

Perancangan Aplikasi Sistem Pengenalan Wajah Dengan Metode Convolutional Neural Network (CNN) Untuk Pencatatan Kehadiran Karyawan

Efanntyo ¹⁾

Teknik Elektro, Fakultas Sains dan Teknologi, Universitas Pelita Harapan
01035190015@student.uph.edu

Aditya Rama Mitra ²⁾

Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pelita Harapan
aditya.mitra@uph.edu

ABSTRAK

Dalam situasi menghadapi pandemi novel coronavirus (COVID-19), pemakaian masker wajah dan menjaga jarak antar sesama menjadi suatu kewajiban dalam beraktivitas. Hal ini diperkuat dengan saran yang diberikan oleh badan dunia yang berkecimpung untuk urusan kesehatan, yaitu WHO (*World Health Organization*), agar penggunaan masker dilakukan secara kontinu selama beraktivitas dengan menjaga jarak antar individu minimal 1 meter. Dengan mengacu pada hal tersebut, terjadi perubahan yang mencakup juga sistem pencatatan kehadiran (presensi) secara khusus karyawan perusahaan X. Bila sebelumnya sistem presensi perusahaan X menggunakan sidik jari (*fingerprint*) kini beralih ke sistem presensi berbasis pengenalan wajah (*face recognition*) dengan memanfaatkan salah satu pendekatan dalam *deep learning*, yaitu metode *Convolutional Neural Network* (CNN) untuk identifikasi wajah seseorang. Keunggulan sistem ini adalah memungkinkan orang bisa menjaga jarak saat melakukan presensi. Berdasarkan hasil observasi sistem presensi yang berjalan dan hasil studi kepustakaan dilakukan perancangan dan pengembangan aplikasi pencatatan kehadiran berbasis pengenalan wajah mengikuti RAD (*Rapid Application Development*). Bahasa pemrograman yang digunakan adalah bahasa Python dengan model FaceNet yang dapat digunakan untuk pengembangan sistem pengenalan wajah disediakan TensorFlow. Aplikasi sistem pengenalan wajah yang telah dibangun memiliki tingkat akurasi pengenalan wajah yang dipengaruhi jarak antara kamera dengan wajah karyawan pada tingkat pencahayaan 24 lux. Pengukuran pada jarak 30 cm memberi hasil rata-rata tingkat akurasi sebesar 81%; sementara pengukuran pada jarak 60 cm, 90 cm, dan 120 cm memberikan hasil rata-rata untuk masing-masing jarak secara berurutan adalah 81%, 72%, 69%. Sebagai kesimpulan, aplikasi yang dibangun memungkinkan terhindarinya kontak langsung antara karyawan dengan perangkat presensi. Notifikasi bagi karyawan yang belum melakukan presensi telah ditunjukkan berfungsi sebagaimana yang diharapkan.

Kata Kunci : *Pengenalan wajah, deep learning, Convolutional Neural Network, Python, TensorFlow, FaceNet*

I. PENDAHULUAN

Perkembangan teknologi kecerdasan buatan (*artificial intelligence*) tampak kian cepat dan menarik perhatian berbagai pihak yang datang dari berbagai bidang yang kian beragam untuk mengikuti dan mendalaminya. Salah satu kajian kecerdasan buatan yang terus mengalami perkembangan secara signifikan adalah *deep learning* yang diterapkan untuk mengenali wajah manusia. Dalam prakteknya, sistem pengenalan wajah banyak dimanfaatkan dalam kebutuhan sehari-hari karena memiliki tingkat efektifitas dan efisiensi lebih baik bila dibandingkan dengan sistem konvensional lainnya seperti pengenalan biometrik sidik jari (*fingerprint recognition*). Setidaknya untuk konteks pengenalan secara waktu nyata, maka pengenalan wajah menawarkan satu keunggulan [1].

Salah satu keunggulan yang dimiliki oleh sistem pengenalan wajah, terlihat nyata pada sistem presensi yang banyak diterapkan oleh berbagai organisasi ataupun instansi. Hal ini diindikasikan dari proses pencatatan kehadiran yang dapat dilakukan tanpa bersentuhan langsung dengan perangkat presensi pada jarak tertentu. Selaras dengan saran yang diberikan oleh WHO mengenai tata cara untuk mencegah penularan dan penyebaran dari *novel coronavirus* (COVID-19), hal ini menjadi salah satu opsi untuk dapat diimplementasikan di perusahaan X dengan tujuan mencegah terjadinya penularan melalui kontak langsung ataupun potensi yang menyebabkan terjadinya kerumunan pada saat melakukan presensi kehadiran.

Pada penelitian ini dilakukan studi mengenai proses otentikasi berbasis pengenalan wajah (*face recognition*) dengan menerapkan salah satu metode dalam domain *deep learning*, yang disebut *Convolutional Neural Network* (CNN) melalui media kamera yang mampu mengenali wajah karyawan dengan jarak mencapai 1 meter dari kamera. Perancangan aplikasi sistem pengenalan wajah diharapkan dapat bermanfaat bagi karyawan untuk menghindari kontak langsung antar karyawan dengan perangkat saat melakukan presensi sehingga dapat mencegah karyawan terpapar dari virus novel coronavirus (COVID-19). Sedangkan bagi perusahaan dapat mengurangi potensi penyebaran virus novel coronavirus (COVID-19) selain hal tersebut pihak perusahaan dapat melihat kehadiran dari karyawan dalam satu aplikasi terintegrasi yang mampu memberikan notifikasi ketidakhadiran kepada karyawan yang tidak melakukan presensi.

Adapun berbagai studi literatur yang dilakukan sebagai panduan mendasar dalam penelitian ini, di antaranya:

Mundial [2] dalam artikelnya yang berjudul "*Towards Facial Recognition Problem in COVID-19 Pandemic*" memuat hasil investigasinya mengenai proses penerapan metode CNN untuk dapat mengenali wajah pengguna dengan masker. Sebagai hasilnya, dengan data terlatih pengenalan wajah yang dilakukan memberi akurasi mendekati 99% pada data set LFW (*Labelled Faces in the Wild*). Hasil ini mengkonfirmasi bahwa model dapat dilatih secara akurat pada data set berbeda yang juga merupakan koleksi tangkapan wajah manusia sesuai prinsip "*in the wild*", yaitu VGGFace2. Fakta ini dijadikan dasar algoritma sebagai pilihan dari berbagai metode *deep learning* yang akan digunakan sebagai salah satu komponen utama dalam penelitian [2].

