

DAFTAR PUSTAKA

- Bin Amir, M. A. (2013). IP Camera dan Aplikasinya. Elex Media Komputindo.
- Belift, P. (n.d.). Mengenal Cara Kerja Elevator, Jenis-Jenis, dan Fungsinya. Retrieved from <https://belift.co.id/artikel/mengenal-cara-kerja-elevator-jenis-jenis-dan-fungsinya>
- Edwards, B. C., & Westling, E. A. (2003). The Space Elevator. BC Edwards.
- Frans. (n.d.). Apa Itu Nodemcu: Pengertian, Sejarah, dan Versinya. Retrieved from <https://www.anakteknik.co.id/rahasia1/articles/apa-itu-nodemcu-pengertian-sejarah-dan-versinya>
- Kafila, R. I., & Rahajoeningroem, T. (2022). Rancang Bangun Sistem Peringatan Batas Penumpang pada Lift Menggunakan Buzzer dan Motor Servo Berbasis Pengolahan Citra di Masjid Al-Karomah. *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali, dan Elektronika Terapan*, 10(1), 38-46.
- Kho, Dickson. (2020). Teknik Elektronika: Pengertian Relay dan Fungsinya. *Ejurnal Teknik ITB*.
- Mobilenet-ssd. (n.d.). Retrieved from https://docs.opencv.ai/2023.3/omz_models_model_mobilenet_ssd.html
- Nadiansyah, R. (2018). Sistem Pengendali Kipas Angin Berbasis NodeMCU ESP8266 (Tesis Doktoral, STMIK AKAKOM Yogyakarta).
- Nugroho, D., & Raharja, B. D. (2020). Modul Komputer dan Masyarakat. Pedomannya Penyusunan Skripsi, Fakultas Teknik, Universitas PGRI Adi Buana Surabaya, 2020.
- Pratama, G. N. I. P., & Sumarjo, H. (2018). Aksesibilitas Tata Letak Elevator Penumpang Gedung Kantor Pusat Layanan Terpadu (KPLT) Fakultas Teknik UNY. *INERSIA: Informasi dan Ekspose Hasil Riset Teknik Sipil dan Arsitektur*, 14(1), 26-35.
- Rahul, A. E. (2022). Face Detection and Recognition System Using Raspberry Pi. *International Journal of Research in Engineering, Science, and Management*, 1(12).
- Richardson, M., & Wallace, S. (2012). Getting Started with Raspberry Pi. O'Reilly Media, Inc.
- Yudhistira. (2024). √ Fungsi Router WiFi Serta Cara Kerja dan Jenisnya. Retrieved from <https://www.bhinneka.com/blog/fungsi-router/>

DAFTAR LAMPIRAN

Program Esp8266

```
#include <ESP8266wifi.h>
#include <ESPAsyncWebServer.h>
#include <Hash.h>
#include <WebSocketsServer.h>

const char* ssid = "UNIPA_AR";
const char* password = "UNIPA_AR";
int valWeb, valRaspi;
AsyncWebServer server(80);
WebSocketsServer webSocket = WebSocketsServer(81);
#include <MD_Parola.h>
#include <MD_MAX72xx.h>
#include <SPI.h>

// Uncomment according to your hardware type
#define HARDWARE_TYPE MD_MAX72XX::FC16_HW
// #define HARDWARE_TYPE MD_MAX72XX::GENERIC_HW

// Defining size, and output pins
#define MAX_DEVICES 4
#define CS_PIN 15

// Create a new instance of the MD_Parola class with hardware SPI connection
MD_Parola myDisplay = MD_Parola(HARDWARE_TYPE, CS_PIN,
MAX_DEVICES);
#define buzzer D3
uint8_t scrollSpeed = 60; // default frame delay value
textEffect_t scrollEffect = PA_SCROLL_LEFT;

textPosition_t scrollAlign = PA_LEFT;
uint16_t scrollPause = 10;
#define BUF_SIZE 175
char curMessage[BUF_SIZE] = { "" };
char c[100];
// HTML content stored in PROGMEM
const char html[] PROGMEM = R"rawliteral(
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>NodeMCU WebSocket</title>
  <style>
  body {
    display: flex;

