APPENDIX

CODING DECLARATION

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
#include <Wire.h>

const char* ssid = "iphonefitria"; //nama wifi
const char* password = "12345678"; //password wifi
String chatId = "1160943523"; //id telegram
String BOTtoken = "1705881550:AAGjOm-wCGODKxoyBhbCS4Jt0P_pg_mnQQE"; //token bot
bool sendPhoto = false;

WiFiClientSecure clientTCP;

 UniversalTelegramBot bot(BOTtoken, clientTCP);
//Library
//CAMERA_MODEL_AI_THINKER
#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
//

#define FLASH_LAMPU 4 // definisi flash dan buzzer
#define pir 2 // definisi pir
bool flashLed = LOW;
bool sensor = 0; //posisi pir ketika hidup lsg mati

int botRequestDelay = 1000;   // mean time between scan messages
long lastTimeBotRan;     // last time messages' scan has been done

```

```
48.  void handleNewMessages(int numNewMessages);
49.  String sendGambarTelegram();
50.
51.  void setup(){
52.    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
53.    Serial.begin(115200);
54.    pinMode(FLASH_LAMPU, OUTPUT);
55.    pinMode(pir,INPUT);
56.    digitalWrite(FLASH_LAMPU, flashLed);
57.  //library
58.    WiFi.mode(WIFI_STA);
59.    Serial.println();
60.    Serial.print("Connecting to ");
61.    Serial.println(ssid);
62.    WiFi.begin(ssid, password);
63.    while (WiFi.status() != WL_CONNECTED) {
64.      Serial.print(".");
65.      delay(500);
66.    }
67.    Serial.println();
68.    Serial.print("ESP32-CAM IP Address: ");
69.    Serial.println(WiFi.localIP());
70.  //
71.    //library
72.    camera_config_t config;
73.    config.ledc_channel = LEDC_CHANNEL_0;
74.    config.ledc_timer = LEDC_TIMER_0;
75.    config.pin_d0 = Y2_GPIO_NUM;
76.    config.pin_d1 = Y3_GPIO_NUM;
77.    config.pin_d2 = Y4_GPIO_NUM;
78.    config.pin_d3 = Y5_GPIO_NUM;
79.    config.pin_d4 = Y6_GPIO_NUM;
80.    config.pin_d5 = Y7_GPIO_NUM;
81.    config.pin_d6 = Y8_GPIO_NUM;
82.    config.pin_d7 = Y9_GPIO_NUM;
83.    config.pin_xclk = XCLK_GPIO_NUM;
84.    config.pin_pclk = PCLK_GPIO_NUM;
85.    config.pin_vsync = VSYNC_GPIO_NUM;
86.    config.pin_href = HREF_GPIO_NUM;
87.    config.pin_sscb_sda = SIOD_GPIO_NUM;
88.    config.pin_sscb_scl = SIOC_GPIO_NUM;
89.    config.pin_pwdn = PWDN_GPIO_NUM;
90.    config.pin_reset = RESET_GPIO_NUM;
91.    config.xclk_freq_hz = 20000000;
92.    config.pixel_format = PIXFORMAT_JPEG;
93.  //
94.    //init with high specs to pre-allocate larger buffers
95.    if(psramFound()){
96.      config.frame_size = FRAMESIZE_UXGA;
97.      config.jpeg_quality = 10;  //0-63 lower number means higher quality
98.      config.fb_count = 2;
99.    } else {
100.     config.frame_size = FRAMESIZE_SVGA;
101.     config.jpeg_quality = 12;  //0-63 lower number means higher quality
102.     config.fb_count = 1;
103.   }
```

```
104.
105.   // camera init
106.   esp_err_t err = esp_camera_init(&config);
107.   if (err != ESP_OK) {
108.     Serial.printf("Camera init failed with error 0x%x", err);
109.     delay(1000);
110.     ESP.restart();
111.   }
112.
113.   // Drop down frame size for higher initial frame rate
114.   sensor_t * s = esp_camera_sensor_get();
115.   s->set_framesize(s,           FRAMESIZE_SVGA);                    //
   UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
116.
117. }
118.
119. void loop(){
120.   while (WiFi.status() != WL_CONNECTED) {
121.     WiFi.begin(ssid, password);
122.     Serial.print(".");
123.     delay(2000);
124.   }
125.
126.   if (sendGambar){
127.     Serial.println("Preparing photo");
128.     sendGambarTelegram();
129.     sendGambar = false;
130.     flashLed = !flashLed;
131.     digitalWrite(FLASH_LAMPU, flashLed);
132.   }
133.
134.   if (sensor){
135.     delay(1000);
136.     if(digitalRead(pir) == 1){
137.       flashLed = !flashLed;
138.       digitalWrite(FLASH_LAMPU, flashLed);
139.       Serial.print("Motion Detected, Value = ");
140.       Serial.println(digitalRead(pir));
141.       String motion = "Terdeteksi gerakan!!\n";
142.       motion += "Foto akan segera dikirim\n";
143.       bot.sendMessage(chatId, motion, "");
144.       sendGambarTelegram();
145.       flashLed = !flashLed;
146.       digitalWrite(FLASH_LAMPU, flashLed);
147.     }
148.   }
149.
150.   if (millis() > lastTimeBotRan + botRequestDelay){
151.     int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
152.     while (numNewMessages){
153.       Serial.println("got response");
154.       handleNewMessages(numNewMessages);
155.       numNewMessages = bot.getUpdates(bot.last_message_received + 1);
156.     }
157.     lastTimeBotRan = millis();
158.   }
```

