

BAB III

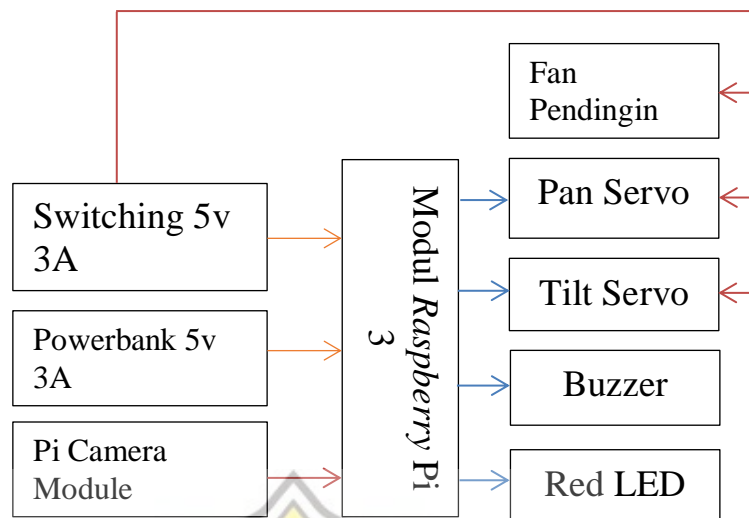
PERANCANGAN ALAT

3.1. Pendahuluan

Pada bab ini akan dijelaskan mengenai perancangan perangkat keras (*hardware*) yang dibutuhkan untuk konstruksi sistem *Pan-Tilt Camera* beserta komponen yang lain seperti *power supply* yang digunakan, perangkat *output* tambahan seperti lampu LED dan *buzzer* kemudian perangkat lunak (*software*) seperti *Raspbian OS* sebagai sistem operasi, *VNC Viewer* sebagai aplikasi untuk mengakses *Raspberry Pi*, *OpenCV* sebagai *library* utamanya serta penggunaan bahasa pemrograman python yang mendukung untuk sistem pendeteksi objek berdasarkan warna yang sudah ditentukan.

3.2. Perancangan *Hardware* yang akan digunakan

Pada alat ini menggunakan input atau sensor utama yaitu *Raspberry Pi Camera* yang dikoneksikan langsung ke *Raspberry Pi* dan beberapa output berupa dua buah servo untuk mekanisme *Pan-Tilt* yang menggerakkan kamera. Berikut pada (Gambar-3.1) merupakan diagram rancangan alat yang dibuat sebagai berikut:



Gambar-3.1 Diagram Perancangan alat

Dua servo pada *Pan-Tilt* yang terdiri dari servo pertama bergerak ke kanan atau kekiri sebanyak 180 derajat dan servo kedua bergerak ke atas dan kebawah sebanyak 90 derajat menyesuaikan dengan rangka konstruksi sistem *Pan-Tilt*.

Servo pertama yang digunakan yaitu micro servo SG90 dan servo kedua menggunakan micro servo MG90 yang sama-sama berbobot 9 gram dengan torka 1,8 kgf.cm seperti pada (Gambar-3.2).



(a)



(b)

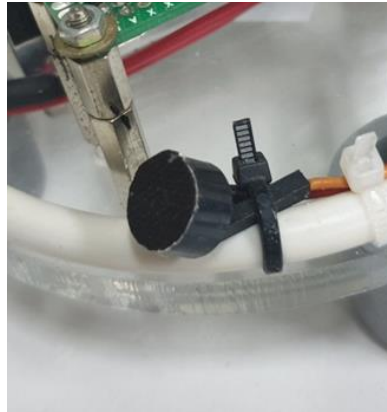
Gambar-3.2 Servo (a) SG90 (b) MG90

Sumber : <http://buaya-instrument.com/analog-servo-towerpro-sg90-9g-16kg-12sec-micro-mssg90.html>

Kedua servo beserta konstruksi *Pan-Tilt* ini menyesuaikan dengan bobot *Raspberry Pi Camera* yang berukuran 2,5 x 2,4 cm dengan tebal 9 mm yang ringan serta lampu led dan *buzzer* pada (Gambar-3.2) sebagai penanda ketika objek yang dituju sudah terdeteksi.



