DAFTAR PUSTAKA

[1] S. Mindasari, M. As'ad, and D. Meilantika, "Sistem Keamanan Kotak Amal di Musala Sabilul Khasanah Berbasis Arduino UNO," *J. Tek. Inform. Mahakarya*, vol. 5, no. 2, pp. 7–13, 2022.

[2] T. Rahayu, "Kuras Kotak Amal di Masjid Sumberlawang Sragen, Residivis Jatim Dibekuk Warga," soloraya. Accessed: Jul. 13, 2024. [Online]. Available: https://soloraya.solopos.com/kuras-kotak-amal-di-masjid-sumberlawang-sragen-residivis-jatim-dibekuk-warga-1844224

[3] M. Fauza and M. A. Muthalib, "Sistem Pengaman Pintu Otomatis Menggunakan Sensor Radio Frequency Identification (Rfid) Berbasis Arduino Uno," *J. Energi Elektr.*, vol. 11, no. 1, p. 30, 2022, doi: 10.29103/jee.v11i1.8185.

[4] M. Yunus, "PROTOTIPE SISTEM KEAMANAN KAMAR KOS BERBASIS INTERNET OF THINGS MENGGUNAKAN SENSOR PASSIVE INFRARED RECEIVER DENGAN ESP32-CAM DAN TELEGRAM SEBAGAI NOTIFIKASI," 2021.

[5] ALFITH, "PERANCANGAN PROTOTYPE SISTEM PENGAMAN MENGGUNAKAN RFID YANG DIENSKRIPSI DAN TERMONITOR BERBASIS MICROCONTROLLER ARDUINO," vol. 5, no. 1, pp. 188–192, 2022.

[6] R. A. Mali, G. Tjahjono, F. F. G. Ray, and I. Fahmi, "PENGUKUR JARAK AMAN MOBIL PADA AREA TEMPAT PARKIR UMUM MENGGUNAKAN SENSOR ULTRASONIC HC-SR04 DAN ARDUINO UNO," vol. 4, pp. 1–7, 2021.

[7] M. R. -Alfariski, M. Dhandi, and A. Kiswantono, "Automatic Transfer Switch (ATS) Using Arduino Uno, IoT-Based Relay and Monitoring," *JTECS J. Sist. Telekomun. Elektron. Sist. Kontrol Power Sist. dan Komput.*, vol. 2, no. 1, p. 1, 2022, doi: 10.32503/jtecs.v2i1.2238.

[8] N. I. Qalbi *et al.*, "Rancang Bangun Kotak Amal Cerdas Sebagai Solusi Ketidak Efisienan Pendistribusi Kotak Amal Di Masjid," *J. Media Elektr.*, vol. 17, no. 2, pp. 25–32, 2020.

[9] Y. Fauzan, "Kotak Penerima Paket Berbasis IoT Menggunakan Modul Esp32-Cam," *Institutional Repos. UIN Syarif Hidayatullah Jakarta*, pp. 1–66, 2020, [Online]. Available:

https://repository.uinjkt.ac.id/dspace/handle/123456789/56069

[10] R. Widiaratih, A. A. Dwi, N. Bui, T. My, and T. Nguyen, "Internet of Things (IoT) Security Alarms on Internet of Things (IoT) Security Alarms on ESP32-CAM," 2021.

[11] A. Restu Mukti, C. Mukmin, E. Randa Kasih, D. Palembang Jalan Jenderal Ahmad Yani No, S. I. Ulu, and S. Selatan, "Perancangan Smart Home Menggunakan Konsep Internet of Things (IOT) Berbasis Microcontroller," vol. 14, no. 2, pp. 516–522, 2022.

[12] F. N. Nirmala, "IMPLEMENTASI SISTEM PENGAMAN KUNCI BRANKAS OTOMATIS DENGAN NOTIFIKASI TELEGRAM BERBASIS ARDUINO UNO," 2021.

[13] H. Khan, "Modul Sensor Tegangan DC 0 Hingga 25V." Accessed: Aug. 06, 2024. [Online]. Available: https://www.datasheethub.com/0-to-25v-dc-voltage-sensor-module/

[14] S. A. Arrahma and R. Mukhaiyar, "Pengujian Esp32-Cam Berbasis Mikrokontroler ESP32," *JTEIN J. Tek. Elektro Indones.*, vol. 4, no. 1, pp. 60–66, 2023.

