

## BAB II

### METODE PENELITIAN

#### 2.1 Lokasi Penelitian

Penelitian ini dilakukan di Kota Samarinda Kecamatan Samarinda Utara Kabupaten Kalimantan Timur. Penelitian ini akan dilakukan pada bulan Mei 2024.

#### 2.2 Objek Penelitian

Objek penelitian ini adalah sistem keamanan cerdas untuk pencegahan pencurian motor menggunakan *ESP32-CAM* dan sensor getar. Sistem keamanan cerdas ini dirancang untuk diterapkan pada kendaraan roda dua, agar bisa memantau kondisi motor secara real-time, mendeteksi getaran yang mencurigakan, dan memberikan notifikasi kepada pemilik motor.

#### 2.3 Alat dan Bahan

Alat dan bahan yang digunakan untuk penelitian dalam menjelaskan metode ini diantaranya :

##### 1. *ESP32-CAM*

*ESP32-CAM* adalah mikrokontroler yang dilengkapi dengan fitur tambahan seperti *Bluetooth*, *Wifi*, kamera, dan slot *microSD*. *ESP32-CAM* biasanya digunakan dalam proyek internet of Things (IoT) yang membutuhkan fitur kamera. Dibandingkan dengan modul *ESP32* sebelumnya seperti *ESP32 Wroom*, *ESP32-CAM* memiliki lebih sedikit pin I/O (Andesta & Ferdian, 2018).

Modul *ESP32-CAM* memiliki dua sisi dalam rangkaiannya. Di bagian atas terdapat modul kamera yang bisa dilepas-pasang, slot *microSD* yang bisa diisi, serta lampu flash yang berfungsi sebagai lampu tambahan untuk kamera jika diperlukan (Andesta & Ferdian, 2018).



Gambar 2. 1 ESP32-CAM

## 2. Sensor Getar

Sensor getar digunakan untuk mendeteksi getaran pada kondisi tertentu di sepeda motor. Getaran yang dimaksud terjadi ketika seorang pencuri mencoba membobol sepeda motor di area tempat sensor dipasang. Sensor getar ini menggunakan modul *SW-420* tipe NC dengan tegangan kerja antara 3.3V hingga 5V. (Hamdani et al., 2019).



Gambar 2.2 SW-420

## 3. Kabel Jumper

Kabel jumper adalah kabel elektrik yang digunakan untuk menghubungkan komponen-komponen di *breadboard* tanpa perlu menyolder. Kabel jumper biasanya memiliki konektor atau di kedua ujungnya. Konektor yang digunakan untuk menusuk disebut *male connector*, sedangkan konektor yang digunakan untuk ditusuk disebut *female connector* (Williams, 1930).



Gambar 2.3 Kabel jumper

#### 4. Powerbank

Power bank, juga dikenal sebagai portable charger atau baterai cadangan, adalah perangkat yang digunakan untuk menyimpan energi dan mengisi ulang gadget saat sedang berpergian. Konsep portable external baterai pertama kali dipamerkan pada tahun 2001 di *Las Vegas International Consumer Electronics Show*, dan sejak itu semakin populer dengan munculnya smartphone pada tahun 2009 yang membutuhkan pengisian daya yang terus-menerus untuk tetap terhubung (Lak'apu et al., 2023).



Gambar 2.4 Powerbank

Power bank tersedia dalam berbagai kapasitas, mulai dari 3000 mAh hingga 11000 mAh, yang memungkinkan mereka untuk mengisi ulang gadget dengan kapasitas baterai yang lebih rendah. Sebagai contoh, power bank dengan kapasitas 3000 mAh dapat mengisi ulang smartphone dengan kapasitas baterai 1500 mAh hingga dua kali penuh (Lak'apu et al., 2023).

#### 5. Kabel *USB Micro*

*Micro USB* memiliki desain yang lebar dan pipih, dan masih umum ditemukan pada smartphone dan perangkat pendukung seperti powerbank. Selain itu, *Micro USB* juga digunakan pada berbagai perangkat elektronik lainnya, misalnya untuk transfer data dan pengisian daya baterai kamera digital (Alfi et al., 2019).