(a)



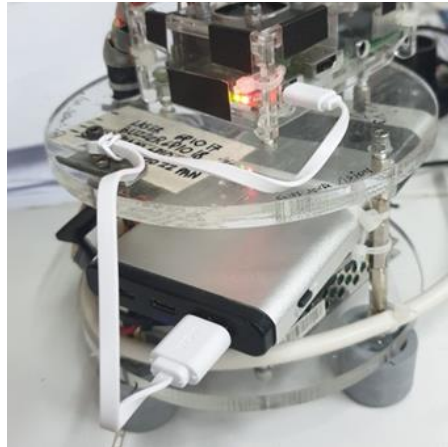
(b)

Gambar-3.3 (a) Lampu LED dan (b) Buzzer yang terpasang

Seperti pada (Gambar-3.4), perancangan alat ini menggunakan dua *power supply* 5V 3A yang terpisah, yang pertama *power supply switching* AC ke DC sebagai power untuk servo *pan-tilt* dan yang kedua menggunakan powerbank sebagai power untuk *Raspberry Pi* beserta kamera modulnya. Dikarenakan agar ketika terjadi listrik mati maka *Raspberry Pi* tidak akan langsung mati yang beresiko menyebabkan *corrupt file* pada sistem perangkat lunak *Raspberry Pi* akibat tidak di *shutdown* terlebih dahulu.



(a)



(b)

Gambar-3.4 Power Supply (a) Switching 5V 3A (b) Powerbank 5v 3A

Tabel-3.1 Data Input dan Output pada Raspberry Pi

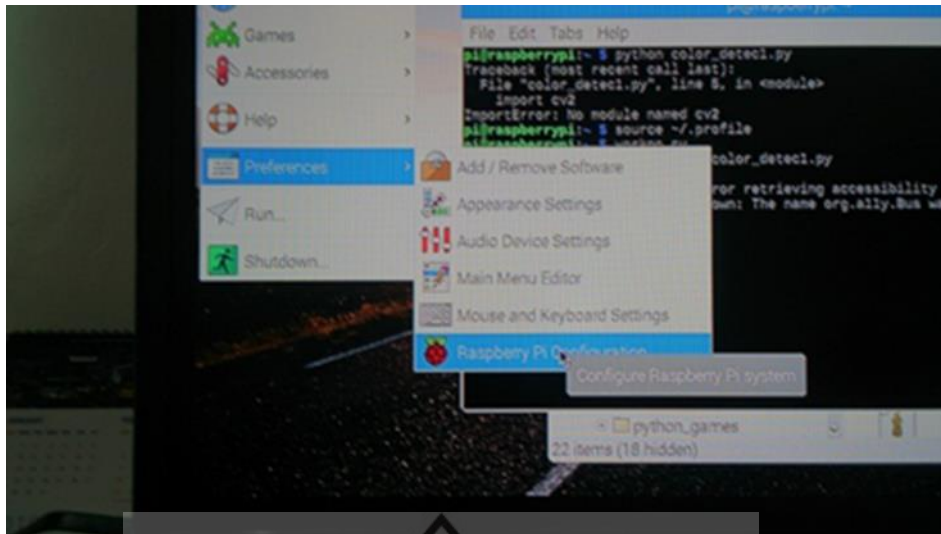
PIN GPIO	JENIS	KOMPONEN
GPIO17	Output	Red LED
GPIO18	Output	Buzzer
GPIO22	Output	Pan Servo
GPIO27	Output	Tilt Servo
GND	Output	Red LED
GND	Output	Buzzer
CSI Port	Input	Pi Camera

(Tabel-3.1) diatas merupakan data input dan output pada komponen yang akan dipasang ke *Raspberry Pi*. Pada power dan ground dari 2 buah servo di atas berasal dari *power supply switching 5v 3A* sedangkan port PWM servo dihubungkan pada Pin GPIO pada *Raspberry Pi*. Power pada *Buzzer* dan *Red LED* dihubungkan langsung pada Pin GPIO dan *ground* pada *Raspberry Pi*. Setelah seluruh komponen sudah terpasang dengan benar maka selanjutnya

mempersiapkan *software* atau perangkat-perangkat lunak yang dibutuhkan oleh *Raspberry Pi*.

