

LAMPIRAN A
(HASIL PENGUJIAN FUNGSIONALITAS
SISTEM)

LAMPIRAN B
(LISTING PROGRAM)

```
from __future__ import print_function

from imutils.video import VideoStream

import argparse

import imutils

import cv2

import cv2 as cv

import numpy as np

import RPi.GPIO as GPIO

import pigpio

import time

import os

os.system ("sudo modprobe bcm2835-v4l2")

time.sleep(1)

# construct the argument parse and parse the arguments

ap = argparse.ArgumentParser()

ap.add_argument("-p", "--picamera", type=int, default=-1,

                help="whether or not the Raspberry Pi camera should be used")

args = vars(ap.parse_args())

# initialize the video stream and allow the cammera sensor to

# warmup

print("[INFO] waiting for camera to warmup...")

vs = VideoStream(usePiCamera=args["picamera"] > 0).start()
```

```
time.sleep(2.0)

# define the lower and upper boundaries of the "blue red yellow"

# ball in the HSV color space

blueLower = (100, 50, 50)

blueUpper = (120, 255, 255)

redLower = (160,100,100)

redUpper = (180,255,255)

yellowLower = (27,100,100)

yellowUpper = (30,255,255)

# initialize

#ESC

MOTOR = 17

pi = pigpio.pi()

SERVO = 14

#SERVO

pi1 = pigpio.pi()

pi1.set_servo_pulsewidth(SERVO, 1500)

time.sleep(1)

Muter = False

#BLUE

# find contours in the mask and initialize the current

# (x, y) center of the ball

cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
```

```

cv2.CHAIN_APPROX_SIMPLE)

cnts = cnts[0] if imutils.is_cv2() else cnts[1]

center = None

# only proceed if at least one contour was found
if len(cnts) > 0:

    # find the largest contour in the mask, then use
    # it to compute the minimum enclosing circle and
    # centroid
    c = max(cnts, key=cv2.contourArea)

    ((x, y), radius) = cv2.minEnclosingCircle(c)

    M = cv2.moments(c)

    center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))

# only proceed if the radius meets a minimum size
if radius > 10:

    # draw the circle and centroid on the frame
    cv2.circle(frame, (int(x), int(y)), int(radius),
                (255,0,0), 2)

    cv2.circle(frame, center, 5, (255,0,0), -1)

# if the led is not already on, raise an alarm and
# turn the LED on
if not Muter:

    #ESC

```

```

        pi.set_servo_pulsewidth(MOTOR, 1300)
        time.sleep(1)
        #SERVO
        pi1.set_servo_pulsewidth(SERVO, 1500)
        time.sleep(1)
        Muter = True

# if the ball is not detected, turn off the Motor
elif Muter:
    pi.set_servo_pulsewidth(MOTOR, 1000)
    time.sleep(1)
    ##         pi1.set_servo_pulsewidth(SERVO, 1500)
    ##         time.sleep(1)
    Muter = False

# show the frame to our screen
        cv2.imshow("Frame", frame)
        key = cv2.waitKey(1) & 0xFF

        # if the 'q' key is pressed, stop the loop
        if key == ord("q"):
            break

# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()

```

LAMPIRAN C
(SPESIFIKASI ALAT)

RASPBERRY PI

- SoC: Broadcom BCM2836 (CPU, GPU, DSP, SDRAM)
- CPU: 900 MHz quad-core ARM Cortex A7 (ARMv7 instruction set)
- GPU: Broadcom VideoCore IV @ 250 MHz
- GPU info: OpenGL ES 2.0 (24 GFLOPS); 1080p30 MPEG-2 and VC-1 decoder (with license); 1080p30 h.264/MPEG-4 AVC high-profile decoder and encoder
- Memory: 1 GB (shared with GPU)
- USB ports: 4
- Video input: 15-pin MIPI camera interface (CSI) connector
- Video output: HDMI, composite video (PAL and NTSC) via 3.5 mm jack
- Audio input: I²S
- Audio output: Analog via 3.5 mm jack; digital via HDMI and I²S
- Storage: MicroSD
- Network: 10/100Mbps Ethernet
- Peripherals: 17 GPIO plus specific functions, and HAT ID bus
- Power rating: 800 mA (4.0 W)
- Power source: 5 V via MicroUSB or GPIO header
- Ukuran: 85.60mm × 56.5mm
- Bobot: 45g (1.6 oz)

BRUSHLESS MOTOR

Specs:

- **Length:** 60m
- **Diameter:** 28mm
- **Kv:** 3300rpm/v
- **Shaft size:** 3.17mm
- **Max Current:** 60amps
- **Volts:** 11.1
- **Watts:** 660

ELECTRONIC SPEED CONTROL

Specs:

- **Constant Current:** 50A
- **Burst Current:** 70A
- **Battery:** 2-6S Lipoly

- **BEC:** 5.5v / 4A
- **Motor Type:** Sensorless Brushless
- **Cooling:** Watercooled
- **Size:** 87 x 38 x 22mm
- **Weight:** 105g

Programming Functions:

- **Running Mode:** Forward with brake / Forward and reverse with brake / Forward and reverse
- **Acceleration:** 6% / 9% / 12%
- **Low Voltage Protection:** None / 2.6V / 2.8V / 3.0V / 3.2V / 3.4V
- **Start Mode (Punch):** Level 1 / Level 2 / Level 3 / Level 4 / Level 5 / Level 6 / Level 7 / Level 8 / Level 9
- **Reverse Force:** 25% / 50% / 75% / 100%
- **Timing:** 0° / 5° / 10° / 15° / 20° / 25° / 30° / Automatic

LAMPIRAN D
(DOKUMENTASI)





