

DAFTAR PUSTAKA

- [1] A. Taqwa *et al.*, “1) , 2) , 3),” *Ranc. Bangun Kunci Loker Mhs. Di Politek. Negeri Sriwij.* Menggunakan Fingerpr. dan Password Berbas. Arduino Mega 2560 Dengan SIM1900A, vol. 9, no. 2, p. 4045, 2019.
- [2] Annisya, L. Hermanto, and R. Candra, “Sistem Keamanan Buka Tutup Kunci Brankas Menggunakan Sidik Jari Berbasis Arduino Mega,” *J. Inform. dan Komput.*, vol. Volume 22, no. 1, pp. 1–9, 2017.
- [3] H. Ieghecog, “sistem absensi menggunakan face recognition,” *Inf. Kaji. ilmu Komun.*, pp. 1–8, 2019.
- [4] A. Hazarah, “Rancang Bangun Smart Door Lock,” *J. Teknol. Inform. dan Terap.*, vol. 04, no. 01, pp. 5–10, 2017.
- [5] M. Rachman Waluyo, “Desain Ulang Lemari Arsip Fleksibel Dengan Pendekatan Perancangan Generik,” *J. Teknol. Dan Manaj. Ind.*, vol. 4, no. 2, pp. 13–17, 2018, doi: 10.36040/jtmi.v4i2.238.
- [6] C. Lesmana, R. Lim, and L. W. Santoso, “Implementasi Face Recognition menggunakan Raspberry pi untuk akses Ruangang Pribadi,” *J. Infra Petra*, pp. 2–5, 2019.
- [7] R. Pi, “Raspberry Pi 4 Model B,” *Datasheet*, no. June, pp. 1–13, 2019.
- [8] I. A. Nugraha, F. Pradana, and A. Arwan, “Pengembangan Sistem Manajemen Notulensi dan Dokumentasi Rapat Berbasis Web (Studi Kasus : Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya),” *J-Ptiik.Ub.Ac.Id*, vol. 4, no. 4, pp. 1273–1280, 2020.
- [9] M. E. Nurlana, A. Murnomo, and I. A. Abstrak, “Pembuatan Power Supply dengan Tegangan Keluaran Variabel Menggunakan Keypad Berbasis Arduino Uno,” *Edu Elektr. J.*, vol. 8, no. 2, pp. 53–59, 2019, [Online]. Available: <https://journal.unnes.ac.id/sju/index.php/eduel/article/view/27045>.
- [10] ade elbani nugroho farhan, muhammad saleh, “PERANCANGAN SISTEM KENDALI KIPAS ANGIN

- OTOMATIS BERBASIS NodeMCU V3,” no. PERANCANGAN SISTEM KENDALI KIPAS ANGIN OTOMATIS BERBASIS NodeMCU V3 Farhan, p. 10, 2020.
- [11] A. Uno, P. T. Xyz, R. S. K, and G. Sembada, “Perancangan Sistem Keamanan Menggunakan Solenoid *Door Lock* Berbasis Arduino Uno pada Pintu Laboratorium di PT. XYZ,” *J. E-KOMTEK (Elektro-Komputer-Teknik)*, vol. 4, no. 1, pp. 62–74, 2020.
- [12] S. Siswanto, M. Anif, D. N. Hayati, and Y. Yuhefizar, “Pengamanan Pintu Ruangan Menggunakan Arduino Mega 2560, MQ-2, DHT-11 Berbasis Android,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 1, pp. 66–72, 2019, doi: 10.29207/resti.v3i1.797.
- [13] T. Susim and C. Darujati, “Pengolahan Citra untuk Pengenalan Wajah (Face Recognition) Menggunakan OpenCV,” *J. Heal. Sains*, vol. 2, no. 3, pp. 534–545, 2021, doi: 10.46799/jsa.v2i3.202.