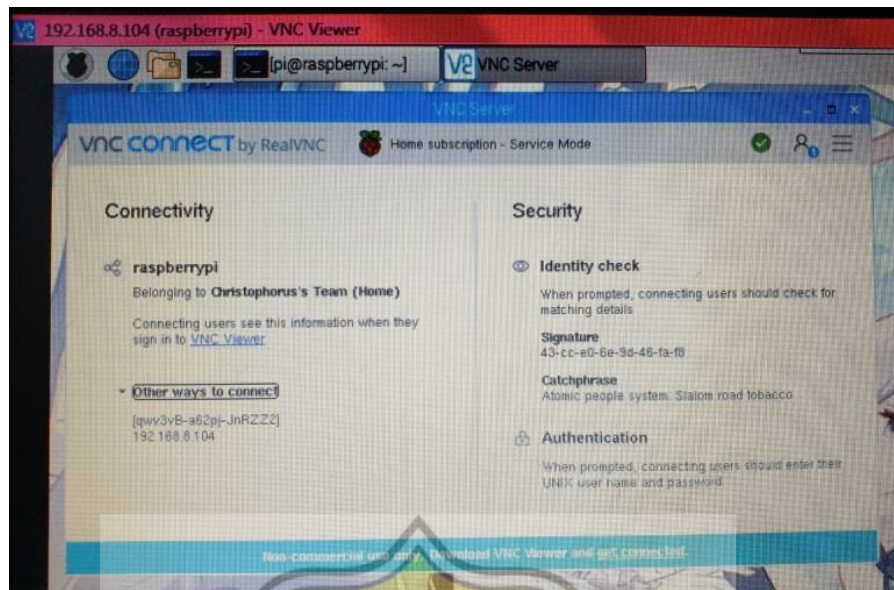
3.3. Persiapan *software* yang akan digunakan

Raspberry Pi merupakan perangkat *Mini Personal Computer* sehingga berbeda dari *mikrokontroller* pada umumnya seperti *Arduino*. Karena itu perangkat ini membutuhkan sistem operasi khusus yaitu *Raspbian OS* yang harus diinstal terlebih dahulu ke *MicroSD Card* yang memang berperan seperti *Harddisk* pada *Personal Computer* ataupun laptop pada umumnya. Setelah dilakukan penginstalan sistem operasi selanjutnya dilakukan pengecekan dengan menghubungkan perangkat *Raspberry Pi* ke layar monitor yang dapat diakses melalui kabel *HDMI (High Definition Multimedia Interface)*. Setelah dipastikan Sistem Operasi dapat berjalan dengan baik, lalu sesuaikan resolusi yang diinginkan pada *Raspberry Pi Preferences* kemudian *Raspberry Pi Configuration* seperti pada (Gambar-3.5). Pada settingan *Raspberry Pi Configuration* aktifkan kamera agar kamera dapat menampilkan *display* ketika diaktifkan. Settingan juga dapat kita rubah melalui perintah *Open LXTerminal* yang juga tersedia pada *Raspberry Pi*.