```

```

justify-content: center;
align-items: center;
height: 100vh; /* Make sure the body takes up the full height of the viewport
*/
margin: 0; /* Remove default margins */
color: gold;
background-color: black; /* Optional: Background color */
}

.content {
text-align: center; /* Optional: Center text within the content */
}
.judul {
color: gold;
font-size: 4em;
font-weight: 600;
/* margin-top: -10%; */
}
.judul2 {
/* margin-top: 10%; */
font-size: 2em;
}
.value1 {
margin-top: 0%;
font-size: 120px;
}
.container {
display: flex;
gap: 100px;
}
.send {
margin-top: -5%;
}
.inputan {
background-color: black;
color: gold;
border: 2px solid gold;
border-radius: 3px;
padding: 10px;
font-size: 16px;
}
.tombol {
background-color: gold;
color: black;
border: 2px solid black;
border-radius: 5px;
padding: 13px;
font-size: 16px;
}

```

```

        cursor: pointer;
    }
</style>
</head>
<body>
    <div class="content">
        <h1 class="judul">Website People Counter</h1>
        <div class="container">
            <div class="wadah">
                <p class="judul2">Hasil pengukuran dari Raspbery :</p>
                <p class="value1" id="data">0</p>
            </div>
            <div class="wadah">
                <p class="judul2">Hasil pengukuran dari Web :</p>
                <p class="value1" id="data2">0</p>
            </div>
        </div>
        <div class="send">
            <input type="text" class="inputan" id="messageInput"
placeholder="Masukan angka"/>
            <button class="tombol" id="sendButton">Kirim</button>
        </div>
    </div>
    <script>
        const gateway = `ws://${window.location.hostname}:81`;
        let websocket;

        window.addEventListener('load', onLoad);

        function onLoad(event) {
            initWebSocket();
            document.getElementById('sendButton').addEventListener('click',
sendMessage);
        }

        function initWebSocket() {
            console.log('Trying to open a WebSocket connection...');
            websocket = new WebSocket(gateway);
            websocket.onopen = onOpen;
            websocket.onclose = onClose;
            websocket.onmessage = onMessage;
        }

        function onOpen(event) {
            console.log('Connection opened');
        }

        function onClose(event) {

```

```

        console.log('Connection closed');
        setTimeout(initWebSocket, 2000);
    }

    function onMessage(event) {
        console.log('Message received');
        console.log(event.data);
        let split = event.data.split(",");

        document.getElementById('data').innerHTML = split[1];
        document.getElementById('data2').innerHTML = split[0];

    }

    function sendMessage() {
        const message = "0,"+document.getElementById('messageInput').value;
        websocket.send(message);
    }
</script>
</body>
</html>
)rawliteral";

void setup() {
    Serial.begin(115200);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Connecting to WiFi...");
    }
    pinMode(buzzer,OUTPUT);
    Serial.println("Connected to WiFi");
    Serial.println("success!");
    Serial.print("IP Address is: ");
    Serial.println(WiFi.localIP());
    // Serve HTML file
    server.on("/", HTTP_GET, [(AsyncWebServerRequest *request){
        request->send_P(200, "text/html", html);
    }]);
    myDisplay.begin();

    // Set the intensity (brightness) of the display (0-15)
    myDisplay.setIntensity(10);

    // Clear the display
    myDisplay.displayClear();
    // Start server
    server.begin();

```

```

// WebSocket
webSocket.begin();
webSocket.onEvent(webSocketEvent);
}

void loop() {
  webSocket.loop();
  if(valRaspi>valWeb){

    String lcd = "OVERLOAD ORG:" + String(valRaspi);
    lcd.toCharArray(c, lcd.length() + 1);
    if (myDisplay.displayAnimate())
    {
      strcpy(curMessage, c);
      myDisplay.displayText(curMessage, scrollAlign, scrollSpeed, scrollPause,
scrollEffect, scrollEffect);
      myDisplay.displayReset();
    }
    digitalWrite(buzzer,HIGH);
  }else{
    String lcd = "Org:" + String(valRaspi);
    myDisplay.setTextAlignment(PA_CENTER);
    myDisplay.print(lcd);
    digitalWrite(buzzer,LOW);
  }
}

void webSocketEvent(uint8_t num, WStype_t type, uint8_t * payload, size_t
length) {
  switch (type) {
    case WStype_DISCONNECTED:
      Serial.printf("[%u] Disconnected!\n", num);
      break;
    case WStype_CONNECTED:
      {
        IPAddress ip = webSocket.remoteIP(num);
        Serial.printf("[%u] Connection from ", num);
        Serial.println(ip.toString());
      }
      break;
    case WStype_TEXT:
      Serial.printf("[%u] Text: %s\n", num, payload);
      String payloadString = String((char*)payload);
      int delimiterIndex = payloadString.indexOf(',');
      String value1 = payloadString.substring(0, delimiterIndex);
      String value2 = payloadString.substring(delimiterIndex + 1);
      if(value1.toInt()==0){

