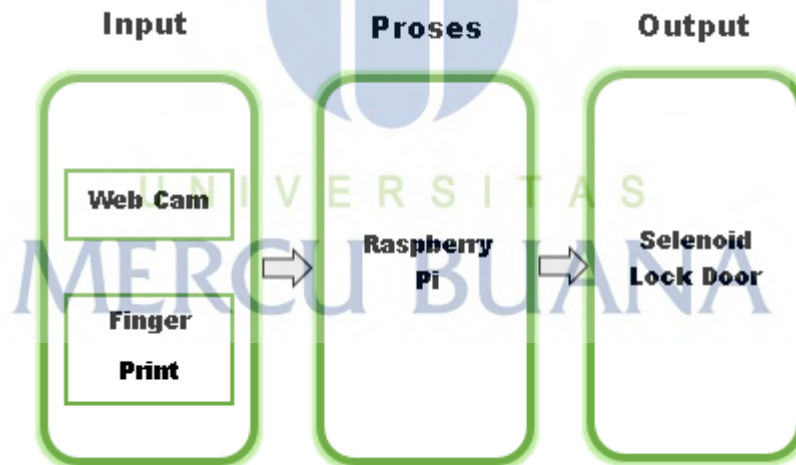


## BAB III

### PERANCANGAN DAN PEMBUATAN ALAT

#### 3.1 Perancangan Blok Diagram

Rancang Bangun *Dual* Sistem Keamanan Pintun Rumah Otomatis Menggunakan Pengenalan Wajah dan Sidik Jari menggunakan dua buah *input* yaitu *Webcam / USB camera* dan *FingerPrint*. Sebelum membuat perangkat lunak, yang perlu diketahui lebih utama adalah susunan sistem alat. Susunan sistem alat yang akan dibuat tersebut seperti pada gambar blok diagram berikut ini.



Gambar 3.1 Blok Diagram

Dari hasil Gambar 3.1 diatas adalah blok diagram rangkaian sistem. Secara garis besar sistem terbagi menjadi tiga bagian, yaitu masukan (Input), proses