Gambar 2.5 USB Micro

## 2.4 Penggunaan Software

### 1. Telegram

*Telegram* adalah salah satu aplikasi perpesanan yang terkenal karna keunggulanya dalam hal kecepatan, privasi, dan berbagai fitur tambahan yang bermanfaat. Salah satunya adalah *Telegram Bot*, yang memungkinkan pengguna untuk berintraksi dengan bot melalui pesan teks dan perintah khusus (Ardiansyah et al., 2015)

Penggunaan *Telegram Bot* untuk berkomunikasi dengan perangkat *mikrokontroler* adalah contoh bagaimana teknologi modern seperti aplikasi perpesanan dapat diaplikasikan dalam konteks, mulai dari sistem keamanan kendaraan bermotor dan kontrol perangkat dari jarak jauh.



Gambar 2.6 Telegram

### 2. Arduino IDE

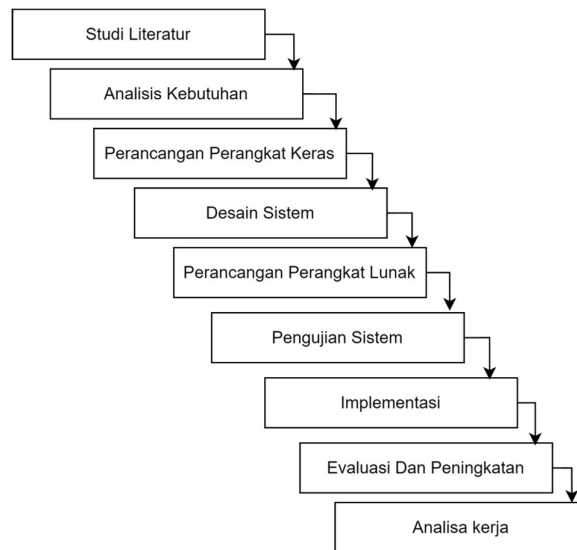
*Arduino IDE* adalah perangkat lunak yang digunakan untuk membuat dan mengedit kode pemrograman, yang disebut sketch, untuk *board Arduino* yang ingin diprogram. Dengan *Arduino IDE*, Anda dapat membuat, mengedit, mengupload kode ke board Arduino yang spesifik, dan mengembangkan program-program khusus. *Arduino IDE* dikembangkan menggunakan bahasa pemrograman *java* dan dilengkapi dengan berbagai pustaka (library) C/C++ yang disebut *WIRING*, yang mempermudah operasi input dan output pada board Arduino (Mahanin Tyas et al., 2023).



Gambar 2.7 Arduino IDE

## 2.5 Prosedur Penelitian

Adapun tahap penelitian yang di gunakan dalam penelitian ini adalah sebagai berikut:



Gambar 2.8 Tahapan Penelitian

### 2.5.1 Studi Literatur

Identifikasi literatur terkait sistem keamanan cerdas, teknologi *ESP32-CAM*, sensor getar, dan pencegahan pencurian motor. Tinjau riset-riset terkait untuk memahami konsep dan teknologi yang relevan.

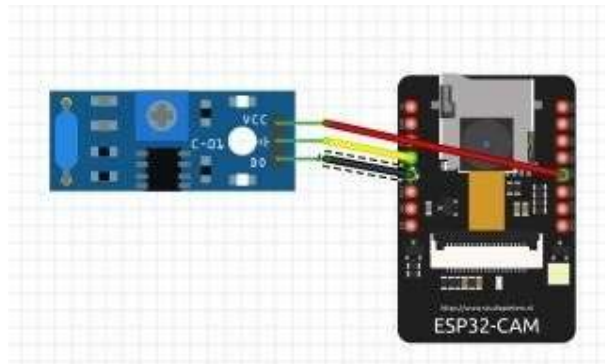
### 2.5.2 Analisis Kebutuhan

Kebutuhan yang akan digunakan dalam membuat sistem keamanan motor ini menggunakan *ESP32-CAM* dan sensor getar untuk melakukan proses analisis kebutuhan sistem yang bertujuan agar sistem keamanan yang dipakai nantinya sesuai dengan hasil yang akan di buat.