Bhatti [3] dalam artikelnya yang berjudul "*Smart Attendance Management System Using Face Recognition*", menuliskan hasil penelitiannya mengenai penggunaan sistem presensi dalam konteks pembelajaran yang berfungsi untuk mencatat kehadiran siswa pertopik pembelajaran. Pencatatan dokumen akan disimpan dalam bentuk excel dokumen yang secara otomatis akan mencatat detail informasi seperti identitas, tanggal, waktu, topik pelajaran yang diambil dan status kehadiran. Dalam penelitian yang dilakukan, hal ini menjadi ide yang akan direalisasi dalam perekaman kehadiran karyawan saat mereka melintas masuk melewati kamera pengenalan [3]. Ide ini akan dibandingkan dengan objek yang berhenti sejenak di depan kamera pengenalan.

Hartanto [4] dalam artikelnya yang berjudul "*Face recognition for attendance system detection*" mengkaji proses pengembangan sistem pengenalan wajah sebagai sebuah sistem presensi. Fokus penelitiannya ada pada investigasi lebih lanjut dari jarak dan kemampuan membaca kamera. Sebagai hasilnya, tingkat akurasi pengenalan wajah yang dicapai adalah 98,2% pada jarak kamera dan objek sejauh 40 cm dengan pencahayaan 24 lux. Selain hal tersebut analisa jarak juga dilakukan hingga jarak 90 cm yang memiliki variasi dalam akurasinya. Temuan-temuan ini juga akan dijajaki dalam penelitian yang akan dilakukan untuk mengetahui kemampuan sistem pengenalan wajah baik untuk jarak minimal dan terjauh yang dapat terbaca oleh sistem [4].

Wang [5] dalam artikelnya yang berjudul "*Masked face recognition dataset and application*" membahas seputar penelitian lebih lanjut mengenai kemampuan sistem pengenalan wajah dalam mengidentifikasi wajah yang menggunakan

berbagai jenis masker berbeda dengan memanfaatkan berbagai jenis dataset (*masked face detection dataset, real-world masked face recognition dataset and simulated masked face recognition dataset*). Penelitian yang dilakukan akan menjadi dasar dalam penelitian untuk meningkatkan akurasi sistem dalam mengidentifikasi pengguna dengan berbagai jenis masker [5].

Anwar [6] dalam artikelnya yang berjudul “*Masked Face Recognition for Secure Authentication*” membahas tentang sistem yang dapat memberikan masker ke berbagai wajah pengguna dengan memiliki tingkat akurasi sampai dengan 38% pada rasio positif dalam mengidentifikasi wajah yang menggunakan masker atau tanpa masker dalam sistem FaceNet. Dalam penelitian, akan memanfaatkan sistem yang telah dibuat untuk melakukan pemberian masker pada wajah sample data sebagai bahan training dengan tujuan dapat meningkatkan akurasi dari sistem [6].

Islami [7] dalam artikelnya yang berjudul “*Pengembangan Aplikasi Manajemen Pelatihan Laboratorium Software Engineering di Fakultas Teknik Sistem Komputer*” membahas tentang penerapan sistem aplikasi pelatihan yang memberikan informasi lengkap tentang pelatihan dan pendaftaran di laboratorium *software engineering*. Pada artikel ini menjelaskan mengenai konsep pemrograman *multiple level user* yang bertujuan memudahkan untuk pengelolaan aplikasi. Pada artikel ini untuk menguji kapabilitas program yang telah dibuat, dilakukan pengujian *blackbox* yang contohnya dapat dilihat pada Tabel 1 [7]. Pengujian *blackbox* ini akan digunakan sebagai salah satu bentuk pengujian fungsionalitas sistem dalam penelitian yang dilakukan.

Tabel 1. Pengujian *blackbox* [7]

Nama Pengujian	Bentuk Pengujian	Hasil yang diharapkan	Hasil Pengujian
----------------	------------------	-----------------------	-----------------

Dengan bertujuan mendapati pemahaman dasar terhadap perancangan aplikasi, dijabarkan juga landasan teori yang digunakan pada penelitian, di antaranya:

A. Python

Python adalah salah satu bahasa pemrograman yang telah dibuat oleh Guido van Rossum yang telah rilis pertama kali pada tahun 1991. *Python* hingga saat ini dikenal dengan bahasa pemrograman tingkat tinggi yang memiliki arti *syntax* yang digunakan merupakan bahasa yang mudah dimengerti dengan perintah yang menggunakan bahasa Inggris. Pada umumnya,

logika perintah yang dituliskan dengan bahasa *python* akan lebih mudah digunakan dibandingkan dengan bahasa pemrograman tingkat rendah seperti halnya bahasa *assembly* [8].

Python banyak digunakan dan menjadi salah satu pilihan bahasa pemrograman yang populer, khususnya dalam mengembangkan program yang berhubungan dengan *machine learning*. Hal tersebut dikarenakan *Python* memiliki komunitas yang besar dengan banyaknya pilihan fungsi pustaka yang memberikan dukungan terhadap perancangan program komputasional kompleks kepada para penggunanya. Beberapa contoh fungsi pustaka yang populer untuk *machine learning*, di antaranya *Scikit-learn, NumPy (Numeric Python), Matplotlib*.

B. Anaconda

Anaconda merupakan salah satu aplikasi yang berfungsi sebagai distribusi bahasa pemrograman *Python* dan *R* yang memiliki sifat *open source*. *Python* banyak dimanfaatkan untuk berbagai perhitungan ilmiah, yang di dalamnya berupa *machine learning*, pengolahan data dengan ukuran besar, analisis prediksi, dan lain sebagainya. *Anaconda* memiliki tujuan untuk dapat menyederhanakan berbagai proses manajemen *package* ataupun *deployment*. *Anaconda* memiliki jumlah distribusi lebih dari 1500 *package* yang populer dan dapat diakses oleh berbagai platform sistem operasi seperti halnya *Windows, Linux, dan MacOS* [9].

Dengan memasang *Anaconda* ke sistem operasi, maka dapat menjalankan beberapa *package* yang secara umum dapat langsung digunakan. Sebagai gambaran, dengan melakukan instalasi *Anaconda*, maka pengguna sudah dapat menjalankan berbagai jenis *package* seperti halnya *Pandas, scikit-learn, numpy*, dan lain sebagainya. Salah satu keuntungan yang lainnya, dengan *Anaconda* pengguna dapat memilih versi bahasa pemrograman *python* yang ingin dijalankan.

