

**RANCANG BANGUN KAMERA PENDETEKSI KEHADIRAN TARUNA
BERBASIS AI (*ARTIFICIAL INTELLIGENCE*) DENGAN
METODE *FISHERFACE* PADA LABORATORIUM
POLITEKNIK PENERBANGAN SURABAYA**

PROYEK AKHIR



Oleh:

TEGAR KURNIAWAN AL RASYID
NIT. 30121023

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA**

2024

**RANCANG BANGUN KAMERA PENDETEKSI KEHADIRAN TARUNA
BERBASIS AI (*ARTIFICIAL INTELLIGENCE*) DENGAN
METODE *FISHERFACE* PADA LABORATORIUM
POLITEKNIK PENERBANGAN SURABAYA**

PROYEK AKHIR

Diajukan sebagai Syarat Menempuh Mata Kuliah Proyek Akhir pada
Program Studi Diploma 3 Teknik Listrik Bandara.



Oleh:

TEGAR KURNIAWAN AL RASYID
NIT. 30121023

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA**

2024

LEMBAR PERSETUJUAN

RANCANG BANGUN KAMERA PENDETEKSI KEHADIRAN TARUNA
BERBASIS AI (*ARTIFICIAL INTELLIGENCE*) DENGAN
METODE *FISHERFACE* PADA LABORATORIUM
POLITEKNIK PENERBANGAN SURABAYA

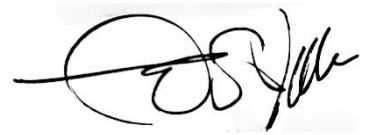
Oleh:

TEGAR KURNIAWAN AL RASYID
NIT. 30121023

Disetujui untuk diujikan pada:
Surabaya, 09 Agustus 2024

Dosen Pembimbing I : YUDHIS THIRO KABUL YUNIOR, ST, M.Kom
NIP. 19870224 202203 1 003

Dosen Pembimbing II : Dr. GUNAWAN SAKTI, ST, MT.
NIP. 19881001 200912 1 003



LEMBAR PENGESAHAN

RANCANG BANGUN KAMERA PENDETEKSI KEHADIRAN TARUNA
BERBASIS AI (*ARTIFICIAL INTELLIGENCE*) DENGAN
METODE *FISHERFACE* PADA LABORATORIUM
POLITEKNIK PENERBANGAN SURABAYA

Oleh:
TEGAR KURNIAWAN ALRASYID
NIT. 30121023

Telah dipertahankan dan dinyatakan lulus pada Ujian Proyek Akhir Program
Pendidikan Diploma 3 Teknik Listrik Bandar Udara
Politeknik Penerbangan Surabaya

Pada tanggal : 09 Agustus 2024

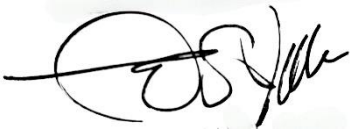
Panitia Penguji:

Ketua : Drs. HARTONO, ST., M. Pd., M.M.
NIP. 196304081989021001

Sekretaris : Dr. GUNAWAN SAKTI, ST, MT.
NIP. 19881001 200912 1 003

Anggota : YUDHIS THIRO KABUL YUNIOR, ST, M.Kom
NIP. 19870224 202203 1 003

Mengetahui
Ketua Program Studi
D3 Teknik Listrik Bandara


Dr. GUNAWAN SAKTI, ST,MT
NIP. 19881001 200912 1 003

PERNYATAAN KEASLIAN HAK CIPTA

Saya yang bertanda tangan di bawah ini:

Nama : Tegar Kurniawan Al Rasyid
NIT : 30121023
Program Studi : Diploma 3 Teknik Listrik Bandara
Judul Proyek Akhir : RANCANG BANGUN KAMERA PENDETEKSI
KEHADIRAN TARUNA BERBASIS AI (ARTIFICIAL
INTELLIGENCE) DENGAN METODE FISHERFACE
PADA LABORATORIUM POLITEKNIK
PENERBANGAN SURABAYA

Dengan ini menyatakan bahwa:

1. Proyek Akhir ini merupakan karya asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Politeknik Penerbangan Surabaya maupun Perguruan Tinggi lain, serta dipublikasikan, kecuali secara tertulis dengan jelas dicantumkan dalam daftar pustaka.
2. Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan Hak Bebas Royalti Non Ekklusif (Non-Exclusive Royalty-Free Right) kepada Politeknik Penerbangan Surabaya beserta perangkat yang ada (jika diperlukan). Dengan hak ini, Politeknik Penerbangan Surabaya berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (database), merawat dan mempublikasikan Proyek Akhir dengan tetap mencantumkan nama saya sebagai pencipta dan sebagai pemilik Hak Cipta.

Pernyataan ini saya buat dengan sebenarnya. Apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Politeknik Penerbangan Surabaya.

Surabaya, 09 Agustus 2024
Yang membuat pernyataan



TEGAR KURNIAWAN AL RASYID
30121023

KATA PENGANTAR

Puji syukur atas kehadiran Tuhan Yang Maha Esa ,atas berkat rahmat dan karunia-nya berhasil menyelesaikan Proyek Akhir ini dapat selesai tepat pada waktu yang telah ditentukan. Proyek akhir dengan judul: “RANCANG BANGUN KAMERA PENDETEKSI KEHADIRAN TARUNA BERBASIS AI (ARTIFICIAL INTELLIGENCE) DENGAN METODE FISHERFACE PADA LABORATORIUM POLITEKNIK PENERBANGAN SURABAYA” diajukan untuk memenuhi salah satu persyaratan menyelesaikan program Diploma 3 Teknik Listrik Bandara, Jurusan Teknik Penerbangan di Politeknik Penerbangan Surabaya.

Dalam penyusunan proyek akhir ini, telah berusaha memberikan yang terbaik dalam penyajian tulisan ini dan dapat mengaplikasikan rancangan proyek akhir ini di Poltekbang Surabaya. Namun peneliti menyadari masih banyak kekurangan baik perancangan alat maupun cara penulisan dalam penyusunan proyek akhir ini. Oleh sebab itu, kritik dan saran yang membangun dari semua pihak untuk penyempurnaan rancangan ini sangat peneliti harapkan.

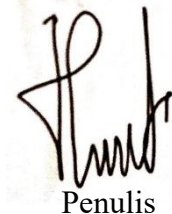
Pada kesempatan ini peneliti mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberi bantuan dan bimbingan dalam menyelesaikan Proyek Akhir ini. Dan dalam kesempatan ini pula peneliti mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua dan kakak Mayang Alindra S.Ak., yang telah memberikan dukungan moral dan material sehingga penulis dapat menyelesaikan laporan Proyek Akhir.
2. Bapak Ahmad Bahrawi, S.E., M.T selaku Direktur Politeknik Penerbangan Surabaya.
3. Dr. Gunawan Sakti, ST, MT.. selaku Ketua Program Studi Teknik Listrik Bandar Udara di Politeknik Penerbangan.
4. Bapak Yudhis Thiro, S.T., M.Kom selaku dosen pembimbing I yang senantiasa membimbing dan membantu dalam pengerjaan alat.
5. Mas Gunawan selaku dosen pembimbing II yang senantiasa membimbing dan membantu dalam penulisan Proyek Akhir.

6. Segenap Dosen dan Staff Pengajar Program Studi Diploma 3 Teknik Listrik Bandara yang telah membantu penulis dalam proses pembelajaran di Politeknik Penerbangan Surabaya.
7. Senior dan Alumni Politeknik Penerbangan Surabaya.
8. Para senior dan rekan selama *On the Job Training* (OJT) I dan II yang telah membagikan ilmu dan pengalamannya.
9. Saudara seperjuangan satu program studi Diploma 3 Teknik Listrik Bandara angkatan XVI, dan adik adik TLB XVII yang senantiasa memberikan dukungan saran dalam penyelesaian Proyek Akhir ini.
10. Serta semua pihak yang telah membantu dan membimbing selama pengerjaan dan penulisan proyek akhir.
11. Ucapan terima kasih yang terakhir ditujukan kepada manusia *special* yang selama ini menemani dan *men-support* penulis dalam penulisan laporan Proyek akhir ini. Terima kasih Adelia Mega Lourenza Ramadhani.