```
  delay(850);
}
//library
String sendGambarTelegram(){
  const char* myDomain = "api.telegram.org";
  String getAll = "";
  String 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));

  if (clientTCP.connect(myDomain, 443)) {
    Serial.println("Connection successful");

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

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

    clientTCP.println("POST /bot"+BOTtoken+"/sendGambar HTTP/1.1");
    clientTCP.println("Host: " + String(myDomain));
    clientTCP.println("Content-Length: " + String(totalLen));
    clientTCP.println("Content-Type: multipart/form-data; boundary=RandomNerdTutorials");
    clientTCP.println();
    clientTCP.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) {
        clientTCP.write(fbBuf, 1024);
        fbBuf += 1024;
      }
      else if (fbLen%1024>0) {
        size_t remainder = fbLen%1024;
        clientTCP.write(fbBuf, remainder);
      }
    }

    clientTCP.print(tail);

    esp_camera_fb_return(fb);
```

```
211.
212.     int waitTime = 10000;   // timeout 10 seconds
213.     long startTimer = millis();
214.     boolean state = false;
215.
216.     while ((startTimer + waitTime) > millis()){
217.       Serial.print(".");
218.       delay(100);
219.       while (clientTCP.available()){
220.           char c = clientTCP.read();
221.           if (c == '\n'){
222.             if (getAll.length()==0) state=true;
223.             getAll = "";
224.           }
225.           else if (c != '\r'){
226.             getAll += String(c);
227.           }
228.           if (state==true){
229.             getBody += String(c);
230.           }
231.           startTimer = millis();
232.        }
233.        if (getBody.length()>0) break;
234.     }
235.     clientTCP.stop();
236.     Serial.println(getBody);
237.   }
238.   else {
239.     getBody="Connected to api.telegram.org failed.";
240.     Serial.println("Connected to api.telegram.org failed.");
241.   }
242.   return getBody;
243. }
244. //
245. void handleNewMessages(int numNewMessages){
246.   Serial.print("Handle New Messages: ");
247.   Serial.println(numNewMessages);
248.
249.   for (int i = 0; i < numNewMessages; i++){
250.     // Chat id of the requester
251.     String chat_id = String(bot.messages[i].chat_id);
252.     if (chat_id != chatId){
253.       bot.sendMessage(chat_id, "Unauthorized user", "");
254.       continue;
255.     }
256.
257.     // Print the received message
258.     String text = bot.messages[i].text;
259.     Serial.println(text);
260.
261.     String fromName = bot.messages[i].from_name;
262.
263.
264.
265.     if (text == "/piron"){ //deklarasi sensor pir untuk hidup
266.       sensor = 1;
```

```
      bot.sendMessage(chatId, "PIR Sensor sudah aktif, Saat terjadi gerakan anda akan dikirimkan foto", "");
    }

    if (text == "/piroff"){ // deklarasi sensor pir untuk mati
      sensor = 0;
      bot.sendMessage(chatId, "PIR sensor sudah mati, Anda tidak akan menerima pemberitahuan lagi saat terjadi gerakan", "");
    }



    if (text == "/start"){
      String welcome = "Selamat datang di My Home Security .\n";
      welcome += "Berikut adalah perintah yang dapat digunakan :\n";
      welcome += "/piron : Mengaktifkan sensor PIR\n"; //perintah sensor pir untuk nyala di tele
      welcome += "/piroff : Mematikan sensor PIR\n"; //perinmtah sensor pir untuk mati di tele
//      welcome += "/readings : request sensor readings\n\n";
      welcome += "Anda juga akan mendapatkan notifikasi saat terjadi gerakan dari sensor PIR\n";
      bot.sendMessage(chatId, welcome, "");
    }
  }
}
```

PLAGIARISM CHECK.ORG

6.56% PLAGIARISM APPROXIMATELY | **1.47%** IN QUOTES

Report #13872419

INTRODUCTION Background The high number of unemployed and the pressure on the necessities of life, causing the crime that occurs continues to increase, especially for the crime of theft in the neighborhood. This incident does not only occur in big cities, but also in the countryside. Thefts often occur at night when the occupants are sleeping, but not infrequently the perpetrators carry out their actions during the day or when the house is quiet. With the continued occurrence of similar incidents, the condition of the house becomes uncomfortable and can cause property losses. If this continues, additional surveillance will be required, such as installing CCTV or hiring a home security guard, but this costs a lot of money. In the current era, the mobility of human needs requires technology that is fast-paced and easy to access and does not interfere with activities, so the implementation of a home security system is designed based on IoT or Internet of things. The system design focuses on being able to carry out

g