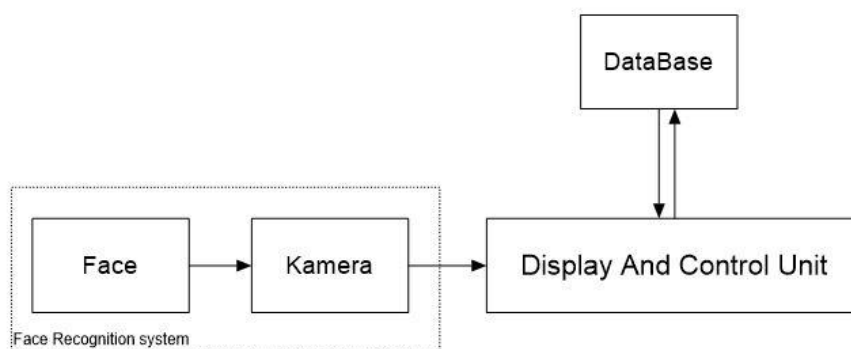


## BAB IV HASIL DAN PEMBAHASAN

Pada perancangan system manajemen kendaraan parkir menggunakan teknologi *Face recognition*, teknologi ini sebagai masukan yang digunakan untuk memasukkan data pengguna dan pada saat pengguna keluar dari tempat parkir, dengan memanfaatkan kamera *video livestream* yang terhubung ke *PC Server*, dan sebuah PC monitor untuk menampilkan GUI *system*.

Sistem manajemen ini diadaptasi dari sistem keamanan kendaraan parkir berbasis *RFID tag* yang biasa digunakan pada suatu lokasi seperti apartemen, universitas, perusahaan, dll. Dengan menggunakan kamera sebagai pendeteksi siapa apa saja melalui *Face Recognition*

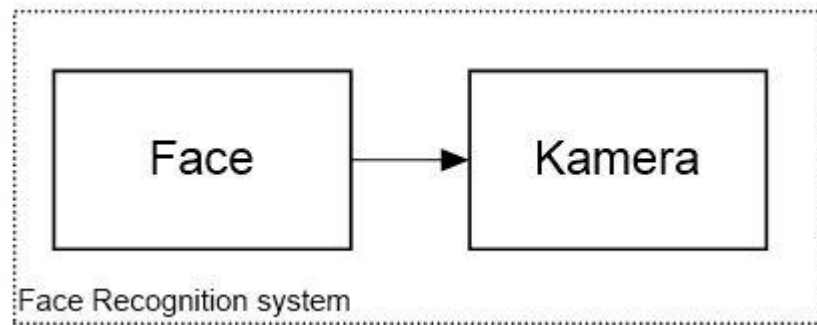


**Gambar 4. 1** Blok Diagram Sistem Manajemen Kendaraan Parkir

Untuk dapat bisa dimengerti, dibawah ini di paparkan keterangan yang bersifat umum dari setiap proses software yang diterapkan pada tugas akhir ini.

### 4.1. Sistem Face Recognition

Sistem keamanan manajemen parkir ini menggunakan sistem *Face recognition* yang memiliki dua bagian penting dalam sistem operasinya seperti gambar dibawah ini.



**Gambar 4. 2** Diagram blok sistem face recognition

#### 4.1.1. Face

Dalam penelitian ini wajah yang dipakai adalah wajah si pengguna yang sudah didaftarkan melalui program juga.berikut adalah contoh gambar wajah yang diambil buat penelitian ini.



**Gambar 4. 3** Contoh sample muka yang dipakai

Supaya bisa mengenali wajah maka dibutuhkan sample wajah yang banyak, maka dari itu dibutuhkan beberapa sample foto. Pada penelitian ini sample foto yang diambil supaya bisa digunakan untuk melatih programnya adalah 21 gambar per orang.

#### **4.1.2. Kamera**

Pada bagian kamera ini berfungsi sebagai alat pengambilan gambar yang diperlukan untuk program ini. Gambar yang diperoleh dari kamera tersebut dapat digunakan sebagai bahan untuk melatih program ini. Pada penelitian ini Kamera yang digunakan adalah webcam logitech C270 HD, Seperti gambar 4.2 ini.



**Gambar 4. 4 Webcam Logitech C270 HD**

Webcam ini memiliki resolusi 1280 x 720 pixels dengan harga yang terjangkau ini cukup untuk mengambil gambar yang jelas untuk dijadikan sample di database.

#### **4.2.Database**

Pada penelitian ini sistem *database* yang digunakan adalah SQLite dikarenakan file yang di buat oleh SQLite ini akan berukuran kecil.

#### **4.3.Perancangan PC Server dan Display**

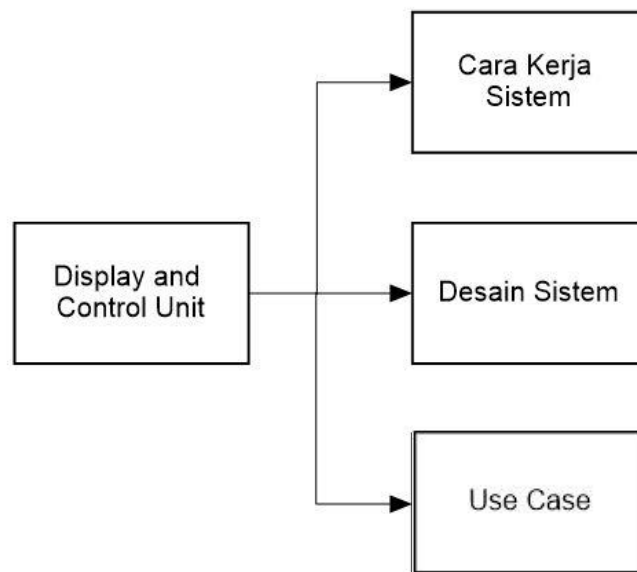
Setelah gambar didapat maka gambar tersebut akan di simpan di suatu tempat / folder di PC Server. PC Server ini juga adalah sebagai tempat program itu berkerja. Program ini akan memproses gambar tersebut apakah benar atau salah itu di PC ini dan akan menampilkan nya di LCD, pada penelitian ini

menggunakan Python 3.7 dan memakai beberapa *library* yang memiliki fungsinya masing-masing

