasi

- Wi-Fi :

```
const char* ssid = "Abi";  
const char* password = "12345678";
```

Digunakan untuk menghubungkan ESP32 ke jaringan Wi-Fi.

- Telegram Bot

```
String BOTtoken =  
"7733208849:AAE16Hjd5YwEnvPLb7vVRpusIN73U63m3jE";  
String CHAT_ID = "1472951612";
```

Token bot Telegram dan ID chat untuk komunikasi dengan bot.

- Konfigurasi Kamera : GPIO pin untuk modul kamera ESP32-CAM didefinisikan sesuai dengan spesifikasi hardware.

## b) Fungsi Setup

Fungsi `setup()` dijalankan sekali saat perangkat dinyalakan :

- Mengatur pin sensor PIR (PIR\_SENSOR\_PIN) sebagai input.
- Mengonfigurasi modul kamera :

```
esp_err_t err = esp_camera_init(&config);
```

Menginisialisasi kamera dengan parameter konfigurasi.

- Menghubungkan ESP32 ke Wi-Fi :

```
WiFi.begin(ssid, password);  
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
}  
Serial.println("WiFi connected");
```

- Memulai server web lokal pada port 80 :

```
server.on("/photo", HTTP_GET, []() {  
    sendPhoto = true;  
});  
server.begin();
```

Menginisialisasi kamera dengan parameter konfigurasi.

## c) Fungsi Utama (Loop)

Fungsi `loop()` berjalan terus-menerus selama perangkat menyala :

- Menangani permintaan klien HTTP :

```
server.handleClient();
```

- Mengirim foto jika ada permintaan :

```
if (sendPhoto) {  
    sendPhotoTelegram();  
    sendPhoto = false;  
}
```

- Mendeteksi gerakan dengan sensor PIR :

```
if (digitalRead(PIR_SENSOR_PIN) == HIGH) {  
    if (!motionDetected) {  
        bot.sendMessage(CHAT_ID, "Motion detected, sending  
photo...", "");  
        sendPhotoTelegram();  
        motionDetected = true;  
    }  
} else {  
    motionDetected = false;  
}
```

- Menghandle pesan Telegram: Mengambil pesan baru menggunakan fungsi :

```
Int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
while (numNewMessages) {
    handleNewMessages(numNewMessages);
    numNewMessages = bot.getUpdates(bot.last_message_received + 1);
}
```

#### d) Fungsi Kirim Foto

sendPhotoTelegram() bertugas mengambil gambar dari kamera dan mengunggahnya ke Telegram menggunakan protokol HTTPS. Fungsi kirim foto digunakan untuk mengunggah atau mengirim gambar yang diambil dari sensor kamera ke server atau platform tertentu, seperti cloud atau aplikasi. Langkah-langkah utama yang dapat dilakukan adalah sebagai berikut :

- Inisialisasi Kamera

Menggunakan library ESP32 Camera atau library lain yang sesuai dengan hardware Anda. Pastikan semua pengaturan seperti resolusi dan format gambar telah diatur sesuai kebutuhan.

#### Code

```
#include "esp_camera.h"

void setupCamera() {
    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;
    // Konfigurasi pin kamera lainnya...
    config.xclk_freq_hz = 20000000;
    config.pixel_format = PIXFORMAT_JPEG;

    if (!esp_camera_init(&config)) {
        Serial.println("Kamera berhasil diinisialisasi");
    } else {
        Serial.println("Kamera gagal diinisialisasi");
    }
}
```

- Mengambil Foto

Ambil foto dari kamera menggunakan fungsi esp\_camera\_fb\_get(). Data gambar akan disimpan dalam buffer yang dapat digunakan untuk dikirimkan ke server.

Code
<pre> camera_fb_t *fb = NULL; fb = esp_camera_fb_get(); if (!fb) {     Serial.println("Gagal mengambil gambar");     return; } </pre>

- Mengunggah Gambar ke Server

Menggunakan protokol HTTP POST untuk mengunggah gambar. Pastikan Anda menggunakan library seperti WiFiClient atau HTTPClient.

Code
<pre> #include &lt;WiFi.h&gt;  #include &lt;HTTPClient.h&gt;  const char* ssid = "Your_SSID";  const char* password = "Your_PASSWORD";  const char* serverUrl = "http://example.com/upload";  void sendPhoto(camera_fb_t *fb) {      WiFiClient client;      HTTPClient http;      http.begin(client, serverUrl);      http.addHeader("Content-Type", "image/jpeg");      int httpResponseCode = http.POST(fb-&gt;buf, fb-&gt;len);      if (httpResponseCode &gt; 0) {          Serial.printf("Gambar berhasil dikirim, kode respon: %d\n", httpResponseCode);      } else { </pre>

```

        Serial.printf("Gagal mengirim gambar, kode
error:                                     %s\n",
http.errorToString(httpResponseCode).c_str());

    }

    http.end();

}

```

- **Membersihkan Buffer**

Setelah selesai, buffer gambar harus dibersihkan untuk membebaskan memori.

