BAB III
HASIL ANALISIS DAN PEMBAHASAN

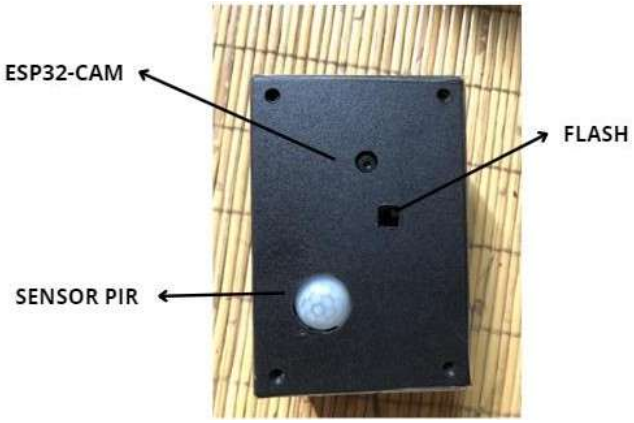
3.1 Hasil

Bab ini membahas hasil implementasi sistem yang telah dirancang pada bab sebelumnya. Implementasi sistem merupakan tahap dimana perancangan sistem diubah menjadi kenyataan melalui pengkodean, konfigurasi komponen-komponen sistem, dan pengujian sistem secara menyeluruh,Perangkat pendukung seperti perangkat keras (hardware) dan perangkat lunak (software) dibutuhkan untuk menjalankan sistem dengan baik.

Meningkatnya pencurian dan pembobolan pada sarang walet yang terletak di lokasi terpencil dan sulit dijangkau merupakan masalah utama dalam penelitian ini. Dengan menerapkan sistem keamanan berbasis Internet of Things (IoT), diharapkan sistem ini dapat memantau sarang walet secara efektif tanpa melakukan pengawasan langsung di lokasi. Implementasi ini menunjukkan bagaimana teknologi dapat membantu pemilik sarang walet mengatasi masalah keamanan.

Rangkaian sistem keamanan sarang walet dibuat menggunakan ESP32-Cam, Sensor PIR (Passive Infrared Sensor) dan Programmer Downloader ESP32-CAM Development Board. Adapun menggunakan cover yang digunakan yaitu black box dengan ukuran lebar 7 cm, Paang 10 cm dengan tinggi 3.5 cm. Berikut gambar rangkaiannya.

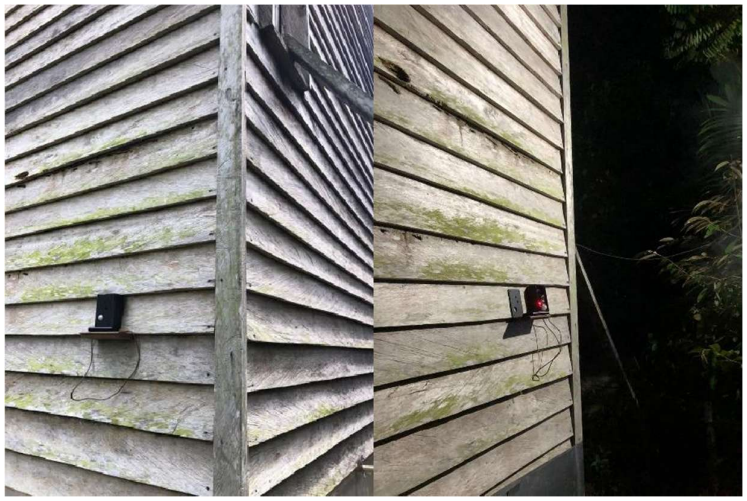
3.1.1 Implementasi Sistem



Gambar 3. 1 *Seluruh Rangkaian Menggunakan Cover Black Box*

Setelah rangkaian dibuat, selanjutkan melakukan implementasi dimana rangkaian alat yang sudah selesai diletakan di lokasi tempat penelitian. Berikut gambar alat yang telah diambil dari Lokasi penelitian.