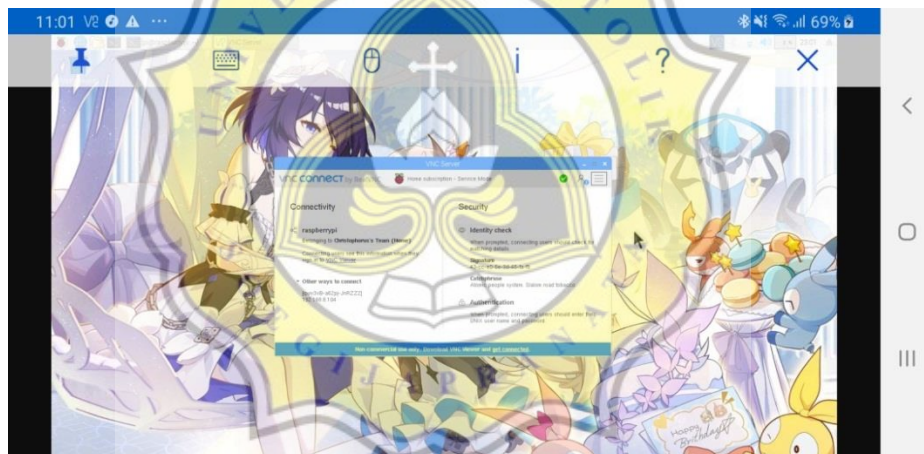


Gambar-3.5 Tampilan *Raspberry Pi Configuration* pada OS *Raspbian*

Selain akses dengan layar monitor, *Raspberry Pi* juga dapat diakses menggunakan ponsel *Android* maupun *Windows PC* dengan aplikasi bernama *VNC Viewer*. Aplikasi ini digunakan untuk mengakses antar gadget elektronik seperti *Raspberry Pi* dari jarak jauh dengan *IP Address* dan konektivitas WiFi yang sama sehingga tidak lagi memerlukan layar monitor. Penginstallannya cukup mudah dengan mengikuti tutorial dari *website* resminya untuk *Raspberry Pi* dan ponsel *Android* maupun *PC* sebagai pengganti *display* monitornya seperti yang ditampilkan pada (Gambar-3.6) .



(a)

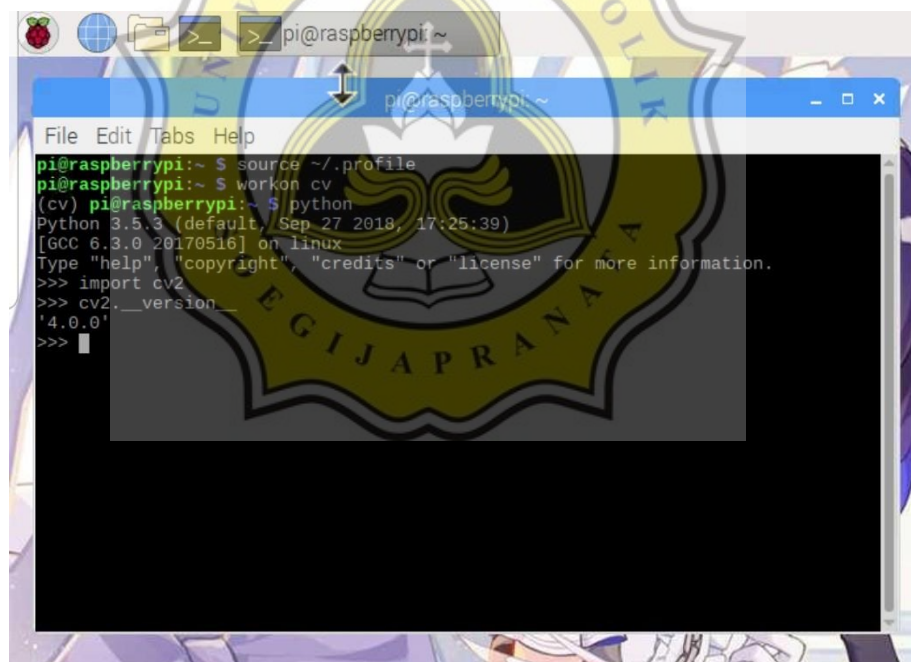


(b)