[15] G. Mahendra, "Rancang Bangun Kontrol Pintu Air Dan Monitoring Ketinggian Air Sungai Berbasis Internet of Things (IoT)," vol. 2, no. 1, pp. 98–106, 2021.

LAMPIRAN A
PROGRAM ALAT

1. Program ESP32 CAM

```
#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>

const char* ssid = "apa";
const char* password = "qwertyuiop";

String BOTtoken1 =
"7447253172:AAF3qXcF7aAK9L6UpRgUsbg7ysmSp26cNco";  String
CHAT_ID1 = "1203423596";
bool sendPhoto = false;
WiFiClientSecure clientTCP;
UniversalTelegramBot bot1(BOTtoken1, clientTCP);
UniversalTelegramBot bot2(BOTtoken2, clientTCP);

#define FLASH_LED_PIN 4
bool flashState = LOW;

#define TRIG_PIN_1 13
#define ECHO_PIN_1 12
#define TRIG_PIN_2 14
#define ECHO_PIN_2 15
#define BUZZER_PIN 2

int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

unsigned long lastUltrasonicCheck = 0;
unsigned long ultrasonicCheckInterval = 5000; // Check every 5 seconds
long lastDistance1 = -1;
long lastDistance2 = -1;
```

```
const long distanceThreshold = 3;

  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot1.messages[i].chat_id);
    if (chat_id != CHAT_ID1 && chat_id != CHAT_ID2) {
      bot1.sendMessage(chat_id, "Unauthorized user", "");
      bot2.sendMessage(chat_id, "Unauthorized user", "");
      continue;    }

    String text = bot1.messages[i].text;
    Serial.println(text);

    if (text == "/start") {
      String welcome = "Welcome!\n";
      welcome += "Use the following commands to interact with the
ESP32-CAM:\n";
      welcome += "/photo : take a new photo\n";
      welcome += "/flash : toggle flash LED\n";
      welcome += "/distance1 : show distance from ultrasonic sensor 1\n";
      welcome += "/distance2 : show distance from ultrasonic sensor 2\n";
      bot1.sendMessage(chat_id, welcome, "");
      bot2.sendMessage(chat_id, welcome, "");
    }
    if (text == "/flash") {
      flashState = !flashState;
      digitalWrite(FLASH_LED_PIN, flashState);
      Serial.println("Changed flash LED state");
    }
    if (text == "/photo") {
      sendPhoto = true;
      Serial.println("New photo request");
    }
    if (text == "/distance1") {
      long distance = readUltrasonicDistance1();
```

```
      String distanceMsg = "Distance from Sensor 1: " + String(distance)
+ " cm";
      bot1.sendMessage(chat_id, distanceMsg, "");
      bot2.sendMessage(chat_id, distanceMsg, "");
      Serial.println("Distance 1 sent: " + distanceMsg);
    }
    if (text == "/distance2") {
      long distance = readUltrasonicDistance2();
      String distanceMsg = "Distance from Sensor 2: " + String(distance)
+ " cm";
      bot1.sendMessage(chat_id, distanceMsg, "");
      bot2.sendMessage(chat_id, distanceMsg, "");
      Serial.println("Distance 2 sent: " + distanceMsg);
    }

void setup() {
  Serial.begin(115200);
  pinMode(FLASH_LED_PIN, OUTPUT);
  digitalWrite(FLASH_LED_PIN, flashState);
  pinMode(TRIG_PIN_1, OUTPUT);
  pinMode(ECHO_PIN_1, INPUT);
  pinMode(TRIG_PIN_2, OUTPUT);
  pinMode(ECHO_PIN_2, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  digitalWrite(BUZZER_PIN, LOW);
  configInitCamera();

long readUltrasonicDistance1() {
  digitalWrite(TRIG_PIN_1, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN_1, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN_1, LOW);
  long duration = pulseIn(ECHO_PIN_1, HIGH);
  long distance = duration * 0.034 / 2; // Convert to distance in cm
  return distance;
}

long readUltrasonicDistance2() {
  digitalWrite(TRIG_PIN_2, LOW);
  delayMicroseconds(2);
```

```
  digitalWrite(TRIG_PIN_2, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN_2, LOW);
  long duration = pulseIn(ECHO_PIN_2, HIGH);
  long distance = duration * 0.034 / 2; // Convert to distance in cm
  return distance;
}

void loop() {
  if (sendPhoto) {
    Serial.println("Preparing photo");
    if (sendPhotoTelegram(BOTtoken1, CHAT_ID1) == "Success" &&
sendPhotoTelegram(BOTtoken2, CHAT_ID2) == "Success") {
      sendPhoto = false;
      delay(2000); }
  }
  if (millis() > lastTimeBotRan + botRequestDelay) {
    int numNewMessages = bot1.getUpdates(bot1.last_message_received
+ 1);
    while (numNewMessages) {
      Serial.println("Received new messages");
      handleNewMessages(numNewMessages);
      numNewMessages = bot1.getUpdates(bot1.last_message_received +
1);
    }
    lastTimeBotRan = millis();
  }

  // Check ultrasonic sensors
  if (millis() - lastUltrasonicCheck >= ultrasonicCheckInterval) {
    long distance1 = readUltrasonicDistance1();
    Serial.print("Distance 1: ");
    Serial.print(distance1);
    Serial.println(" cm");
    if (abs(distance1 - lastDistance1) >= distanceThreshold) {
      sendPhoto = true;
      String distanceMsg = "Distance 1 changed: " + String(distance1)
```

```
+ " cm";
     bot1.sendMessage(CHAT_ID1, distanceMsg, "");
     bot2.sendMessage(CHAT_ID2, distanceMsg, "");
     Serial.println("Distance 1 sent: " + distanceMsg);
     lastDistance1 = distance1;

     // Activate buzzer
     digitalWrite(BUZZER_PIN, HIGH);
     delay(5000);  // Buzzer sound duration
     digitalWrite(BUZZER_PIN, LOW);
    }

   long distance2 = readUltrasonicDistance2();
   Serial.print("Distance 2: ");
   Serial.print(distance2);
   Serial.println(" cm");
   if (distance2 < 20) {
     sendPhoto = true;
     String distanceMsg = "Distance 2 detected less than 20 cm: " +
String(distance2) + " cm";
     bot1.sendMessage(CHAT_ID1, distanceMsg, "");
     bot2.sendMessage(CHAT_ID2, distanceMsg, "");
     Serial.println("Distance 2 sent: " + distanceMsg);
   }
   lastUltrasonicCheck = millis();  }
}
```