Akhir kata dengan segala kerendahan hati, Penulis menyadari bahwa dalam penulisan Proyek Akhir ini masih memiliki banyak kekurangan dan jauh dari kesempurnaan. Oleh karena itu, peneliti mengharapkan kritik dan saran yang membangun dari para pembaca. Penulis berharap semoga laporan ini dapat memberikan manfaat bagi semua pihak. Terima Kasih.

Surabaya, 26 Juli 2024



Penulis

ABSTRAK

RANCANG BANGUN KAMERA PENDETEKSI KEHADIRAN TARUNA BERBASIS AI (*ARTIFICIAL INTELLIGENCE*) DENGAN METODE *FISHERFACE* PADA LABORATORIUM POLITEKNIK PENERBANGAN SURABAYA

Oleh:

TEGAR KURNIAWAN AL RASYID

30121023

Perlu adanya peningkatan kedisiplinan bagi taruna untuk menghindari sikap indiscipliner. Dimulai dari hal yang paling kecil yakni absensi daftar kehadiran taruna, yang membutuhkan pengembangan pada absensi dengan menggunakan pendeteksian wajah sebagai upaya peningkatan kedisiplinan taruna. Dengan menggunakan pendeteksi wajah mempermudah prodi untuk memantau kehadiran taruna. Selain itu pendeteksi wajah memastikan dari tindakan kecurangan absensi.

Penelitian ini bertujuan untuk merancang dan membangun sistem kamera pendeteksi kehadiran taruna berbasis Artificial Intelligence (AI) dengan menggunakan metode Fisherface. Metode Fisherface dipilih karena kemampuannya dalam mengenali wajah dengan akurasi tinggi meskipun terdapat variasi ekspresi wajah, pencahayaan, dan jarak. Sistem yang dibangun terdiri dari beberapa komponen utama, yaitu kamera sebagai alat penangkap citra wajah, dan perangkat lunak yang mengimplementasikan algoritma Fisherface untuk pengenalan wajah.

Hasil dari penelitian ini proses absensi taruna di Politeknik Penerbangan Surabaya dapat dilakukan secara otomatis dan real-time, mengurangi kesalahan manual dan meningkatkan efisiensi administrasi. Sistem ini dapat diterapkan di Laboratorium Politeknik Penerbangan Surabaya maupun ruang prodi untuk mempermudah proses absensi dan memastikan kehadiran taruna dengan lebih tepat waktu. Kelebihan dari monitoring kehadiran taruna ini dari penelitian sebelumnya yaitu jarak yang lebih jauh yakni 2 meter pada lumen 239 dan 2,5 meter pada lumen 279. Kecepatan dalam mendeteksi dan mengirimkan data ke *database* jauh lebih cepat yaitu 1,40 detik. Keakuratan (tingkat kemiripan) dalam mendeteksi sebesar 92,7% dari 69 data percobaan.

Kata Kunci: Pendeteksian Kehadiran, Artificial Intelligence, Fisherface, Pengenalan Wajah, Politeknik Penerbangan Surabaya

ABSTRACT

DESIGN OF AN AI (ARTIFICIAL INTELLIGENCE) BASED CADET ATTENDANCE DETECTION CAMERA WITH FISHERFACE METHOD IN THE LABORATORY AVIATION POLYTECHNIC SURABAYA

By:
TEGAR KURNIAWAN AL RASYID
30121023

It is necessary to increase discipline for cadets to avoid indiscipline. Starting from the smallest thing, namely the attendance list of cadets, which requires development in attendance by using face detection as an effort to increase cadet discipline. Using face detection makes it easier for the study programme to monitor the attendance of cadets. In addition, face detection ensures from the act of cheating attendance.

This study aims to design and build an Artificial Intelligence (AI) based cadet attendance detection camera system using the Fisherface method. The Fisherface method was chosen because of its ability to recognise faces with high accuracy despite variations in facial expressions, lighting, and distance. The system built consists of several main components, namely the camera as a face image capture tool, and software that implements the Fisherface algorithm for face recognition.

The result of this research is that the attendance process of cadets at the Aviation Polytechnic of Surabaya can be done automatically and in real-time, reducing manual errors and increasing administrative efficiency. This system can be applied in the Surabaya Aviation Polytechnic Laboratory and study programme rooms to simplify the attendance process and ensure the attendance of cadets in a more timely manner. The advantage of monitoring the attendance of cadets from previous research is a longer distance of 2 metres at lumen 239 and 2.5 metres at lumen 279. The speed in detecting and sending data to the database is much faster at 1.40 seconds. The accuracy (similarity rate) in detecting is 92.7% of 69 experimental data.

Keywords: Attendance Detection, Artificial Intelligence, Fisherface, Face Recognition, Surabaya Aviation Polytechnic

MOTTO

JANGAN TERLALU SIBUK UNTUK MEMIKIRKAN HAL YANG ADA DI
DUNIA, KARENA SESUNGGUHNYA DUNIA ADALAH
HUKUMAN BAGI NABI ADAM.



DAFTAR ISI

LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN.....	iii
PERNYATAAN KEASLIAN HAK CIPTA	iv
KATA PENGANTAR.....	v
ABSTRAK	vii
<i>ABSTRACT</i>	viii
MOTTO	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	Error! Bookmark not defined.
1.1 Latar Belakang Masalah.....	Error! Bookmark not defined.
1.2 Rumusan Masalah	Error! Bookmark not defined.
1.3 Batasan Masalah	Error! Bookmark not defined.
1.4 Tujuan Penelitian.....	Error! Bookmark not defined.
1.5 Manfaat Penelitian	Error! Bookmark not defined.
1.6 Sistematika Penulisan	Error! Bookmark not defined.
BAB 2 LANDASAN TEORI.....	Error! Bookmark not defined.
2.1 Bahasa Pemrograman.....	Error! Bookmark not defined.
2.1.1 Python	Error! Bookmark not defined.
2.3 QT.....	Error! Bookmark not defined.
2.4 <i>Interface</i>	Error! Bookmark not defined.
2.5 <i>Fisherface Algoritma</i>	Error! Bookmark not defined.
2.6 Visual Studio Code	Error! Bookmark not defined.
2.7 Kamera atau Webcam.....	Error! Bookmark not defined.
2.8 Komputer Server	Error! Bookmark not defined.
2.9 Pendeteksi Wajah	Error! Bookmark not defined.
2.11 Aplikasi Mobile.....	Error! Bookmark not defined.
2.12 Algoritma Haar Cascade.....	Error! Bookmark not defined.
2.13 Sublime Text	Error! Bookmark not defined.

2.14 DLIB.....	Error! Bookmark not defined.
2.15 Kajian Penelitian yang Relevan	Error! Bookmark not defined.
BAB 3 METODOLOGI PENELITIAN.....	Error! Bookmark not defined.
3.1 Desain Penelitian	Error! Bookmark not defined.
3.2 Perancangan Alat	Error! Bookmark not defined.
3.2.1 Desain Alat	Error! Bookmark not defined.
4.2.2 Cara Kerja Alat	Error! Bookmark not defined.
3.3 Komponen Alat.....	Error! Bookmark not defined.
3.3.1 Perangkat Keras (Hardware)	Error! Bookmark not defined.
3.3.2 Perangkat Lunak (Software)	Error! Bookmark not defined.
3.4 Teknik Pengujian.....	Error! Bookmark not defined.
3.5 Teknik Analisis Data.....	Error! Bookmark not defined.
3.6 Tempat dan Waktu Penelitian	Error! Bookmark not defined.
BAB IV HASIL DAN PEMBAHASAN	Error! Bookmark not defined.
4.1 Hasil Penelitian.....	Error! Bookmark not defined.
4.1.1 Pembuatan Perangkat Lunak.....	Error! Bookmark not defined.
4.1.2 Sinkronisasi Perangkat Keras dan Aplikasi	Error! Bookmark not defined.
4.2 Pembahasan Hasil Penelitian.....	Error! Bookmark not defined.
4.2.1 Pengujian Kamera Face Recognition.....	Error! Bookmark not defined.
4.2.2 Pengujian <i>Enroll</i> dan daftar sampel.....	Error! Bookmark not defined.
4.2.3 Pengujian Kecocokan Pendeteksi Wajah yang Bekerja.....	Error! Bookmark not defined.
4.2.4 Pengujian Jarak Akurasi Pendeteksi dan Pengaruh Cahaya terhadap Objek Wajah.....	Error! Bookmark not defined.
4.2.5 Pengujian rentang waktu pendeteksian terhadap wajah dan pengambilan gambar <i>output</i>	Error! Bookmark not defined.
4.2.5 Implementasi Mock Up dan Pengujian Keseluruhan Pendeteksi Wajah dan Analisa Data	Error! Bookmark not defined.
4.3 Perbandingan Data Sebelum dan Sesudah Adanya Alat ..	Error! Bookmark not defined.
BAB V PENUTUP.....	Error! Bookmark not defined.
5.1 Kesimpulan	Error! Bookmark not defined.
5.2 Saran	Error! Bookmark not defined.
LAMPIRAN	1