C. Database

Database merupakan kumpulan dari berbagai jenis data yang berisikan informasi dari satu atau beberapa *object* yang saling berhubungan dan berinteraksi. Relasi antar setiap data dapat dilihat dengan kunci dari setiap kolom. Pengolahan *database* banyak dilakukan untuk menyusun, mengatur, memperbaharui data yang dapat ditampilkan dalam bentuk laporan ataupun pengolahan data internal lainnya sehingga informasi dapat ditampilkan secara rapi [10].

Database diperlukan untuk mengatasi kendala yang umumnya sering terjadi ketika melakukan pengolahan data, seperti [10]:

1. Redudansi dan Inkonsistensi data

Dengan melakukan penyimpanan data yang sama pada satu tempat penyimpanan yang sama akan menyebabkan pemborosan memori penyimpanan. Sedangkan penyimpanan data yang berulang dapat menyebabkan terjadinya ketidak konsistenan (inkonsistensi).

2. Keamanan akses data

Dengan memanfaatkan manajemen *database*, maka dapat meningkatkan sistem keamanan akses data, sebagai contoh akses *database* keuangan hanya dapat diakses oleh pihak keuangan.

3. Kesulitan dalam mengakses data

Dengan memanfaatkan *database*, dapat melakukan pengambilan data secara langsung dengan perintah yang dimasukkan ke dalam program aplikasi.

4. Isolasi data untuk standarisasi

Pada penyimpanan *database*, format data penyimpanan diseragamkan sehingga akan mempermudah ketika ingin mengambil dan menyimpan data tersebut dalam format lain.

D. Machine Learning

Machine learning merupakan salah satu bentuk pendekatan terhadap kecerdasan buatan yang banyak dimanfaatkan untuk menggantikan kinerja dari manusia dalam melakukan suatu proses kerja. Dalam prosesnya, *machine learning* akan melakukan proses pelatihan, pembelajaran dan pelatihan. Dua karakteristik utama dari *machine learning* adalah klasifikasi dan prediksi. Klasifikasi merupakan metode yang untuk memilah dan mengelompokkan obyek sesuai dengan ciri-ciri tertentu, sedangkan Prediksi merupakan proses untuk menentukan keluaran sesuai dengan data yang telah dilatih [11].

E. Neural Network

Neural Network yang lebih dikenal dengan jaringan saraf tiruan adalah salah satu jenis teknik dalam *machine learning* yang berperan untuk menirukan saraf manusia sebagai bagian dari proses. Pada jaringan saraf tiruan tersusun atas bagian lapis masukan dan lapis keluaran. Pada setiap lapis akan tersusun oleh beberapa neuron yang memiliki fungsi sebagai penentu atau hasil dari keluaran unit. Dalam prosesnya dapat juga ditambahkan lapisan tersembunyi guna meningkatkan kemampuan dari jaringan saraf tiruan. Semakin banyak data pelatihan yang dikumpulkan dan diproses, maka hasilnya juga semakin baik [11].

F. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan satu jenis algoritma dari *Deep Learning* (DL) dengan pengembangan dari *Multi Layer Perception* (MLP) yang umumnya digunakan pada jenis data gambar. CNN banyak digunakan untuk mendeteksi dan mengenali obyek pada suatu gambar. Sebagai contoh penerapan CNN digunakan pada klasifikasi gambar medis, yang di dalamnya terdapat tiga teknik utama yang di antaranya [12]:

1. Sesi pelatihan, pada tahap ini dilakukan pelatihan “CNN dari *scratch*”.

2. Menggunakan fitur “*off-the shelf CNN*” sebagai saluran informasi pelengkap untuk fitur *hand-crafted* guna rontgen dada dan identifikasi nodul pada paru CT.

3. Melakukan pra-pelatihan *unsupervised* pada gambar alami atau gambar medis untuk penyesuaian halus terhadap gambar target medis menggunakan CNN.

Pada prosesnya, CNN menerapkan proses konvolusi yang menggerakkan suatu kernel konvolusi (*filter*) dengan ukuran yang ditentukan pada suatu gambar, dengan demikian akan mendapatkan informasi-informasi yang memiliki representatif baru berdasar pada hasil perkalian gambar dengan filter yang digunakan. Berikut ini merupakan tahapan dalam proses CNN, di antaranya [13]:

1. Proses untuk memecah gambar menjadi ukuran gambar yang saling menimpa dengan ukuran lebih kecil.

2. Gambar yang telah dipecah sebelumnya, dimasukkan ke *small neural network*, hal ini akan dijadikan masukan guna menghasilkan suatu representasi dari fitur.

3. Menyimpan hasil proses yang sebelumnya dari ukuran gambar yang kecil ke bentuk array yang baru.

4. *Downsampling* atau yang dikenal dengan *max pooling*, berperan untuk mengambil nilai *pixel* terbesar dari *pooling kernel*. Dengan demikian akan meringankan parameter proses yang ada dan hanya mengambil informasi terpentingnya.

5. Melakukan proses prediksi, dengan *array* yang merupakan kumpulan angka dimasukkan ke dalam jaringan saraf lain.

G. FaceNet

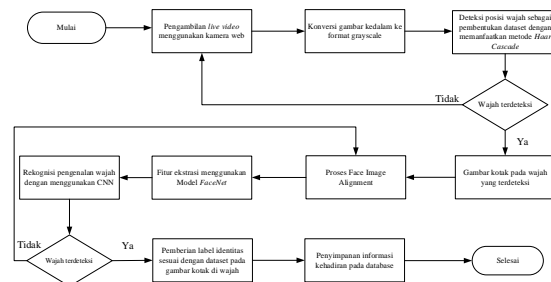
FaceNet merupakan salah satu model dengan jenis metode ekstraksi wajah yang memanfaatkan *Deep Convolutional Neural Network*. *FaceNet* dipertimbangkan sebagai model yang terbaik dikembangkan oleh *Google*. *FaceNet* menggunakan modul awal dalam blok untuk mengurangi jumlah parameter yang dapat dilatih.

Pada model ini mengambil gambar jenis RGB dengan ukuran pixel 160 x 160 dan menghasilkan penyisipan ukuran 128 bit untuk suatu gambar. Pada kumpulan data Labeled Faces in the Wild (LFW), jenis metode FaceNet memiliki tingkat akurasi sebesar 99,63% [14].