2. Program Arduino Uno

```
#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

#define SS_PIN 10
#define RST_PIN 9
#define LED_G 4 //define green LED pin
#define LED_R 5 //define red LED
#define RELAY 7 //relay pin
#define BUZZER 2 //buzzer pin
#define ACCESS_DELAY 2000
#define DENIED_DELAY 1000
```

```
MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.

void setup()
{
  lcd.init();
  Serial.begin(9600);   // Initiate a serial communication
  SPI.begin();          // Initiate  SPI bus
  mfrc522.PCD_Init();   // Initiate MFRC522
  pinMode(LED_G, OUTPUT);
  pinMode(LED_R, OUTPUT);
  pinMode(RELAY, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  noTone(BUZZER);
  digitalWrite(RELAY, HIGH);
  Serial.println("Put your card to the reader...");
  Serial.println();
  lcd.backlight();
  lcd.setCursor(1, 0);
  lcd.print("TAP KARTU ANDA");
}

void loop()
{
   // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  //Show UID on serial monitor
  Serial.print("UID tag :");
  String content = "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
     Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
     Serial.print(mfrc522.uid.uidByte[i], HEX);
```

```
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  Serial.println();
  Serial.print("Message : ");
  content.toUpperCase();

  if (content.substring(1) == "84 15 B3 72" || content.substring(2) == "05
89 95 F5 5B D1 00")
  {
    Serial.println("Authorized access");
    Serial.println();
    lcd.backlight();
    lcd.setCursor(1, 1);
    lcd.print("AKSES DITERIMA");

    digitalWrite(RELAY, LOW);
    digitalWrite(LED_G, HIGH);
    delay(ACCESS_DELAY);
    digitalWrite(RELAY, HIGH);
    digitalWrite(LED_G, LOW);
  }
  else
  {
    Serial.println("Access denied");
    lcd.backlight();
    lcd.setCursor(1, 1);
    lcd.print("AKSES DITOLAK");

    digitalWrite(LED_R, HIGH);
    tone(BUZZER, 300);
    delay(DENIED_DELAY);
    digitalWrite(LED_R, LOW);
    noTone(BUZZER);  }
}
```