Pada penelitian ini system manajemen adalah sebuah aplikasi *prototype* yang memiliki beberapa fungsi yang disediakan yaitu:

- A. Merekam Data yang diperlukan mengenai pengguna
- B. Keamanan yang lebih simple
- C. Mengidentifikasi data user yang telah terekam dalam *database* pada saat keluar dari lokasi melalui face recognition

Pada penelitian ini ada tahapan yang diperlukan untuk terbentuknya *prototype* dalam system manajemen keamanan kendaraan parkir ini yang akan dijabarkan dibawah ini:



Gambar 4. 5 Rancangan pembuatan sistem

#### 4.3.1. Perancangan Cara Kerja Sistem

Pada penelitian ini cara kerja system nya cukup sederhana yang dengan memanfaatkan teknologi identifikasi *biometric* berupa *Face recognition* menciptakan system keamanan parkir kendaraan bermotor yang lebih fleksibel dan aman

#### A. Cara Kerja Sistem Face Recognition untuk yang pertama

Sistem *Face recognition* yang pertama ini digunakan untuk mendaftarkan pengguna ke database dengan cara memasukkan data diri dan gambar nya tersebut. Kamera nya akan di taruh di dalam ruangan. Detail cara kerjanya akan dijelaskan pada gambar diagram arus 4.3 dibawah ini.



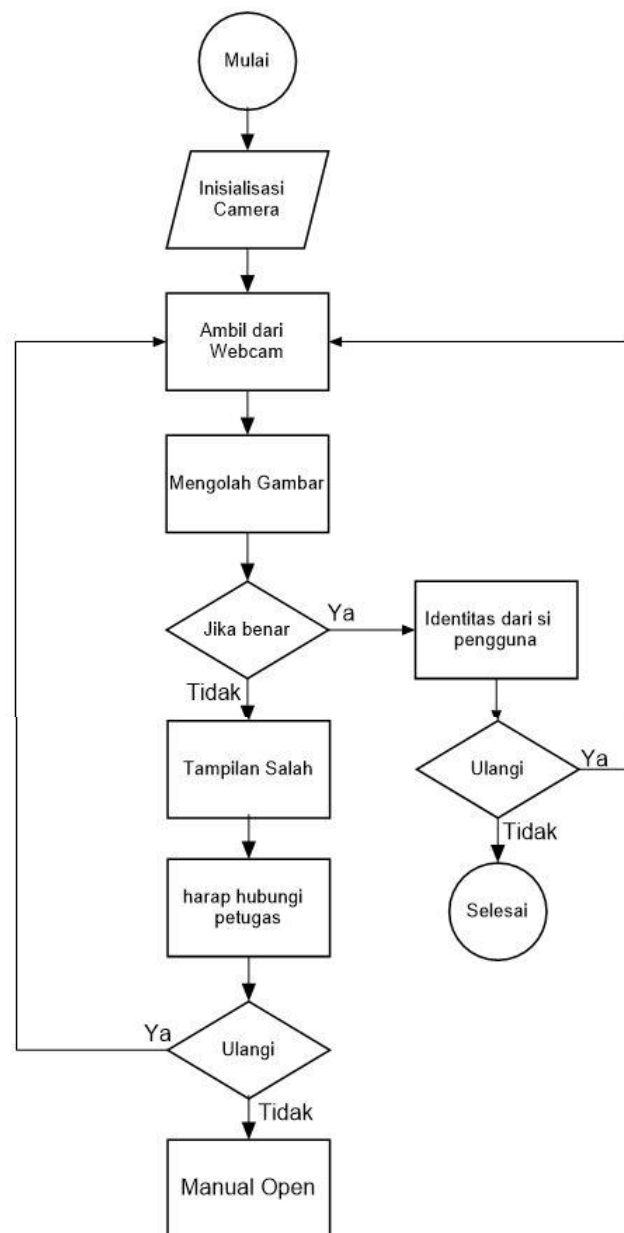
**Gambar 4. 6** flowchart system face recognition bagian input data

Sistem *face recognition* ini perlu dilakukan karena jika tidak dilakukan maka pengguna tidak akan bisa keluar pada saat dia masuk ke dalam lokasi parkir

#### B. Cara Kerja Sistem Face Recognition untuk yang kedua

Pada Sistem *face recognition* ini kamera di letakkan di pintu keluar nya kendaraan bermotor yang pintunya ini adalah satu-satunya jalan keluar dari area parkir ini, Pada system ini bekerja dengan cara pengguna yang akan keluar dari lokasi perlu mengidentifikasi dirinya sebagai pemilik kendaraan

dengan menggunakan wajah dari si pengguna yang telah tersimpan di *database*, pengguna diminta untuk menunjukkan wajahnya ke kamera yang telah disediakan. Jika muka tersebut terdaftar dalam *database*, pada display akan menunjukkan *Graphic User Interface (GUI)* data diri yang telah ada di *database*, jika wajah pengguna belum terdaftar di *database*, maka pengguna harus melakukan prosedur administrasi yang berlaku di lokasi tersebut. Di bawah ini akan di jelaskan system *Face recognition* melalui gambar diagram arus 4.4 dibawah ini



**Gambar 4. 7** flowchart system face recognition bagian security check

### 4.3.2. Perancangan Desain Sistem

Perancangan desain system ini terdiri dari program GUI, *video livestream* dan *database* yang saling terhubung satu sama lain dengan menggunakan aplikasi *Python 3.7*. GUI di penelitian ini dipakai pada saat memasukkan data diri ke *database* dan pada saat security check supaya bisa memudahkan pengguna dan operator.