DAFTAR GAMBAR

Gambar 2. 1 Bahasa pemrograman Python	Error! Bookmark not defined.
Gambar 2. 2 Open Cv.....	Error! Bookmark not defined.
Gambar 2. 3 Fisherface algoritma.....	Error! Bookmark not defined.
Gambar 2. 4 Vs Code	Error! Bookmark not defined.
Gambar 2. 5 Kamera/Webcam	Error! Bookmark not defined.
Gambar 2. 6 Komputer.....	Error! Bookmark not defined.
Gambar 2. 7 AI.....	Error! Bookmark not defined.
Gambar 2. 8 Haar Cascade.....	Error! Bookmark not defined.
Gambar 2.9 Sublime text.....	Error! Bookmark not defined.
Gambar 2.10 DLIB.....	Error! Bookmark not defined.
Gambar 3. 1 Blog Diagram Rancangan Alat.....	Error! Bookmark not defined.
Gambar 3. 2 Desain Alat	Error! Bookmark not defined.
Gambar 3. 3 Cara Kerja Alat.....	Error! Bookmark not defined.
Gambar 3.4 Kamera sebagai media face recognition.....	Error! Bookmark not defined.
Gambar 3.5 Laptop.....	Error! Bookmark not defined.
Gambar 4.1 File kode sistem dalam satu folder	Error! Bookmark not defined.
Gambar 4.2 Dashboard sublime text.....	Error! Bookmark not defined.
Gambar 4.3 Kode sinkronisasi perangkat lunak dengan kamera..	Error! Bookmark not defined.
Gambar 4.4 Pengujian kamera dengan aplikasi	Error! Bookmark not defined.
Gambar 4.5 Pengujian kamera dengan sistem GUI	Error! Bookmark not defined.
Gambar 4. 6 Proses pendaftaran wajah pengguna	Error! Bookmark not defined.
Gambar 4. 7 Proses training berhasil	Error! Bookmark not defined.
Gambar 4. 8 Database pada GUI.....	Error! Bookmark not defined.
Gambar 4. 9 alur pendeteksian wajah	Error! Bookmark not defined.
Gambar 4. 10 Sistem tidak mendeteksi wajah	Error! Bookmark not defined.
Gambar 4. 11 Sistem mendeteksi wajah yang belum terdaftar ...	Error! Bookmark not defined.
Gambar 4. 12 kecocokan sistem secara real time ..	Error! Bookmark not defined.
Gambar 4. 13 Sistem mendeteksi adanya wajah yang terdaftar..	Error! Bookmark not defined.
Gambar 4. 14 Kode video capture.....	Error! Bookmark not defined.
Gambar 4. 15 Proses pendaftaran pengguna	Error! Bookmark not defined.
Gambar 4. 16 Proses training sistem berhasil	Error! Bookmark not defined.
Gambar 4. 17 Sistem mendeteksi wajah dengan tampilan kotak hijau.....	Error! Bookmark not defined.
Gambar 4. 18 Sistem mengcapture wajah.....	Error! Bookmark not defined.
Gambar 4. 19 Rekap data sistem.....	Error! Bookmark not defined.
Gambar 4. 20 Data excel.....	Error! Bookmark not defined.

DAFTAR TABEL

Tabel 2. 1 Kajian Peneliti yang Relevan	Error! Bookmark not defined.
Tabel 3. 1 Jadwal Kegiatan	Error! Bookmark not defined.
Tabel 4. 1 Pengujian kamera	Error! Bookmark not defined.
Tabel 4. 2 Pendaftaran wajah pengguna.....	Error! Bookmark not defined.
Tabel 4. 3 Pengujian pendeteksi wajah	Error! Bookmark not defined.
Tabel 4. 4 Pengujian rentang jarak pada lumen 239	Error! Bookmark not defined.
Tabel 4. 5 Pengujian rentang jarak pada lumen 279	Error! Bookmark not defined.
Tabel 4. 6 Pengujian kecepatan mendeteksi.....	Error! Bookmark not defined.
Tabel 4. 7 Pengujian kecocokan data output.....	Error! Bookmark not defined.
Tabel 4. 8 Data perbandingan sebelum dan sesudah adanya sistem	Error! Bookmark not defined.
Tabel 4. 9 Uji coba metode eigenface pada tingkat kecerahan ...	Error! Bookmark not defined.
Tabel 4. 10 Hasil ujicoba metode fisherface pada tingkat kecerahan	Error! Bookmark not defined.
Tabel 4. 11 Ujicoba jarak 50cm dan 100cm pada metode eigenface	Error! Bookmark not defined.
Tabel 4. 12 Ujicoba jarak 50 cm dan 100 cm pada metode fisherface.....	Error! Bookmark not defined.
Tabel 4. 13 Perbandingan pada fitur ekspresi wajah metode eigenface dan fisherface	Error! Bookmark not defined.
Tabel 4. 14 Perbandingan metode eigenface dan fisherface	Error! Bookmark not defined.



DAFTAR PUSTAKA

- Al-Fatta, H. (2009). *Rekayasa Sistem Pengenalan Wajah*. Yogyakarta: Andi.
- Alwendi Alwendi, M. M. (2021). *APLIKASI PENGENALAN WAJAH MANUSIA PADA CITRA MENGGUNAKAN METODE FISHERFACE*.
- Ardhito, D. Y. (2023). *Optimasi Face Recognition Untuk Presensi Pegawai*. Surakarta.
- Arlando, R. (2007). *Pengenalan Wajah Menggunakan Metode Fisherface*.
- Bachtiar, A. H. (2022). *Rancang Bangun Dual Keamanan Sistem Rumah Menggunakan Pengenalan Wajah dan Sidik Jari Berbasis IOT*. Jurnal Power Elektronik.
- Bekti Maryuni Susanto, F. E. (t.thn.). *Sistem Keamanan Pintu Berbasis Pengenalan Wajah Menggunakan Metode Fisherface*.
- Bharata, H. K. (2020). *Pemanfaatan Kamera TOF Smartphone untuk Pencatatan Kehadiran Mahasiswa dengan Metode Face Recognition*.
- Darujati, C. (2022). *Pengolahan Citra untuk Pengenalan Wajah (Face Recognition) Menggunakan OpenCV*. Surabaya: Jurnal Syntax Admiration.
- Dedi Leman, M. R. (2021). *PERANCANGAN ABSENSI SISWA MENGGUNAKAN RFID DENGAN KARTU PELAJAR SEBAGAI PENDETEKSI SISWA MASUK DAN KELUAR (STUDI KASUS: SMK TRITECH INFORMATIKA MEDAN)*. Jurnal Computer Science Research Its Development Journal.
- Dedy Irawan, J. d. (2019). *Ruang Kuliah Pintar Pemantau Tingkat Efektivitas Pembelajaran Yang Dapat Mendeteksi Mahasiswa Bosan Dan Mengantuk*. Jurnal Engineering ITN.
- Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis*. (2018). Jurnal Teknologi Elektikal.