LAMPIRAN A

LISTING PROGRAM

A. PROGRAM PYTHON

```
from ast import And
from cgitb import enable
import face_recognition
import cv2
import numpy as np
import os
import socketio
import base64
import time
import os
import serial
import time

enable_serial = True
enable_socket = True

known_face_encodings = []
known_face_names = []

doorOpen = 0
timeOpen = 0

if enable_serial :
    ser = serial.Serial('/dev/ttyACM0', timeout=0.1)
    time.sleep(1)

listFile = os.listdir('dataset')
for file in listFile :
```

```

faces =
face_recognition.face_encodings(face_recognition.load_image_file("dataset/" + file))
if len(faces) > 0 :
    known_face_encodings.append(faces[0])
    known_face_names.append(file.split('.')[0])

video_capture = cv2.VideoCapture(0)

sio = socketio.Client()

def listToString(s):
    str1 = ""
    for ele in s:
        str1 += ele + ','
    return str1

@sio.event
def connect():
    print('connection established')

@sio.event
def capture(data):
    if data != "Unknown" and data != "-":
        doorOpen = str(int(data[-1]) - 1)
        timeOpen = time.time()
        if enable_serial :
            ser.write((doorOpen + '=1\n').encode())

@sio.event
def add(data):
    print('message received with ', data)
    ret, frame = video_capture.read()
    cv2.imwrite("dataset/" + data + ".jpg", frame)
    os.system("pm2 restart main")

```

```

@sio.event
def delete(data):
    print(data)
    os.system("rm 'dataset/' + data + '.jpg'")
    os.system("pm2 restart main")

if enable_socket :
    sio.connect("http://localhost:3000")

while True :

    ret, frame = video_capture.read()
    frame = cv2.flip(frame, 1)
    small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

    rgb_small_frame = small_frame[:, :, :-1]

    face_recognition.api
    face_locations = face_recognition.face_locations(rgb_small_frame)
    face_encodings = face_recognition.face_encodings(rgb_small_frame,
face_locations)

    face_names = []

    if len(known_face_names) > 0 :
        for face_encoding in face_encodings:
            matches =
face_recognition.compare_faces(known_face_encodings,
face_encoding)
            name = "Unknown"
            face_distances =
face_recognition.face_distance(known_face_encodings, face_encoding)
            best_match_index = np.argmin(face_distances)
            best_match_value = np.min(face_distances)
            if matches[best_match_index] and best_match_value < 0.35 :
                name = known_face_names[best_match_index]

```

```

        face_names.append(name)
    for (top, right, bottom, left), name in zip(face_locations,
face_names):
        top *= 4
        right *= 4
        bottom *= 4
        left *= 4
        cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
        cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0,
255), cv2.FILLED)
        font = cv2.FONT_HERSHEY_DUPLEX
        cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255,
255, 255), 1)

    resized = cv2.resize(frame, (int(frame.shape[1] * 0.75),
int(frame.shape[0] * 0.75)), interpolation = cv2.INTER_AREA)
    retval, buffer = cv2.imencode('.jpg', resized)
    jpg_as_text = base64.b64encode(buffer).decode()
    if enable_socket :
        sio.emit('frame', str(jpg_as_text))
        sio.emit("face_list", listToString(known_face_names))

    if len(face_names) > 0 :
        if enable_socket :
            sio.emit('value', face_names[0])
        else :
            if enable_socket :
                sio.emit('value', '-')

    if not enable_socket :
        cv2.imshow('Video', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

    if doorOpen != 0 :
        if time.time() > timeOpen + 3 :

```

```

if enable_serial :
    ser.write((doorOpen + '=0\n').encode())
doorOpen = 0

if enable_serial :
    line = ser.readline()
    if line :
        statusDoor = (line.decode("utf-8")).replace('\n', '')
        print(statusDoor)
        if enable_socket :
            sio.emit('door', statusDoor)

```

B. PROGRAM SERVER

```

const express = require('express');
const mysql = require('mysql')
const app = express();
const http = require('http');
const server = http.createServer(app);
const { Server } = require("socket.io");
const io = new Server(server);

const db = mysql.createConnection({
  host: 'localhost',
  user: 'dwita',
  password: 'cilacap',
  database: 'ta_dwita'
})

db.connect((err) => {
  if (err) throw err
  else {
    console.log("database terhubung")
  }
})

```

```

app.set('view engine', 'ejs');

app.use('/public', express.static('public'))

app.get('/', function (req, res) {
  db.query(`
select
  history.*,
  date_format(history.ts_created, "%Y-%m-%d %H:%i:%s") as
ts_created_string
from history
order by id desc
`, (err, result) => {
  if (err) {
    throw (err)
  }
  else {
    res.render('index', { history: result });
  }
})
});

app.get('/history/clear', (req, res) => {
  db.query(`truncate table history`, (err) => {
    if (err) {
      throw (err)
    }
    else {
      res.send('ok')
    }
  })
})

var lastValue = ""
io.on('connection', (socket) => {
  socket.on('value', (msg) => {