Tahap perancangan pertama dalam desain system ini merupakan pembuatan *database* menggunakan *python 3.7* dengan library *sqlite3* yang merupakan library yang mengimplementasikan mesin basis data atau *database* SQL kecil, cepat, mandiri, memiliki keandalan yang tinggi dan, berfitur lengkap. *SQLite* ini *database*.

```
import sqlite3

conn = sqlite3.connect('database.db')

c = conn.cursor()

sql = """
DROP TABLE IF EXISTS users;
CREATE TABLE users (
|         id integer unique primary key autoincrement,
|         nama text, date Birth integer
);
"""
c.executescript(sql)

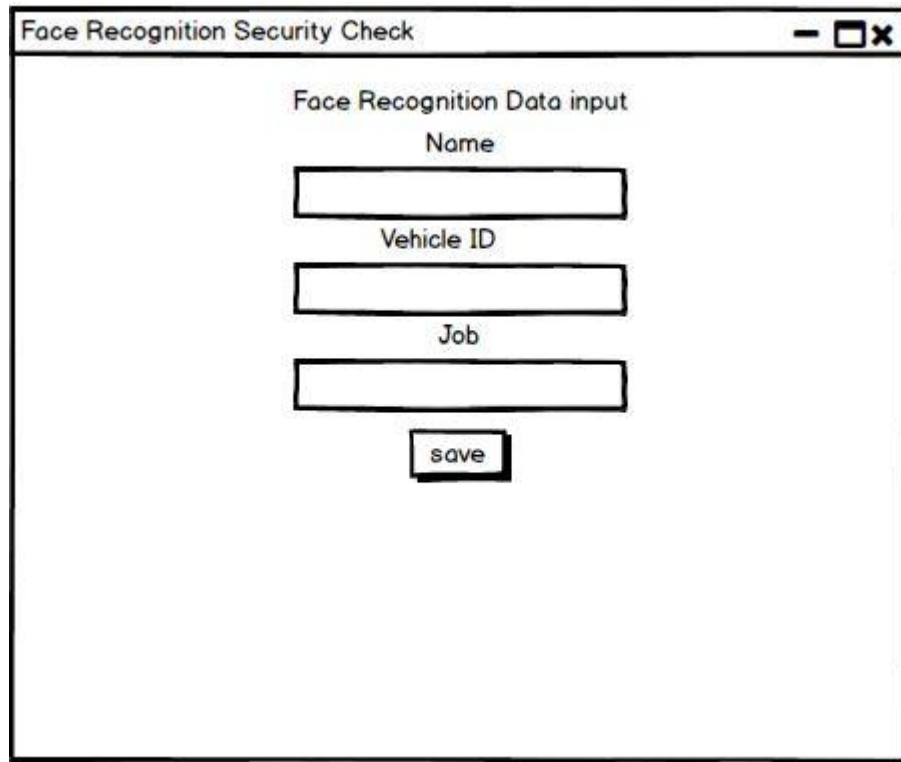
conn.commit()

conn.close()
```

**Gambar 4. 8** *Source Code* untuk membuat *database*

Gambar diatas merupakan cara membuat *database* setelah *Source Code* ini telah di jalankan maka akan terbuat file untuk *database* yang isinya tentang data diri pengguna yaitu nama dan plat nomor

Tahap perancangan kedua adala desain input data untuk *Face recognition* yang akan di implementasikan untuk memudahkan pengguna dalam memasukan data diri merka, desain sistemnya akan terlihat seperti gambar dibawah ini.



Gambar 4. 9 Desain Sistem face Recognition Data Input

GUI Yang berupa window yang digunakan untuk memasukkan data dari sang pengguna, setelah pengguna Selesai maka secara otomatis kamera akan menyala dan mengambil gambar dari si pengguna itu tersebut dan akan di ambil sampai 21 kali

Dalam pengerjaannya sistem *Face Recognition* ini menggunakan *Python IDLE 3.7 (64-bit)*, dan dalam sistem ini memerlukan beberapa *Library* algoritma Python 3.7, *Library* yang dipakai adalah:

- a) OpenCV (*Open Source Computer Vision Library*)

*Library* ini digunakan sebagai pengolahan citra dinamis secara *real-time*. Dalam program ini digunakan sebagai *video livestream* yang akan menangkap wajah si pengguna. Untuk bisa menangkap wajah tersebut dibutuhkan file tambahan yaitu *Haarcascade\_frontalface\_default*



b) Numpy

*Numpy* merupakan *library* untuk *python programming* yang digunakan untuk menambah dukungan array dan matriks multi dimensi yang besar, bersama dengan koleksi besar fungsi matematika tingkat tinggi untuk beroperasi pada array ini

c) Os

Os module ini *library* yang menyediakan cara untuk menggunakan system operasi secara fungsionalitas. Fungsi-fungsi yang disediakan memungkinkan pengguna dapat berinteraksi dengan system

d) Tkinter

Tkinter ini merupakan *Library* yang digunakan supaya bisa membuat GUI (*Graphic User Interface*).

