# DAFTAR PUSTAKA


A. Prasetyo and R. Susanti, "Sistem Informasi Penjualan Berbasis Web Pada PT. Cahaya Sejahtera Sentosa Blitar," J. Ilm. Teknol. Inf. Asia, vol. 10, no. 2, pp. 1–16, 2016, [Online]. Available: https://jurnal.stmikasia.ac.id/index.php/jitika/article/view/57/45.

Abdullah, M. Z., Astiningrum, M., Ariyanto, Y., Puspitasari, D., & Asri, A. N. (2020). Rancang Bangun Sistem Informasi Akuntansi Berbasis Website menggunakan Framework Laravel. SITEKIN: Jurnal Sains, Teknologi dan Industri, 18(1), 49-56.

Abuarqoub, A., Abusaimeh, H., Hammoudeh, M., Uliyan, D., Abu-Hashem, M. A., Murad, S., ... & Al-Fayez, F. 2017. A survey on internet of things enabled smart campus applications. *In Proceedings of the International Conference on Future Networks and Distributed Systems* (pp. 1-7).

Adhitama, M. R., & Putra, I. G. E. W. (2019). Analisis Pembacaan dan Penulisan Data Buku Menggunakan RFID Mifare RC-522 untuk Perpustakaan. *Jurnal Sistem dan Informatika (JSI)*, *14*(1), 23–34

Alfina, R., & Yusran, A. M. (2023). Rancang Bangun Sistem Keamanan Parkir Kendaraan Roda Dua dengan Teknologi Face Recognition (Doctoral dissertation, Politeknik Negeri ujung Pandang).

Areni, I. S., Palantei, E., Suyuti, A., Baharuddin, M., Samman, F. A., & Umraeni, A. E. (2019). Pengembangan dan Implementasi Smart Campus Berbasis Smart Card Di Institut Agama Islam Negeri Bone. *JURNAL TEPAT: Teknologi Terapan untuk Pengabdian Masyarakat*, *2*(1), 1–7.

Arifin, A., Dengen, N., Setyadi, H. J., Prafanto, A., & Putra, G. M. (2019, September). Analisis Penerapan Metode Search Engine Optimization (SEO) Untuk Meningkatkan Traffic Website Berbayar dan Tidak Berbayar. In *Prosiding Seminar Nasional Ilmu Komputer dan Teknologi Informasi* (Vol. 4, No. 2).

Clinton, R. M., & Sengkey, R. (2019). Purwarupa Sistem Daftar Pelanggaran Lalulintas Berbasis Mini-Komputer Raspberry Pi. *Jurnal Teknik Elektro dan Komputer*, *8*(3), 181-192.

Cordiaz, Muhammad. 2017. PENERAPAN SMART CAMPUS SEBAGAI PENDUKUNG KEGIATAN PENDIDIKAN DALAM TRI DHARMA PERGURUAN TINGGI. Teknik Informatika, Universitas Pamulang

Effendi, S., Heriansyah, H., Haryansyah, H., & Pamungkas, M. S. (2017). Rekayasa Sistem Parkir Berlangganan Berbasis RFID (Radio Frequency Identification). Journal of Applied Microcontroller and Autonomous System,3(1), 27-37

Hafizah, V. P. N., & Periyadi, P. (2020). Perancangan Aplikasi Mobile Smart Identification Berbasis Android. eProceedings of Applied Science, 6(2).

Herlambang, M. Linto. Menjadi Master Konfigurasi Backbone. Koneksi Internet tuk Warnet, Website dan ISP. Penerbit CV. Andi Offset. Yogyakarta. )9.