```

```
    valWeb = value2.toInt();
  }else if(value1.toInt()==1){
    valRaspi = value2.toInt();
  }

  // websocket.broadcastTXT(payloadString, num);

  String msg = String(valWeb)+"," +String(valRaspi);

  Serial.println(msg);
  websocket.broadcastTXT(msg.c_str(),msg.length());

  break;
}
```

Program Main Raspberry PI menggunakan MobilenetSSD

```
from imutils.video import VideoStream
from imutils.video import FPS
import numpy as np
import imutils
import time
import cv2
import asyncio
import websockets

# Define the classes and colors
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
           "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
           "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
           "sofa", "train", "tvmonitor"]

COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

# Load the pre-trained model
print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe("MobileNetSSD_deploy.prototxt.txt",
                              "MobileNetSSD_deploy.caffemodel")

# Start the video stream
print("[INFO] starting video stream...")
vs = VideoStream(src=1).start()
time.sleep(2.0)
fps = FPS().start()

# Define the WebSocket URI
uri = "ws://192.168.100.112:81"

async def send_message(message):
    async with websockets.connect(uri) as websocket:
        await websocket.send(message)
        print(f"Sent: {message}")

# Main loop
async def main():
    while True:
        a = 0
        frame = vs.read()
        frame = imutils.resize(frame, width=400)

        (h, w) = frame.shape[:2]
        blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)),
                                     0.007843, (300, 300), 127.5)
```



```

net.setInput(blob)
detections = net.forward()

for i in np.arange(0, detections.shape[2]):
    confidence = detections[0, 0, i, 2]
    if confidence > 0.8:
        idx = int(detections[0, 0, i, 1])
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")
        label = "{}: {:.2f}%".format(CLASSES[idx], confidence * 100)

        if CLASSES[idx] == 'person':
            cv2.rectangle(frame, (startX, startY), (endX, endY), COLORS[idx],
2)
                y = startY - 15 if startY - 15 > 15 else startY + 15
                cv2.putText(frame, label, (startX, y),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)
                a += 1

print(a)

# Send the count of detected persons to the WebSocket server

frame_up = cv2.pyrUp(frame)
cv2.imshow("Frame", frame_up)
message = "1," +str(a)
await send_message(message)
key = cv2.waitKey(1) & 0xFF

if key == ord("q"):
    break

fps.update()

fps.stop()
print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))
cv2.destroyAllWindows()
vs.stop()

# Run the main loop
asyncio.get_event_loop().run_until_complete(main())

```

Foto Alat



Standar Operasional Prosedur (SOP)

1. Mengaktifkan Alat

Langkah pertama adalah mengaktifkan alat dengan memberikan tegangan sebagai sumber DC untuk menyalakan semua komponen yang terdapat pada alat tersebut. Pastikan semua koneksi terhubung dengan benar dan tegangan yang diberikan sesuai dengan spesifikasi alat.

2. Menghubungkan ke WiFi

Aktifkan hotspot WiFi pada perangkat Android sebagai sumber sinyal koneksi dengan alat. Setelah hotspot aktif, buka browser di perangkat yang akan digunakan untuk mengakses alat dan ketik alamat IP 192.168.0.24.

- **ID:** Adin_R
- **Password:** Adin_Rohman

3. Mengatur Jumlah Penumpang

Setelah berhasil terhubung, langkah berikutnya adalah melakukan pengaturan jumlah penumpang sesuai dengan yang telah ditentukan. Pastikan jumlah penumpang diatur dengan benar untuk mendapatkan hasil yang akurat.

4. Melihat Hasil Visual

Untuk melihat hasil visual dari alat, download dan install aplikasi Moboxtrim pada perangkat yang akan digunakan. Setelah aplikasi terinstall, buka dan masukkan alamat IP 192.168.0.225 untuk login ke Raspberry Pi.

- **ID:** pi
- **Password:** Adin@123



UNIVERSITAS PGRI ADI BUANA SURABAYA

FAKULTAS TEKNIK

Program Studi : Teknik Lingkungan – Perencanaan Wilayah Kota
Teknik Industri – Teknik Elektro - PVKK

KAMPUS II: Jl. Dukuh Menanggal XII/4 ☎ (031) 8281181 Surabaya 60234 Website
: www.ft.unipasby.ac.id E-mail : ft@unipasby.ac.id