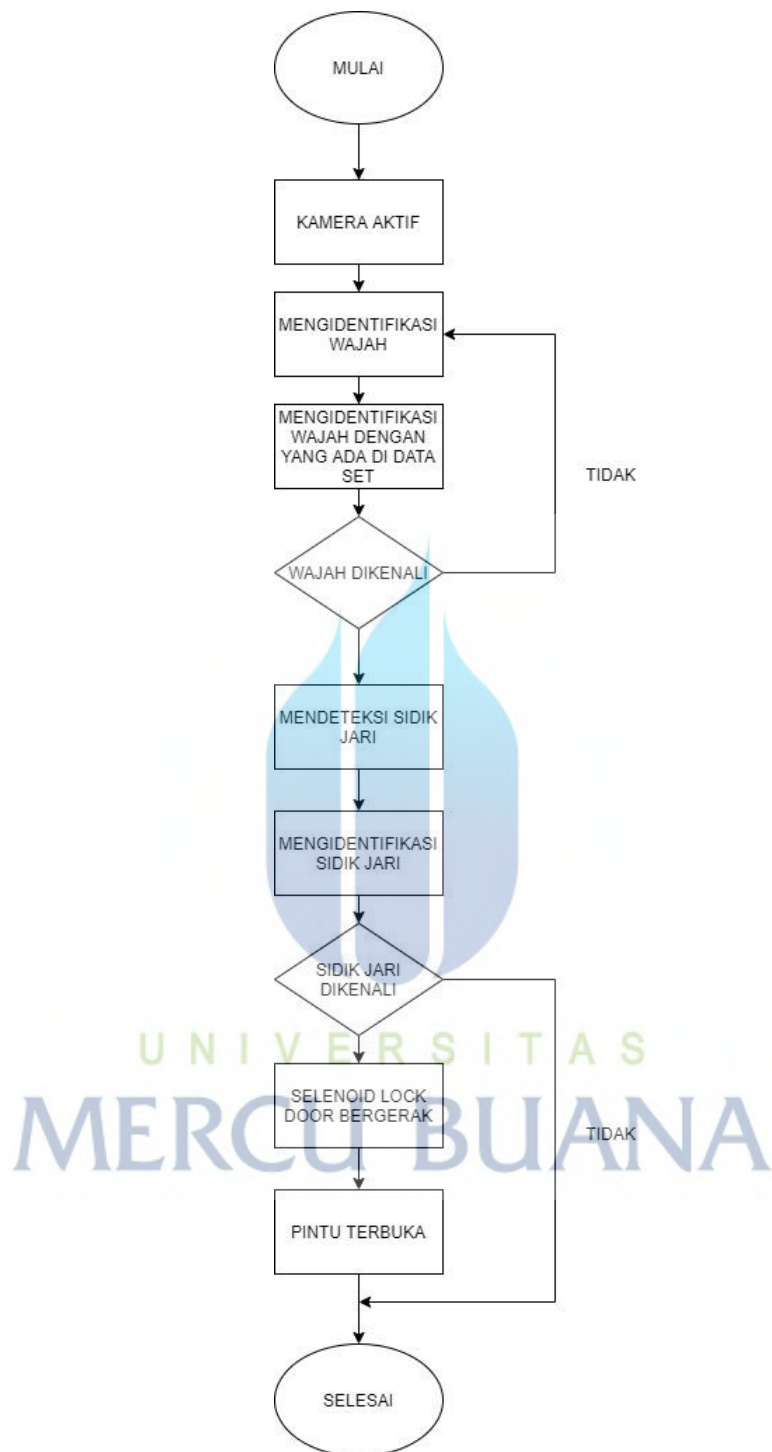
data/program dan keluaran (*output*). Bagian masukan (*input*) yaitu *Webcam / USB camera* dan *FingerPrint*. sementara untuk keluaran (*output*) yaitu *Solenoid Lock Door*. *Raspberri Pi* digunakan sebagai kontrol utama untuk mengolah program data sekaligus sebagai penyimpanan dataset untuk masukan (*input*). Fungsi dari tiap blok diagram diatas dijelaskan di bawah ini:

1. *Webcam / USB camera* sebagai inputan yang gunanya untuk menampilkan kotak video sekaligus untuk mengambil gambar wajah.
2. *FingerPrint* sebagai inputan yang digunakan untuk membaca atau mengidentifikasi sidik jari.
3. *Raspberry Pi* sebagai kontrol utama pengolahan data yang berfungsi untuk menghubungkan *Webcam / USB camera* dan *fingerprint* dengan *solenoid lock door*.
4. *Solenoid lock door* adalah motor yang digunakan untuk membuka atau menutup slot kunci pada pintu.

### **3.2 Perancangan Flow Chart**

Untuk Flow Chart perancangan pada “RANCANG BANGUN DUAL SISTEM KEAMANAN PINTU RUMAH OTOMATIS MENGGUNAKAN PENGENALAN WAJAH DAN SIDIK JARI”, ditunjukkan pada gambar 3.2