Gambar 3. 2 *Sebelum Alat Dipasang di Lokasi Sarang Walet*



Gambar 3. 3 *Implementasi di Lokasi Sarang Walet*

3.1.2 Kode Program

Kode program untuk sistem ini ditulis menggunakan bahasa pemrograman C++ dan Arduino IDE. Kode program ini meliputi:

```
#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>
#include "FS.h"                // SD Card ESP32
#include "SD_MMC.h"            // SD Card ESP32
#include "EEPROM.h"            // read and write from flash memory

// WiFi credentials
const char* ssid = "Ginott";
const char* password = "Ginamaulidina";

// Telegram Bot Token and Chat ID
String BOTtoken = "7164150383:AAFwTyj9x54IDFq1tt1mK5d6KkoIkdT2Ntg";
String chatId = "1336135591";

bool sendPhoto = false;
bool wifiConnected = false;

WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);

#define FLASH_PIN 4
#define PIR_PIN 13

// Camera model AI-Thinker pin definitions
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
```

Gambar 3. 4 *Program 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

int botRequestDelay = 1000;   // Every 1 second check bot
long lastTimeBotRan;
bool adaGerakan = false;

#define EEPROM_SIZE 1
RTC_DATA_ATTR int bootCount = 0;
int pictureNumber = 0;

void handleNewMessages(int numNewMessages);
String sendPhotoTelegram();
void savePhotoSDCard(camera_fb_t * fb);
bool initializeCamera();
void checkWiFiAndReconnect();

static void IRAM_ATTR detectsMovement(void * arg){
  adaGerakan = true;
}
```

Gambar 3. 5 *Program 2*

```
void setup(){
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  Serial.begin(115200);

  pinMode(FLASH_PIN, OUTPUT);
  digitalWrite(FLASH_PIN, LOW);

  // Initialize EEPROM
  EEPROM.begin(EEPROM_SIZE);
  pictureNumber = EEPROM.read(0) + 1;

  // Connect to WiFi
  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Connecting to WiFi: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT);
  int retries = 30;
  while (WiFi.status() != WL_CONNECTED && retries > 0) {
    Serial.print(".");
    delay(500);
    retries--;
  }
  if (WiFi.status() == WL_CONNECTED) {
    wifiConnected = true;
    Serial.println();
    Serial.println("WiFi connected");
    Serial.print("ESP32-CAM IP Address: ");
    Serial.println(WiFi.localIP());
  } else {
    wifiConnected = false;
```

Gambar 3. 6 *Program 3*

```
    Serial.println();
    Serial.println("Failed to connect to WiFi");
  }

  // Initialize Camera
  if (!initializeCamera()) {
    Serial.println("Camera init failed");
    delay(1000);
    ESP.restart();
  }

  // Initialize SD Card
  Serial.println("Starting SD Card");
  delay(500);
  if (!SD_MMC.begin()) {
    Serial.println("SD Card Mount Failed");
  } else {
    uint8_t cardType = SD_MMC.cardType();
    if (cardType == CARD_NONE) {
      Serial.println("No SD Card attached");
    }
  }

  // PIR Motion Sensor mode INPUT_PULLUP
  esp_err_t err = gpio_isr_handler_add((gpio_num_t)PIR_PIN, &detectsMovement, (void *) PIR_PIN);
  if (err != ESP_OK) {
    Serial.printf("Handler add failed with error 0x%x \r\n", err);
  }
  err = gpio_set_intr_type((gpio_num_t)PIR_PIN, GPIO_INTR_POSEDGE);
  if (err != ESP_OK) {
    Serial.printf("Set intr type failed with error 0x%x \r\n", err);
```

Gambar 3. 7 *Program 4*

```
  }
}

void loop(){
  if (sendPhoto) {
    Serial.println("Preparing to send photo");
    sendPhotoTelegram();
    sendPhoto = false;
  }

  if (adaGerakan) {
    if (wifiConnected) {
      bot.sendMessage(chatId, "Ada Gerakan!", "");
      Serial.println("Ada Gerakan");
      sendPhotoTelegram();
    } else {
      camera_fb_t * fb = esp_camera_fb_get();
      if (!fb) {
        Serial.println("Camera capture failed");
      } else {
        savePhotoSDCard(fb);
        esp_camera_fb_return(fb);
      }
    }
    adaGerakan = false;
  }

  if (millis() > lastTimeBotRan + botRequestDelay) {
    if (wifiConnected) {
      int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
      while (numNewMessages) {
        Serial.println("Got response");
```

Gambar 3. 8 *Program 5*

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

String sendPhotoTelegram() {
  const char* myDomain = "api.telegram.org";
  String getAll = "";
  String getBody = "";

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

  Serial.println("Connecting to " + String(myDomain));
  if (clientTCP.connect(myDomain, 443)) {
    Serial.println("Connected");

    String head = "--Random\r\nContent-Disposition: form-data; name=\"chat_id\"; \r\n\r\n" + chatId + "\r\n--Random\r\
    String tail = "\r\n--Random--\r\n";
```