(2020). "Analisis Akurasi Pengenalan Wajah Menggunakan Algoritma iola Jones Dan Modified Self Organizing Map". Tesis. Universitas

SumateraUtara.
Iqbal, Muhammad. 2018. Pemberdayaan Smart Campus dengan menggunakan Sistem IBMS (Intelligent Building Management System). Telkom University
Islamiyah A. N. 2023. Interkoneksi dan Interoperabilitas Perangkat Reader Hybrid Cerdas Menggunakan Website. Skripsi. Makassar: Universitas Hasanuddin
Iswanto, R. (2020). Pemanfaatan kertas daur ulang dalam dunia percetakan dan desain grafis. Seminar Nasional Envisi, 98–105.
Jadid, A., Zulhelmi, Z., & Ardiansyah, A. (2017). Rancang Bangun Sistem Absensi Perkuliahan Auto ID Berbasis RFID yang Terintegrasi dengan Database Berbasis WEB. *Jurnal Komputer, Informasi Teknologi, dan Elektro*, *2*(2).
Jaini, N., Asri, E., & Nova, F. (2021). Sistem Manajemen Kehadiran Menggunakan Metode Face Recognition Berbasis Web. JITSI : Jurnal Ilmiah Teknologi Sistem Informasi, 2 (2), 48 - 55. https://doi.org/10.30630/jitsi.2.2.39
Khatimah, K. 2022. Sistem Presensi Berbasis *Smart Hybrid Reader*. Skripsi. Makassar: Program Studi Teknik Elektro Universitas Hasanuddin.
Kinkar, S., Hennessy, M. & Ray, S. 2016. An ontology and integration framework for smart communities. *Journal of Computing and Information Science in Engineering*, 16(1), 011003. doi:10.1115/1.4032218.
Kurniawan, Akuwan, A. S., & Ramadijanti, N. (2014). Aplikasi Absensi Kuliah Berbasis Identifikasi Wajah Menggunakan Metode Gabor Wavelet. Jurnal ICT, Face Regocnition, 6.
Kurniawan, Tedi & Ismail, Ari & Sumarsono, Fikri & Nn, Nuryana. 2019. Perancangan Website Data Karyawan Dengan Menggunakan PHP dan MySQL.
Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," Bus. Horiz., vol. 58, no. 4, pp. 431–440, 2015
Lestari, R. (2023). *Penerapan Algoritma Fuzzy Logic pada Sistem Pengaman Pintu Menggunakan E-KTP Berbasis Arduino Uno R3* (Doctoral dissertation, Fakultas Teknik Universitas Islam Sumatera Utara).
Madakam, S., Lake, V., Lake, V. and Lake, V., 2015. Internet of Things (IoT): A literature review. Journal of Computer and Communications, 3(05), p.164.
Manullang, A. B. P., Saragih, Y., & Hidayat, R. (2021). Implementasi Nodemcu Esp8266 Dalam Rancang Bangun Sistem Keamanan Sepeda Motor Berbasis Iot. *Jurnal Informatika dan Rekayasa Elektronik*, *4*(2), 163-170.
Martunus, F. (2020). IMPLEMENTASI FACE RECOGNITION DENGAN OPENCV "SMART CCTV" UNTUK KEAMANAN BERANGKAS BERBASIS IOT.Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. 2015. Pengujian aplikasi menggunakan black box testing boundary value analysis (studi kasus: Aplikasiprediksi kelulusan SMNPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3).
O. K. Sulaiman and A. Widarma, "Sistem Internet of Things (IoT) Berbasis *Cloud Computing dalam Campus Area Network," ReseachGate*, no. April, pp.9–12, 2017.
A. Y., & Romadlon, P. (2019). Perancangan Sistem Informasi Penjualan erumahan Mengunakan Metode Sdlc Pada Pt. Mandiri Land Prosperous erbasis Mobile. *Jurnal Sigma*, *10*(2), 153-167.

Purnomo Fitrianto. (2007), "Komputerisasi Presensi Siswa Berbasis SMS dan Barcode di SMA Negeri 3 Purworejo" Skripsi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sebelas Maret Surakarta

Putra, I. S., Ferdinandus, F. X., & Bayu, M. (2019). Sistem Pendukung Keputusan Pemilihan Paket Pernikahan Dengan Metode Saw Berbasis Web. *CAHAYAtech*, *8*(2), 136-149.