```



```

    io.emit("value", msg)
  })
  socket.on('frame', (msg) => {
    io.emit("frame", msg)
  })
  socket.on('add', (msg) => {
    io.emit("add", msg)
  })
  socket.on('face_list', (msg) => {
    io.emit("face_list", msg)
  })
  socket.on('delete', (msg) => {
    io.emit("delete", msg)
  })
  socket.on('door', (msg) => {
    io.emit("door", msg)
  })
  socket.on('door_force', (msg) => {
    db.query(`insert into history (name) values('Buka Paksa Pintu
    ${msg}')`)
  })
  socket.on('capture', (msg) => {
    db.query(`insert into history (name) values('${msg}')`)
    io.emit("capture", msg)
  })
});

server.listen(3000, () => {
  console.log('listening on *:3000');
}); })
socket.on('add', (msg) => {
  io.emit("add", msg)
})
socket.on('face_list', (msg) => {
  io.emit("face_list", msg)
})

```

```

socket.on('delete', (msg) => {
  io.emit("delete", msg)
})
socket.on('door', (msg) => {
  io.emit("door", msg)
})
});

server.listen(3000, () => {
  console.log('listening on *:3000');
});

```

C. PROGRAM ARDUINO MEGA

```

int pinSwitch[4] = {2, 3, 4, 5};
int pinRelay[4] = {6, 7, 8, 9};

void setup() {
  Serial.begin(9600);
  for (int I = 0; I < 4; i++) {
    pinMode(pinSwitch[i], INPUT_PULLUP);
    pinMode(pinRelay[i], OUTPUT);
    digitalWrite(pinRelay[i], HIGH);
  }
}

long lastSend = 0;
String lastData = "";
void loop() {
  if (millis() > lastSend + 100) {
    lastSend = millis();
    String data = "";
    data += (String) digitalRead(pinSwitch[0]);
    data += (String) digitalRead(pinSwitch[1]);
    data += (String) digitalRead(pinSwitch[2]);

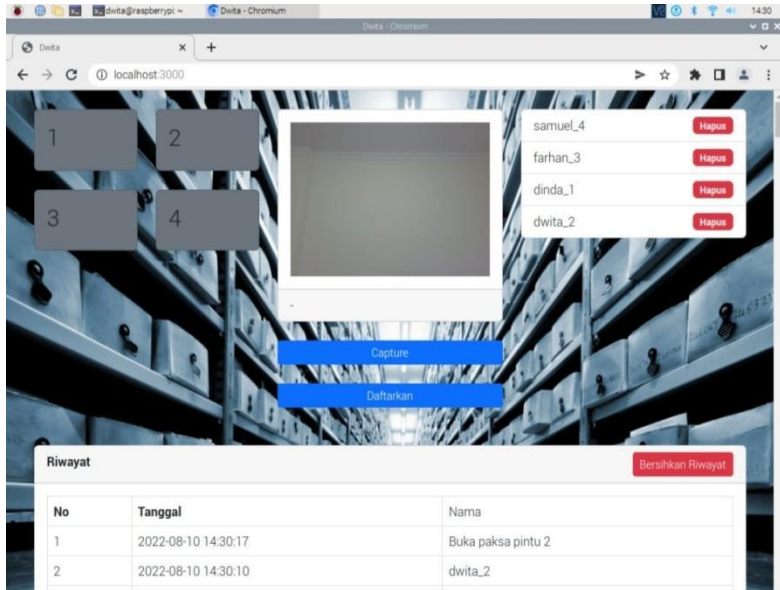
```

```
data += (String) digitalRead(pinSwitch[3]);
if(data != lastData) {
    lastData = data;
    Serial.println(data);
}
}
}
```

```
void serialEvent() {
    String data = Serial.readStringUntil('\n');
    int key = data.substring(0, 1).toInt();
    int val = data.substring(2).toInt();
    switch(key) {
        case 1 :
            digitalWrite(pinRelay[1], !val);
            digitalWrite(pinRelay[2], !val);
            digitalWrite(pinRelay[3], !val);
            break;
        default :
            digitalWrite(pinRelay[key], !val);
            break;
    }
}
```