- Fauzi, A. (t.thn.). *Identifikasi Dan Pengujian keakuratan Pengenalan* . Jakarta.
- Hendrico, A. (2023). *Sistem presensi otomatis dengan Face Recognition berbasis kamera Smartphone menggunakan metode Haar Cascade*.
- Hidayah, N. (2022). *Perbedaan dan Contoh Absensi Kehadiran Online dan Manual*. bussines & economy.
- Khoswanto, H. (2022). *Sistem Untuk Mengklasifikasikan Emosi Dan Mendeteksi Wajah Pada Pembelajaran Daring*.
- Nata, G. N. (2023). *SISTEM INFORMASI KEHADIRAN SISWA BERBASIS SMS GATEWAY DENGAN QR CODE*. Jurnal Manajemen Informatika.
- Perbawa, A. K. (2024). *Identifikasi Wajah Individu Tunanetra dan Non-Tunanetra dengan menggunakan Fitur Wavelet dan PCA*.
- Putra, R. (2021). *Sistem Monitoring Pegawai Dimasa Adaptasi Kebiasaan Baru Berbasis Face Recognition*.
- Rachmadi, T. (2020). *Mengenal Apa Itu Internet of tugas aThings* . Tiga Ebook.
- Reza Hidayat, S. S. (2018). *PERANCANGAN SISTEM KEAMANAN RUMAH BERBASIS IoT DENGAN NodeMCU ESP8266 MENGGUNAKAN SENSOR PIR HC- SR501 DAN SENSOR SMOKE DETECTOR*.
- Romi Wiryadinata1, U. I. (2021). *Sistem Presensi Menggunakan Algoritme Eigenface*.
- Rosid, J. (2022). *Face Recognition Dengan Metode Haar Cascade dan Facenet*.
- Skripsi, F. T. (2011). *“Analisis Algoritma Eigenface (Pengenalan Wajah) Pada Aplikasi Kehadiran Pengajaran Dosen,. Jakarta: UIN Syarif Hidayatullah*.
- Subita, R. J. (2023). *Pengujian Identifikasi Jumlah Kerumunan Face Recognition Menggunakan Haar Cascade Clasifier*.

Surya, M. A. (2024). *Sitem Keamanan Ruangan Dengan Human Detection Menggunakan Sensor Kamera Berbasis Deep Learning*. Lampung: <https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index>.

SYARIF, M. A. (2024). *IMPLEMENTASI ALGORITMA HAAR CASCADE DAN CONVOLUTIONAL NEURAL NETWORK UNTUK PRESENSI KARYAWAN*.

Zata, N. (2021). *PENCOCOKAN PLAT KENDARAAN DENGAN ALGORITMA HAARCASCADE DAN TEMPLATE MATCHING*.



LAMPIRAN

Lampiran A. SCRIPT KODE PROGRAM

```
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget,
QVBoxLayout, QTabWidget, QComboBox
from PyQt5.QtWidgets import QLabel, QGroupBox, QGridLayout, QLineEdit,
QPushButton, QProgressBar
from PyQt5.QtWidgets import QTableWidgetItem, QTableWidgetItem, QFormLayout,
QMessageBox, QFileDialog
from PyQt5.QtWidgets import QTabWidget, QHBoxLayout, QScrollArea
from PyQt5.QtCore import pyqtSignal, pyqtSlot, Qt, QThread, QRect
from PyQt5.QtGui import QPixmap, QPainter, QFont, QImage, QPen
from PyQt5 import QtGui
import shutil, os, subprocess
import pickle
import cv2
import os, time
import numpy as np
import csv
import pandas as pd

current_directory = os.path.dirname(os.path.abspath(__file__))
from utils import get_image_paths
from utils import face_encodings
from utils import face_rects
from utils import face_encodings
from utils import nb_of_matches

from datetime import datetime
##### 1 baris
import serial.tools.list_ports
import pickle
from thread import *
with open("{}encodings.pickle".format(current_directory), "rb") as f:
    name_encodings_dict = pickle.load(f)
threshold = 0.5

class CardWidget(QWidget):
    def __init__(self, image_path, text, parent=None):
        super().__init__(parent)
        self.image_path = image_path
        self.text = text
        self.setMinimumSize(200, 150) # Set minimum size for the card
```

```
self.setStyleSheet("background-color: white; border: 2px solid black; border-
radius: 10px;")
```

```
def paintEvent(self, event):
```

```
    painter = QPainter(self)
```

```
    painter.setRenderHint(QPainter.Antialiasing)
```

```
    # Draw outline around the card
```

```
    outline_rect = QRect(0, 0, self.width()-1, self.height()-1)
```

```
    painter.setPen(QPen(Qt.black, 2, Qt.SolidLine))
```

```
    painter.drawRect(outline_rect)
```

```
    # Draw image (adjust position and size as needed)
```

```
    pixmap = QPixmap(self.image_path).scaled(100, 100, Qt.KeepAspectRatio,
Qt.SmoothTransformation)
```

```
    painter.drawPixmap(10, 10, pixmap)
```

```
    # Draw text (adjust font and position as needed)
```

```
    painter.setFont(QFont('Arial', 10))
```

```
    text_rect = QRect(120, 10, self.width()-130, self.height()-20)
```

```
    painter.drawText(text_rect, Qt.AlignLeft | Qt.AlignTop, self.text)
```

```
    painter.end()
```

```
class MyApp(QMainWindow):
```

```
    def __init__(self):
```

```
        super().__init__()
```

```
        self.judul = "SISTEM PENGENALAN WAJAH"
```

```
        self.display_width, self.display_height = 640, 480
```

```
        self.captured_images = []
```

```
        self.files = []
```

```
        self.initUI()
```

```
    def initUI(self):
```

```
        # Widget utama
```

```
        self.tabs = QTabWidget()
```

```
        self.setCentralWidget(self.tabs)
```

```
        self.main_tab = QWidget()
```

```
        self.data_tab = QWidget()
```

```
        self.rekap_tab = QWidget()
```

```
        self.tabs.addTab(self.main_tab, "Main")
```

```
        self.tabs.addTab(self.data_tab, "Data")
```

```
        self.tabs.addTab(self.rekap_tab, "Rekap")
```

```
        main_layout = QGridLayout()
```

```
        self.main_tab.setLayout(main_layout)
```

```

data_layout = QGridLayout()
self.data_tab.setLayout(data_layout)

rekap_layout = QVBoxLayout()
self.rekap_tab.setLayout(rekap_layout)
#####

gbRekap = QGroupBox("Rekap")

# Create scroll area
self.scroll_area = QScrollArea()
self.scroll_area.setWidgetResizable(True)

# Create a widget for scroll area contents
self.scroll_content = QWidget(self.scroll_area)
self.scroll_area.setWidget(self.scroll_content)

# Create a grid layout inside scroll content widget with 2 columns
self.rekap_grid_layout = QGridLayout(self.scroll_content)
self.rekap_grid_layout.setSpacing(20)

# Example data for cards (replace with your data)
card_data = [
    {"image": "example.jpg", "text": "Card 1 Text"},
    {"image": "example.jpg", "text": "Card 2 Text"},
    {"image": "example.jpg", "text": "Card 3 Text"},
    {"image": "example.jpg", "text": "Card 4 Text"},
    {"image": "example.jpg", "text": "Card 5 Text"},
    {"image": "example.jpg", "text": "Card 6 Text"},
    {"image": "example.jpg", "text": "Card 7 Text"},
    # Add more cards as needed
]

# Create cards dynamically and add them to the grid layout
# row = 0
# col = 0
# for data in card_data:
#     card_widget = CardWidget(data["image"], data["text"])
#     self.rekap_grid_layout.addWidget(card_widget, row, col)
#     col += 1
#     if col == 2: # Move to the next row after 2 columns
#         col = 0
#         row += 1
rekap_layout.addWidget(self.scroll_area)
#####

