

LAMPIRAN

Lampiran 1. Surat Keterangan Melakukan Penelitian



UMKT
Program Studi
Teknik Informatika
Fakultas Sains dan Teknologi

Telp. 0541-748511 Fax. 0541-766832
Website <http://informatika.umkt.ac.id>
email: informatika@umkt.ac.id

UNIVERSITAS MUHAMMADIYAH
Kalimantan Timur
Berkeadilan | Berkeadilan | Berkeadilan

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 056-003/KET/FST.1/A/2024
Lampiran : -
Perihal : Keterangan Melakukan Penelitian

Assalamu'alaikum Warrahmatullahi Wabarrakatuh

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Amin.

Dengan surat ini, kami menerangkan bahwa mahasiswa berikut:

No	Nama	NIM
1	Fajar Magda	1911102441134
2	Anugrah Fiansyah	1911102441155
3	Sulhijja	1911102441146
4	Difha Prayudha	1911102441030
5	Gina Maulidina	2011102441037

Melakukan penelitian dengan membuat sebuah alat IoT di Laboratorium Hardware & Networking.

Demikian hal ini disampaikan, atas kerjasamanya kami ucapkan terima kasih.

Wassalamu'alaikum Warrahmatullahi Wabarrakatuh

Samarinda, 19 Dzulhijjah 1445 H
26 Juni 2024 M

Dua Program Studi S1 Teknik Informatika



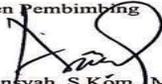
Arhansyah, S.Kom., M.TI
NIDN. 1118019203

Lampiran 2. Kartu Kendali Bimbingan Skripsi

KARTU KENDALI BIMBINGAN SKRIPSI

Nama : Gina Maulidina
 Nim : 2011102441037
 Nama Dosen Pembimbing : Arbansyah, S.Kom., M.TI
 Judul Penelitian : Sistem Keamanan Sarang Walet Berbasis Internet of Things menggunakan ESP32-CAM dan Sensor PIR (Passive Infrared Sensor)

No	Tanggal	Uraian Pembimbing	Paraf Dosen
1	20 Januari 2024	Pengajuan Judul Skripsi	
2	24 Januari 2024	Membuat Format Canva	
3	27 Febuari 2024	Pengajuan mengganti judul	
4	17 Maret 2024	Format Pembuatan ab 1 dan bab 2	
5	20 Maret 2024	Revisi proposal bab 1 dan bab perbaikan penulisan format sesuai pedoman	
6	25 Maret 2024	Revisi proposal bab 1 pada bagian pendahuluan dan tujuan masalah	
7	28 Maret 2024	Rangkaian alat dan bahan yang akan digunakan untuk pembuatan sistem	
8	18 April 2024	Revisi proposal bab 2 pada bagain perancangan sistem	
9	28 April 2024	Revisi proposal bab 1 dan bab 2 untuk penyelesaian sebelum di submit ke simpel	
10	10 Juni 2024	Membahas terakit hasil yang terdapat pada bab 3	
11	18 Juni 2024	Merakit alat, membuat skema rangakain dan program yang akan di gunakan.	
12	26 Juni 2024	Membahas terkait hasil dari alat yang sudah dirakit serta hasil dari uji coba alat	
13			
14			
15			

Dosen Pembimbing

 Arbansyah, S.Kom., M.TI
 NIDN.1118019203

Mengetahui

 Gina Maulidina, S.Kom., M.TI
 NIDN.1118019203

Lampiran 3. Lokasi Sarang Walet



Lampiran 4. Program 1 & 2

```
#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

#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;
}
```

Lampiran 5. Program 3 & 4

```
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;

    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);
  }
}
```

Lampiran 6. Program 5 & 6

```
    }
}

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");

                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\n";
        String tail = "\r\n--Random--\r\n";
    }
}
```

Lampiran 7. Program 7 & 8

```
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;

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) {
```

Lampiran 8. Program 9 & 10

```
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;

        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;
}
```

Lampiran 9. Program 11

```
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;
  }
}
```

RIWAYAT HIDUP



Nama Lengkap Penulis Gina Maulidina lahir tanggal 16 Juni 2001 di Tenggarong Seberang, Provinsi Kalimantan Timur. Agama Islam dan merupakan anak ketiga dari empat bersaudara dari pasangan bapak Eko Naryono dan ibu Isnawati. Penulis memulai Pendidikan di Sekolah Dasar Negeri 010 Tenggarong Seberang tahun 2008 lulus pada tahun 2014 melanjutkan ke sekolah Menengah Pertama Negeri 2 Bongan pada tahun 2014 dan lulus pada tahun 2017. Tahun 2017 masuk Sekolah Menengah Kejuruan Negeri 1 Samarinda dan lulus pada tahun 2020. Pendidikan tinggi dimulai pada tahun 2020 di Universitas Muhammadiyah Kalimantan Timur, Fakultas Sains dan Teknologi, Program Studi Teknik Informatika.

Pada tahun 2020 penulis resmi terdaftar sebagai mahasiswa Fakultas Sains Dan Teknologi, Jurusan Teknik Informatika Universitas Muhammadiyah Kalimantan Timur jenjang studi Starta satu (S1) kemudian pada tahun 2023 penulis telah melakukan program maagang di sebuah Perusahaan Kelapa Sawit di Kampung Muara Gusik Kecamatan Bongan Kabupaten Kutai Barat selama 3 bulan.