Gambar 3. 9 *Program 6*

```
    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=Random");
    clientTCP.println();
    clientTCP.print(head);

    uint8_t * fbBuf = fb->buf;
    size_t fbLen = fb->len;
    for (size_t n = 0; n < fbLen; 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);

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

Gambar 3. 10 *Program 7*

```
    while ((startTimer + waitTime) > millis()) {
      Serial.print(".");
      delay(100);
      while (clientTCP.available()) {
        char c = clientTCP.read();
        if (state == true) 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("Connection to api.telegram.org failed.");
  }
  return getBody;
}

void savePhotoSDCard(camera_fb_t * fb) {
  String path = "/picture" + String(pictureNumber) + ".jpg";
  fs::FS &fs = SD_MMC;
  Serial.printf("Picture file name: %s\n", path.c_str());
  File file = fs.open(path.c_str(), FILE_WRITE);
  if (!file) {
```

Gambar 3. 11 *Program 8*

```
void savePhotoSDCard(camera_fb_t * fb) {
  String path = "/picture" + String(pictureNumber) + ".jpg";
  fs::FS &fs = SD_MMC;
  Serial.printf("Picture file name: %s\n", path.c_str());
  File file = fs.open(path.c_str(), FILE_WRITE);
  if (!file) {
    Serial.println("Failed to open file in writing mode");
  } else {
    file.write(fb->buf, fb->len); // payload (image), payload length
    Serial.printf("Saved file to path: %s\n", path.c_str());
    EEPROM.write(0, pictureNumber);
    EEPROM.commit();
  }
  file.close();
  pictureNumber++;
}

void handleNewMessages(int numNewMessages) {
  Serial.print("Handle New Messages: ");
  Serial.println(numNewMessages);

  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != chatId) {
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }

    String text = bot.messages[i].text;
    Serial.println(text);

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

Gambar 3. 12 *Program 9*

```
      if (text == "/flash") {
        bool flashState = digitalRead(FLASH_PIN);
        flashState = !flashState;
        digitalWrite(FLASH_PIN, flashState ? HIGH : LOW);
        bot.sendMessage(chatId, flashState ? "Flash ON" : "Flash OFF", "");
      }
      if (text == "/foto") {
        sendPhoto = true;
        Serial.println("New photo request");
      }
      if (text == "/start") {
        String welcome = "Sistem Keamanan ESP32-CAM\n";
        welcome += "Klik Tulisan Biru:\n";
        welcome += "/foto : Mengambil Gambar\n";
        welcome += "Sistem Ini Otomatis Kirim Gambar Saat Terjadi Gerakan.\n";
        bot.sendMessage(chatId, welcome, "");
      }
    }
  }

  bool initializeCamera() {
    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;
```

Gambar 3. 13 *Program 10*

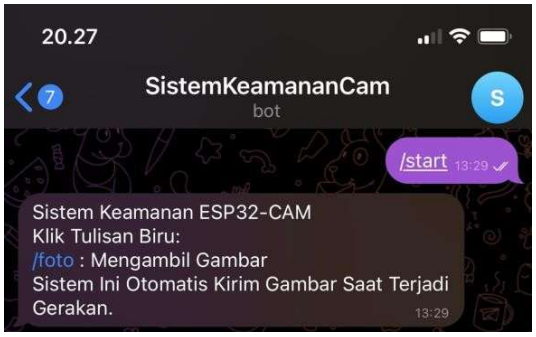
```
  sensor_t * s = esp_camera_sensor_get();
  s->set_framesize(s, FRAMESIZE_CIF);
  return true;
}

void checkWiFiAndReconnect() {
  if (WiFi.status() != WL_CONNECTED) {
    wifiConnected = false;
    WiFi.disconnect();
    WiFi.begin(ssid, password);
    int retries = 30;
    while (WiFi.status() != WL_CONNECTED && retries > 0) {
      Serial.print(".");
      delay(500);
      retries--;
    }
    if (WiFi.status() == WL_CONNECTED) {
      wifiConnected = true;
      Serial.println();
      Serial.println("WiFi reconnected");
      Serial.print("ESP32-CAM IP Address: ");
      Serial.println(WiFi.localIP());
    } else {
      wifiConnected = false;
      Serial.println();
      Serial.println("Failed to reconnect to WiFi");
    }
  } else {
    wifiConnected = true;
  }
}
```

Gambar 3. 14 *Program 11*

3.1.3 Halaman Bot Aplikasi Telegram

Pada halaman bot telegram ada menu yang bisa diakses yaitu untuk mengambil foto secara langsung yang terdapat pada menu telegram dapat di lihat pada gambar 3.15.