```

```

# title_label = QLabel(self.judul)
# title_label.setAlignment(Qt.AlignCenter)
# main_layout.addWidget(title_label, 0, 0, 1, 2)

gbVid = QGroupBox("Camera Stream")
image_layout = QVBoxLayout()
self.vid_label = QLabel()
pixmap = QPixmap("{}no-image.jpg".format(current_directory))
self.vid_label.setPixmap(pixmap)
image_layout.addWidget(self.vid_label, alignment=Qt.AlignCenter)
gbVid.setLayout(image_layout)
main_layout.addWidget(gbVid, 0, 0, 2, 1)

gbDeteksi = QGroupBox("Deteksi")
det_layout = QVBoxLayout()
self.img_label = QLabel()
pixmap = QPixmap("{}no-image.jpg".format(current_directory))
self.img_label.setPixmap(pixmap)
det_layout.addWidget(self.img_label, alignment=Qt.AlignCenter)
gbDeteksi.setLayout(det_layout)
main_layout.addWidget(gbDeteksi, 0, 1, 2, 1)

status_deteksi_box = QGroupBox("Status Deteksi")
status_deteksi_layout = QVBoxLayout()
self.status_deteksi_label = QLabel("-")
status_deteksi_layout.addWidget(self.status_deteksi_label)
status_deteksi_box.setLayout(status_deteksi_layout)
main_layout.addWidget(status_deteksi_box, 0, 2)

hasil_deteksi_box = QGroupBox("Hasil Deteksi")
hasil_deteksi_layout = QVBoxLayout()
self.label_nama = QLabel("Nama : -")
self.label_tanggal = QLabel("Tanggal : -")
self.label_waktu = QLabel("Waktu : -")
self.label_conf = QLabel("Conf : -")
hasil_deteksi_labels = [self.label_nama, self.label_tanggal, self.label_waktu,
self.label_conf]

for label in hasil_deteksi_labels:
    hasil_deteksi_layout.addWidget(label)

hasil_deteksi_box.setLayout(hasil_deteksi_layout)
main_layout.addWidget(hasil_deteksi_box, 1, 2)

## Status Serial GroupBox

```

```

# status_serial_box = QGroupBox("Status Serial")
# status_serial_layout = QVBoxLayout()
# self.status_serial_label = QLabel("-")
# status_serial_layout.addWidget(self.status_serial_label)
# status_serial_box.setLayout(status_serial_layout)
# main_layout.addWidget(status_serial_box, 2, 2)
#####

gbDt = QGroupBox("Data")
dt_layout = QVBoxLayout()
self.table = QTableWidgetItem()
dt_layout.addWidget(self.table)
gbDt.setLayout(dt_layout)
data_layout.addWidget(gbDt, 0, 0, 2, 1)

# Grup Box untuk tampilan kamera
camera_groupbox = QGroupBox("Camera")
camera_layout = QVBoxLayout()
self.vid_label2 = QLabel()
pixmap = QPixmap("{}no-image.jpg".format(current_directory))
pixmap = pixmap.scaled(500, 500, Qt.KeepAspectRatio,
Qt.SmoothTransformation)
self.vid_label2.setPixmap(pixmap)
camera_layout.addWidget(self.vid_label2)
camera_groupbox.setLayout(camera_layout)
data_layout.addWidget(camera_groupbox, 0, 1, 3, 1)

gbForm = QGroupBox("Form")
group_layout = QVBoxLayout()
self.name_label = QLabel("Nama:")
self.name_edit = QLineEdit()
self.kelas_label = QLabel("Kelas:")
self.kelas_edit = QLineEdit()
self.file_label = QLabel("File gambar:")
self.file_edit = QLineEdit()
self.file_edit.setReadOnly(True)
group_layout.addWidget(self.name_label)
group_layout.addWidget(self.name_edit)
group_layout.addWidget(self.kelas_label)
group_layout.addWidget(self.kelas_edit)
group_layout.addWidget(self.file_label)
group_layout.addWidget(self.file_edit)
self.file_button = QPushButton("Pilih Gambar dari File")
self.file_button.clicked.connect(self.select_files)
self.file_button2 = QPushButton("Ambil Gambar dari Kamera")
self.file_button2.clicked.connect(self.take_5pics)

```



```

group_layout2 = QHBoxLayout()
group_layout2.addWidget(self.file_button)
group_layout2.addWidget(self.file_button2)
group_layout.addLayout(group_layout2)
self.save_button = QPushButton("Simpan")
self.save_button.clicked.connect(self.save_data)
group_layout.addWidget(self.save_button)
gbForm.setLayout(group_layout)
data_layout.addWidget(gbForm, 2, 0, 2, 1)

```

```

gbAct = QGroupBox("Training")
act_layout = QVBoxLayout()
self.progress_bar = QProgressBar()
self.progress_bar.setMinimum(0)
self.progress_bar.setMaximum(100)
self.progress_bar.setValue(0)
self.btn_train = QPushButton('Training Model')
self.btn_train.clicked.connect(self.train_model)
# self.btn_database = QPushButton('Database')
# self.btn_database.clicked.connect(self.database)
# act_layout.addWidget(self.btn_database)
act_layout.addWidget(self.progress_bar)
act_layout.addWidget(self.btn_train)
gbAct.setLayout(act_layout)
data_layout.addWidget(gbAct, 4, 0)

```

```

# Grup Box untuk tampilan kamera
preview_groupbox = QGroupBox("Preview")
self.preview_layout = QHBoxLayout()
# self.image_label2 = QLabel()
# pixmap2 = QPixmap("{}no-image.jpg".format(current_directory))
# self.image_label2.setPixmap(pixmap2)
# self.preview_layout.addWidget(self.image_label2)
preview_groupbox.setLayout(self.preview_layout)
data_layout.addWidget(preview_groupbox, 3, 1, 2, 1)

```

```

self.tabs.currentChanged.connect(self.tab_changed)
self.setWindowTitle(self.judul)

```

```

self.image_thread = ImageThread()
self.image_thread.change_pixmap_signal.connect(self.update_image)
self.image_thread.change_pixmap_signal2.connect(self.update_image2)
self.image_thread.change_status_signal.connect(self.update_status_deteksi)
self.image_thread.change_hasil_signal.connect(self.update_status_hasil)
self.image_thread.start()

```

```

# self.serial_thread = SerialThread()
# # self.image_thread.change_pixmap_signal.connect(self.update_image)
# # self.image_thread.change_pixmap_signal2.connect(self.update_image2)
# self.serial_thread.change_status_signal.connect(self.update_status_serial)
# # self.image_thread.change_hasil_signal.connect(self.update_status_hasil)
# self.serial_thread.start()

def update_rekap(self):
    # Baca file CSV
    df = pd.read_csv('{} /output/data.csv'.format(current_directory))

    # Iterasi dan cetak setiap baris
    # for index, row in df.iterrows():
    #     print(row)

    # Atau jika ingin mencetak setiap nilai kolom secara terpisah
    card_data = []
    for index, row in df.iterrows():
        data = {}
        print(f'Baris ke-{index + 1}:')
        for col, value in row.items():
            print(f'{col}: {value}')
            data[col] = value
        print()
        card_data.append(data)

    # print(card_data)

    row = 0
    col = 0
    for data in card_data:
        if data["nama"] != "Unknown":
            image = "{} /output/{_}_{_}_{_}.png".format(current_directory,
data["tanggal"].replace("-", ""), data["datang"].replace(":", ""), data["kelas"],
data["nama"])
            else:
                image = "{} /output/{_}_{_}_{_}.png".format(current_directory,
data["tanggal"].replace("-", ""), data["datang"].replace(":", ""), data["nama"])
                text = "Nama : {}\nKelas : {}\nTanggal : {}\nDatang : {}\nPulang :
{}\nKeterangan : {}".format(data["nama"], data["kelas"], data["tanggal"],
data["datang"], data["pulang"], data["keterangan"])
                print(text)
                card_widget = CardWidget(image, text)
                self.rekap_grid_layout.addWidget(card_widget, row, col)
                col += 1