R. H. Suki, Nurussa'adah, and A. Zainuri, "IMPLEMENTASI RFID SEBAGAI PENGAMAN PADA SEPEDA MOTOR UNTUK MENGURANGI TINDAK PENCURIAN," J. Mhs. TEUB, vol. 1, no. 7, pp. 1–5, 2014.

S. Mulyono, M. Qomaruddin, and M. S. Anwar, "Penggunaan Node-RED pada Sistem Monitoring dan Kontrol Green House berbasis Protokol MQTT," J. Transistor Elektro dan Inform. (TRANSISTOR EI, vol. 3, no. 1, pp. 31– 44, 2018.

Sabran, Purnamawati, and Nasruddin. 2020. "Penerapan Smart Lab Automation Berbasis IoT Pada Laboratorium Digital Jurusan Pendidikan Teknik Elektronika FT-UNM." *Elektronika Telekomunikasi & Computer* 15.2.

Salendah, J., Kalele, P., Tulenan, A., & Joshua, J. S. R. (2022, September). Penentuan Beasiswa Dengan Metode Fuzzy Tsukamoto Berbasis Web Scholarship Determination Using Web Based Fuzzy Tsukamoto Method. In *Seminar Nasional Ilmu Komputer (SNASIKOM)* (Vol. 2, No. 1, pp. 81-90).

Saluky. 2016. Pengembangan Bahan Ajar Matematika Berbasis Web Dengan Menggunakan Wordpress. EduMa. Vol. 5. No. 1. Hal. 5

Santoso, S., & Nurmalina, R. (2017). Perencanaan dan Pengembangan Aplikasi Absensi Mahasiswa Menggunakan Smart Card Guna 85 Pengembangan Kampus Cerdas (Studi Kasus Politeknik Negeri Tanah Laut). Jurnal Integrasi, 9(1), 84–91.

Sari, A. O., & Abdilah, A. (2015). Buku Web Programming I. Universitas Bina Sarana Informatika

Setiawan, E. B., & Kurniawan, B. (2015). Perancangan Sistem Absensi Kehadiran Perkuliahan dengan Menggunakan Radio Frequency Identification (RFId). *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi*, *1*(2), 44-49.

Shull, H. (1977). The overhead headache. Science, 195(4279), 639.

Solichin, A. (2016). *Pemrograman web dengan PHP dan MySQL*. Penerbit Budi Luhur.

Srivastava, L. 2006. Pervasive, ambient, ubiquitous: the magic of radio, In Proceedings of European Commission Conference ''From RFID to the Internet of Things", Bruxelles, Belgium.

Sugiyono. 2009. Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif, dan R & D. Bandung : Alfabeta.

Susim, T., & Darujati, C. (2021). Pengolahan citra untuk pengenalan wajah (face recognition) menggunakan opencv. *Jurnal Syntax Admiration*, *2*(3), 534-545.

B., Rais, M., & Sirad, M. A. H. (2019). Aplikasi Point OF Sales enggunakan Framework Laravel. *JIKO (Jurnal Informatika dan omputer)*, *2*(2), 55-59.

Turang, D. A. O. 2015. Pengembangan Sistem *Relay* Pengendalian Dan Penghematan Pemakaian Lampu Berbasis Mobile. *Seminar Nasional Informatika (SEMNASIF)* (Vol. 1, No. 1).

Utomo, B. T., Fitri, I., & Mardiani, E. (2021). Penerapan Face Recognition Pada Aplikasi Akademik Online. Jurnal Teknologi Informasi Dan Komunikasi), 5(4), 2021. Https://Doi.Org/10.35870/Jti

V. Geetanjali, I. Subramanian, G. Kannan, S. B. Prathiba and G. Raja, "IoTexpert: Interconnection, Interoperability and Integration of IoT Platforms," 2019 11th International Conference on Advanced Computing (ICoAC), 2019, pp. 212-219, doi: 10.1109/ICoAC48765.2019.246842.

Y. Efendi, "Internet Of Things (IoT) Sistem Pengendalian Lampu Menggunakan *Raspberry* Pi Berbasis Mobile," J. Ilm. Ilmu Komput., vol. 4, no. 2, pp. 21–27, 2018, doi: 10.35329/jiik.v4i2.41.