3. Program ESP32

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
```

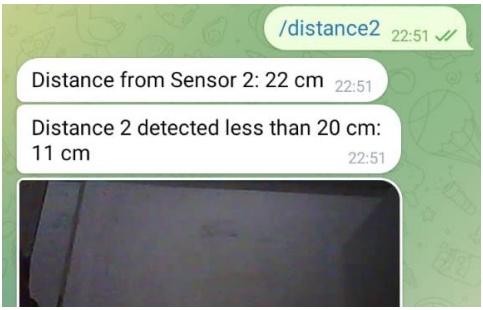
```
#include <ArduinoJson.h>
const char* ssid = "apa";
const char* password = "qwertyuiop";
#define BOTtoken1
"7163693843:AAFjbMm2idR4MSjtOiX3m4M42jinEuEx7F0"
#define CHAT_ID1 "6065623983"

WiFiClientSecure clientTCP;
UniversalTelegramBot bot1(BOTtoken1, clientTCP);
#define RELAY1_PIN 27
#define RELAY2_PIN 26
#define VOLTAGE_SENSOR_PIN 14

bool lastVoltageStatus = HIGH;
void setup() {
  Serial.begin(115200);
  pinMode(RELAY1_PIN, OUTPUT);
  pinMode(RELAY2_PIN, OUTPUT);
  pinMode(VOLTAGE_SENSOR_PIN, INPUT);
  digitalWrite(RELAY1_PIN, HIGH); // Turn off relay 1
  digitalWrite(RELAY2_PIN, HIGH); // Turn off relay 2

  // Serial2 for communication with Arduino (RX=14, TX=15)
  Serial2.begin(9600, SERIAL_8N1, 32, 33);
}

void loop() {
  int voltageStatus = digitalRead(VOLTAGE_SENSOR_PIN);

  if (voltageStatus == HIGH) {
    digitalWrite(RELAY1_PIN, LOW);  // Activate relay 1 (Power
supply)
    digitalWrite(RELAY2_PIN, HIGH); // Deactivate relay 2 (Battery)
    Serial.println("Power supply Connected");

    // Reset lastVoltageStatus
    lastVoltageStatus = HIGH;
  } else if (voltageStatus == LOW) {
    digitalWrite(RELAY1_PIN, HIGH); // Deactivate relay 1 (Power
supply)
    digitalWrite(RELAY2_PIN, LOW);  // Activate relay 2 (Battery)
```

```
    Serial.println("Power supply Disconnected");

    // Send message to Telegram if voltage status changes to LOW
    if (lastVoltageStatus == HIGH) {
      String message = "Warning: Power supply Disconnected, switching
to Battery!";
      bot1.sendMessage(CHAT_ID1, message, "");
      lastVoltageStatus = LOW; }
  }
  if (Serial2.available()) {
    String message = Serial2.readStringUntil('\n');
    Serial.println("Received: " + message);

    if (message.startsWith("AUTHORIZED")) {
      String uid = message.substring(message.indexOf(',') + 1);
      String telegramMessage = "User with UID " + uid + " has
accessed.";
      bot2.sendMessage(CHAT_ID1, telegramMessage, ""); }  }

  delay(1000); // Wait for 1 second
}
```

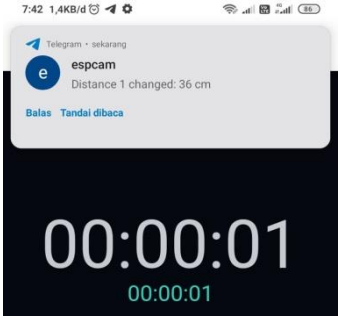
LAMPIRAN B
DOKUMENTASI

1. Pengujian Sensor Ultrasonik 1

2. Pengujian Sensor Ultrasonik 2



3. Pengujian Delay Pengiriman Gambar (sample ke 5)



BIODATA PENULIS

Nama : Tri Rahmawati
Kelas : Teknik Elektronika B (Angkatan 2021)
E-mail : rahmawatitri003@gmail.com
No. telp : +6285736871779

Riwayat Pendidikan

1. SD Negeri 1 Bajing Kulon Tahun 2009-2015
2. SMP Negeri 2 Kroya Tahun 2015-2018
3. SMA Negeri 1 Kroya Tahun 2018-2021
4. Politeknik Negeri Cilacap Tahun 2021-2024

Penulis telah mengikuti seminar Tugas Akhir pada tanggal 29 juli 2024 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md).