```

```

        if col == 2: # Move to the next row after 2 columns
            col = 0
            row += 1

    def clear_layout(self):
        while self.rekap_grid_layout.count():
            item = self.rekap_grid_layout.takeAt(0)
            widget = item.widget()
            if widget is not None:
                widget.deleteLater()
            else:
                # if it's a layout, recursively clear it
                layout = item.layout()
                if layout is not None:
                    self.clear_layout(layout)

    def tab_changed(self, index):
        if index == 2:
            self.image_thread.stop()
            self.clear_layout()
            self.update_rekap()
        elif index == 1:
            self.captured_images = []
            self.files = []
            self.image_thread.stop()
            self.update_table()
        elif index == 0:
            self.image_thread = ImageThread()
            self.image_thread.change_pixmap_signal.connect(self.update_image)
            self.image_thread.change_pixmap_signal2.connect(self.update_image2)

self.image_thread.change_status_signal.connect(self.update_status_deteksi)
self.image_thread.change_hasil_signal.connect(self.update_status_hasil)
self.image_thread.start()
QApplication.processEvents()

    def take_5picts(self):
        while self.preview_layout.count():
            item = self.preview_layout.takeAt(0)
            widget = item.widget()
            if widget is not None:
                widget.deleteLater()

self.image_thread2 = ImageThread2()
self.image_thread2.change_pixmap_signal.connect(self.update_imageA)
self.image_thread2.change_pixmap_signal2.connect(self.update_imageA2)

```

```

self.image_thread2.stop_signal.connect(self.stop_thread)
self.image_thread2.start()

def stop_thread(self):
    pixmap = QPixmap("{}no-image.jpg".format(current_directory))
    pixmap = pixmap.scaled(500, 500, Qt.KeepAspectRatio,
Qt.SmoothTransformation)
    self.vid_label2.setPixmap(pixmap)

def select_files(self):
    while self.preview_layout.count():
        item = self.preview_layout.takeAt(0)
        widget = item.widget()
        if widget is not None:
            widget.deleteLater()

    file_dialog = QFileDialog()
    file_dialog.setFileMode(QFileDialog.ExistingFiles)
    file_dialog.setNameFilter("Images (*.png *.jpg *.jpeg *.bmp *.gif)")
    self.files = file_dialog.getOpenFileNames(self, "Pilih Berkas Gambar", "",
"Images (*.png *.jpg *.jpeg *.bmp *.gif)")[0]
    print("Berkas yang dipilih:", self.files[:5]) # Ambil lima berkas pertama
    fileTxt = []
    for f in self.files:
        f = f.split("/")[-1]
        fileTxt.append(f)
    result = ', '.join(fileTxt)
    # print(result)
    self.file_edit.setText(result)

    for f in self.files:
        image = QPixmap(f)
        image = image.scaled(100, 100, Qt.KeepAspectRatio,
Qt.SmoothTransformation)
        image_label = QLabel()
        image_label.setPixmap(image)
        self.preview_layout.addWidget(image_label)

def save_data(self):
    # self.captured_images = []
    # self.files = []
    name = self.name_edit.text()
    kelas = self.kelas_edit.text()
    if name != "" and kelas != "":
        if len(self.files) > 0:
            folder_path = "dataset/train/{}/{_}".format(kelas, name)

```

```

folder_path = os.path.join(current_directory, folder_path)
try:
    # Menghapus folder beserta seluruh isinya
    shutil.rmtree(folder_path)
    print(f'Folder {folder_path} berhasil dihapus.")
except FileNotFoundError:
    print(f'Folder {folder_path} tidak ditemukan.")
except Exception as e:
    print(f'Terjadi kesalahan saat menghapus folder: {e}')

dest = "{} /dataset/train/{}_{}".format(current_directory, kelas, name)
os.mkdir(dest)
for f in self.files:
    print(f)
    shutil.copy(f, dest)
    print("Nama yang disimpan:", name, kelas)
    QMessageBox.information(self, "Information", "Gambar berhasil
disimpan")
    elif len(self.captured_images) > 0:
        # print("kkkkkkk")
        folder_path = "dataset/train/{}".format(name)
        folder_path = os.path.join(current_directory, folder_path)
        try:
            # Menghapus folder beserta seluruh isinya
            shutil.rmtree(folder_path)
            print(f'Folder {folder_path} berhasil dihapus.")
        except FileNotFoundError:
            print(f'Folder {folder_path} tidak ditemukan.")
        except Exception as e:
            print(f'Terjadi kesalahan saat menghapus folder: {e}')

        dest = "{} /dataset/train/{}".format(current_directory, name)
        os.mkdir(dest)
        for i, image in enumerate(self.captured_images):
            # print(i)
            image.save(f'{folder_path}/captured_image_{i}.png')
            QMessageBox.information(self, "Information", "Gambar berhasil
disimpan")
            self.update_table()
            self.name_edit.setText("")
            self.kelas_edit.setText("")
            self.file_edit.setText("")

self.files = []
self.captured_images = []
while self.preview_layout.count():

```



```

        item = self.preview_layout.takeAt(0)
        widget = item.widget()
        if widget is not None:
            widget.deleteLater()

def edit_data(self, row):
    name_item = self.table.item(row, 0)
    if name_item:
        name = name_item.text()
        print(f'Edit data: {name}')
        self.name_edit.setText(name)
        self.file_edit.setText("")

def delete_data(self, row):
    # self.table.removeRow(row)
    name_item = self.table.item(row, 0)
    name_item2 = self.table.item(row, 1)
    if name_item:
        name = name_item.text()
        name2 = name_item2.text()
        print(f'Hapus data: {name} {name2}')
        folder_path = "dataset/train/{_}_{_}".format(name, name2)
        folder_path = os.path.join(current_directory, folder_path)
        try:
            # Menghapus folder beserta seluruh isinya
            shutil.rmtree(folder_path)
            print(f'Folder {folder_path} berhasil dihapus.")
        except FileNotFoundError:
            print(f'Folder {folder_path} tidak ditemukan.")
        except Exception as e:
            print(f'Terjadi kesalahan saat menghapus folder: {e}')
        print("Data dihapus.")
        self.update_table()

def update_table(self):
    self.table.setRowCount(0)
    folder_name = "dataset/train"
    dataset_folder_path = os.path.join(current_directory, folder_name)

    folders_inside_dataset = []

    for name in os.listdir(dataset_folder_path):
        if os.path.isdir(os.path.join(dataset_folder_path, name)):
            if os.listdir(os.path.join(dataset_folder_path, name)):
                folders_inside_dataset.append(name)
            else:

```

```

        shutil.rmtree(os.path.join(dataset_folder_path, name))

self.table.setColumnCount(3)
# self.table.setRowCount(len(folders_inside_dataset))
self.table.setHorizontalHeaderLabels(["Kelas", "Nama", "Aksi"])

current_row = self.table.rowCount()
for name in folders_inside_dataset:
    self.table.insertRow(current_row)
    self.table.setItem(current_row, 0, QTableWidgetItem(name.split("_")[0]))
    self.table.setItem(current_row, 1, QTableWidgetItem(name.split("_")[1]))
    # self.table.setItem(current_row, 1, QTableWidgetItem("")) # Kolom usia
    dikosongkan

    # Tambahkan tombol edit dan hapus
    edit_button = QPushButton("Edit")
    delete_button = QPushButton("Hapus")

    # edit_button.clicked.connect(lambda _, row=current_row:
self.edit_data(row))
    delete_button.clicked.connect(lambda _, row=current_row:
self.delete_data(row))

    cell_widget = QWidget()
    layout = QHBoxLayout()
    # layout.addWidget(edit_button)
    layout.addWidget(delete_button)
    layout.setContentsMargins(0, 0, 0, 0)
    cell_widget.setLayout(layout)

    self.table.setCellWidget(current_row, 2, cell_widget)
    current_row += 1

def database(self):
    print("database")
    self.image_thread.stop()
    time.sleep(10)
    self.image_thread = ImageThread()
    self.image_thread.change_pixmap_signal.connect(self.update_image)
    self.image_thread.change_pixmap_signal2.connect(self.update_image2)
    self.image_thread.change_status_signal.connect(self.update_status_deteksi)
    self.image_thread.change_hasil_signal.connect(self.update_status_hasil)
    self.image_thread.start()
    QApplication.processEvents()

@pyqtSlot(np.ndarray)