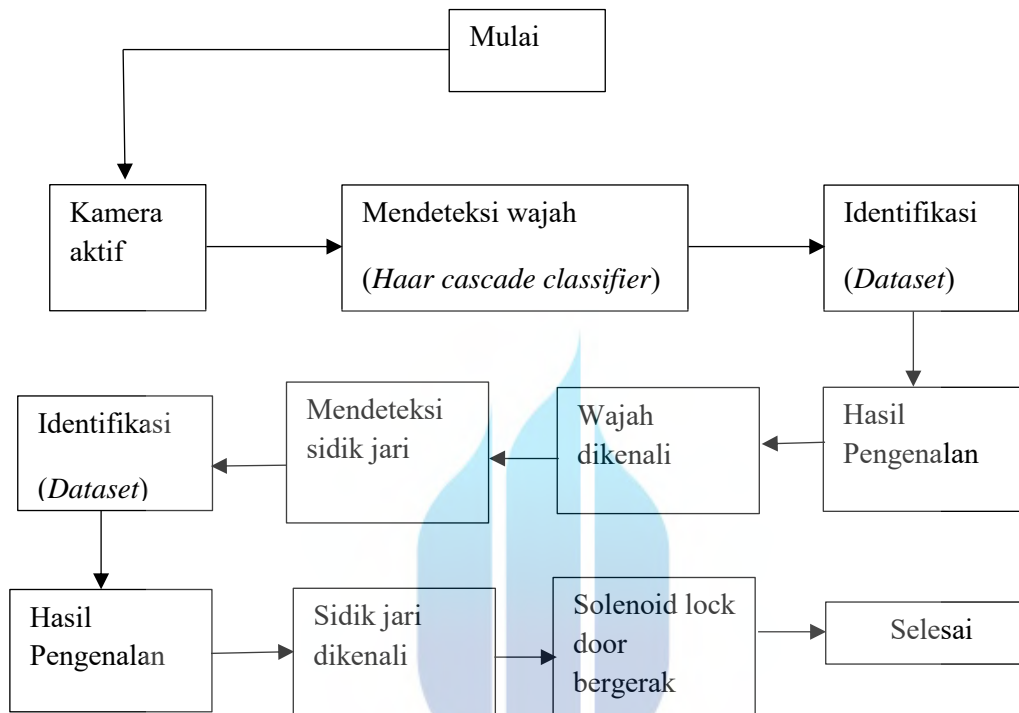
UNIVERSITAS  
MERCU BUANA



Gambar 3.2 Flow Chart

Dari hasil Flow Chart dan blok sistem dari cara kerja alat ini. Secara garis besar dapat dijelaskan dibawah ini :

#### Tahap Pengenalan



#### Tahap penyimpanan

1. Diawali dengan mulai kemudian membuka aplikasi *VNC Viewer* lalu memasukkan IP address yang sudah diintegrasikan terlebih dahulu pada *Raspberry Pi*.
2. Lalu tampilan kamera akan muncul pada layar monitor atau laptop kemudian *Webcam / USB camera* akan bekerja untuk meng-*capture* gambar .
3. Hasil gambar yang telah *capture* di simpan sebagai data masukan pada dataset.

#### Tahap pengenalan

1. Diawali dengan mulai kemudian membuka aplikasi *VNC Viewer* lalu memasukkan IP address yang sudah diintegrasikan terlebih dahulu pada *Raspberry Pi*.

2. Lalu tampilan kamera akan muncul pada layar komputer atau laptop kemudian *Webcam / USB camera* akan mendeteksi wajah.
3. Kemudian wajah yang sudah dideteksi oleh *webcam / USB camera* menggunakan metode *Haar cascade classifier* akan dibandingkan hasilnya dengan gambar yang sudah tersimpan sebelumnya pada dataset.
4. Jika wajah yang dibandingkan hasilnya sesuai dengan gambar yang sudah tersimpan maka wajah akan dikenali.
5. Jika wajah yang dibandingkan hasilnya tidak sesuai dengan gambar yang sudah tersimpan maka wajah tidak dikenali dan akan di deteksi ulang.
6. Setelah itu wajah yang sudah dikenali maka akan langsung membuka sensor sidik jari.
7. Kemudian melakukan pendeteksian sidik jari dan melakukan perbandingan dengan sidik jari yang sudah tersimpan dalam dataset.
8. Jika sidik jari yang dibandingkan hasilnya sesuai maka akan langsung memberikan sinyal kepada solenoid lock door dan pintu akan dapat dibuka.
9. Dan jika sidik jari yang dibandingkan hasilnya tidak sesuai maka program akan tertutup dan harus memulai dari tahap awal lagi.
10. Selesai.