Tahap ketiga dari perancangan desain sistem ini adalah melatih program ini dengan data yang telah dimasukkan di program sebelumnya. Program ini sangat di butuhkan dalam program *Face Recognition* ini jika tidak dilatih terlebih dahulu maka data yang terbaru tidak akan muncul pada saat dibutuhkan

```
import os
import cv2
import numpy as np
from PIL import Image

recognizer = cv2.face.LBPHFaceRecognizer_create()
path = 'dataset'
if not os.path.exists('./recognizer'):
    os.makedirs('./recognizer')

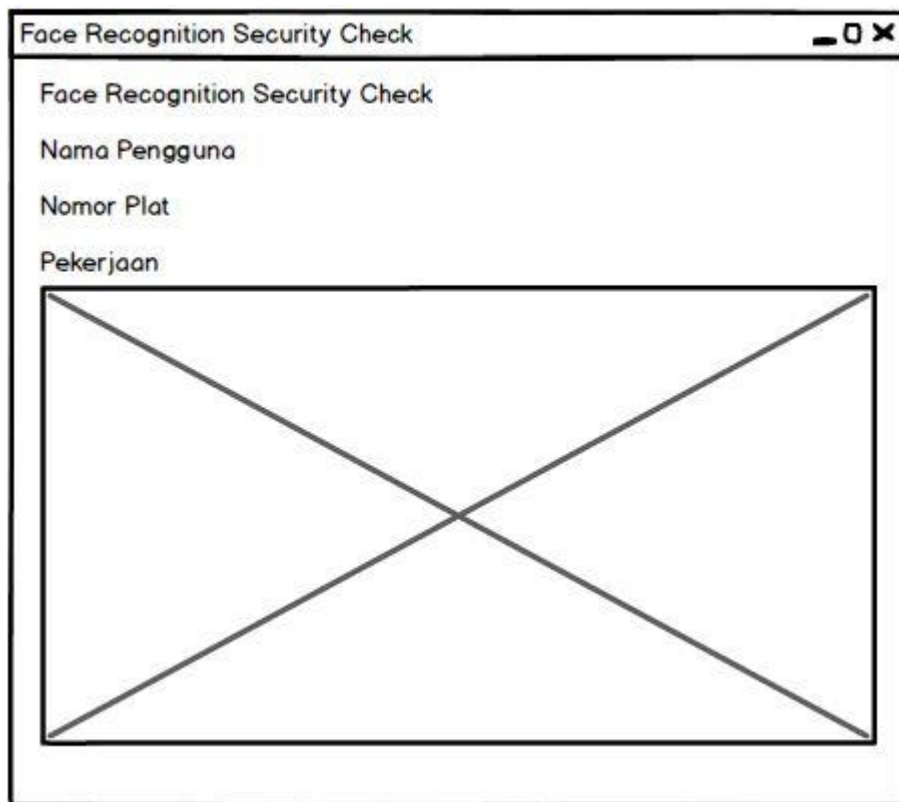
def getImagesWithID(path):
    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
    faces = []
    IDs = []
    for imagePath in imagePaths:
        faceImg = Image.open(imagePath).convert('L')
        faceNp = np.array(faceImg,'uint8')
        ID = int(os.path.split(imagePath)[-1].split('.')[1])
        faces.append(faceNp)
        IDs.append(ID)
        cv2.imshow("training", faceNp)
        cv2.waitKey(10)
    return np.array(IDs), faces

Ids, faces = getImagesWithID(path)
recognizer.train(faces, Ids)
recognizer.save('recognizer/trainingData.yml')
cv2.destroyAllWindows()
```

**Gambar 4. 10** Source Code untuk Training Data

Gambar diatas merupakan *Source Code* yang digunakan untuk mentraining data dengan file keluaran nya yaitu *trainingData.yml* yang merupakan hasil dari training data itu sendiri

Tahap yang terakhir ini dari perancangan desain sistem ini adalah program *Face Recognition* ini yang akan diimplementasikan kedalam pintu keluar lokasi, desain sistem akan di lihat di bawah ini



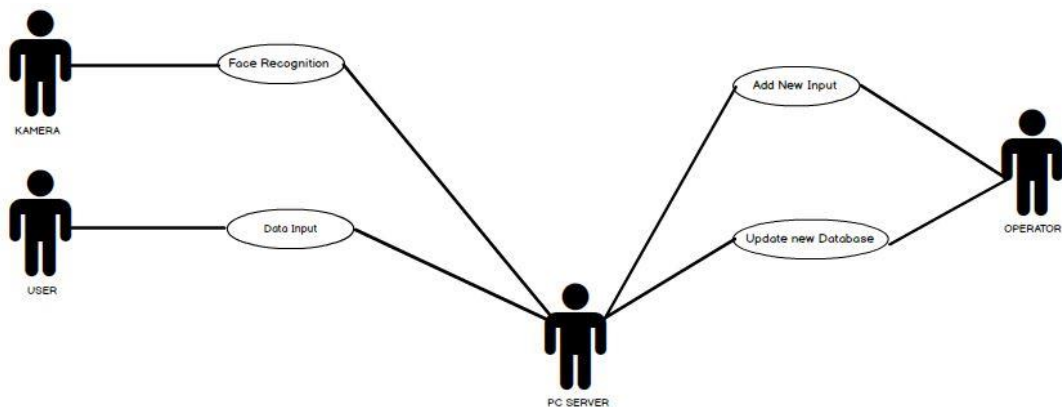
*Gambar 4. 11* Desain Sistem *Face Recognition Security Check*

GUI disini berupa window yang berisi tentang data diri dan hasil foto dari kamera yang ditangkap dengan sistem *Face recognition*. semuanya dilakukan secara otomatis demi meningkatkan fleksibilitas akses masuk bagi pengguna.

Dalam pengerjaannya sistem *Face Recognition Security check* ini sama juga bmenggunakan *Python IDLE 3.7 (64-bit)*, dan dalam sistem ini memerlukan beberapa *Library* algoritma Python 3.7 yang sama dengan program sebelumnya.