### 2.5.3 Perancangan Perangkat Keras

Perancangan perangkat keras merupakan tahapan krusal dalam pengembangan sistem keamanan cerdas untuk mencegah pencurian motor. Melalui perancangan yang matang, kita dapat menetapkan spesifikasi yang tepat, mengoptimalkan kinerja sistem, mengidentifikasi dan mengatasi tantangan teknis, serta mengelola biaya dengan efektif. Dibawah ini adalah hasil dari rangkaian keseluruhan sistem yang dapat dijelaskan bahwa rangkaian terdiri dari *ESP32-CAM* dan *Sw-420* berfungsi sebagai sistem

keamanan pencurian motor, yaitu mulai dari mendeteksi getaran lalu mengambil gambar dan mengirimkan notifikasi seperti gambar. Berikut rangkaian keseluruhan dapat dilihat pada Gambar 2.6 Perancangan perangkat Keras.



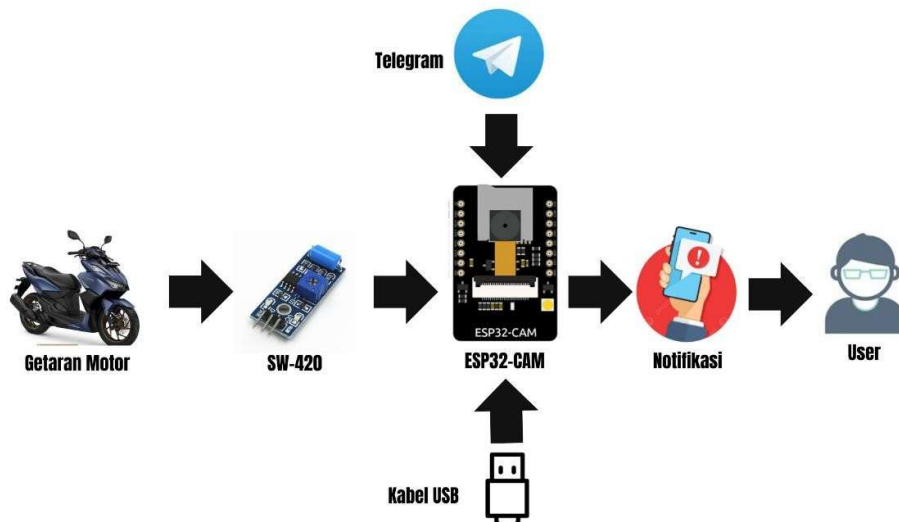
Gambar 2.9 Perancangan perangkat keras

Tabel 2.1 Pin-pin I/O ESP32-CAM

No	Nama Perangkat Keras	Pin pada ESP32-CAM
1	Sensor <i>SW-420</i>	GPIO 13

Tabel 2.1 hanya memiliki 1 sensor yang terhubung ke *ESP32-CAM* yaitu Sensor Getar (*SW-420*) pin yang digunakan adalah pin 13 untuk terhubung ke sensor getar, yang digunakan untuk mendeteksi getaran yang mungkin menandakan aktivitas atau perubahan kondisi pada motor.