```

```

def update_image(self, cv_img):
    qt_img = self.convert_cv_qt(cv_img)
    self.vid_label.setPixmap(qt_img)

@pyqtSlot(np.ndarray)
def update_imageA(self, cv_img):
    qt_img = self.convert_cv_qt(cv_img)
    self.vid_label2.setPixmap(qt_img)

@pyqtSlot(np.ndarray)
def update_image2(self, cv_img):
    global name_encodings_dict
    # try:
    #     qt_img = self.convert_cv_qt2(cv_img)
    #     self.img_label.setPixmap(qt_img)
    # except:
    #     pass
    self.update_status_deteksi("Sedang diproses")

    frame = cv2.imread('{} /temp/pic.png'.format(current_directory))
    encodings, rects = face_encodings(frame)
    # print(rects)
    name = ""
    names = []
    confs = []

    for encoding in encodings:
        counts = {}
        for (name, encodings) in name_encodings_dict.items():
            print(name, len(encodings), len(encoding))
            _, counts[name] = nb_of_matches(encodings, encoding)
        print("-----", counts, "ini")
        if all(count == 0 for count in counts.values()):
            name = "Unknown"
            confs.append(" ")
        else:
            name = max(counts, key=counts.get)
            if counts[name] < threshold:
                name = "Unknown"
                confs.append(" ")
            else:
                confs.append(counts[name])
        names.append(name)
    nama = ""
    conf = ""
    rect = ""

```

```

for n, name in enumerate(names):
    nama = name
    conf = confs[n]
    rect = rects[n]
    try:
        conf = "{:.2f}".format(conf)
    except:
        pass
    print("{} : {} {}".format(name, confs[n], rects[n]))
    left = rects[n].left()
    top = rects[n].top()
    right = rects[n].right()
    bottom = rects[n].bottom()
    # cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
    cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
    cv2.putText(frame, "{} {}".format(nama, conf), (left, top),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 2)

    tgl = datetime.now().strftime("%d-%m-%Y")
    waktu = datetime.now().strftime("%H:%M:%S")

    # print("Current Date:", tgl)
    # print("Current Time:", waktu)
    try:
        # x, y, w, h = rect.left(), rect.top(), rect.width(), rect.height()
        # cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 255, 0), 2)
        # cv2.putText(frame, "{} {}".format(nama, conf), (x, y),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 2)

        # cv2.imwrite("ini.png", frame)
        # self.update_image(frame)
        self.image_thread.update_name(nama, conf)
    except:
        pass
    qt_img = self.convert_cv_qt(frame)
    self.img_label.setPixmap(qt_img)
    self.update_status_hasil([names, str(tgl), str(waktu), confs, rects])

@pyqtSlot(np.ndarray)
def update_imageA2(self, cv_img):
    image = self.convert_cv_qt(cv_img)
    image = image.scaled(100, 100, Qt.KeepAspectRatio,
Qt.SmoothTransformation)
    image_label = QLabel()
    image_label.setPixmap(image)
    self.preview_layout.addWidget(image_label)

```

```

# Simpan gambar ke daftar gambar yang telah di-capture
self.captured_images.append(image)

@pyqtSlot(list)
def update_status_hasil(self, text):
    nama = text[0]
    tgl = text[1]
    waktu = text[2]
    conf = text[3]
    rect = text[4]

    conf = [str(elemen) for elemen in conf]

    # nama = ".join(char for char in nama if char.isalpha())

    self.label_nama.setText( "Nama : \n{}".format('\n\t'.join(nama)))
    self.label_tanggal.setText( "Tanggal : {}".format(tgl))
    self.label_waktu.setText( "Waktu : {}".format(waktu))
    self.label_conf.setText( "Conf : \n{}".format('\n\t'.join(conf)))

    if (len(nama) == 1 and (nama[0] == "" or nama[0] == "-")) == False:
        # Example usage
        file_path = '{}output/data.csv'.format(current_directory)
        print("namamamamamamamamam",nama)
        for idx, na in enumerate(nama):
            try:
                # new_row_data = [tgl, waktu, na.split("_")[0], na.split("_")[1]]
                # self.add_row_to_csv(file_path, new_row_data)
                try:
                    self.add_or_edit_data(na.split("_")[0], na.split("_")[1], tgl,
waktu=waktu)
                except:
                    self.add_or_edit_data("-", na, tgl, waktu=waktu)

            left = rect[idx].left()
            top = rect[idx].top()
            right = rect[idx].right()
            bottom = rect[idx].bottom()

        # Example usage
        source_file = '{}temp/pic.png'.format(current_directory)
        destination_file = 'output/{}_{}_{}.png'.format(tgl, waktu, na)

        destination_file = destination_file.replace(":", "")
        destination_file = destination_file.replace("-", "")

```



```

destination_file = "{}/{ {}".format(current_directory, destination_file)

img = cv2.imread(source_file)
padd = 10
cropped_image = img[top-padd:bottom+padd, left-padd:right+padd]
# cv2.imwrite()
cv2.imwrite(destination_file, cropped_image)
except:
    pass
    # self.copy_and_rename_file(source_file, destination_file)
    # self.serial_thread.send("~{}!{}@{}#n".format(nama, tgl, waktu))

# if nama == "" or nama.lower() == "unknown":
#     self.serial_thread.send("1\n")
# else:
#     self.serial_thread.send("2\n")

def is_before_8_am(self, time_str):
    # Konversi string waktu menjadi objek waktu
    time_obj = datetime.strptime(time_str, "%H:%M:%S").time()

    # Tentukan jam 8 pagi sebagai objek waktu
    eight_am = datetime.strptime('08:00:00', '%H:%M:%S').time()

    # Bandingkan waktu yang diberikan dengan jam 8 pagi
    return time_obj <= eight_am

def is_after_8_am(self, time_str):
    # Konversi string waktu menjadi objek waktu
    time_obj = datetime.strptime(time_str, "%H:%M:%S").time()

    # Tentukan jam 8 pagi sebagai objek waktu
    eight_am = datetime.strptime('08:00:00', '%H:%M:%S').time()

    # Bandingkan waktu yang diberikan dengan jam 8 pagi
    return time_obj > eight_am

def is_after_4_pm(self, time_str):
    # Konversi string waktu menjadi objek waktu
    time_obj = datetime.strptime(time_str, "%H:%M:%S").time()

    # Tentukan jam 8 pagi sebagai objek waktu
    four_pm = datetime.strptime('16:00:00', '%H:%M:%S').time()

    # Bandingkan waktu yang diberikan dengan jam 8 pagi
    return time_obj >= four_pm