### 4.3.3. Use-Case

Pada bagian *Use-case* ini akan mendefinisikan dari sistem *Face Recognition* pada lokasi area parkir yang telah dibuat dengan cara menampilkan diagram *use-case*.terdapat 2 Diagram yang akan ditampilkan yaitu Digram pada saat data input dan pada saat *Security Check* akan di tampilkan di bawah ini:

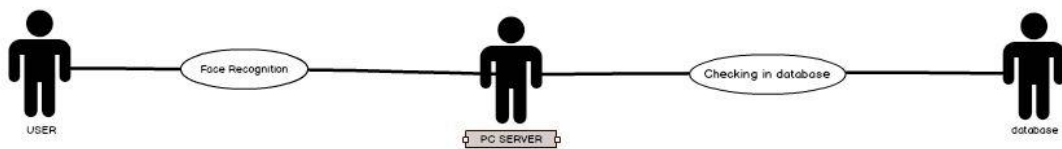


Gambar 4. 12 Use-Case Diagram`

Terdapat beberapa bagian yang ada di *use-case* diagram diatas yang akan di jelaskan dibawah ini:

1. Face Recognition ini merupakan bagian yang digunakan untuk memudahkan kamera dalam mengambil gambar wajah si pengguna
2. *Data Input* ini adalah bagian memasukkan data pribadi oleh si pengguna untuk dimasukkan ke *database*
3. *ADD Data Input* adalah bagian yang memasukkan data dari si pengguna ini ke dalam *DataBase*
4. *Update New Database* adalah bagian melatih program *Face recognition* Pada pintu keluar sama operator dengan data yang baru masuk supaya pengguna yang baru bisa keluar.

Pada diagram dibawah ini adalah diagram *Use-case* yang akan mendefinisikan dari sistem yang *Face Recognition Security Check*, berikut adalah diagramnya.



**Gambar 4. 13** *Use-Case Diagram 2*

Terdapat beberapa bagian yang akan di jelaskan tentang sistem *Security check* dibawah ini:

1. *Face Recognition* adalah pada saat muka dari pengguna akan dideteksi di kamera untuk di cek dengan data yang ada di *Database*
2. *Checking in Database* adalah bagian yang akan melihat apakah data yang diambil dengan kamera sesuai dengan data yang ada di *Database*

#### **4.4.Pengujian dan analisa program**

Setelah tahap perancangan PC server dan display maka akan dilanjutkan dengan tahap pengujian dan analisa yang akan dilakukan untuk menguji program yang telah dibuat dalam penelitian ini dan di analisa tiap program nya termasuk bagian yang jika ada bug atau kerusakan dan sebgainya

##### **4.4.1. Pengujian *input data to DataBase***

Program ini ditujukan untuk memudahkan pengguna untuk memasukkan data dari si pengguna berikut adalah tampilannya jika programnya akan dijalankan:

The screenshot shows a window titled "Face Recognition Data Input" with a light gray background. The window contains the following text and input fields:

- Title: Face Recognition Data Input
- Label: Name
- Input field: M Arief F
- Label: Vehicle id
- Input field: AB 1464 JU
- Label: Job
- Input field: Student
- Button: Save

**Gambar 4. 14** Contoh hasil *data input* Pertama

The screenshot shows a window titled "Face Recognition Data Input" with a light gray background. The window contains the following text and input fields:

- Title: Face Recognition Data Input
- Label: Name
- Input field: Bram Y
- Label: Vehicle id
- Input field: AB 2222 JU
- Label: Job
- Input field: Student
- Button: Save

**Gambar 4. 15** Contoh hasil *data input* Kedua

The screenshot shows a window titled "Face Recognition Data Input" with a light gray background. The window contains the following text and input fields:

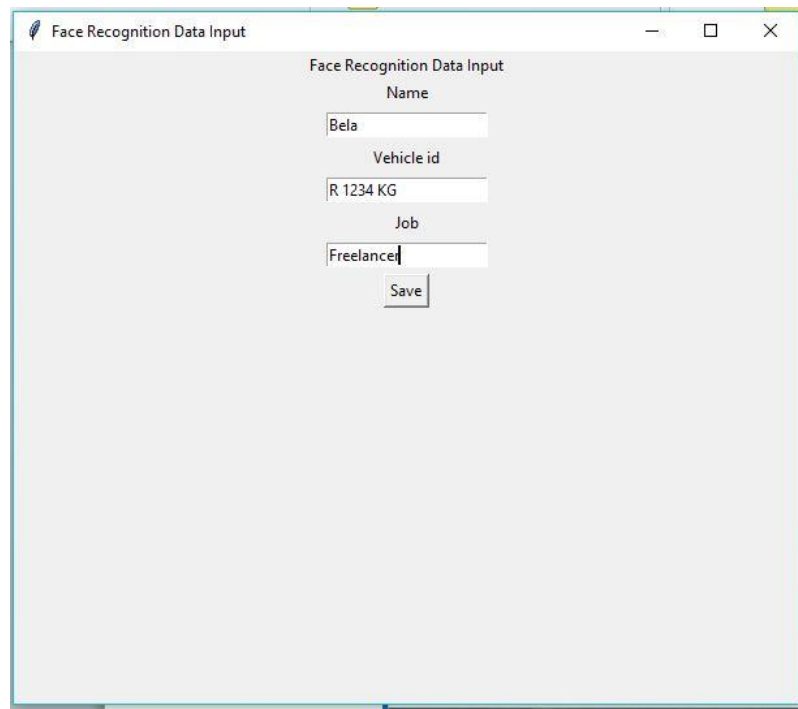
- Title: Face Recognition Data Input
- Label: Name
- Input field: Tasyrif Ikhsanul Jihad
- Label: Vehicle id
- Input field: AB7957JU
- Label: Job
- Input field: Student
- Button: Save