Gambar-3.6 VNC Viewer yang tekoneksi ke (a) PC (b) Ponsel Android

Setelah akses sudah sesuai dengan yang diinginkan, saatnya melakukan penginstalan aplikasi program *Python*, *Library Open CV* dan *Numpy*. *Python* biasanya sudah tersedia dan siap digunakan dalam *OS Raspbian* ketika awal penginstalan. Berbeda dengan *OpenCV* dan *Numpy*, aplikasi ini harus diinstal terlebih dahulu agar dapat kompatibel dengan program *Python*. Penginstalan

OpenCV dan *Numpy* biasanya harus dilakukan bersamaan sehingga memerlukan tata cara dan waktu yang cukup lama (hingga 4 jam), tata caranya dapat dicari melalui banyak sumber di internet. Untuk memastikan *OpenCV* dapat terinstall dengan baik maka kita dapat melakukan perintah pada *LXTerminal* dengan memberi perintah `source ~/.profile` untuk membuka folder dimana *library* disimpan kemudian `workon cv` untuk mengaktifkan *library OpenCV* setelah itu muncul huruf (cv) pada terminal. Untuk mengecek versi dari *OpenCV* dapat dicoba dengan mengetik *Python* kemudian `import cv2` dan `cv2.__version__`. TamPilannya akan muncul seperti pada (Gambar-3.7).

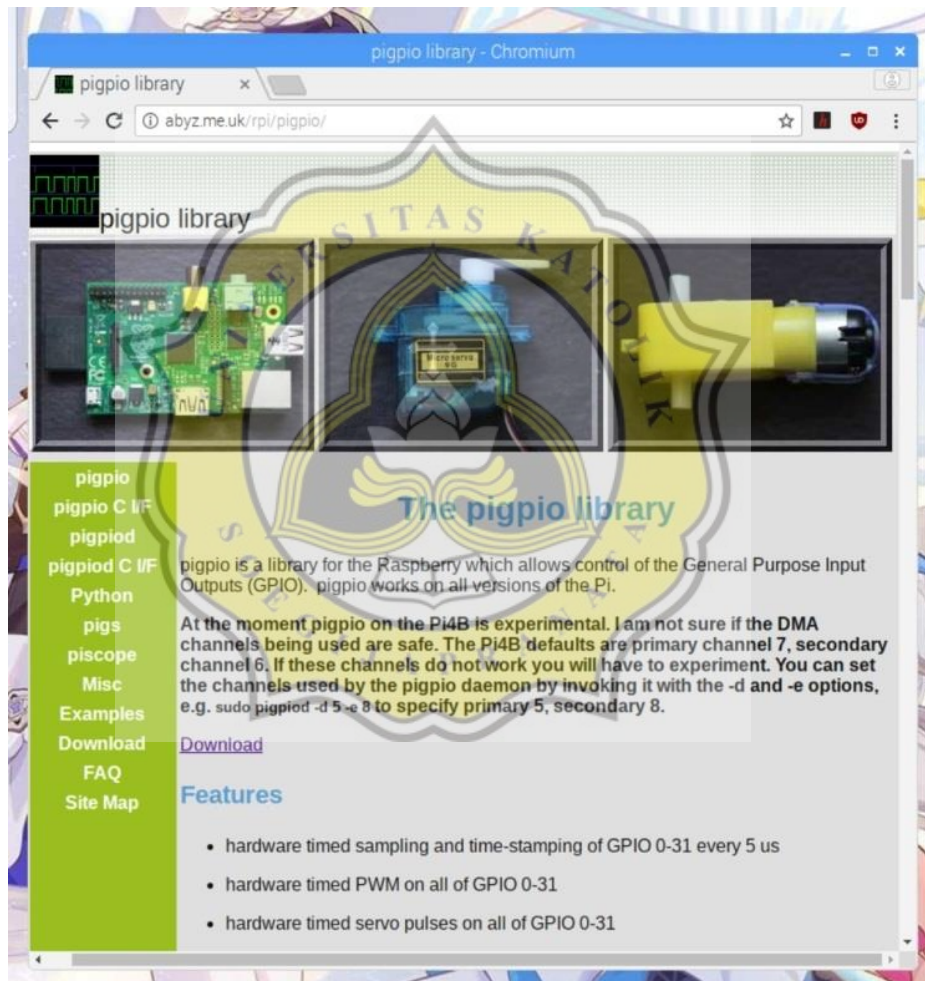


```
pi@raspberrypi:~ $ source ~/.profile
pi@raspberrypi:~ $ workon cv
(cv) pi@raspberrypi:~ $ python
Python 3.5.3 (default, Sep 27 2018, 17:25:39)
[GCC 6.3.0 20170516] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.0.0'
>>>
```