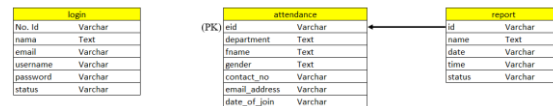
II. METODE PENELITIAN

Metode penelitian yang dilakukan mencakup observasi lapangan untuk mengidentifikasi dan merumuskan masalah bersama dengan pihak perusahaan. Setelah melakukan observasi, dilanjutkan dengan melakukan studi kepustakaan untuk mencari data terkait yang akan digunakan dalam salah satu tahapan penelitian dan mempelajari temuan-temuan dari berbagai artikel ilmiah dengan topik penelitian yang terkini. Selanjutnya menerapkan metode *Rapid Application Development* (RAD) untuk mengembangkan aplikasi yang sesuai dengan kebutuhan perusahaan. Berdasarkan data dan hasil studi yang ada dilakukan perancangan secara eksperimental untuk menyusun program aplikasi sistem pengenalan wajah. Secara khusus dilakukan studi untuk menerapkan salah satu metode dalam pembelajaran mesin yaitu *Convolutional Neural Network* (CNN). Pengujian program dan verifikasi data dilakukan dengan metode kotak hitam (*blackbox*) untuk memastikan keseluruhan aplikasi berfungsi dengan optimal.

Aplikasi sistem pengenalan wajah dibangun menggunakan perangkat keras berupa laptop dengan spesifikasi *Windows 10 Home 64-bit, Processor Intel® Core i7-8750H, RAM 16 GB, VGA 4 GB, Integrated camera (0,92 megapixel and 1280 x 720 (HD))*. Adapun perangkat lunak yang digunakan seperti *anaconda navigator, visual studio code, dan XAMPP*. *Anaconda navigator* menyediakan berbagai fungsi Pustaka seperti *keras* versi 2.4.3, *numpy* versi 1.19.5, *opencv-python* versi 4.2.0.34, *pandas* versi 1.1.5, *tensorflow* versi 2.4.0, dan sebagainya. *Visual studio code* menjadi platform dalam pengkodean program. *XAMPP* menyediakan fasilitas penyimpanan basis data dalam bentuk *MySQL*. Pada Gambar 1 dapat dilihat diagram alir perancangan dari aplikasi sistem pengenalan wajah. Sedangkan aplikasi sistem pengenalan wajah yang dirancang menggunakan basis data sebagai media penyimpanannya yang dapat dilihat pada Gambar 2.



Gambar 1. Diagram alir perancangan



Gambar 2. Struktur basis data

III. HASIL DAN PEMBAHASAN

Perancangan sistem pengenalan wajah menggunakan berbagai fungsi pustaka yang di antaranya:

1. *tkinter* berfungsi untuk mengatur tampilan dari aplikasi.
2. *cv2* berfungsi untuk melakukan proses pelatihan sistem dan pemrosesan gambar.
3. *os* berfungsi untuk membuat, menghapus ataupun mengubah direktori.
4. *pickle* berfungsi untuk mengubah objek Python menjadi suatu file/basis data.
5. *numpy* berfungsi untuk menangani bilangan kompleks atau fungsi operasi aritmatika.
6. *tensorflow* berfungsi untuk melakukan komputasi numerik cepat, dalam aplikasi yang dirancang diterapkan pada model FaceNet.
7. *pandas* berfungsi untuk mengatur baris dan kolom pada suatu bingkai data.
8. *PIL* berfungsi untuk membuka, memanipulasi dan menyimpan gambar dalam berbagai format yang berbeda.

Fungsi pustaka yang digunakan pada perancangan aplikasi sistem pengenalan wajah dapat dilihat pada Gambar 3.

```
#####Importing Tkinter modules and Libraries #####

from tkinter import *
import cv2
import os
from tkinter.ttk import Combobox, Treeview, Scrollbar, Progressbar
from PIL import Image, ImageTk
import pymysql
import csv
from tkinter import messagebox, Message
import numpy as np
from os import listdir
from tkinter import simpledialog
import time
import random
import pandas as pd
from tkinter import filedialog
import gtts
from gtts import gTTS
from extract_embeddings import Extract_Embeddings
import pickle
from training import Training
import os
from datetime import datetime
from statistics import mode
from mark_attendance import Mark_Attendance
import sys
import webbrowser
import re
import shutil
from apscheduler.schedulers.background import BackgroundScheduler
import event_scheduler
import json
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import model_from_json
import tensorflow as tf
from PIL import Image, ImageTk
```

Gambar 3. Fungsi pustaka

Perancangan tampilan aplikasi memanfaatkan fungsi pustaka *tkinter*. Fungsi pustaka *tkinter* dapat melakukan pengaturan terhadap tombol akses aplikasi, jenis huruf, ukuran huruf, posisi tulisan, posisi gambar, dan warna layar aplikasi. Dengan fungsi pustaka *tkinter* memberikan kemudahan dalam melakukan perubahan tampilan sesuai dengan kebutuhan. Contoh kode program tampilan aplikasi dapat dilihat pada Gambar 4.

```
##### Admin login page form #####

bg_icon = PhotoImage(file = "Photos/background9.png", master =
face)

background_image = Label(face, image = bg_icon)
background_image.pack()

title = Label(face, text = "Akses Masuk", font = ("times new
roman", 30, "bold"), bg = "blue", fg = "white", bd = 7, relief =
GROOVE)
title.place(x = 0, y = 0, relwidth = 1)

clock = Label(face, font = ("times", 20, "bold"), bg = "blue", fg =
"white", relief = GROOVE)
clock.place(x = 1250, y = 100)
tick()

login_frame = Frame(face, bg = "white", relief = GROOVE)
login_frame.place(x = 550, y = 200)
logo_icon = PhotoImage(file = "Photos/logo.png", master =
login_frame)
logo_image = Label(login_frame, image = logo_icon, bd = 0).grid(
row = 0, columnspan = 3, pady = 40, padx = 40)
user_icon = PhotoImage(file = "Photos/user.png", master =
login_frame)
password_icon = PhotoImage(file = "Photos/password.png", master =
login_frame)
user_label = Label(login_frame, text = "Nama Pengguna", image =
user_icon, bg = "white", compound = LEFT, font = ("times new
roman", 15, "bold")).grid(row = 1, column = 0, padx = 30, pady
= 5)
user_entry = Entry(login_frame, font = ("times new roman", 15,
"bold"), relief = GROOVE, textvariable = username_var, bg =
"lightgray").grid(row = 1, column = 1, padx = 10, pady = 5)
password_label = Label(login_frame, text = "Kata Sandi",
image = password_icon, bg = "white", compound = LEFT, font =
("times new roman", 15, "bold")).grid(row = 2, column = 0, padx =
30, pady = 5)
password_entry = Entry(login_frame, show = "*", font = ("times new
roman", 15, "bold"), relief = GROOVE, textvariable = password_var,
bg = "lightgray").grid(row = 2, column = 1, padx = 20, pady = 5)
submit_btn = Button(login_frame, text = "Masuk", cursor = "hand2",
width = 10, activebackground = "blue", activeforeground = "white",
command = login, font = ("times new roman", 20, "bold"), relief =
GROOVE, bg = "blue", fg = "white").grid(row = 3, column = 1, pady
= 25, padx = 25)
face.mainloop()
```