Code
<code>esp_camera_fb_return(fb);</code>

e) **Codingan Sistem keamanan Rumah berbasis Esp 32 CAM dan Sensor PIR Menggunakan Aplikasi Telegram untuk Pengawasan *Real-Time***

```

#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <WebServer.h>
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
const char* ssid = "Abi";
const char* password = "12345678";
String BOTtoken =
"7733208849:AAE16Hjd5YwEnvPLb7vVRpusIN73U63m3jE";
String CHAT_ID = "1472951612";
WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);
#define PIR_SENSOR_PIN 2
bool sendPhoto = false;
bool motionDetected = false;
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;
String sendPhotoTelegram();
#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
WebServer server(80);
void handleNewMessages(int numNewMessages) {
  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    String text = bot.messages[i].text;
    if (text == "/photo") {
      sendPhoto = true;
      bot.sendMessage(chat_id, "Taking a photo...", "");
    }
  }
}
String sendPhotoTelegram() {
  const char* myDomain = "api.telegram.org";
  String response;
  camera_fb_t * fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    return "Camera capture failed";
  }
  if (clientTCP.connect(myDomain, 443)) {
    String head = "--boundary\r\nContent-Disposition: form-data;
    name=\"chat_id\"\r\n\r\n" + CHAT_ID + "\r\n--
    boundary\r\nContent-Disposition: form-data; name=\"photo\";
    filename=\"esp32-cam.jpg\"\r\nContent-Type:
    image/jpeg\r\n\r\n";
    String tail = "\r\n--boundary--\r\n";
    clientTCP.println("POST /bot" + BOTtoken + "/sendPhoto
    HTTP/1.1");
    clientTCP.println("Host: " + String(myDomain));
    clientTCP.println("Content-Length: " + String(fb->len +
    head.length() + tail.length()));
    clientTCP.println("Content-Type:          multipart/form-data;
    boundary=boundary");
    clientTCP.println();
    clientTCP.print(head);
    clientTCP.write(fb->buf, fb->len);
    clientTCP.print(tail);
    esp_camera_fb_return(fb);
    while (clientTCP.connected()) {
      String line = clientTCP.readStringUntil('\n');
      if (line == "\r") break;
    }
    response = clientTCP.readString();
  } else {
    response = "Connection to Telegram API failed";
  }
  clientTCP.stop();
  return response;
}
void setup(){
  Serial.begin(115200);
  pinMode(PIR_SENSOR_PIN, INPUT);
  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;
config.fb_count = 2;
} else {
config.frame_size = FRAMESIZE_SVGA;
config.jpeg_quality = 12;
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();
}
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("WiFi connected");
server.on("/photo", HTTP_GET, []() {
sendPhoto = true;
});
server.begin();
}
void loop() {
server.handleClient();
if (sendPhoto) {
sendPhotoTelegram();
sendPhoto = false;
}
if (digitalRead(PIR_SENSOR_PIN) == HIGH) {
if (!motionDetected) {
bot.sendMessage(CHAT_ID, "Motion detected, sending photo...", "");
sendPhotoTelegram();
motionDetected = true;
}
} else {
motionDetected = false;
}
if (millis() > lastTimeBotRan + botRequestDelay) {

```

```
int numNewMessages =  
bot.getUpdates(bot.last_message_received + 1);  
while (numNewMessages) {  
  handleNewMessages(numNewMessages);  
  numNewMessages = bot.getUpdates(bot.last_message_received +  
  1);  
}  
lastTimeBotRan = millis();  
}  
}
```

f) Dokumentasi Demonstrasi





## DAFTAR PUSTAKA

- Cahyaningtyas, I. A., Stefanie, A., & Ibrahim. *Implementasi ESP32 CAM dan Kodular Berbasis Android untuk Monitoring Smart Garden*. Program Studi Teknik Elektro S1, Fakultas Teknik, Universitas Singaperbangsa Karawang, Karawang, Indonesia.
- Fandriana, A., Soebroto, & Yasin, R. P. *Rancangan Motion Detector dengan Sensor PIR (Passive Infrared Receiver) dan Kamera Menggunakan Arduino UNO Berbasis SMS Gateway pada Antenna Distribution Unit Localizer di Bandara Halim Perdanakusuma*. Sekolah Tinggi Penerbangan Indonesia.
- Nusyirwan, D., & Alfarizi. *“Fun Book” Rak Buku Otomatis Berbasis Arduino dan Bluetooth pada Perpustakaan untuk Meningkatkan Kualitas Siswa*. Jurusan Teknik Elektro, Fakultas Teknik, Program Studi Teknik Elektro, Universitas Maritim Raja Ali Haji (UMRAH), Tanjungpinang, Indonesia.