Gambar 3. 15 *Bot Telegram*

3.2 Pengujian Sistem

Pengujian dilakukan untuk memastikan bahwa sistem keamanan sarang walet berbasis internet of things (IoT) yang menggunakan ESP32-CAM dan Sensor PIR berfungsi dengan baik sesuai dengan yang diharapkan. Pengujian yang dilakukan meliputi:

3.2.1 Pengujian ESP32-CAM

Pengujian ini menggunakan kamera ESP32-CAM untuk pengambilan gambar Selanjutnya, foto dikirim melalui platform Telegram.

Tabel 3. 1 *Hasil Pengujian ESP32-CAM*

No	*Kondisi*	*Hasil Pengambilan Gambar*	*Keterangan*
1	*Pagi*		*Gambar berhasil diambil*

2	*Malam*		*Gambar berhasil diambil*

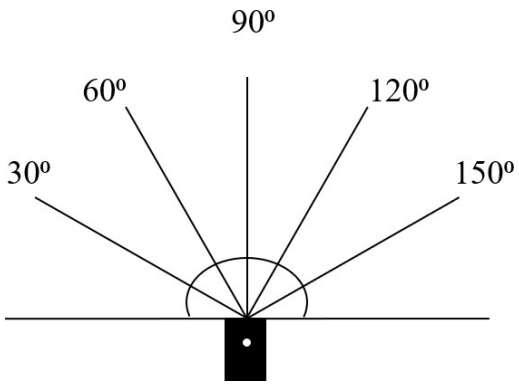
Pada tabel 3.1 hasil pengujian ESP32-CAM menunjukkan bahwa dapat mengambil gambar dengan baik pada kondisi pagi dan malam hari dan dapat dikirim ke Telegram. Pengiriman hanya membutuhkan waktu sepuluh detik jika kondisi jaringan normal, tetapi dapat memakan waktu hingga tiga menit atau lebih jika kondisi jaringan tidak stabil.

3.2.2 Pengujian Sensor PIR

Pengujian sensor PIR ini dilakukan untuk mengetahui sensitivitas sensor dalam mendeteksi gerakan serta menentukan jangkauan deteksinya. Selain itu, serta melibatkan variasi sudut untuk memahami pengaruh posisi relatif objek terhadap sensor. Sudut-sudut pengujian meliputi:

- 30°: Sedikit serong kiri dari sensor.
- 60°: Serong kiri dari sensor, lebih serong dibanding sudut 30°.
- 90°: Tepat di depan sensor.
- 120°: Serong kanan dari sensor.
- 150°: Lebih serong belakang kanan dari sensor dibanding sudut 120°.

Sudut-sudut ini mencakup posisi dari kiri ke kanan relatif terhadap sensor PIR. Berikut ini gambar visual dari pengambilan sudut dari sensor PIR.



Gambar 3. 16 *Pengambilan titik sudut yang terletak pada cover box alat*

Tabel 3. 2 *Hasil Pengujian Sensor PIR*

No	Objek	Jarak	Sudut 30°	Sudut 60°	Sudut 90°	Sudut 120°	Sudut 150°
1	Manusia	± 1-2 meter	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
2	Manusia	± 3-4 meter	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
3	Manusia	5 meter	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Tidak terdeteksi
4	Manusia	6 meter	Tidak terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Tidak terdeteksi
5	Manusia	7 meter	Tidak terdeteksi	Tidak terdeteksi	Terdeteksi	Tidak terdeteksi	Tidak terdeteksi

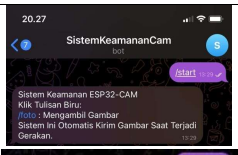
Pada tabel 3.2 hasil pengujian sensor PIR menunjukan bahwa pada jarak ± 1-2 meter dan ± 3-4 meter, sensor mampu mendeteksi gerakan manusia pada semua sudut (30°, 60°, 90°, 120°, 150°), memberikan kinerja yang optimal pada jarak pendek hingga menengah. Namun, pada jarak 5 meter, sensor mulai kehilangan kepekaan terhadap sudut (150°), dan pada jarak 6 meter, hanya sudut 60°, 90°, dan 120° yang terdeteksi dengan baik. Pada jarak 7 meter, sensor hanya dapat mendeteksi gerakan pada sudut 90° (tepat di depan sensor), sementara sudut

lainnya tidak terdeteksi. Kesimpulannya, sensor PIR efektif dalam mendeteksi gerakan hingga 4 meter untuk semua sudut, dengan penurunan kepekaan pada jarak yang lebih jauh, terutama pada sudut yang lebih besar.