Gambar-3.7 Perintah untuk mengaktifkan *OpenCV*

Setelah itu tak lupa juga dilakukan penginstallan *library* khusus bernama *Pigpio*. *Library* ini berfungsi sebagai *filter* untuk membantu kekurangan *Raspberry Pi* di mana Pin *GPIO* pada *Raspberry Pi* memiliki delay tidak seperti

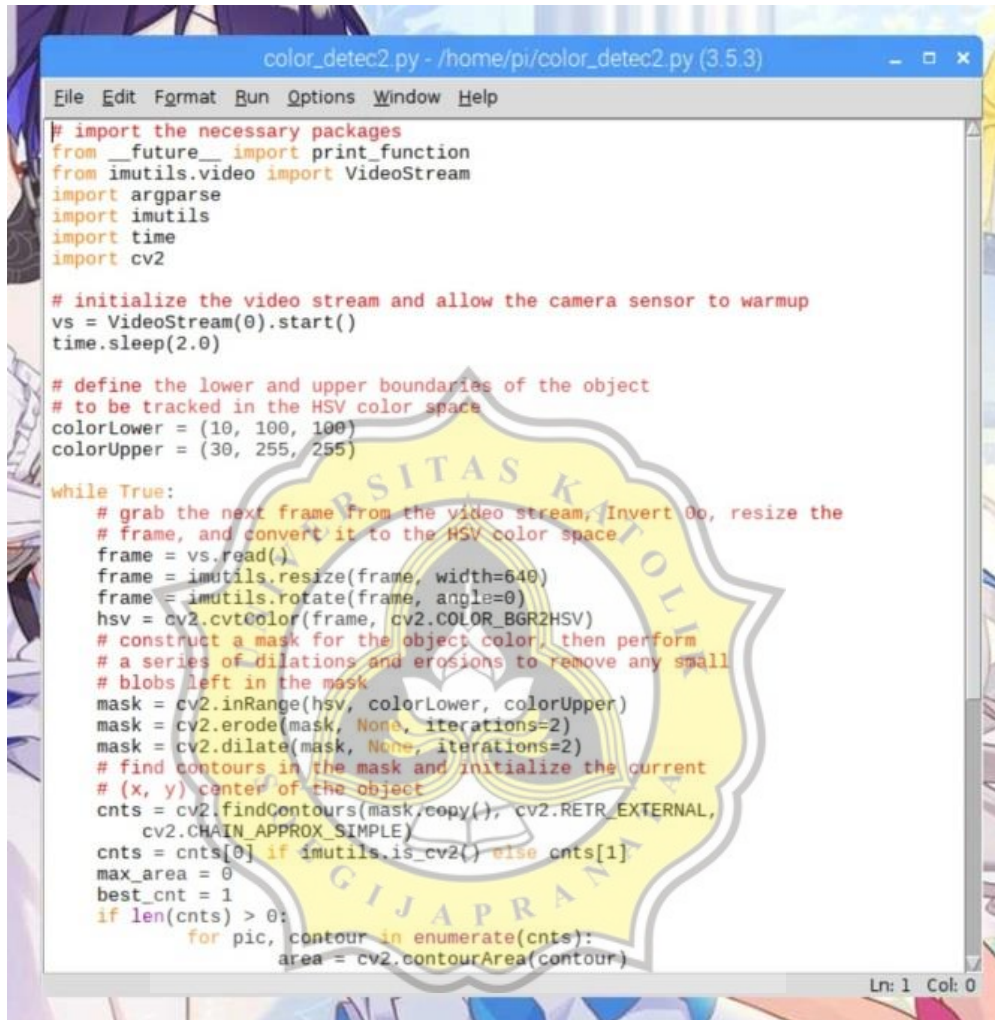
mikrokontroller seperti *Arduino* pada umumnya, karena *Raspberry Pi* juga mengolah data pada sistem operasinya sehingga sedikit terbebani. Seperti halnya *OpenCV*, tata cara penginstalan *Pigpio* cukup mudah dan tersedia di website resminya. Berikut pada (Gambar-3.8) merupakan tampilan *library Pigpio* yang dibuat khusus untuk pemrograman *Raspberry Pi*.