Gambar 4. Kode program tampilan aplikasi

Penambahan baris baru data ke tabel basis data memanfaatkan kode perintah *“insert into”*

pada *MySQL*. Hal ini diterapkan pada penambahan karyawan baru di menu manajemen karyawan. Kode program penambahan data ke tabel basis data terlihat seperti pada Gambar 5.

```
cur = conn.cursor()
cur.execute("insert into attendance(id, department, frame, gender, contact_no, email_address, date_of_join) values (%s,%s,%s,%s,%s,%s)", (
id_var.get(),
fname_var.get(),
gender_var.get(),
contact_no_var.get(),
address_var.get(),
noj_var.get()
))

conn.commit()
display()
conn.close()

cur = conn.cursor()
cur.execute("select eid from attendance where fname=%s", (name))
name = cur.fetchone()
(eid) = name
os.remove(os.path.join(datasets_dir, name), os.path.join(datasets_dir, name + ".") + str(id))
display()
cur()
conn.close()
```

Gambar 5. Penambahan data ke tabel basis data

Pembaharuan data ke tabel basis data memanfaatkan kode perintah *“update”* pada *MySQL*. Hal ini diterapkan pada pembaharuan informasi karyawan di menu manajemen karyawan. Kode program pembaharuan data ke tabel basis data terlihat seperti pada Gambar 6.

```
cur.execute("update attendance set department = %s, fname = %s, gender = %s, contact_no = %s, email_address = %s where eid = %s", (
post_var.get(),
fname_var.get(),
gender_var.get(),
contact_no_var.get(),
address_var.get(),
eid_var.get()
))

conn.commit()
display()
conn.close()
messagebox.showinfo("Berhasil", "Foto dan database berhasil diperbarui", parent = first)
```

Gambar 6. Pembaharuan data ke tabel basis data

Penghapusan data dari tabel basis data memanfaatkan kode perintah *“delete”* pada *MySQL*. Hal ini diterapkan pada penghapusan informasi karyawan di menu manajemen karyawan. Kode penghapusan data dari tabel basis data terlihat seperti pada Gambar 7.

```
try:
input_name = fname_var.get() + "_" + eid_var.get()
staff_input = os.path.join(datasets_dir, input_name)
if not os.path.exists(staff_input):
cur.execute("delete from attendance where eid = %s", eid_var.get())
else:
cur.execute("delete from attendance where eid = %s", eid_var.get())
shutil.rmtree(staff_input)
conn.commit()
conn.close()
display()
clear()
except Exception as e:
messagebox.showerror("Galat", e)
```

Gambar 7. Penghapusan data tabel basis data

Pencarian data dari tabel basis data memanfaatkan kode perintah *“select”* pada *MySQL*. Hal ini diterapkan untuk mencari informasi karyawan di menu manajemen karyawan ataupun status kehadiran di menu laporan kehadiran. Kode penghapusan data ke tabel basis data terlihat seperti pada Gambar 8.

```
def search_data():
conn = pymysql.connect(host = "localhost", user = "root", password = "", database = "recognition")
cur = conn.cursor()
search_data = stringvar()
if str(search_from.get())=="10":
search_data = "id"
if str(search_from.get())=="Nama":
search_data = "fname"
if str(search_from.get())=="Departemen":
search_data = "department"
cur.execute("select * from attendance where " + str(search_data) + " LIKE '%" + str(search_result.get()) + "%'"
)
data = cur.fetchall()
if len(data) == 0:
table.delete(*table.get_children())
for row in data:
table.insert('', END, values = row)
conn.commit()
else:
messagebox.showinfo("Maaf", "Data tidak ditemukan", parent = first)
conn.close()
```

Gambar 8. Pencarian data tabel basis data

Deteksi wajah menggunakan fungsi pustaka cv2. Berikut ini merupakan penjelasan terhadap fungsi kode program deteksi wajah dari Gambar 9, di antaranya:

1. `video_capture.read()` berfungsi mengaktifkan perangkat kamera.
2. `cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)` berfungsi untuk merubah gambar yang diperoleh menjadi warna abu-abu.
3. `face_cascade.detectMultiScale(gray,1.3,5)` berfungsi untuk mendeteksi objek dengan ukuran berbeda dalam gambar masukkan. Pernyataan pertama merupakan gambar yang telah dibuah menjadi warna abu-abu, pernyataan kedua merupakan scalefactor (Banyaknya ukuran gambar yang akan dikurangi pada setiap skala gambar), dan pernyataan yang ketiga merupakan jumlah titik deteksi pada satu wajah.
4. `frame[y-5:y+h+5,x-5:x+w+5]` berfungsi untuk menampilkan frame pada wajah yang terdeteksi.
5. `cv2.resize` berfungsi merubah dimensi dari frame menjadi ukuran yang dibutuhkan (160, 160).
6. `cv2.imwrite(os.path.join(input_directory,name +str(count) '.jpg'),resized_face)` berfungsi untuk menyimpan gambar pada direktori yang telah ditentukan.
7. `cv2.rectangle(frame, (x,y), (x+w, y+h),(0,0,255), 2)` berfungsi untuk membentuk garis persegi pada wajah yang terdeteksi.

```
try:
    check, frame = video_capture.read()
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray,1.3,5)
    for (x,y,w,h) in faces:
        face = frame[y-5:y+h+5,x-5:x+w+5]
        resized_face = cv2.resize(face,(160,160))
        cv2.imwrite(os.path.join(input_directory,name + str(count) + '.jpg'),resized_face)
        cv2.rectangle(frame, (x,y), (x+w, y+h),(0,0,255), 2)
        count += 1
    # show the output frame
    cv2.imshow("frame",frame)
    key = cv2.waitKey(1)
    if key == ord('q'):
        break
except Exception as e:
    pass
```