## 2.5.4 Desain Sistem



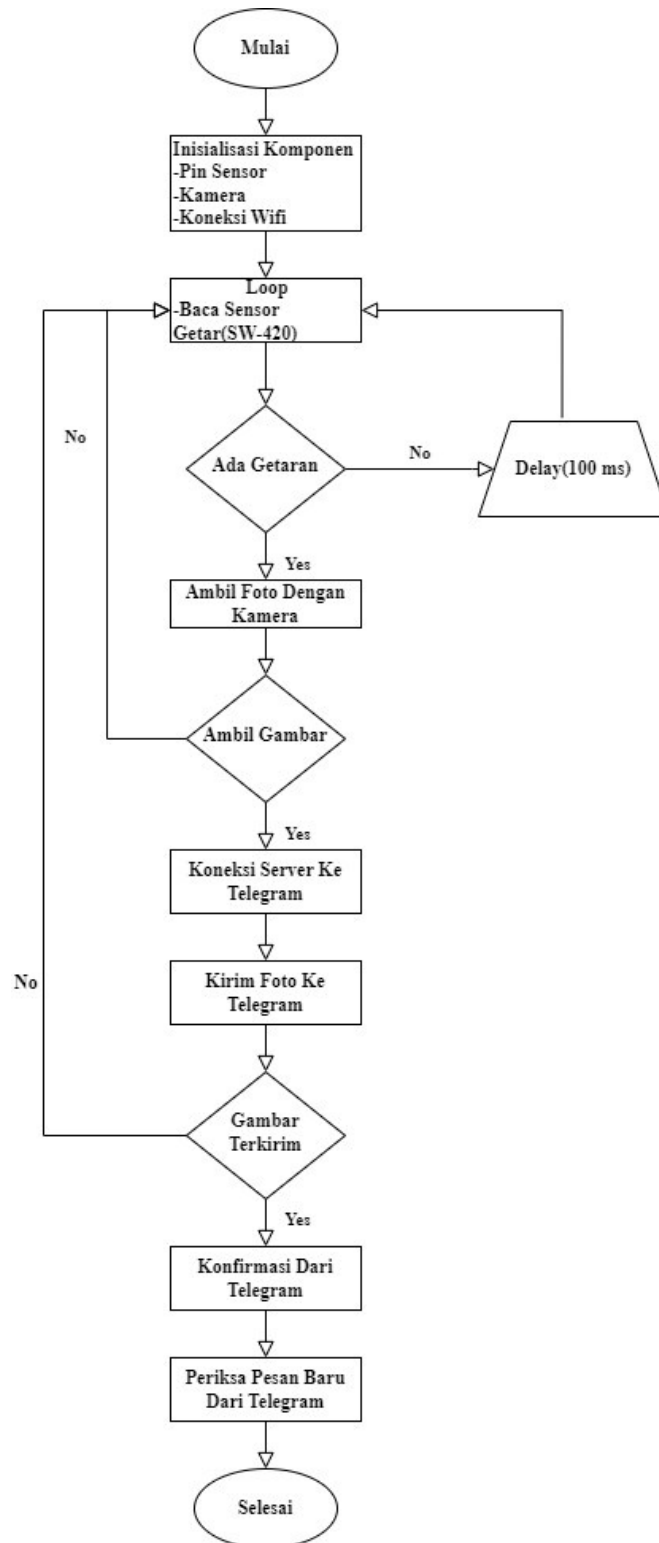
Gambar 2.10 Desain Sistem Keamanan Sepeda Motor

Alur Kerja Sistem pada Gambar 2.10 :

1. Sistem diaktifkan dan menghubungkan ke jaringan Wi-Fi.
2. Sensor mendeteksi getaran pada motor dan mengirimkan sinyal ke *ESP32-CAM*.
3. *ESP32-CAM* mengambil gambar saat menerima sinyal dari sensor getar.
4. Data Gambar dikirim ke server *Telegram* Melalui koneksi Wi-Fi.
5. Server *Telegram* menerima data gambar.
6. Server mengirim ke pengguna melalui *bot Telegram*.

## 2.5.5 Perancangan Perangkat Lunak

Flowchart Sistem Keamanan :



Gambar 2.11 Flowchart Sistem



Pada Gambar 2.11 menjelaskan bagaimana cara kerja sistem keamanan motor berkerja. Berikut penjelasan rinci cara kerja flowchart pada Gambar 2.11 :