**Gambar 4. 16** Contoh Hasil *Data Input* Ketiga

The screenshot shows a window titled "Face Recognition Data Input" with a light gray background. The window contains the following text and input fields:

- Title: Face Recognition Data Input
- Label: Name
- Input field: siti maryam
- Label: Vehicle id
- Input field: E2524CM
- Label: Job
- Input field: Student
- Button: Save

**Gambar 4. 17** Contoh Hasil *Data Input* Keempat



**Gambar 4. 18** Contoh Hasil *Data Input* Kelima

Pada gambar diatas ini merupakan gambar GUI pada saat pengguna memasukkan data diri mereka, yang akan dimasukkan adalah nama dan plat nomor nya setelah data nya sudah dimasukkan dan pengguna menekan tombol **Save** maka secara otomatis kamera akan mengambil gambar wajah dari si pengguna sampai 21 kali. Setelah pengambilan foto selesai maka data tersebut akan di masukkan ke *Database* Untuk supaya bisa terdeteksi dengan program maka program ini harus dilatih dulu setelah dilatih maka akan bisa mendeteksi wajah dari pengguna

Program berikut bisa berjalan jika menggunakan *Source Code* yang benar berikut akan ditampilkan *Source Code* dari program ini

```

import cv2
import numpy as np
import sqlite3
import os
from tkinter import *

conn = sqlite3.connect('database.db')
if not os.path.exists('./dataset'):
    os.makedirs('./dataset')

c = conn.cursor()

face_cascade = cv2.CascadeClassifier(r'F:\try2\haarcascade_frontalface_default.xml')

cap = cv2.VideoCapture(0)

drawGUI() #It will invoke the method drawGUI

c.execute('INSERT INTO users (nama,platonomer) VALUES (?,?)', (uname,platonomer))

uid = c.lastrowid
sampleNum = 0
while True:
    ret, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        sampleNum = sampleNum+1
        cv2.imwrite("dataset/User."+str(uid)+"."+str(sampleNum)+".jpg",gray[y:y+h,x:x+w])
        cv2.rectangle(img, (x,y), (x+w, y+h), (255,0,0), 2)
        cv2.waitKey(100)

    cv2.imshow('img',img)
    cv2.waitKey(10);
    if sampleNum > 20:
        print(uname,platonomer) # just to check the input is printed corrected
        break
cap.release()
conn.commit()
conn.close()
cv2.destroyAllWindows()

```

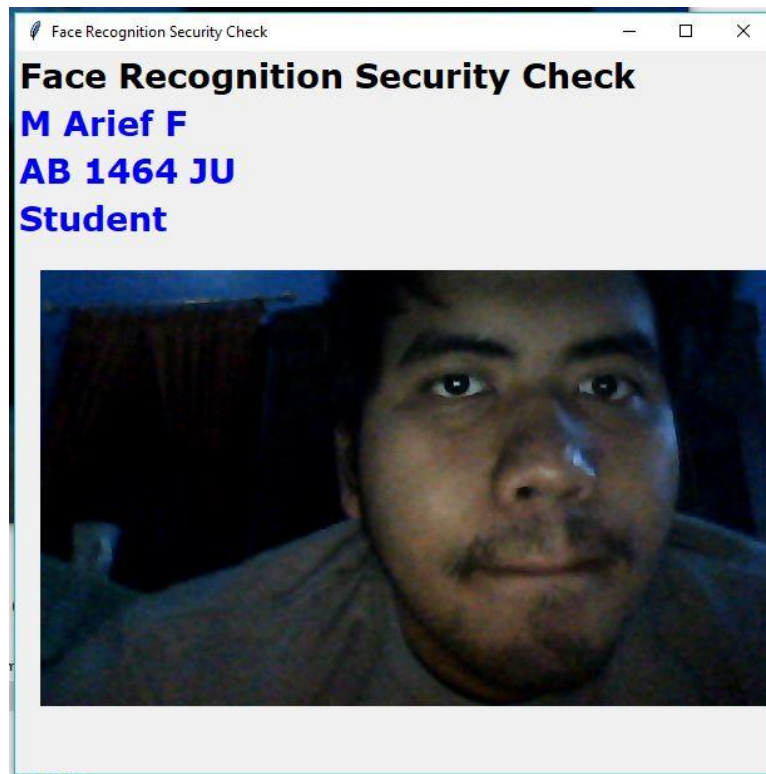
**Gambar 4. 19** Source Code Program Data Input

Bisa dilihat di atas Program ini memakai 5 *Library* yaitu OpenCv, numpy, sqlite, Os, Tkinter yang sudah di jelaskan di sebelumnya fungsinya buat apa. Program ini akan membuat folder *dataset* yang akan berisi sample foto yang telah di tangkap dan akan dipakai, setelah di sambungkan dengan database yang di buat dengan sqlite dan disambungkan dengan kamera untuk menangkap wajah maka selanjutnya akan di munculkan GUI yang seperti diatas untuk memasukan nama dan platnomer. setelah kedua data tersebut dimasukkan dan di tekan tombol *Save* maka secara otomatis kamera akan mengambil gambar wajah dengan warna abu-abu sebanyak 21 kali.