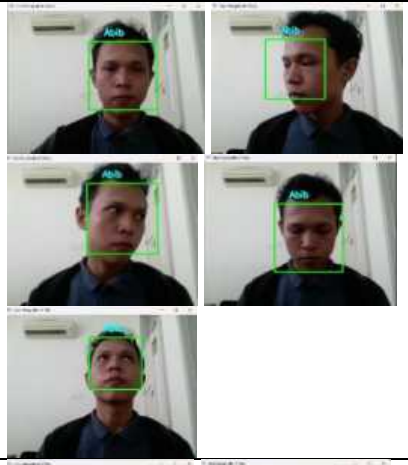
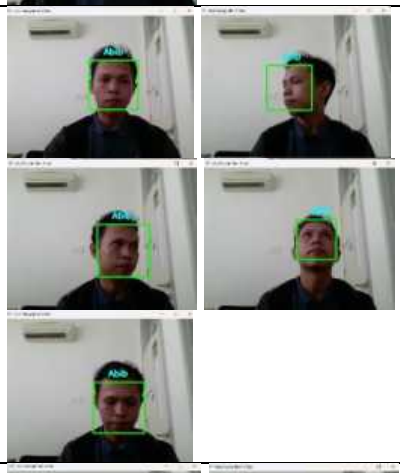
Y. Fauzan, "Kotak Penerima Paket Berbasis IoT Menggunakan Modul ESP32-CAM," 2020

Zein A. (2018). PENDETEKSIAN MULTI WAJAH DAN RECOGNITION SECARA REAL TIME MENGGUNAKAN METODA PRINCIPAL COMPONENT ANALYSIS (PCA) DAN EIGENFACE. 1–6.

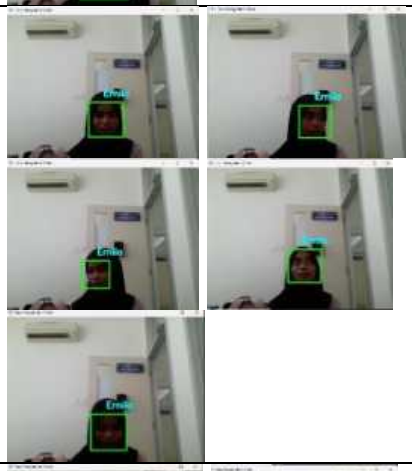
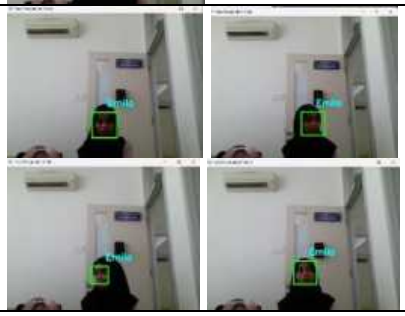
Zhamanov, A., Sakhiyeva, Z., Suliyev, R., & Kaldykulova, Z. (2017, November). IoT smart campus review and implementation of IoT applications into education process of university. In 2017 13th International Conference on Electronics, Computer and Computation (ICECCO) (pp. 1-4). IEEE.

# LAMPIRAN

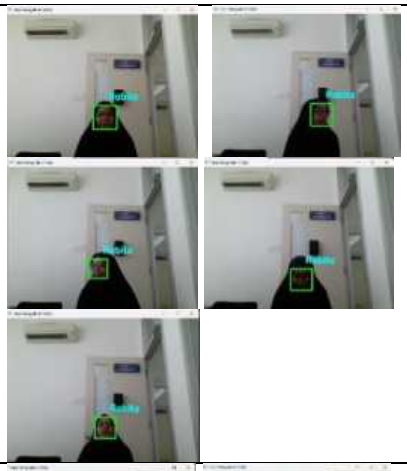
## LAMPIRAN 1 Jarak *face recognition* pada wajah yang terdaftar dalam *database*