### 3.3 Perancangan Umum Alat

Rancang bangun *dual* Sistem Keamanan Pintu Rumah Otomatis Menggunakan *Face Recognition* Dan *Fingerprint*, ini akan membahas mengenai proses perancangan mekanik serta penyusunan rangkaian untuk merealisasikan sistem alat. Adapun sistem alat yang dibuat dan dirancang sesuai blok diagram. Perancangan alat ini menggunakan *Raspberry Pi* terdapat tiga tahap perancangan yaitu :

1. Tahap mekanikal
2. Tahap elektrikal dan
3. Tahap pemrograman.

### **3.3.1 Perancangan Mekanik**

Dalam perancangan mekanik dual sistem keamanan pintu rumah otomatis ini, penulis menggunakan kayu sebagai rangka utama untuk membuat sebuah pintu. Kayu yang digunakan untuk membuat sebuah pintu mempunyai dimensi yang berbeda – beda. Pintu yang dibuat mempunyai ukuran tinggi 40 cm, panjang 25cm dan lebar 10 cm. penulis memilih badan dasar kayu agar alat ini dapat terkesan seperti pintu pada aslinya dan agar tidak mudah patah atau rusak.

### **3.3.2 Perancangan Elektrik**

Pada perancangan keamanan pintu rumah otomatis ini juga memiliki komponen – komponen elektrikal yang terhubung dari bagian input dan output. Oleh karena itu, rangkaian elektrikal ini dibagi menjadi dua bagian, yaitu :

1. Rangkaian Input
2. Rangkaian Output

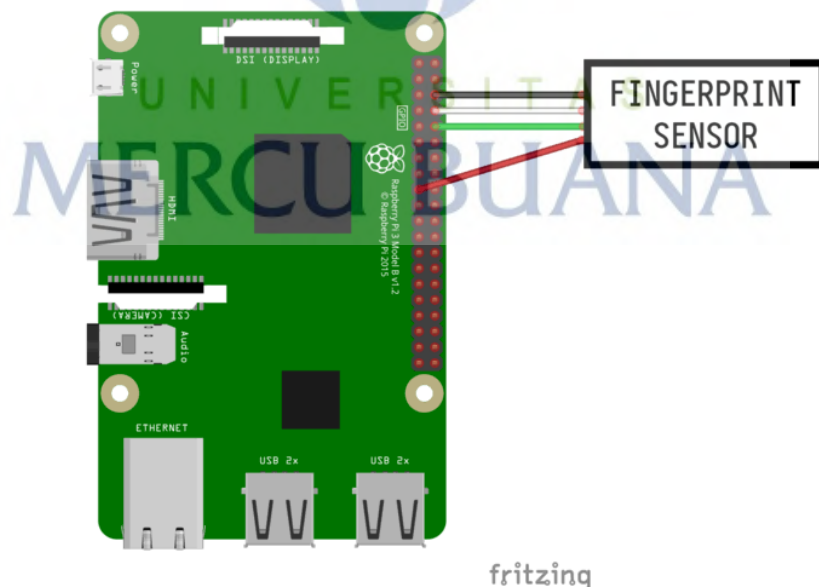
#### **3.3.2.1 Rangkaian input**

Perancangan dan pembuatan *wiring* rangkaian input dual sistem keamanan pintu rumah otomatis, meliputi prinsip kerja alat untuk sistem komunikasi *Raspberry Pi* dengan input *Webcam / USB camera* dan



prinsip kerja alat untuk system komunikasi *Raspberry Pi* dengan input sensor *fingerprint*. *Wiring* rangkaian input dual sistem keamanan pintu rumah otomatis dapat dilihat pada gambar 3.5.

