LAMPIRAN 1 SOURCE CODE

```
#!/usr/bin/python
# coding: utf-8
import cv2.cv as cv
import cv2 as cv2
import time
import numpy as np
import RPi.GPIO as gpio

# Menggunakan gpio tipe board
gpio.setmode(gpio.BOARD)
# Matikan peringatan
gpio.setwarnings(False)
#--------------------------------------
# Menyatakan pin GPIO sebagai output - Motor A

# Mesin aktivasi pin via Rasp 1
gpio.setup(7, gpio.OUT)

# Mesin aktivasi pin via Rasp 2
gpio.setup(11, gpio.OUT)

# Mulai pin 13 sebagai output - Motor A
gpio.setup(13, gpio.OUT)

# Mulai pin 15 sebagai output - Motor A
gpio.setup(15, gpio.OUT)

#--------------------------------------
# Menyatakan pin GPIO sebagai output - Motor B

# Motor B melalui aktivasi pin Rasp 1
gpio.setup(26, gpio.OUT)

# Motor B melalui aktivasi pin Rasp 2
gpio.setup(16, gpio.OUT)

# Mulai Pin 5 sebagai output - Motor B
gpio.setup(18, gpio.OUT)

# Mulai pin 22 sebagai output - Motor B
gpio.setup(22, gpio.OUT)

#--------------------------------------
# Memungkinkan L298N dikendalikan oleh GPIO:
#--------------------------------------
# nilai awal - Benar - Motor yang diaktifkan
```

```
gpio.output(7, True) #Motor A - Rasp 1
gpio.output(11, True) #Motor A - Rasp 2
#--------------------------------------
# nilai awal - Benar - Motor B diaktifkan
gpio.output(26, True) #Motor B - Rasp 1
gpio.output(16, True) #Motor B - Rasp 2
#--------------------------------------

# Maju
def Frente():
# Motor 1
 gpio.output(13, False)
 gpio.output(15, True)
# Motor 2
 gpio.output(18, False)
 gpio.output(22, True)

# Mundur
def Tras():
# Motor 1
 gpio.output(13, True)
 gpio.output(15, False)
# Motor 2
 gpio.output(18, True)
 gpio.output(22, False)

# Berhenti
def Parar():
# Motor 1
 gpio.output(18, False)
 gpio.output(22, False)
# Motor 2
 gpio.output(13, False)
 gpio.output(15, False)

# Kanan
def Direita():
# Motor 1
 gpio.output(13, True)
 gpio.output(15, False)
# Motor 2
 gpio.output(18, False)
 gpio.output(22, True)

# Kiri
def Esquerda():
# Motor 1
 gpio.output(13, False)
```

```
 gpio.output(15, True)
# Motor 2
 gpio.output(18, True)
 gpio.output(22, False)

# Jarak mendeteksi objek
def ajusteZ(area):
  if(area<=120):
     Frente()
  elif(area>=600):
     Tras()
  else:
     Parar()

#---------------------------------------------------------------------------
#
#---------------------------------------------------------------------------
# parameter:
# x -> posisi saat ini pada sumbu X
# max -> limit sentral
# centrox -> posisi statis pusat

# Fungsi Belok
def ajusteX(x, max, centrox):
          # Jika jarak di sisi kanan lebih besar dari ambang batas dibuat jarak
       if (x - centrox) > max:
          cv2.line(entrada, (int(x),int(y)), (centrox,centroy),(0,0,255), 1)
        Direita()
          time.sleep(0.05)
          Parar()

          # Jika jarak di sisi kiri lebih besar dari pengaturan ambang batas dibuat
       elif (centrox - x) > max:
          cv2.line(entrada, (int(x),int(y)), (centrox,centroy),(0,0,255), 1)
        Esquerda()
          time.sleep(0.05)
          Parar()
       else:
          Parar()
#---------------------------------------------------------------------------

# coklat
#Hmin = 33
#Hmax = 17
#Smin = 100
#Smax = 255
#Vmin = 56
#Vmax = 144
```

```
# hijau
#Hmin = 42
#Hmax = 92
#Smin = 62
#Smax = 255
#Vmin = 63
#Vmax = 235

# Merah
Hmin = 0
Hmax = 179
Smin = 131
Smax = 255
Vmin = 126
Vmax = 255


# Menciptakan HSV nilai array (minimum dan maksimum)
rangeMin = np.array([Hmin, Smin, Vmin], np.uint8)
rangeMax = np.array([Hmax, Smax, Vmax], np.uint8)

# fungsi pengolahan gambar
def processamento(entrada):
    imgMedian = cv2.medianBlur(entrada,5)
    imgHSV = cv2.cvtColor(imgMedian,cv2.cv.CV_BGR2HSV)
    imgThresh = cv2.inRange(imgHSV, rangeMin, rangeMax)
    imgErode = cv2.erode(imgThresh, None, iterations = 3)
    return imgErode
#--------------------------------------------------------------------------

cv.NamedWindow("Input")
#cv.NamedWindow("HSV")
#cv.NamedWindow("Thre")
cv.NamedWindow("Erosao")
capture = cv2.VideoCapture(0)
```

```
#-----------------------------------------------------------------------
# S E T T I N G A N   L A Y A R / J A R A K
#-----------------------------------------------------------------------
# Ukuran parameter pengambilan gambar
largura = 160
altura = 120

# luas minimum untuk dideteksi
minArea = 50 # Sekitar 80 cm

# Sumbu Pusat
centrox = largura/2
centroy = altura/2

# Limit Sentral
max = largura/2.5  # 64 pixels
#-----------------------------------------------------------------------



# Mengatur ukuran untuk frame (membuang Piramida Bawah)
if capture.isOpened():
  capture.set(cv2.cv.CV_CAP_PROP_FRAME_WIDTH, largura)
  capture.set(cv2.cv.CV_CAP_PROP_FRAME_HEIGHT, altura)


while True:
    ret, entrada = capture.read()
    imagem_processada = processamento(entrada)
    moments = cv2.moments(imagem_processada, True)
    area = moments['m00']
    if moments['m00'] >= minArea:
     x = moments['m10'] / moments['m00']
     y = moments['m01'] / moments['m00']
     cv2.circle(entrada, (int(x), int(y)), 5, (0, 255, 0), -1)
     cv2.line(entrada, (int(x),int(y)), (int(centrox),int(centroy)),(0,255,0), 1)

     if(area<=120):
      cv2.circle(entrada, (int(x), int(y)), 5, (255, 0, 0), -1)
      Frente()
     elif(area>=600):
      cv2.circle(entrada, (int(x), int(y)), 5, (0, 0, 255), -1)
      Tras()
     else:
      Parar()
      ajusteX(x, max, centrox)
    else:
     Parar()
```

```
cv2.imshow("Input",entrada)
#cv2.imshow("HSV", imgHSV)
#cv2.imshow("Thre", imgThresh)
cv2.imshow("Erosao", imagem_processada)

if cv.WaitKey(10) == 27:
   break
      cv.DestroyAllWindows()
      gpio.cleanup()
```