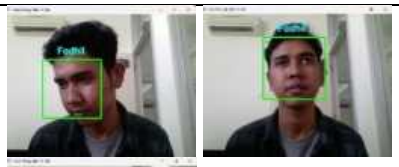
| No | Sampel | Jarak (cm) | Lampiran Foto |
|---|---|---|---|
| 1. | Abib | 30 |  |
| | | 50 |  |
| | | 70 |  |

| No | Sampel | Jarak (cm) | Lampiran Foto |
|---|---|---|---|
| | | 100 |  |
| 2. | Pati | 30 |  |
| | | 50 |  |
| | | 70 |  |
| | | 100 |  |
| 3. | Ernila | 30 |  |

| No | Sampel | Jarak (cm) | Lampiran Foto |
|---|---|---|---|
| | | |  |
| | | 50 |  |
| | | 70 |  |
| | | 100 |  |

| No | Sampel | Jarak (cm) | Lampiran Foto |
|---|---|---|---|
| | | |  |
| 4. | Rabita | 30 |      |
| | | 50 |      |
| | | 70 |      |

| No | Sampel | Jarak (cm) | Lampiran Foto |
|---|---|---|---|
| | | 100 |  |
| 5. | Yuar | 30 |  |
| | | 50 |  |
| | | 70 |  |
| | | 100 |  |
| 6. | Fadhil | 30 |  |

| No | Sampel | Jarak (cm) | Lampiran Foto |
|---|---|---|---|
| | | |   |
| | | 50 |  |
| | | 70 |  |
| | | 100 |  |

**LAMPIRAN 2 Jarak *face recognition* pada wajah yang tidak terdaftar dalam *database***

| No | Sampel | Lampiran Foto |
|---|---|---|
| 1. | Yusuf |   |
| 2. | Kholik |   |

| | | |
|---|---|---|
| 3. | Regita |   |

## LAMPIRAN 3 Program Code

### 1. Kode program untuk modul kamera ESP32 CAM

```
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//            Ensure ESP32 Wrover Module or other board with PSRAM is selected
//            Partial images will be transmitted if image exceeds buffer size
//

// Select camera model
//#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

const char* ssid = "Patioo";
const char* password = "patirzky_17";

void startCameraServer();

void setup() {
  Serial.begin(115200);
        setDebugOutput(true);
        println();

        a_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;

  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
  //                     for larger pre-allocated frame buffer.
  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 defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
          ı;



          _t * s = esp_camera_sensor_get();
```

```
  // initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
  }
  // drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}
```