1. **Mulai:** Program dimulai dari titik ini.
2. **Inisialisasi Komponen:**
  - a. Inisialisasi pin sensor getar, kamera, dan koneksi WIFI.
  - b. Konfigurasi kamera menggunakan *ESP32-CAM*.
3. **Loop:**
  - a. Program memasuki loop utama untuk terus melakukan pemantauan dan pengambilan tindakan berdasarkan kondisi sensor dan pesan Telegram.
4. **Baca Sensor Getaran:**
  - a. Program membaca sensor getaran (*SW-420*) untuk mendeteksi adanya getaran.
5. **Ada Getaran?:**
  - a. Jika sensor mendeteksi getaran maka akan memasuki langkah untuk mengambil gambar.
6. **Tunggu 100 ms:**
  - a. Setelah tidak ada getaran yang terdeteksi, program menunggu selama 100 ms sebelum membaca sensor lagi. Ini membantu menghindari pembacaan berlebihan.
7. **Ambil Foto dengan Kamera:**
  - a. Program melalui proses untuk mengambil gambar menggunakan kamera *ESP32-CAM*.
  - b. Jika gagal mengambil gambar (misalnya karena kegagalan koneksi atau masalah lain), proses kembali ke langkah “Baca Sensor Getar”.
8. **Gambar Diambil?:**
  - a. Program memeriksa apakah gambar berhasil diambil.
  - b. Jika tidak berhasil, kembali ke langkah “Baca Sensor Getar”.
  - c. Jika berhasil, lanjut ke langkah selanjutnya untuk mengirim gambar ke Telegram.
9. **Koneksi Ke Sever Telegram:**
  - a. Program membuat koneksi ke server Telegram menggunakan protokol HTTPS.
  - b. Mengirim gambar yang telah diambil sebagai pesan dengan format multipart/form-data.
10. **Kirim Foto Ke Telegram:**
  - a. Proses pengiriman gambar ke Telegram dimulai.
  - b. Program menunggu konfirmasi atau respons dari sever Telegram setelah mengirimkan gambar.
11. **Gambar Terkirim?:**
  - a. Program memeriksa apakah gambar berhasil terkirim ke Telegram.
  - b. Jika tidak berhasil, kembali ke langkah “Baca Sensor Getaran” untuk memulai ulang proses.
  - c. Jika berhasil, lanjut ke langkah untuk memeriksa pesan baru dari Telegram.
12. **Periksa Pesan Baru Dari Telegram:**

- a. Program memeriksa apakah ada pesan baru atau respons dari Telegram setelah pengiriman gambar.
- b. Tanggapan dari Telegram bisa berupa konfirmasi pengiriman, pesan dari pengguna, atau intruksi lain.

13. **Selesai:**

- a. Program berakhir setelah selesai menjalankan tugasnya.

Flowchart di atas menggambarkan Langkah-langkah utama dalam sistem keamanan cerdas untuk pencegahan pencurian motor, ini mencakup inisialisasi perangkat keras dan layanan, deteksi getaran, pengambilan gambar, pengiriman notifikasi ke Telegram. Berikut adalah program yang telah disediakan untuk sistem keamanan yang menggunakan *ESP32-CAM* dan sensor getar, dengan pengiriman notifikasi melalui Telegram:

1. ***Inklusi Header dan Definisi Variabel***

```
#include <Arduino.h>
#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>
```

Serial Monitor ×

Gambar 2.12 Program 1

Gambar 2.12 adalah pustaka yang diperlukan untuk koneksi Wifi, kamera, dan komunikasi dengan *bot Telegram*.

## 2. Konfigurasi Koneksi Wifi dan *Bot Telegram*

```
// WiFi credentials
const char* ssid = "Bobli";
const char* password = "apamserawa";
// Telegram Bot Token and Chat ID
String BOTtoken = "7442583602:AAHuLiRU-hAuJbHII1KN8jmhF-uyYvUzxR0";
String CHAT_ID = "6842233029";
// Telegram and WiFi client initialization
WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);
```

Serial Monitor ×

Gambar 2.13 Program 2

Gambar 2.13 ini berisi kredensial Wifi dan token *bot Telegram* serta menginisialisasi klien Wifi dan *bot Telegram*.

## 3. Konfigurasi Pin Kamera dan Sensor Getar

```
// Camera pins definition
#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

// Vibration sensor pin definition
#define VIBRATION_SENSOR_PIN 13
```

Serial Monitor ×

Gambar 2.14 Program 3

Gambar 2.14 adalah deifinisi pin yang digunakan oleh modul kamera dan pin yang digunakan untuk sensor getar *SW-420*.