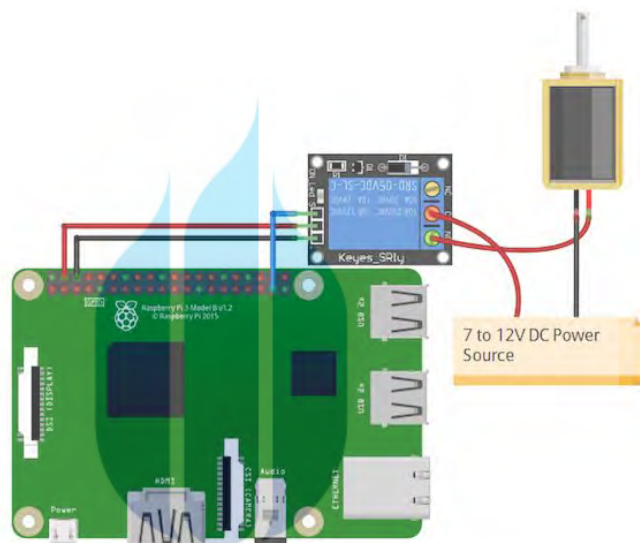
Gambar 3.5 *Wiring* Rangkaian input Sistem Keamanan Pintu Rumah Otomatis



Gambar 3.6 *Wiring* Rangkaian input Sistem Keamanan Pintu Rumah Otomatis

### 3.3.2.2. Rangkaian output

Perancangan dan pembuatan *wiring* rangkaian output sistem keamanan pintu rumah, meliputi prinsip kerja alat untuk sistem komunikasi *Raspberry Pi* dengan output *solenoid lock door*. *Wiring* rangkaian input sistem keamanan pintu rumah otomatis dapat dilihat pada gambar 3.6.



**Gambar 3.6** *Wiring* Rangkaian output Sistem Keamanan Pintu Rumah Otomatis

Pin GPIO dari *Raspberry Pi* dapat memberikan output 3.3V tetapi *solenoid lock door* membutuhkan 7-12V untuk beroperasi. Karena itu, kita perlu menggunakan sumber daya eksternal dan relay untuk mengoperasikan kunci. Hubungkan VCC dan GND dari modul relay ke 5V dan GND dari *Raspberry Pi*. Kemudian hubungkan pin sinyal dari modul relay ke GPIO 26 dari *Raspberry Pi*. Kemudian pada modul *relay*, sambungkan sumber daya DC bentuk *negatif* ke *negatif* dari *solenoid lock door*. Hubungkan positif dari sumber daya DC ke common dari modul *relay* dan kemudian menghubungkan keluaran dari modul *relay* ke *positif* dari kunci pintu *solenoid*.

Gambar 3.6 menjelaskan seluruh komponen yang dirangkai sesuai dengan pin – pin agar menjadi suatu sistem yang dapat berfungsi sesuai



dengan perencanaan. Adapun pin – pin yang digunakan pada *Raspberry Pi* dapat dilihat pada tabel 3.1 di bawah ini.

Tabel 3.1 Pin GPIO dan Port pada Raspberry Pi

NO	Pin atau Port	Keterangan
1.	Pin 5V	Relay
2.	Pin GND	Relay
3.	Pin GPIO 26	Relay
4.	Port USB 1	Webcam / USB camera
5.	Port USB 2	Sidik jari

Tabel 3.1 menjelaskan pin/port pada Raspberry Pi yang digunakan oleh komponen – komponen agar dapat terinput dan terproses dengan baik.

### 3.4 Pemrograman Perangkat Lunak

Secara garis besar alat akan bekerja sesuai dengan perintah yang berawal melalui *Webcam / USB camera* dan sidik jari sebagai inputan. Kemudian dari inputan tersebut data akan dikirim menuju *Raspberry Pi*. Lalu *Raspberry Pi* yang akan mengolah data menjadi berbagai macam perintah-perintah ke beberapa komponen pendukung kerja alat sesuai program yang dibuat, yang berakhir pada *output* pada *Solenoid lock door*.