LAMPIRAN B

TAMPILAN WEB PADA LAYAR MONITOR



LAMPIRAN C
DESAIN MEKANIK ALAT



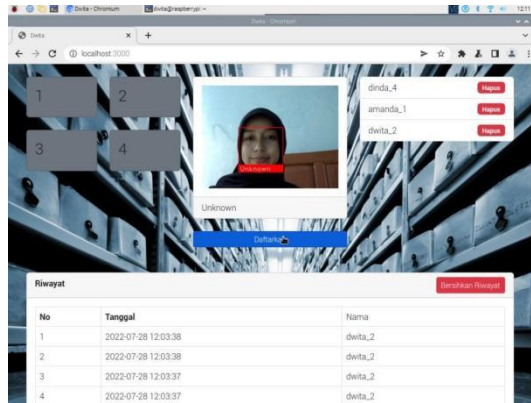
C-1

LAMPIRAN D

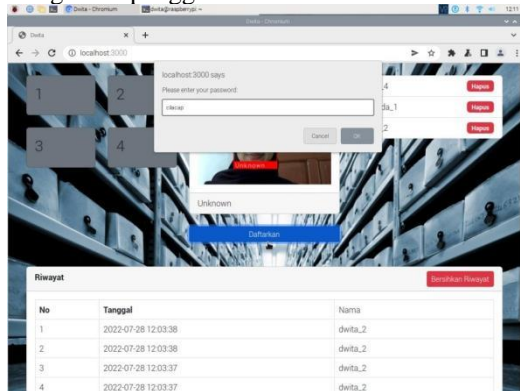
MANUAL BOOK SISTEM KEAMANAN LOKER ARSIP

Langkah pertama untuk menggunakan loker arsip adalah dengan colokkan Adaptor ke dalam sumber listrik. Kemudian melakukan langkah-langkah pendaftaran pada layar monitor sebagai berikut :

1. Langkah awal pendaftaran wajah yaitu dengan mengklik “Daftarkan”

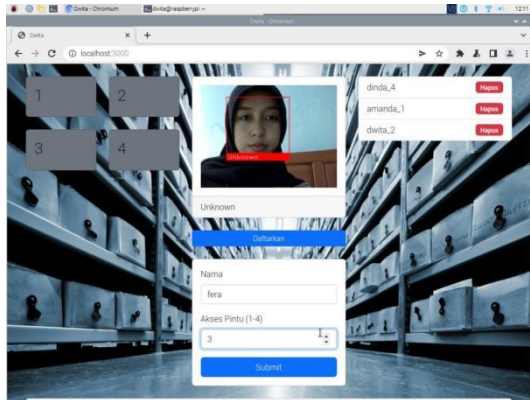


2. Kemudian masukan *password* “cilacap” lalu akan muncul tampilan untuk mengisi ID pengguna loker.

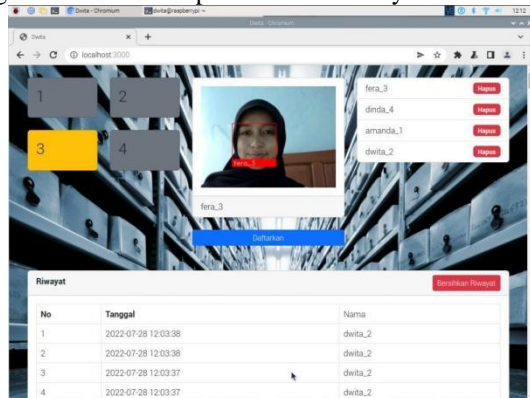


3. Pada tampilan pengisian ID pengguna loker, masukan nama

pengguna dan nomor loker yang akan digunakan, kemudian klik enter.



4. Jika wajah berhasil mendaftarkan maka sistem akan *restart* dan pengguna akan muncul pada *list user* di layar monitor.



LAMPIRAN E

DOKUMENTASI ALAT



Gambar Mekanik Tampak Luar



Gambar *wiring* komponen

BIODATA PENULIS



Nama : Dwita Nuraini Diatma
Tempat/Tanggal Lahir : Cilacap, 24 Juli 2001
Alamat : Jalan Kroya-Nusawungu, Desa
Danasri RT 01 RW 03, Kecamatan
Nusawungu, Kabupaten Cilacap
Email : dwita.diatma07@gmail.com
Telepon/HP : 0856-0071-4894
Hobi : Catur
Motto : Tak pernah ada kata terlambat untuk
menjadi apa yang kamu impikan
Riwayat Pendidikan :

- TK Islam Bustanul Athfal Tahun 2006-2007
- SD Negeri Danasri 01 Tahun 2007-2013
- SMP Negeri 1 Nusawungu Tahun 2013-2016
- SMK Negeri 1 Binangun Tahun 2016-2019