3.2.3 Pengujian Menu Bot Telegram

Pengujian ini dilakukan untuk mengetahui respon ESP32-cam Ketika perintah pada menu bot telegram, apakah pesan atau perintah yang dikirim melalui Telegram direspon dengan baik.

Tabel 3. 3 *Hasil Pengujian Menu Bot Telegram*

Pesan	Hasil	Tampilan telegram
/start	Tampilan menu bot telegram	
/foto	Foto berhasil diambil dan di kirim ke telegram	
Otomatis kirim notifikasi dan gambar jika ada gerakan	Berhasil mengirim notifikasi dan gambar ke telegram	

Pada tabel 3.3 pengujian yang dilakukan pada bot Telegram menunjukkan bahwa menunya berfungsi dengan baik dan memberikan respons yang tepat. Pengguna dapat mengambil foto, menerima notifikasi, dan dengan mudah mengakses menu bot melalui aplikasi.

3.2.4 Pengujian Kartu SD

Pengujian ini dilakukan untuk memastikan gambar dapat disimpan pada kartu MicroSD ketika koneksi internet tidak tersedia.

Tabel 3. 4 *Hasil Pengujian Kartu MicroSD*

Kondisi	Keterangan
Terhubung internet	Gambar tidak tersimpan di SD
Tidak terhubung internet	Gambar berhasil disimpan di SD

Pengujian ini menunjukkan bahwa sistem beroperasi sesuai yang diharapkan. Ketika koneksi internet tersedia, prioritas pengiriman gambar ke Telegram dilakukan. Namun, saat koneksi internet tidak ada, gambar akan disimpan pada kartu microSD sebagai alternatif.

3.3 Analisis Hasil Pengujian

Analisis pengujian yang dilakukan agar mengetahui kelayakan alat yang digunakan dapat bekerja Ketika digunakan.

Tabel 3. 5 *Analisis Hasil Pengujian Sistem Keamanan*

No	Pengujian	Proses	Hasil yang diharapkan	Hasil Pengujian
1	Pengujian ESP32-CAM	Mengambil gambar menggunakan ESP32-CAM dan mengirimkan ke Telegram.	ESP32-CAM mengambil gambar baik di pagi maupun di malam hari, gambar dapat dikirim ke Telegram dalam waktu kurang dari 10 detik dalam kondisi jaringan normal.	Gambar berhasil diambil dan dikirim ke Telegram dalam waktu yang sesuai dengan kondisi jaringan (kurang dari 10 detik pada kondisi normal), dan pengiriman gambar berhasil dilakukan baik di pagi hari maupun di malam hari.
2	Pengujian Sensor PIR	Menguji sensitivitas dan jangkauan deteksi sudut pada sensor PIR	Sensor PIR mampu mendeteksi gerakan manusia pada jarak 1-5 meter dengan berbagai sudut.	Sensor PIR memiliki sensitivitas yang baik untuk mendeteksi gerakan manusia pada jarak 1-4 meter pada semua sudut, pada jarak 5 meter sensor mulai menurun sensivitas pada sudut 150 º, jarak 6 meter hanya mendeteksi gerakan pada sudut 60º, 90º, dan 120º lalu pada jarak 7 meter hanya dapat mendeteksi gerakan dengan sudut 90º.

3	Pengujian Menu Bot Telegram	Memastikan bot Telegram merespons perintah dengan benar dan mengirim notifikasi saat ada gerakan.	Menu bot Telegram berfungsi dengan baik: /start menampilkan menu, /foto mengambil, mengirim gambar ke Telegram, dan otomatis mengirim notifikasi serta gambar saat ada gerakan.	Menu bot Telegram merespons dengan benar terhadap perintah /start, /foto, dan mengirim notifikasi serta gambar sesuai saat ada gerakan terdeteksi.
4	Pengujian Kartu MicroSD	Memastikan gambar dapat disimpan di Kartu microSD saat tidak ada koneksi internet.	Gambar berhasil disimpan di Kartu microSD tidak terhubung pada internet. Tidak ada penyimpanan gambar di Kartu microSD jika terhubung dengan internet.	Sistem mengirimkan gambar ke Telegram saat terhubung ke internet tanpa menyimpannya di Kartu microSD. Jika tidak terhubung ke internet, gambar berhasil disimpan di Kartu SD.

Hasil analisis menunjukkan bahwa sistem keamanan sarang walet berbasis IoT dengan Sensor PIR dan ESP32-CAM berfungsi dengan baik untuk mengambil gambar, mendeteksi gerakan, berintegrasi dengan platform Telegram, dan menyimpan pada Kartu MicroSD.