```

```

def add_or_edit_data(self, kelas, nama, tanggal, waktu=None):
    filename = '{}'/output/data.csv'.format(current_directory)
    print("filename == ", filename)
    # Cek apakah file sudah ada
    if os.path.exists(filename):
        # Jika file ada, baca file tersebut
        print("ada")
        df = pd.read_csv(filename)
    else:
        print("tidak ada")
        # Jika file tidak ada, buat DataFrame baru dengan kolom yang sesuai
        df = pd.DataFrame(columns=['no', 'kelas', 'nama', 'tanggal', 'datang',
        'pulang', 'keterangan'])

    # Cek apakah ada data dengan nama dan tanggal yang sama
    existing_data = df[(df['nama'] == nama) & (df['tanggal'] == tanggal) &
    (df['kelas'] == kelas)]
    print(existing_data)

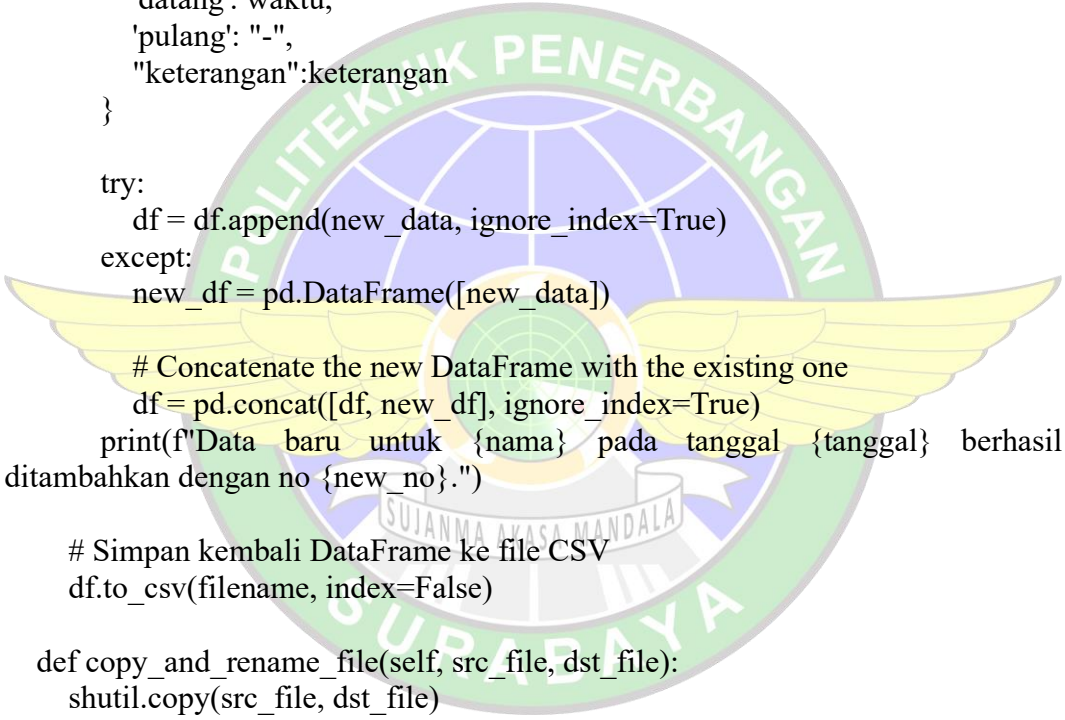
    if not existing_data.empty:
        print("ada data")
        # Jika data ada, update data tersebut
        index = existing_data.index[0]

        if self.is_after_4_pm(waktu):
            keterangan = "Pulang setelah jam 16:00"
            df.at[index, 'pulang'] = waktu
            df.at[index, 'keterangan'] += ", {}".format(keterangan)
        else:
            pass

        # if jam_masuk != None:
        #     df.at[index, 'jam masuk'] = jam_masuk
        # if jam_keluar != None:
        #     df.at[index, 'jam keluar'] = jam_keluar
        print(f'Data untuk {nama} pada tanggal {tanggal} berhasil diupdate.")
    else:
        print("tidak ada data")
        # Jika data tidak ada, tambahkan data baru
        if df.empty:
            new_no = 1
        else:
            new_no = df['no'].max() + 1

        if self.is_before_8_am(waktu):

```



```

        keterangan = "Datang sebelum jam 08:00"
    elif self.is_after_4_pm(waktu):
        keterangan = "Pulang setelah jam 16:00"
    elif self.is_after_8_am(waktu):
        keterangan = "Datang terlambat"

    print(keterangan)

    new_data = {
        'no': new_no,
        'kelas': kelas,
        'nama': nama,
        'tanggal': tanggal,
        'datang': waktu,
        'pulang': "-",
        "keterangan": keterangan
    }

    try:
        df = df.append(new_data, ignore_index=True)
    except:
        new_df = pd.DataFrame([new_data])

        # Concatenate the new DataFrame with the existing one
        df = pd.concat([df, new_df], ignore_index=True)
        print(f>Data baru untuk {nama} pada tanggal {tanggal} berhasil
        ditambahkan dengan no {new_no}.")

        # Simpan kembali DataFrame ke file CSV
        df.to_csv(filename, index=False)

    def copy_and_rename_file(self, src_file, dst_file):
        shutil.copy(src_file, dst_file)

    def add_row_to_csv(self, file_path, data):
        with open(file_path, 'a', newline='') as csvfile:
            writer = csv.writer(csvfile)
            writer.writerow(data)

    @pyqtSlot(str)
    def update_status_deteksi(self, status):
        self.status_deteksi_label.setText(status)

    # @pyqtSlot(str)
    # def update_status_serial(self, status):
    #     self.status_serial_label.setText(status)

```

```

def convert_cv_qt(self, cv_img):
    rgb_image = cv2.cvtColor(cv_img, cv2.COLOR_BGR2RGB)
    h, w, ch = rgb_image.shape
    bytes_per_line = ch * w
    convert_to_Qt_format = QImage(rgb_image.data, w, h, bytes_per_line,
QImage.Format_RGB888)
    p = convert_to_Qt_format.scaled(int(self.display_width/2),
int(self.display_height/2), Qt.KeepAspectRatio)
    return QPixmap.fromImage(p)

def convert_cv_qt2(self, cv_img):
    rgb_image = cv2.cvtColor(cv_img, cv2.COLOR_BGR2RGB)
    h, w, ch = rgb_image.shape
    bytes_per_line = ch * w
    convert_to_Qt_format = QImage(rgb_image.data, w, h, bytes_per_line,
QImage.Format_RGB888)
    p = convert_to_Qt_format.scaled(int(self.display_width/4),
int(self.display_height/4), Qt.KeepAspectRatio)
    return QPixmap.fromImage(p)

def train_model(self):
    global name_encodings_dict
    self.image_thread.stop()
    root_dir = "{}dataset/train".format(current_directory)
    class_names = os.listdir(root_dir)

    image_paths = get_image_paths(root_dir, class_names)
    name_encodings_dict = {}

    nb_current_image = 1

    for n, image_path in enumerate(image_paths):
        try:
            print(f"Image processed {nb_current_image}/{len(image_paths)}")
            print(image_path)
            image = cv2.imread(image_path)
            encodings, _ = face_encodings(image)
            name = image_path.split(os.path.sep)[-2]
            e = name_encodings_dict.get(name, [])
            e.extend(encodings)
            name_encodings_dict[name] = e
            nb_current_image += 1

        progres = int(n/len(image_paths) * 100)
        self.progress_bar.setValue(progres)

```

```

        QApplication.processEvents()
    except:
        print("Fail {}".format(image_path))

    with open("{}encodings.pickle".format(current_directory), "wb") as f:
        pickle.dump(name_encodings_dict, f)

    with open("{}encodings.pickle".format(current_directory), "rb") as f:
        name_encodings_dict = pickle.load(f)
    self.progress_bar.setValue(100)
    QMessageBox.information(self, "Information", "Training Berhasil")

    # self.image_thread = ImageThread()
    # self.image_thread.change_pixmap_signal.connect(self.update_image)
    # self.image_thread.change_pixmap_signal2.connect(self.update_image2)
    # self.image_thread.change_status_signal.connect(self.update_status_deteksi)
    # self.image_thread.change_hasil_signal.connect(self.update_status_hasil)
    # self.image_thread.start()
    QApplication.processEvents()

    # def closeEvent(self, event):
    #     self.conn.close()
    #     # self.threadCamera1.stop()
    #     # self.threadCamera2.stop()
    #     # event.accept()

if __name__ == "__main__":
    app = QApplication(sys.argv)
    my_gui = MyApp()
    # my_gui.showMaximized()
    my_gui.show()
    sys.exit(app.exec_())

```


Lampiran B : Perhitungan Intensitas Cahaya



DAFTAR RIWAYAT HIDUP



TEGAR KURNIAWAN AL RASYID, lahir di Surabaya pada tanggal 26 Juli 2003, anak kedua dari dua bersaudara, dari pasangan Alik Tri Djasmoko dan Ibu Indra Wahyu Pujiningsih. Mempunyai satu saudara kandung, Mayang Alindra Nur Kholifah. Beragama Islam. Bertempat tinggal di JL.Gerbong 19B No 3, Surabaya.

Dengan Pendidikan formal yang pernah diikuti sebagai berikut :

1. SD Negeri KAPASARI VIII Surabaya Lulus tahun 2013
2. SMP Negeri 41 Surabaya Lulus tahun 2016
3. SMA Negeri 19 Surabaya Lulus tahun 2019

Pada bulan September 2020 diterima sebagai Taruna di Politeknik Penerbangan Surabaya pada Program Studi Diploma III Teknik Listrik Bandar Udara Angkatan XVI. Melaksanakan On The Job Training 1 di UPBU Sangia Nibandera, Kolaka dan On The Job Training 2 di Bandar Udara Husein Sastranegara, Bandung. Telah melaksanakan Proyek Akhir sebagai syarat kelulusan dalam Pendidikan di Politeknik Penerbangan Surabaya.