Gambar 9. Kode deteksi wajah

Ekstraksi wajah merupakan tahapan awal yang sangat penting dalam pengenalan wajah. Ekstraksi wajah berfungsi untuk mengekstraksi fitur komponen biologis yang dimiliki oleh sampel wajah. Hasil ekstraksi dapat dibandingkan dengan vektor yang dihasilkan untuk wajah lain. Keluaran dari proses ekstraksi wajah berupa `embeddings.pickle`. Kode program ekstraksi wajah dapat terlihat pada Gambar 10.

```
for (i,face_pixel) in enumerate(face_pixels):
    i = i+1
    percent.set(str(int(i)/100)+"%")
    text.set(str(i))+"risr(i)"+tasks_completed()
    update('value', i)
    # encode()
    sample = cv2.expand_dims(face_pixel,axis=0)
    embedding = embedding_model.predict(sample)
    new_embedding = embedding.reshape(-1)
    embeddings.append(new_embedding)
    data = [{"path":image_path, "name":name,"face_id":face_id, "image_id":image_id,"embedding":embedding}]
    # save embeddings to pickle
    f = open('models/embeddings.pickle', 'wb')
    pickle.dump(embeddings,f)
    f.close()
    # show message
    messagebox.showinfo("Berhasil", "Pencarian berhasil diakhiri... file pickle baru dibuat untuk menyimpan penyematan", parent = attendance)
```

Gambar 10. Kode deteksi wajah

Pelatihan sistem merupakan tahapan lanjutan setelah proses ekstraksi wajah. Pelatihan sistem berfungsi untuk mengklasifikasikan fitur dari data ekstraksi wajah. Pelatihan sistem menerapkan pendekatan statistik berbasis SVM (*Support Vector Machine*). Keluaran dari proses pelatihan sistem berupa `recognizer.pickle`. Kode program ekstraksi wajah dapat terlihat pada Gambar 11.

```
try:
    training_obj = Training(embedding_path='models/embeddings.pickle')
    [train,train_embeddings,[i] = training_obj.load_embeddings_and_labels()
    recognizer = training_obj.create_recognizer(train_embeddings,train_labels,embedding_dimension)
    f = open('models/recognizer.pickle', 'wb')
    pickle.dump(recognizer,f)
    f.close()
    messagebox.showinfo("Berhasil", "Pelatihan selesai... file pickle baru telah dibuat untuk menyimpan model pengenalan wajah", parent = attendance)
    second_start = time.time()
    second_end = time.time()
```

Gambar 11. Kode pelatihan sistem

Pengenalan wajah berfungsi untuk mengenali wajah karyawan sesuai dengan ekstraksi sampel wajah yang dikenali dan diklasifikasikan oleh sistem aplikasi. Proses pengenalan wajah diawali dengan mendeteksi wajah aktual dan membandingkannya dengan sistem yang telah dilatih. Keluaran dari proses pengenalan wajah berupa pencatatan status kehadiran dengan tingkat kepercayaan minimal 60%. Kode program ekstraksi wajah seperti terlihat pada Gambar 12.

```
try:
    (ret,frame) = vs.read()
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray,1.3,5)
    for (x,y,w,h) in faces:
        face = frame[y-5:y+h+5,x-5:x+w+5]
        resized_face = cv2.resize(face,(160,160))

        processed_face = resized_face.astype("float") / 255.0
        processed_face = img_to_array(resized_face)
        processed_face = np.expand_dims(processed_face, axis=0)
        preds = liveness_model.predict(processed_face)[0]
        print(preds)
        if preds > 0.9:
            label_name = 'spoof'
            liveness_predictor.append(label_name)
        elif preds < 0.25:
            label_name = "real"
            liveness_predictor.append(label_name)
        #else:
        #    label_name = "none"
        #    liveness_predictor.append(label_name)

        face_pixel = embedding_obj.normalize_pixels(imagearrays-resized_face)
        sample = np.expand_dims(face_pixel,axis=0)
        embedding = embedding_model.predict(sample)
        embedding = embedding.reshape(-1)
        COLORS = np.random.randint(0, 255, size=(len(label.classes_), 3), dtype="uint8")
        # perform classification to recognize the face
        preds = recognizer.predict_proba(embedding)[0]
        p = np.argmax(preds)
        proba = preds[p]
        id = label.classes_[p]
        name = getkey(id,staff_details)
        if proba >= 0.6:
            color = [int(c) for c in COLORS[p]]
            cv2.rectangle(frame,(x,y),(x+w,y+h),color,2)
            text = "{} {} {}".format(name,id,proba)
            cv2.putText(frame,text,(x,y - 5),cv2.FONT_HERSHEY_SIMPLEX,0.5,color,2)
            predictions.append(id)
        else:
            name = "NONE"
            id = "NONE"
            color = (255,255,0)
            text = "{} {} {}".format(name,id,proba)
            cv2.putText(frame,text,(x,y - 5),cv2.FONT_HERSHEY_SIMPLEX,0.5,color,2)
            cv2.rectangle(frame,(x,y),(x+w,y+h),color,2)

    cv2.imshow("capture",frame)
    key = cv2.waitKey(1)
    if key == ord('q'):
        break
except Exception as e:
    messagebox.showerror("Gagal",e)
    break
```

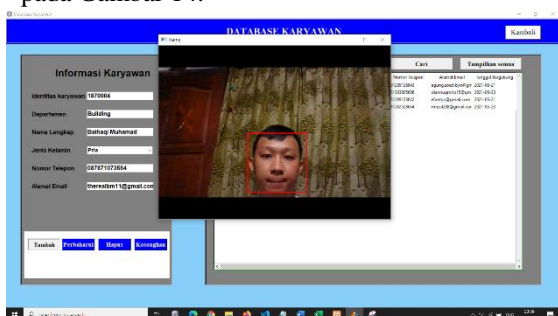
Gambar 12. Kode pengenalan wajah

Aplikasi yang dirancang memiliki halaman utama atau dasbor aplikasi yang berisikan informasi-informasi mengenai fungsi dari menu yang terdapat pada aplikasi. Pada halaman utama akan disajikan beberapa tombol yang akan mengacu pada fungsinya masing-masing seperti terlihat pada Gambar 13.