#### 4. Thershold Getaran

```
const int vibrationThresholdLow = 1;
const int vibrationThresholdMedium = 2;
const int vibrationThresholdHigh = 3;
```

Serial Monitor ×

Gambar 2.15 Program 4

Gambar 2.15 Mendefinisikan nilai ambang untuk mendeteksi level getaran rendah, sedang, dan tinggi.

#### 5. Fungsi 'handleNewMessages()'

```
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;
    String from_name = bot.messages[i].from_name;

    if (text == "/start") {
      String welcome = "Welcome , " + from_name + "\n";
      bot.sendMessage(chat_id, welcome, "");
    } else if (text == "/photo") {
      sendPhoto = true;
      bot.sendMessage(chat_id, "New photo request received.");
    }
  }
}
```

Serial Monitor ×

Gambar 2.16 Program 5

Fungsi pada gambar 2.16 adalah memeriksa pesan baru yang diterima oleh *bot telegram*.

## 6. Mengirim Foto Ke *Telegram*

```
String sendPhotoTelegram(int vibrationLevel) {
    const char* myDomain = "api.telegram.org";
    String getAll = "";
    String getBody = "";
    digitalWrite(4, HIGH);
    camera_fb_t * fb = esp_camera_fb_get();
    delay(200);
    digitalWrite(4, LOW);

    if (!fb) {
        Serial.println("Camera capture failed");
        delay(1000);
        ESP.restart();
        return "Camera capture failed";
    }
}
```

Serial Monitor ×

Gambar 2.17 Program 6

```
if (clientTCP.connect(myDomain, 443)) {
    String head = "--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"chat_i";
    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 + "/sendPhoto 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);
}
```

Serial Monitor ×

Gambar 2.18 Program 7

```

uint8_t *fbBuf = fb->buf;
size_t fbLen = fb->len;
for (size_t n = 0; n < fbLen; n += 1024) {
    size_t chunkSize = (n + 1024 < fbLen) ? 1024 : fbLen % 1024;
    clientTCP.write(fbBuf, chunkSize);
    fbBuf += chunkSize;
}

clientTCP.print(tail);
esp_camera_fb_return(fb);

int waitTime = 10000;
long startTimer = millis();
boolean state = false;

```

Serial Monitor ×

Gambar 2.19 Program 8

```

while ((startTimer + waitTime) > millis()) {
    delay(100);
    while (clientTCP.available()) {
        char c = clientTCP.read();
        if (state) getBody += String(c);
        if (c == '\n') {
            if (getAll.length() == 0) state = true;
            getAll = "";
        } else if (c != '\r') getAll += String(c);
        startTimer = millis();
    }
    if (getBody.length() > 0) break;
}
clientTCP.stop();
Serial.println(getBody);
} else {
    getBody = "Connection to api.telegram.org failed.";
    Serial.println(getBody);
}
return getBody;
}

```

Serial Monitor ×

Gambar 2.20 Program 9

Pada gambar 2.17, 2.18 dan 2.19 yaitu berfungsi sebagai mengambil foto dan menyimpannya di buffer, menghubungkan ke server *API Telegram* dan mempersiapkan data foto untuk diunggah. Mengirim data foto ke server Telegram dan mengembalikan hasilnya dan jika kamera gagal mengambil foto, *ESP32-CAM* akan restart.

## 7. Fungsi 'setup()'

```
void setup() {  
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);  
  Serial.begin(115200);  
  pinMode(VIBRATION_SENSOR_PIN, INPUT);  
  pinMode(4, OUTPUT);  
  digitalWrite(4, LOW);  
  
  WiFi.mode(WIFI_STA);  
  WiFi.begin(ssid, password);  
  clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT);  
  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
}
```

Serial Monitor ×

Gambar 2.21 Program 10

```
Serial.println("\nWiFi connected");  
Serial.println("IP address: ");  
Serial.println(WiFi.localIP());  
  
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;
```

Serial Monitor ×

Gambar 2.22 Program 11

```
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;
}

if (esp_camera_init(&config) != ESP_OK) {
  Serial.printf("Camera init failed");
  delay(1000);
  ESP.restart();
}

sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_CIF);
}

Serial Monitor x
```

Gambar 2.23 Program 12

Pada gambar 2.21, 2.22 dan 2.23 untuk mencegah masalah saat tegangan rendah, untuk debugging, mengatur pin sensor getar dan *LED*, menghubungkan *ESP32-CAM* ke jaringan Wifi, serta mengatur konfigurasi kamera dan menginisialisasinya.