#### Pemrograman pada Webcam / USB camera

Dalam membuat sebuah program Raspberry Pi, maka langkah yang paling dasar adalah menginisialisasikan atau mengenalkan pin yang akan digunakan. Pada pengenalan ini, pin akan ditentukan sebagai import

```
# import library yang di perlukan
from imutils.video import VideoStream
from imutils.video import FPS
import face_recognition
import argparse
import imutils
import pickle
import time
import cv2
import hashlib
from pyfingerprint.pyfingerprint import PyFingerprint
import RPi.GPIO as GPIO
```

Gambar 3.7 Pengenalan import library yang dibutuhkan

Menginisialisasi *library* yang dibutuhkan pada sistem agar *raspberry pi* bisa mengakses modul kamera dan *fingerprint* serta melakukan pengenalan pada objek.

```
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

Relay = 17
GPIO.setup(Relay, GPIO.OUT)
GPIO.output(Relay, 1)
```

Gambar 3.8 Inisialisasi Pin GPIO untuk *Relay Selenoid*

Gambar 3.8 ialah inisialisasi pin yang dibutuhkan untuk menggunakan *relay 5V* yang digunakan untuk menggerakkan *Selenoid Lockdoor*

```
while (wajahdikenali == 0):
    # Dapatkan frame yang akan diproses
    # Dapatkan FPS
    frame = vs.read()
    frame = imutils.resize(frame, width=500)

    # Konversi ke grayscale dan konversi ke hue
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # Deteksi wajah dari frame grayscale
    rects = detector.detectMultiScale(gray, scaleFactor=1.1,
    minNeighbors=5, minSize=(30, 30),
    flags=cv2.CASCADE_SCALE_IMAGE)

    # Tampilkan kotak di wajah yang dideteksi
    boxes = [(y, x + w, y + h, x) for (x, y, w, h) in rects]
    encodings = face_recognition.face_encodings(rgb, boxes)
    names = []

    # Loop di semua wajah yang terdeteksi
    for encoding in encodings:
        matches = face_recognition.compare_faces(data["encodings"],
        encoding)
        name = "Unknown"
        wajahdikenali = 0

        # Periksa apakah ada wajah yang di kenali
        if True in matches:
            matchedIdxs = [i for (i, b) in enumerate(matches) if b]
            counts = {}
            for i in matchedIdxs:
                name = data["names"][i]
                counts[name] = counts.get(name, 0) + 1
            name = max(counts, key=counts.get)
            wajahdikenali = 1
            names.append(name)
```

Gambar 3.9 Looping deteksi dan pengenalan wajah

```

## Tries to initialize the sensor
try:
    f = PyFingerprint('/dev/ttyUSB0', 57600, 0xFFFFFFFF, 0x00000000)
    if ( f.verifyPassword() == False ):
        raise ValueError('The given fingerprint sensor password is wrong!')
except Exception as e:
    print('The fingerprint sensor could not be initialized!')
    print('Exception message: ' + str(e))
    exit(1)

## Tries to search the finger and calculate hash
try:
    print('Tempelkan Jari anda...')
    ## Wait that finger is read
    while ( f.readImage() == False ):
        pass

    ## Converts read image to characteristics and stores it in charbuffer 1
    f.convertImage(0x01)

    ## Searches template
    result = f.searchTemplate()

    positionNumber = result[0]
    accuracyScore = result[1]

    if ( positionNumber == -1 ):
        print('No match found!')
        exit(0)
    else:
        sidikjari = 1
        print('Found template at position #' + str(positionNumber))
        print('Tingkat Akurasi: ' + str(accuracyScore))

    ## OPTIONAL stuff
    ##
    ## Loads the found template to charbuffer 1
    f.loadTemplate(positionNumber, 0x01)

    ## Downloads the characteristics of template loaded in charbuffer 1
    characteristics = str(f.downloadCharacteristics(0x01)).encode('utf-8')

    ## Hashes characteristics of template
    print('Sidikjari ditemukan = ' + str(sidikjari))

```

Gambar 3.10 Fungsi Deteksi Sensor Sidik Jari

```

if ((wajahdikenali == 1)and(sidikjari == 1)):
    print('Pintu Terbuka')
    GPIO.output(Relay,0)
    exit(1)
else:
    GPIO.output(Relay,1)

```

Gambar 3.11 Fungsi Untuk Menggerakkan Selenoid Lockdoor

*Solenoid Lockdoor* akan bergerak apabila dari sensor *Fingerprint* dan Camera berhasil mengenali objek.