## 2. Kode untuk *training* wajah *database* (LPBH)

```
import cv2
import os
import numpy as np

    name = 'CobaData'
    scade_file = 'Cascade Classifier/haarcascade_frontalface_default.xml'
    scade = cv2.CascadeClassifier(face_cascade_file)
```

```
total_images = 5
counter = 1
ids = 1

cam = cv2.VideoCapture(0)                                   # Akses Kamera
while True:
   ret, frame = cam.read()                                  # Membaca setiap frame dari stream
kamera
   frame_copy = frame.copy()                                # Copy frame
   gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)           # Mengubah mode BGR
ke GRAY (hitam putih)

                                                # Proses pencarian wajah
   faces = face_cascade.detectMultiScale(gray, 1.3, 3)      #
<cascade_file>.detectMultiScale(<frame>, <scale_factor>, <min_neighbors>)
   for x, y, w, h in faces:                                 # Looping semua wajah yang terdeteksi
      cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 3)    # Gambar box untuk setiap wajah

      if cv2.waitKey(1) & 0xff == ord('c'):                 # Menunggu tombol c di tekan
         roi_face = frame_copy[y:y+h, x:x+w]                # region of interest dari frame
         cv2.imwrite(f'{folder_name}/wajah.{ids}.{counter}.jpg',# write region wajah
                  roi_face)
         gray_face = gray[y:y+h, x:x+w]                     # region of interest dari frame
         cv2.imwrite(f'{folder_name}/wajah_gray.{ids}.{counter}.jpg',# write region wajah
                  gray_face)
         counter += 1
         if counter > total_images:
            print(f'[INFO] {total_images} IMAGE CAPTURED!')     # info done proses

   cv2.imshow('Face Detect Video', frame)                   # Jendela untuk menampilkan hasil

   if cv2.waitKey(1) & 0xff == ord('x'):                    # Exit dengan tombol x
      break

cam.release()                                               # Menyudahi akses kamera
cv2.destroyAllWindows()                                     # Menutup jendela

recognizer = cv2.face.LBPHFaceRecognizer_create()           # Create recognizer object
images = os.listdir(folder_name)                            # List semua path data wajah pada
folder train data

image_arrays = []                                           # Container semua array data wajah
image_ids = []                                              # Container semua ID data wajah
for image_path in images:                                   # Looping semua path data wajah
   if not image_path.startswith('.'):                       # Pengecualian untuk file yang dimulai
dengan "."
      splitted_path = image_path.split('.')
      print(splitted_path)
      image_id = int(splitted_path[1])
      image = cv2.imread(os.path.join(folder_name, image_path))
      image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

      age_array = np.array(image, 'uint8')                  # Ambil array data wajah

      age_arrays.append(image_array)                        # Store array data wajah ke
      tainer
      age_ids.append(image_id)                              # Store ID data wajah ke
```

```
list/container

recognizer.train(image_arrays, np.array(image_ids))          # Train recognizer
recognizer.save('recognizer/faces_data.yml')                 # Save model recognizer
print('[INFO] TRAIN RECOGNIZER SUCCESS!')

recognizer.read('recognizer/faces_data.yml')                 # Load recognizer
font = cv2.FONT_HERSHEY_SIMPLEX                              # Specify jenis font dari
OpenCV

known_names = ['Yuar', 'Ernila', 'Rabita', 'Rehan', 'Abib', 'Pati', 'Fadhlan'] # List untuk nama
yang ada di model

cam = cv2.VideoCapture(0)                                    # Akses Kamera
while True:
    ret, frame = cam.read()                                  # Membaca setiap frame dari stream
kamera
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)           # Mengubah mode BGR
ke GRAY (hitam putih)

                                                 # Proses pencarian wajah
    faces = face_cascade.detectMultiScale(gray, 1.3, 3)      #
<cascade_file>.detectMultiScale(<frame>, <scale_factor>, <min_neighbors>)
    for x, y, w, h in faces:                                 # Looping semua wajah yang terdeteksi
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 3)    # Gambar box untuk setiap wajah

        roi_gray = gray[y:y+h, x:x+w]
        ids, dist = recognizer.predict(roi_gray)             # Prediksi wajah siapoa
        cv2.putText(frame, f'{known_names[ids-1]}',
                    (x+50, y-20), font, 1 ,(255, 255, 0), 3)        # Menaruh text pada frame


    cv2.imshow('Face Recognition Video', frame)              # Jendela untuk menampilkan
hasil

    if cv2.waitKey(1) & 0xff == ord('x'):                    # Exit dengan tombol x
        break

cam.release()                                                # Menyudahi akses kamera
cv2.destroyAllWindows()
```