#### 4.4.2. Pengujian *Face Recognition Security Check*

Program ini di perlukan pada lokasi pintu keluar ini akan menampilkan GUI jika data dari pengguna sudah dimasukkan ke *Database* dan dilatih di program *Face Recogiton* berikut adalah contoh tampilan GUI





Gambar 4. 20 Contoh hasil dari *Security check* Pertama



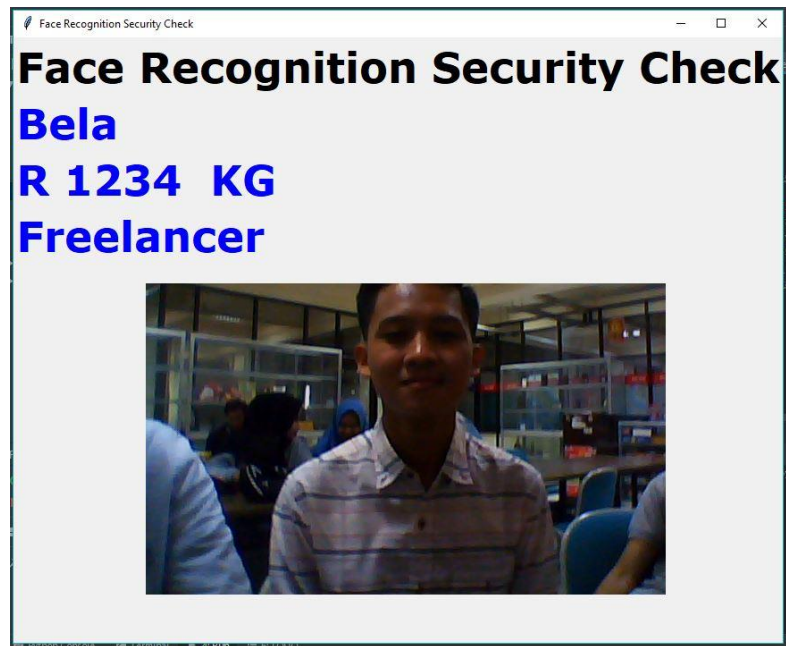
Gambar 4. 21 Contoh hasil dari *Security Check* Kedua.



Gambar 4. 22 Contoh hasil dari *Security Check* Ketiga

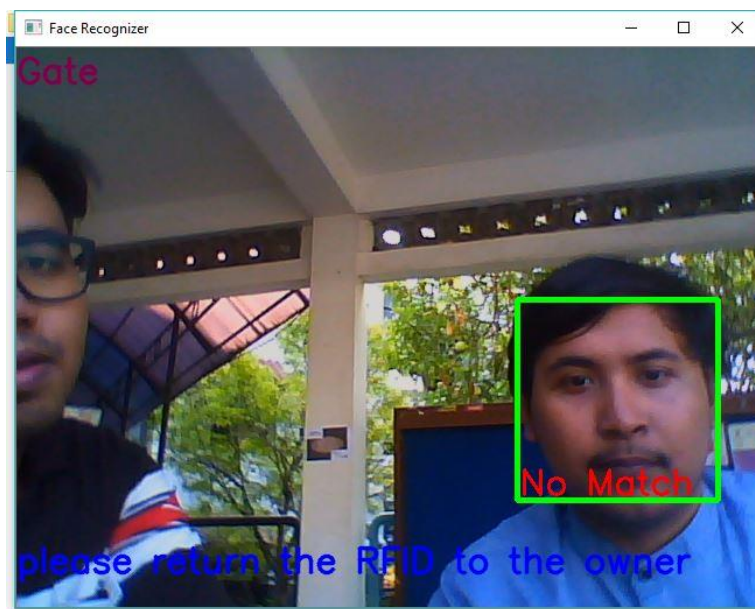


Gambar 4. 23 Contoh hasil dari *Security Check* Keempat



Gambar 4. 24 Contoh hasil dari *Security Check* Kelima

Jika wajah yang ada di depan kamera tidak ada di dataset atau program belum dilatih dengan data yang ada maka yang akan muncul di window adalah seperti berikut



Gambar 4. 25 Contoh hasil dari *Security Check* yang error pertama



**Gambar 4. 26** Contoh hasil dari *Security Check* yang error Kedua

Program berikut bisa berjalan jika menggunakan *Source Code* yang benar, Pada *Source Code* ini memakai 4 *Library* yang dipakai yaitu OpenCV, Numpy, sqlite, dan os. setelah disambungkan ke *database* dan kamera yang akan dipakai, maka kamera tersebut akan menyala dan mulai mendeteksi wajah pada saat wajah muncul didalam tangkapan kamera tersebut maka akan di check apakah wajah yang didalam tersebut ada didalam *Dataset*. Jika ada maka muncul GUI yang akan menampilkan data dari Wajah yang ditangkap yaitu nama dan nomor plat. Jika tidak ada maka GUI tidak akan muncul, jika itu terjadi maka pengguna harus mengurusnya dengan petugas yang sedang berjaga. Berikut akan ditampilkan *Source Code* dari program ini:

```

import cv2
import numpy as np
import sqlite3
import os

conn = sqlite3.connect('database.db')
c = conn.cursor()
fname = "recognizer/trainingData.yml"
if not os.path.isfile(fname):
    print("Please train the data first")
    exit(0)
face_cascade = cv2.CascadeClassifier('F:\\Komputasi berbasis peranti mobile\\haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0)
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read(fname)
while True:
    ret, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(img, 'Gate 1A', (0,30), font, 1, (75,0,130), 2, cv2.LINE_AA)
    for (x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0),3)
        ids,conf = recognizer.predict(gray[y:y+h,x:x+w])
        c.execute("select nama from users where id = (?)", (ids,))
        result = c.fetchall()
        name = result[0][0]
        font = cv2.FONT_HERSHEY_SIMPLEX
        if conf < 50:
            cv2.putText(img, name, (x+2,y+h-5), cv2.FONT_HERSHEY_SIMPLEX, 1, (150,255,0),2)
            cv2.putText(img, 'Thank you for coming', (0,450), font, 1, (251,140,1), 2, cv2.LINE_AA)
        else:
            cv2.putText(img, 'No Match', (x+2,y+h-5), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255),2)
            font = cv2.FONT_HERSHEY_SIMPLEX
            cv2.putText(img, 'please return the RFID to the owner', (0,450), font, 1, (255,0,0), 2, cv2.LINE_AA)
    cv2.imshow('Face Recognizer',img)
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break
cap.release()
cv2.destroyAllWindows()