Gambar-3.8 Website Resmi Library Pigpio

Setelah berbagai macam *software* yang diperlukan sudah terinstall pada *Raspberry Pi*, selanjutnya pemrograman untuk bisa langsung dibuat menggunakan

software Python yang sudah terinstall sebelumnya. Berikut (Gambar-3.9) merupakan tampilan *software* pemrograman *Python* pada *Raspberry Pi*.



```
color_detec2.py - /home/pi/color_detec2.py (3.5.3)
File Edit Format Run Options Window Help
# import the necessary packages
from __future__ import print_function
from imutils.video import VideoStream
import argparse
import imutils
import time
import cv2

# initialize the video stream and allow the camera sensor to warmup
vs = VideoStream(0).start()
time.sleep(2.0)

# define the lower and upper boundaries of the object
# to be tracked in the HSV color space
colorLower = (10, 100, 100)
colorUpper = (30, 255, 255)

while True:
    # grab the next frame from the video stream, invert 00, resize the
    # frame, and convert it to the HSV color space
    frame = vs.read()
    frame = imutils.resize(frame, width=640)
    frame = imutils.rotate(frame, angle=0)
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    # construct a mask for the object color, then perform
    # a series of dilations and erosions to remove any small
    # blobs left in the mask
    mask = cv2.inRange(hsv, colorLower, colorUpper)
    mask = cv2.erode(mask, None, iterations=2)
    mask = cv2.dilate(mask, None, iterations=2)
    # find contours in the mask and initialize the current
    # (x, y) center of the object
    cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
    cnts = cnts[0] if imutils.is_cv2() else cnts[1]
    max_area = 0
    best_cnt = 1
    if len(cnts) > 0:
        for pic, contour in enumerate(cnts):
            area = cv2.contourArea(contour)
```

Gambar-3.9 Tampilan pemrograman *Python* pada *Raspberry Pi*

Setelah pembuatan program *Python* selesai dibuat, dilakukan pembuatan file *bash shell script* atau yang disebut dengan file dengan format *.sh*. File dengan format ini berisi bahasa pemrograman yang digunakan untuk menyusun perintah-perintah untuk menjalankan suatu aplikasi tertentu. Seperti sebelumnya, ketika ingin menjalankan program yang berisi *library OpenCV*, maka harus diberi perintah `source ~/.profile` untuk membuka folder *library* dan workon

cv untuk mengaktifkan *library* pada *LXTerminal*. Pada kali ini, file *.sh* yang dibuat akan berisi perintah-perintah diatas secara bersamaan dengan program *Python* yang sudah dibuat sebelumnya, sehingga hanya dengan memberikan satu perintah nama file *.sh* dengan contoh `./on_reboot.sh` tersebut pada *LXTerminal*, maka program akan dapat langsung berjalan. Berikut pada (Gambar -3.10) merupakan isi program file *.sh* untuk menjalankan program dalam penelitian ini.



```
<on_reboot.sh>
File Edit Search Options Help
#!/bin/bash
source /home/pi/.profile
workon cv
cd /home/pi/pi-reboot
sudo python3 detec_final2.py
```

Gambar-3.10 Isi program file *.sh* yang sudah dibuat