FORM REVISI SKRIPSI

Nama Mahasiswa : ADIN ROHMAN
 NIM : 203600018
 Fakultas / ProgdI : TEHNIK ELEKTRO
 Judul Skripsi : PEMBATAS KAPASITAS PENUMPANG PADA ELEVATOR MENGGUNAKAN CAMERA SEBAGAI SENSOR COUNTING BERBASIS RASPBERRY PI

Ujian Tanggal : 2 Juli 2024

| No Bab. | Tanggal | Materi Konsultasi | Keterangan Catatan | Tanda Tangan Penguji |
|---------|---------|-------------------|--------------------|----------------------|
| I | 4-7-24 | Template | acc | |
| II | 6-7-24 | cover | acc | |
| III | 8-7-24 | Abstrak | Acc | |
| IV | 11-7-24 | Daftar pustaka | ACC | |
| V | 15-7-24 | SOP. Skripsi | acc | |
| | | | | |

Disetujui Dosen Penguji

Pada Tanggal, 2 Juli 2024

Penguji I,

(Atmiasri, S.T., M.T.)

Penguji II,

(Rasyida Shabihah 2.A, S.T., M.T.)

1. a. Penyelesaian Revisi paling lambat 2 minggu dari pelaksanaan Ujian Skripsi.
 b. Pengetikan, penjilidan, penandatanganan Skripsi dan mengumpulkan Skripsi paling lambat 2 minggu dari revisi.
2. Apabila sampai batas waktu tersebut (point 1, a dan b) mahasiswa belum menyelesaikan revisi dan tanda tangan, maka Ujian dinyatakan Gugur.
3. a. Foto copy Form Revisi diserahkan ke Program Studi.
 b. Skripsi yang sudah direvisi diserahkan ke Fakultas tiga eksemplar untuk dijilid.




UNIVERSITAS PGRI ADI BUANA SURABAYA FAKULTAS TEKNIK

Program Studi : Teknik Lingkungan – Perencanaan Wilayah Kota
Teknik Industri – Teknik Elektro - PVKK

KAMPUS II: Jl. Dukuh Menanggal XII/4 ☎ (031) 8281181 Surabaya 60234
Website : www.ft.unipasby.ac.id E-mail : ft@unipasby.ac.id

BERITA ACARA BIMBINGAN SKRIPSI

Form Skripsi-03

| | | | | |
|---|---|--|---|----------|
| Nama | : ADIN ROHMAN | |  | |
| NIM | : 203600018 | | | |
| Program Studi | : Teknik Elektro | | | |
| Pembimbing | : <i>Akhmad Solikin, ST., M.Tom</i> | | | |
| Periode Bimbingan | : Genap / Genap*) Tahun 20 <i>23</i> / 20 <i>24</i> | | | |
| Judul Skripsi | PEMBATAS KAPASITAS PENUMPANG PADA ELEVATOR MENGGUNAKAN CAMERA SEBAGAI SENSOR COUNTING BERBASIS RASPBERRY PI | | | |
| KEGIATAN KONSULTASI / BIMBINGAN | | | | |
| No | Tanggal | Materi pembimbingan | Keterangan | Paraf |
| 1 | <i>18-4-2024</i> | <i>Revisi latar belakang</i> | <i>Acc</i> | <i>h</i> |
| 2 | <i>23-4-2024</i> | <i>Revisi Rumusan masalah dan tujuan</i> | <i>Acc</i> | <i>h</i> |
| 3 | <i>30-4-2024</i> | <i>Revisi kajian pustaka</i> | <i>Acc</i> | <i>h</i> |
| 4 | <i>7-5-2024</i> | <i>Revisi Desain Alat</i> | <i>Acc</i> | <i>h</i> |
| 5 | <i>14-5-2024</i> | <i>Revisi Daftar pustaka</i> | <i>Acc</i> | <i>h</i> |
| 6 | <i>29-5-2024</i> | <i>Revisi teknik pengambilan data</i> | <i>Acc</i> | <i>h</i> |
| 7 | <i>4-6-2024</i> | <i>Revisi Teknik analisa dan tabel-tabel</i> | <i>Acc</i> | <i>h</i> |
| 8 | <i>15-6-2024</i> | <i>Revisi kesimpulan dan saran.</i> | <i>Acc</i> | <i>h</i> |
| Dinyatakan selesai tanggal <i>20</i> <i>juni</i> 20 <i>24</i> | | | | |

Surabaya, *20-6-2024*
Mahasiswa,

Mengetahui,
Ketua Program Studi,

Pembimbing,

[Signature]
.....*Sigit Rochman, S.T., M.Si*.....

[Signature]
.....*Akhmad Solikin, ST., M.Tom.*.....

[Signature]
.....*ADIN K*.....