Gambar 13. Halaman utama aplikasi

Manajemen karyawan berfungsi untuk mengelola informasi detail dari karyawan yang dijadikan sebagai master data aplikasi. Pada manajemen karyawan memiliki fitur penambahan karyawan baru, pembaharuan informasi karyawan, penghapusan data karyawan serta pencarian data karyawan ke sistem basis data. Setiap dilakukannya proses penambahan karyawan baru ataupun pembaharuan identitas karyawan, maka akan dilakukan pengambilan foto wajah dari karyawan yang akan dijadikan sebagai foto sampel (sebanyak 50 sampel) dengan ukuran foto yang telah dikecilkan sebagai bahan pelatihan sistem pengenalan wajah. Selain hal tersebut, terdapat juga fitur menu pencarian informasi karyawan dari basis data yang ditampilkan pada tabel. Tampilan pengambilan foto sampel wajah terlihat seperti pada Gambar 14.



Gambar 14. Manajemen karyawan

Foto sampel berfungsi untuk mengakses dan melihat kumpulan foto sampel yang tersimpan pada direktori setelah proses pendaftaran karyawan baru dilakukan. Direktori penyimpanan foto sampel diberikan penamaan sesuai dengan nama dan identitas karyawan. Tampilan foto sampel terlihat seperti pada Gambar 15.



Gambar 15. Foto sampel

Ekstraksi dan simpan penyematan berfungsi untuk mengekstrak data dari foto sampel wajah karyawan dan mengolahnya menjadi model *embeddings.pickle* yang akan menjadi data untuk mengenali wajah karyawan dalam sistem. Tampilan ekstraksi dan simpan penyematan terlihat seperti pada Gambar 16.



Gambar 16. Ekstraksi dan simpan penyematan

Pelatihan sistem berfungsi untuk untuk melatih sistem dalam mengenali wajah karyawan dari ekstraksi dan penyematan data wajah yang dilakukan pada proses sebelumnya. Hasil keluaran dari proses ini berupa *recognizer.pickle* yang berfungsi mengenali wajah karyawan pada sistem. Tampilan pelatihan sistem terlihat seperti pada Gambar 17.



Gambar 17. Pelatihan Sistem

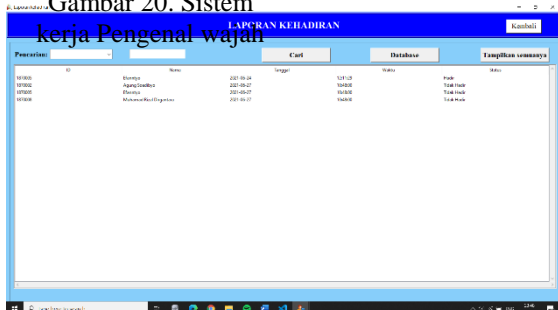
Pengenal wajah berfungsi untuk mengenali wajah dari karyawan. Bilamana wajah karyawan tersimpan dalam direktori dan teridentifikasi, maka sistem akan mengenalinya dan mencatat dengan status hadir, namun bila wajah karyawan belum tersimpan dalam sistem, maka sistem tidak akan mengenalinya. Pada sistem ini menerapkan metode

Facenet-Keras sebagai algoritma untuk mendeteksi dan mengenali wajah dari karyawan. Pencatatan status kehadiran karyawan dilakukan satu kali setiap harinya dan disimpan kedalam sistem. Tampilan pengenalan wajah dapat dilihat pada Gambar 18.



Gambar 18. Pengenal Wajah

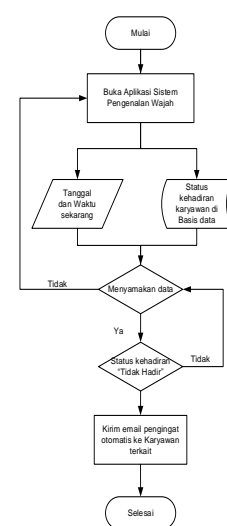
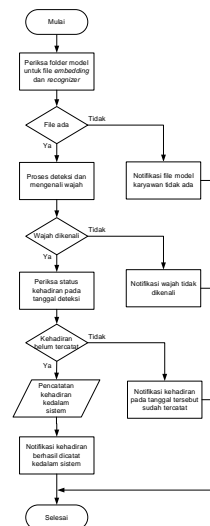
Laporan kehadiran berfungsi untuk melaporkan catatan kehadiran dan ketidakhadiran dari karyawan. bilamana terdapat karyawan yang tidak melakukan proses presensi, maka sistem akan mendeteksinya sebagai tidak hadir. Pada sistem ini juga menambahkan fitur untuk memberikan informasi berupa email ke karyawan yang tidak melakukan presensi setiap harinya (sesuai dengan pengaturan yang dilakukan) melalui email. Tampilan dari laporan kehadiran terlihat seperti pada Gambar 19.



Gambar 19. Laporan kehadiran

Aplikasi sistem pengenalan wajah merupakan suatu aplikasi yang terbentuk dari kumpulan sistem yang saling terintegrasi. Pada aplikasi memiliki fitur utama pembacaan wajah yang berperan sebagai inti dari aplikasi. Pembacaan wajah dilakukan dengan model FaceNet yang merupakan salah satu model dikembangkan langsung oleh pihak Google dengan tingginya tingkat akurasi dalam pengenalan wajah. Salah satu komponen penting dalam pengenalan wajah dengan metode FaceNet yaitu pengolahan gambar wajah, pada proses pengolahan akan dilakukan deteksi wajah, pemotongan wajah yang terdeteksi dan merubah ukuran file wajah menjadi ukuran yang diperkecil. Data olahan tersebut akan diekstraksi kedalam file 128bit dengan model FaceNet. Selanjutnya ketika mengenali wajah akan dibandingkan dengan data

olahan tersebut. Sistem kerja menu pengenalan wajah seperti terlihat pada Gambar 20. Selain sistem pengenalan wajah, terdapat juga sistem notifikasi yang berfungsi untuk memberikan pesan kepada karyawan yang belum melakukan presensi pada tanggal terkait. Dengan demikian dapat mengingatkan karyawan untuk selalu melakukan presensi. Pengaturan untuk tanggal dan waktu pengingat dapat disesuaikan pada aplikasi. Sistem kerja pemberi notifikasi seperti terlihat pada Gambar 21.