### 3. Kode program untuk *face recognition*

```
from flask import Flask, render_template, Response
import cv2
import requests

app = Flask(__name__)
global_name = "unknown"
stream_url = "http://192.168.18.210:81/stream"
api_endpoint = "http://telcomlaboratoryunhas.com/api/iot-input"  # Replace this with your API
                ndpoint

                l_name_to_api(name):
                pare data to send to the API
```

```
    data = {
        "mac": "7A-EC-20-97-FF-9E",
        "value": name
    }

    try:
        # Send a POST request to the API endpoint
        response = requests.post(api_endpoint, json=data)

        # Check if the request was successful
        if response.status_code == 200:
            print("Name sent successfully to API")
        else:
            print(f"Failed to send name to API. Status code: {response.status_code}")
            print("Response content:", response.content)
    except Exception as e:
        print("Error sending name to API:", e)

def gen_frames():
    face_cascade_file = 'Cascade Classifier/haarcascade_frontalface_default.xml'
    face_cascade = cv2.CascadeClassifier(face_cascade_file)
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('recognizer/faces_data.yml')                    # Load recognizer
    font = cv2.FONT_HERSHEY_SIMPLEX

    known_names = ['Yuar', 'Ernila', 'Rabita', 'Rehan', 'Abib', 'Pati', 'Fadhil']
        # List untuk nama yang ada di model

    cam = cv2.VideoCapture(stream_url)                                  # Akses Kamera
    name_sent = False  # Flag to track whether the name has been sent to the API
    while True:
        ret, frame = cam.read()                                 # Membaca setiap frame dari stream
          kamera
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)              # Mengubah mode
          BGR ke GRAY (hitam putih)

        # Proses pencarian wajah
        faces = face_cascade.detectMultiScale(gray, 1.3, 3)          #
          <cascade_file>.detectMultiScale(<frame>, <scale_factor>, <min_neighbors>)
        for x, y, w, h in faces:                                # Looping semua wajah yang terdeteksi
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 3)    # Gambar box untuk setiap
          wajah

            roi_gray = gray[y:y+h, x:x+w]
            ids, dist = recognizer.predict(roi_gray)                # Prediksi wajah siapoa
            name = known_names[ids-1]
            cv2.putText(frame, f'{name}',
                        (x-20, y-20), font, 1 ,(255, 255, 0), 3)        # Menaruh text pada frame

            if not name_sent:
                send_name_to_api(name)  # Send detected name to API
                name_sent = True  # Set flag to True to indicate that the name has been sent

            2.imshow('Face Recognition Video', frame)                # Jendela untuk menampilkan
            asil

            y = cv2.waitKey(1) & 0xFF
```

```
        if key == ord('x'):                    # Exit dengan tombol x
            break
        elif key == ord('r'):                  # Press 'r' to resend the name to API
            name_sent = False  # Reset the flag to allow resending the name

    cam.release()                                          # Menyudahi akses kamera
    cv2.destroyAllWindows()

gen_frames()


if __name__ == "__main__":
    app.run(debug=True)
```