## 8. Fungsi 'Loop()'

```
void loop() {
  int vibrationLevel = analogRead(VIBRATION_SENSOR_PIN);

  if (vibrationLevel >= vibrationThresholdHigh) {
    Serial.println("High vibration detected!");
    sendPhoto = true;
  } else if (vibrationLevel >= vibrationThresholdMedium) {
    Serial.println("Medium vibration detected!");
    sendPhoto = true;
  } else if (vibrationLevel >= vibrationThresholdLow) {
    Serial.println("Low vibration detected but not triggering photo.");
  }
}
```

Serial Monitor ×

Gambar 2.24 Program 13

```
if (sendPhoto) {
  bot.sendMessage(CHAT_ID, "Sending photo...");
  sendPhotoTelegram(vibrationLevel);
  sendPhoto = 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();
}
delay(50);
}
```

Serial Monitor ×

Gambar 2.25 Program 14

Pada gambar 2.24 dan 2.25 menjelaskan program tentang membaca nilai dari sensor getaran. Misalnya jika getaran tinggi dan sedang terdeteksi maka akan mengambil foto dan mengirimkannya ke telegram. Jika getaran rendah terdeteksi, hanya mencetak pesan ke serial monitor

### **2.5.6 Pengujian sistem**

Setelah perancangan hardware dan software, langkah berikutnya adalah menjalankan program dan menguji setiap untuk memastikan semua yang berfungsi sesuai yang diinginkan. Pengujian ini meliputi beberapa aspek penting, seperti mengukur respon sistem, mengevaluasi cakupan sistem, dan memastikan keseluruhan rangkaian bekerja dengan baik. Proses pengujian melibatkan pemeriksaan konektivitas Wifi, respon *bot Telegram*, dan deteksi getaran oleh sensor *SW-420*. Semua fungsi sistem, termasuk pengambilan dan pengiriman gambar, harus diuji untuk memastikan sistem beroperasi dengan baik dan sesuai harapan.

### **2.5.7 Implementasi**

Setelah melakukan pengujian selanjutnya akan dilakukan implementasi pada alat yang telah dibuat. Pada tahap ini hasil yang sudah dirancang akan diimplementasikan untuk menjadi sebuah system yang nyata. Dalam konteks implementasi system keamanan cerdas untuk mencegah pencurian motor dengan integrasi *ESP32-CAM* dan sensor getar, berikut adalah beberapa Langkah implementasi yang dilakukan:

- A. Perancangan Fisik: Memasang komponen-komponen yang sudah di siapkan.
- B. Instalasi perangkat keras: Pasang perangkat keras ke motor sesuai dengan desain yang telah ditentukan.
- C. Konfigurasi perangkat lunak: Muat program yang telah dibuat kedalam *ESP32-CAM* menggunakan Arduino ID
- D. Dengan pengujian awal: Uji setiap komponen perangkat keras untuk memastikan fungsionalitasnya. Pastikan *ESP32-CAM* dapat mengambil gambar, sensor getar dapat mendeteksi getaran.
- E. Uji coba lapangan: Uji system secara menyeluruh dilapangan untuk memastikan semua system dapat mendeteksi dan memberikan respon yang tepat.

### **2.5.8 Evaluasi dan peningkatan**

Melakukan Evaluasi sistem setelah digunakan dalam jangka waktu tertentu. Identifikasi area-area yang perlu ditingkatkan atau diperbaiki, dan melakukan perbaikan atau peningkatan sesuai kebutuhan .

### **2.5.9 Analisa Kerja**

Analisa Kerja sistem keamanan akan melibatkan pemahaman tentang bagaimana setiap komponen berintraksi satu sama lain dan bagaimana sistem secara keseluruhan bekerja untuk mencapai tujuan perlindungan terhadap pencurian.