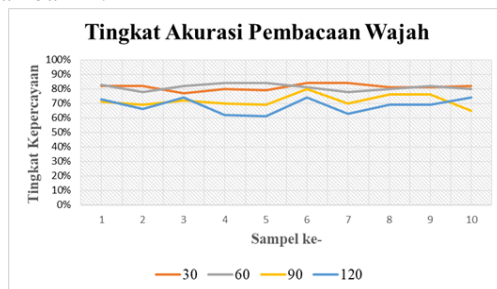
Gambar 21. Sistem kerja

Analisa tingkat akurasi pembacaan wajah bertujuan untuk menguji kemampuan sistem dalam mengenali wajah dengan berbagai kondisi yang ditentukan. Proses analisa dilakukan dengan mengambil sejumlah sampel yang wajahnya akan dideteksi oleh sistem pada jarak 30 cm, 60 cm, 90 cm, dan 120 cm dari posisi kamera. Pencahayaan yang digunakan dalam pengambilan sampel sebesar 24 lux. Hasil pengambilan data analisa dapat dilihat pada Tabel 2.

Tabel 2. Analisa tingkat akurasi pembacaan wajah

Orang ke-	Jarak wajah dari kamera (cm)			
	30	60	90	120
1	82%	83%	71%	73%
2	82%	78%	69%	66%
3	77%	82%	72%	74%
4	80%	84%	70%	62%
5	79%	84%	69%	61%
6	84%	81%	80%	74%
7	84%	78%	70%	63%
8	81%	80%	76%	69%
9	81%	82%	76%	69%
10	82%	80%	65%	74%
Rata-rata	81%	81%	72%	69%

Berdasarkan pada hasil analisa data, semakin jauh jarak wajah dari kamera, maka berpengaruh terhadap tingkat akurasi pembacaan wajah yang semakin menurun. Hal ini dikarenakan apabila wajah semakin jauh dari kamera maka ukuran area wajah yang terbaca oleh kamera juga akan semakin mengecil yang berarti banyaknya piksel pada area wajah yang terdeteksi juga semakin sedikit. Dari hal ini dapat diketahui bahwa banyaknya piksel yang terdeteksi pada area wajah akan mempengaruhi fitur ekstraksi wajah. Semakin sedikit piksel yang dideteksi pada area wajah maka fitur wajah yang dapat diekstraksi juga akan semakin sedikit. Pada pengambilan data yang dilakukan tingkat akurasi rata-rata pada jarak 30 cm memiliki tingkat kepercayaan sebesar 81%, dan semakin jauh jarak wajah maka tingkat akurasinya menurun seperti pada jarak 120 cm memiliki tingkat kepercayaan sebesar 69%. Hasil olahan data analisis dalam bentuk grafik terlihat seperti pada Gambar 22.



Gambar 22. Tingkat akurasi pengenalan wajah

IV. KESIMPULAN

Berdasarkan seluruh kegiatan penelitian yang telah dilakukan, dapat disimpulkan bahwa aplikasi yang dirancang mampu mengenali wajah karyawan pada jarak 1,2 meter dari kamera dengan akurasi rata-rata 69%. Aplikasi sistem pengenalan wajah yang telah dibangun memiliki tingkat akurasi pengenalan wajah yang dipengaruhi jarak antara kamera dengan wajah karyawan pada tingkat pencahayaan 24 lux. Pengukuran pada jarak 30 cm memberi hasil rata-rata tingkat akurasi sebesar 81%; sementara pengukuran pada jarak 60 cm, 90 cm, dan 120 cm memberikan hasil rata-rata untuk masing-masing jarak secara berurutan adalah 81%, 72%, 69%.

DAFTAR PUSTAKA

- [1] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face recognition systems: A survey," *Sensors (Switzerland)*, vol. 20, no. 2, 2020, doi: 10.3390/s20020342.
- [2] I. Q. Mundial, M. S. Ul Hassan, M. I. Tiwana, W. S. Qureshi, and E. Alanazi, "Towards facial recognition problem in COVID-19 pandemic," *2020 4th Int. Conf. Electr. Telecommun. Comput. Eng. ELTICOM 2020 - Proc.*, pp. 210–214, 2020, doi: 10.1109/ELTICOM50775.2020.9230504.
- [3] K. Bhatti, L. Mughal, F. Khuhawar, and S. Memon, "Smart Attendance Management System Using Face Recognition," *EAI Endorsed Trans. Creat. Technol.*, vol. 5, no. 17, p. 159713, 2018, doi: 10.4108/eai.13-7-2018.159713.
- [4] R. Hartanto and M. N. Adji, "Face recognition for attendance system detection," *Proc. 2018 10th Int. Conf. Inf. Technol. Electr. Eng. Smart Technol. Better Soc. ICITEE 2018*, pp. 376–381, 2018, doi: 10.1109/ICITEED.2018.8534942.
- [5] Z. Wang *et al.*, "Masked Face Recognition Dataset and Application," pp. 1–3, 2020, [Online]. Available: <http://arxiv.org/abs/2003.09093>.
- [6] A. Anwar and A. Raychowdhury, "Masked Face Recognition for Secure Authentication," pp. 1–8, 2020, [Online]. Available: <http://arxiv.org/abs/2008.11104>.
- [7] F. R. Islami, K. I. Satoto, and R. Kridalukmana, "Pengembangan Aplikasi Manajemen Pelatihan Laboratorium Software Engineering Di Fakultas Teknik Sistem Komputer," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 2, p. 223, 2016, doi: 10.14710/jtsiskom.4.2.2016.223-231.
- [8] D. Kuhlman, "A Python Book," *A Python B.*, pp. 1–227, 2013.
- [9] Anaconda, "Anaconda | The World's Most Popular Data Science Platform," *Anaconda*, 2021. <https://www.anaconda.com/> (accessed Jul. 03, 2021).
- [10] R. S. dan J. Febio, "Membangun Aplikasi E-Library Menggunakan HTML, PHP Script, dan MySQL Database," *Processor*, vol. 6, no. 2, pp. 38–54, 2011.
- [11] T. A. M. Trijaya, "Mengenal Artificial Intelligence, Machine Learning, & Deep Learning," [Online]. Available: <https://amt-it.com/mengenal-perbedaan-artificial-intelligence-machine-learning-deep-learning/>.
- [12] H. C. Shin *et al.*, "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016, doi: 10.1109/TMI.2016.2528162.
- [13] Medium.com, "Apa itu Convolutional Neural Network? | by QOLBIYATUL LINA | Medium." <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4> (accessed Jul. 03, 2021).
- [14] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," *Proc. IEEE*

*Comput. Soc. Conf. Comput. Vis. Pattern
Recognit.*, vol. 07-12-June, pp. 815–823,
2015, doi: 10.1109/CVPR.2015.7298682.