## 4. Kode program RFID Reader dan pembukaan pintu

```
#include <SPI.h>
#include <MFRC522.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>  // Untuk mengurai JSON
#include <WiFiManager.h>
#define LED_STDBY 4  // led standby
#define LED_FALSE 5  // led False
#define LED_TRUE 15  // led true
#define RST_PIN D1
#define SDA_PIN D2
#define relay D4


const char* ssid = "Lorbaks";
const char* password = "Lorbax19";
const char* serverUrl = "http://telcomlaboratoryunhas.com/api/iot-input";  // Replace with
your server URL
const char* WebURL = "https://telcomlaboratoryunhas.com/api/commands";
const char* mac_system;  // dari giga 98-29-A6-93-3C-BB
//const int relay = 13;
String key = "{\"message\":\""
             "0"
             "\"}";
String idMac;
int id_new = 0;
short perintah = 3;
unsigned long previousMillis = 0;  // Waktu sebelumnya
          ong interval = 1000;     // Interval waktu (1 detik)
          ! mfrc522(SDA_PIN, RST_PIN);


          () {
```

```
  Serial.begin(115200);
  Serial.println();
  WiFi.begin(ssid, password);
  Serial.println(WiFi.macAddress());
  idMac = WiFi.macAddress();

  WiFiManager wifiManager;
  wifiManager.autoConnect("Lorbaks", "Lorbax19");  // "ESP8266-AP" adalah nama AP
WiFi yang akan dibuat jika gagal terhubung ke WiFi sebelumnya
  Serial.println("Terhubung ke jaringan WiFi!");
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  pinMode(LED_STDBY, OUTPUT);
  pinMode(LED_FALSE, OUTPUT);
  pinMode(LED_TRUE, OUTPUT);
  pinMode(relay, OUTPUT);
  SPI.begin();
  mfrc522.PCD_Init();
  Serial.println("Dekatkan Kartu RFID Anda ke Reader");
  Serial.println();
  digitalWrite(LED_STDBY, HIGH);
  digitalWrite(LED_FALSE, LOW);
  digitalWrite(LED_TRUE, LOW);
  //digitalWrite(relay, LOW);
}


void loop() {
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  unsigned long currentMillis = millis();
  unsigned long currentMicros = micros();  // Waktu mikrodetik saat ini
  if (currentMillis - previousMillis >= interval) {
    web();
    previousMillis = currentMillis;
  }

  // if (currentMicros - previousMicros >= intervalMicros) {
  if (mfrc522.PICC_IsNewCardPresent() && mfrc522.PICC_ReadCardSerial()) {
            ();

            ɔusMicros = currentMicros;
```

```
  switch (perintah) {
    case 1:
      digitalWrite(LED_STDBY, LOW);
      digitalWrite(LED_FALSE, LOW);
      digitalWrite(LED_TRUE, HIGH);
      Serial.println("1");
      digitalWrite(relay, HIGH);
      Serial.println("Pintu Terbuka");
      delay(10000);

      digitalWrite(LED_STDBY, HIGH);
      digitalWrite(LED_FALSE, LOW);
      digitalWrite(LED_TRUE, LOW);
      perintah = 0;
      digitalWrite(relay, LOW);
      break;

    case 2:
      digitalWrite(LED_STDBY, LOW);
      digitalWrite(LED_FALSE, HIGH);
      digitalWrite(LED_TRUE, LOW);
      Serial.println("2");
      delay(500);
      digitalWrite(LED_STDBY, HIGH);
      digitalWrite(LED_FALSE, LOW);
      digitalWrite(LED_TRUE, LOW);
      perintah = 0;
      break;

  }
}

void web() {
  WiFiClient client;
  HTTPClient http;
  http.begin(client, WebURL);

  // Mengeksekusi permintaan dan menyimpan hasil
  int httpCode = http.GET();
  if (httpCode == 200) {
    if (httpCode == HTTP_CODE_OK) {
          payload = http.getString();
          println("Data JSON diterima:");
          println(payload);
```

```
    // Mengurai data JSON
    DynamicJsonDocument doc(1024);  // Ubah sesuai kebutuhan
    deserializeJson(doc, payload);

    // Memeriksa setiap objek dalam JSON
    JsonArray data = doc.as<JsonArray>();
    for (JsonObject item : data) {
      int id = item["id"];
      String mac = item["mac"];
      String status = item["status"];
      // Menampilkan data per ID
      // Serial.println("ID: " + String(id));
      // Serial.println("MAC: " + mac);
      // Serial.println("Status: " + status);
      // Serial.println();

      // Melakukan sesuatu dengan data yang diambil
      if (id > id_new) {
        if (mac == idMac) {
          if (status == "0") {
            Serial.println("terbuka");
            perintah = 2;
          }
          if (status == "1") {
            Serial.println("id salah");
            perintah = 1;
          }
        }
        id_new = id;
      }
    }
  }
 } else {
  Serial.println("Gagal mengambil data dari endpoint.");
 }
 Serial.println(id_new);

 http.end();
 client.stop();
}

void kirimid() {
        nt client;
        ient http;
        n(client, serverUrl);
        nt("UID tag :");
```

```
  String rfid = "";
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    rfid += mfrc522.uid.uidByte[i];
  }
  Serial.println(rfid);
  http.addHeader("Content-Type", "application/json");
  String mac = "6E-1B-ED-BE-13-B3";
  String postData = "{\"mac\":\"" + mac + "\",\"value\":\"" + rfid + "\"}";
  int httpResponseCode = http.POST(postData);
  delay(1000);
  String response = http.getString();
  Serial.println("Response: " + response);
  Serial.println("HTTP Response code: " + String(httpResponseCode));
  if (httpResponseCode == 200) {
    perintah = 1;
  } else if (httpResponseCode == 400) {
    perintah = 2;
  }
  http.end();
  client.stop();
}
```

**5. Kode program untuk website**

Kode program dapat diakse pada pada link https://github.com/agthm3/website-face