```

Gambar 4. 27 Source Code program Security Check

#### 4.4.3. Percobaan keberhasilan program

Pada bagian ini akan dilakukan percobaan pada program apakah program akan bisa mendeteksi wajah dengan akurat akan ada beberapa sampel wajah yang akan digunakan. Untuk yang pertama berikut adalah sample muka yang di pakai:



Gambar 4. 28 Contoh Sample muka di dataset

Percobaan pertama dilakukan dengan wajah *user 1* yang sampel nya dari user.1.1 sampai user.1.21 berikut adalah hasil dari percobaannya:

**Tabel 4. 1** Tabel Percobaan 1

NO	Input	Output	kebenaran
1	Wajah <i>user 1</i>	Data diri dan gambar <i>user 1</i>	1
2	Wajah <i>user 1</i>	Data diri dan gambar <i>user 1</i>	1
3	Wajah <i>user 1</i>	Data diri dan gambar <i>user 1</i>	1
4	Wajah <i>user 1</i>	Data diri dan gambar <i>user 1</i>	1
5	Wajah <i>user 1</i>	Data diri dan gambar <i>user 1</i>	1
6	Wajah <i>user 1</i>	No Detect	0
7	Wajah <i>user 1</i>	Data diri dan gambar <i>user 1</i>	1
8	Wajah <i>user 1</i>	Data diri dan gambar <i>user 1</i>	1
9	Wajah <i>user 1</i>	Data diri dan gambar <i>user 1</i>	1
10	Wajah <i>user 1</i>	Data diri dan gambar <i>user 1</i>	1

$$Akurasi = \frac{Jumlah\ nilai\ kebenaran\ 1}{Jumlah\ data\ testing} \times 100$$

$$Akurasi = \frac{9}{10} \times 100$$

$$Akurasi = 90\%$$

Percobaan kedua dilakukan dengan wajah si *user 2* yang sampel nya dari user.2.1 sampai user.2.21 berikut adalah hasil dari percobaannya:

**Tabel 4. 2** Tabel Percobaan 2

NO	Input	Output	kebenaran
1	Wajah <i>user 2</i>	Data diri dan gambar <i>user 2</i>	1
2	Wajah <i>user 2</i>	Data diri dan gambar <i>user 2</i>	1
3	Wajah <i>user 2</i>	Data diri dan gambar <i>user 2</i>	1
4	Wajah <i>user 2</i>	Data diri dan gambar <i>user 2</i>	1
5	Wajah <i>user 2</i>	Data diri dan gambar <i>user 2</i>	1

NO	Input	Output	Kebenaran
6	Wajah <i>user 2</i>	No Detect	0
7	Wajah <i>user 2</i>	Data diri dan gambar <i>user 2</i>	1
8	Wajah <i>user 2</i>	Data diri dan gambar <i>user 2</i>	1
9	Wajah <i>user 2</i>	No Detect	0
10	Wajah <i>user 2</i>	Data diri dan gambar <i>user 2</i>	1

$$Akurasi = \frac{\text{Jumlah nilai kebenaran 1}}{\text{Jumlah data testing}} \times 100$$

$$Akurasi = \frac{8}{10} \times 100$$

$$Akurasi = 80\%$$

Percobaan ketiga ini dilakukan menggunakan 3 wajah sekaligus yang ditujukan untuk mengetahui apakah pada saat mendeteksi wajah akan berhasil mengeluarkan data diri dari ketiga orang ini berikut hasil percobaannya:

**Tabel 4. 3** Tabel Percobaan 3

NO	Input	Output	kebenaran
1	Wajah <i>user 3</i>	Data diri dan gambar <i>user 3</i>	1
2	Wajah <i>user 3</i>	Data diri dan gambar <i>user 3</i>	1
3	Wajah <i>user 3</i>	Data diri dan gambar <i>user 3</i>	1
4	Wajah <i>user 3</i>	Data diri dan gambar <i>user 3</i>	1
5	Wajah <i>user 3</i>	Data diri dan gambar <i>user 3</i>	1
6	Wajah <i>user 4</i>	Data diri dan gambar <i>user 4</i>	1
7	Wajah <i>user 4</i>	Data diri dan gambar <i>user 4</i>	1
8	Wajah <i>user 4</i>	Data diri dan gambar <i>user 4</i>	1
9	Wajah <i>user 4</i>	Data diri dan gambar <i>user 4</i>	1
10	Wajah <i>user 4</i>	Data diri dan gambar <i>user 4</i>	1
11	Wajah <i>user 5</i>	Data diri dan gambar <i>user 5</i>	1
12	Wajah <i>user 5</i>	Data diri dan gambar <i>user 5</i>	1

NO	Input	Output	Kebenaran
13	Wajah <i>user 5</i>	Data diri dan gambar <i>user 5</i>	1
14	Wajah <i>user 5</i>	Data diri dan gambar <i>user 5</i>	1
15	Wajah <i>user 5</i>	Data diri dan gambar <i>user 5</i>	1

$$Akurasi = \frac{Jumlah\ nilai\ kebenaran}{Jumlah\ data\ testing} \times 100$$

$$Akurasi = \frac{15}{15} \times 100$$

$$Akurasi = 100\%$$

Setelah dilakukan percobaan diatas maka didapatkan bahwa program bisa berjalan dengan baik dengan *Chance* kesalahan yaitu antara 0-20% dengan 10-15 kali percobaan.