

LAMPIRAN

Program Object Tracking

```
import cv2
import numpy as np
import math
from tkinter import *
from imutils.video import VideoStream
import RPi.GPIO as GPIO
from time import sleep

##### Fungsi untuk Pengaturan PiCamera
#####

usingPiCamera = True

#frameSize = (320, 240)
frameSize = (640, 360)

cap = VideoStream(src=0, usePiCamera=usingPiCamera, resolution=frameSize,
                  framerate=32).start()
sleep(2.0)

b=0

teks1=None
teks2=None
teks3=None
teks4=None
teks5=None
teks6=None
teks7=None
#define Servos GPIO

Servo_atas= 32
Servo_bawah = 33

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(Servo_atas, GPIO.OUT)
GPIO.setup(Servo_bawah, GPIO.OUT)

##### Fungsi untuk Menyimpan Data Excel
#####

def tampilan():
    global c
    c= teks1.get()
    root.t1.set(c)
    f=open("Percobaan" + str(c) + ".csv", "a")
    f.write(str('Posisi_Target'))
    f.write(', ')
    f.write(str('Posisi_Laser'))
    f.write("\r\n")

def save(data3):
```



```

global c
c= teks1.get()
f=open("Percobaan" + str(c) + ".csv", "a")
f.write(str(data3))
f.write("\r\n")
f.flush()
f.close()

##### Fungsi untuk Pengaturan Servo
#####

def setServoAngle(ServoA):
    assert ServoA >=0 and ServoA <= 180
    pwm = GPIO.PWM(Servo_atas, 50)
    pwm.start(0)
    #    dutyCycle= 1./18.*(ServoA)+2.5
    dutyCycle= 1./18.*(ServoA)
    pwm.ChangeDutyCycle(dutyCycle)
    sleep(0.36)
    pwm.stop()

def setServoAngle2(ServoB):
    assert ServoB >=0 and ServoB <= 180
    pwm1 = GPIO.PWM(Servo_bawah, 50)
    pwm1.start(0)
    #    dutyCycle= 1./18.*(ServoB)+2
    dutyCycle= 1./18.*(ServoB)
    pwm1.ChangeDutyCycle(dutyCycle)
    sleep(0.36)
    pwm1.stop()

##### Fungsi untuk Tampilan Gambar
#####

def stackImages(scale,imgArray):
    rows = len(imgArray)
    cols = len(imgArray[0])
    rowsAvailable = isinstance(imgArray[0], list)
    width = imgArray[0][0].shape[1] #jika mau 4 tampilan gambar nilai shape
    height = imgArray[0][0].shape[0]
    diganti '1'
    if rowsAvailable:
        for x in range ( 0, rows):
            for y in range(0, cols):
                if imgArray[x][y].shape[:2] == imgArray[0][0].shape[:2]:
                    imgArray[x][y] = cv2.resize(imgArray[x][y], (0, 0),
None, scale, scale)
                else:
                    imgArray[x][y] = cv2.resize(imgArray[x][y],
(imgArray[0][0].shape[1], imgArray[0][0].shape[0]), None, scale, scale)
                    if len(imgArray[x][y].shape) == 2: imgArray[x][y]=
cv2.cvtColor( imgArray[x][y], cv2.COLOR_GRAY2BGR)
                    imageBlank = np.zeros((height, width, 3), np.uint8)
                    hor = [imageBlank]*rows
                    hor_con = [imageBlank]*rows
                    for x in range(0, rows):
                        hor[x] = np.hstack(imgArray[x])
                    ver = np.vstack(hor)
            else:
                for x in range(0, rows):

```

```

        if imgArray[x].shape[:2] == imgArray[0].shape[:2]:
            imgArray[x] = cv2.resize(imgArray[x], (0, 0), None, scale,
scale)
        else:
            imgArray[x] = cv2.resize(imgArray[x],
(imgArray[0].shape[1], imgArray[0].shape[0]), None, scale, scale)
            if len(imgArray[x].shape) == 2: imgArray[x] =
cv2.cvtColor(imgArray[x], cv2.COLOR_GRAY2BGR)
            hor= np.hstack(imgArray)
            ver = hor
            return ver

```

```

##### Fungsi untuk Deteksi Objek
#####

```

```

def getContours(img, imgContour):
    contours, hierarchy = cv2.findContours(img, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_NONE)
    for cnt in contours:
        area = cv2.contourArea(cnt)
        #areaMin = cv2.getTrackbarPos("Area", "Parameters")
        areaMin = w3.get()
        if area > areaMin:
            continue
        cv2.drawContours(imgContour, cnt, -1, (255, 0, 255), 1)
        peri = cv2.arcLength(cnt, True)
        approx = cv2.approxPolyDP(cnt, 0.02 * peri, True)
        x , y , w , h = cv2.boundingRect(approx)
        cv2.rectangle(imgContour, (x , y ), (x + w , y + h ), (0, 255, 0),
1)
        M = cv2.moments(cnt)
        cx,cy = int (M['m10']/M['m00']),int (M['m01']/M['m00'])
        cv2.circle(imgContour, (cx,cy), 1, (0, 255, 0), -1)
        cv2.putText (imgContour, (" "+str(cx)+" "+str(cy)+" ",
((cx,cy) [0]+1, (cx,cy) [1]+20), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 125,
255), 2)

```

```

##### Fungsi untuk Mengolah Format Data
#####

```

```

def format_data():
    global cx, cy

    if cx < 10:
        cx = '0'+ '0'+str(cx)
    elif cx < 100:
        cx = '0'+str(cx)
    if cy < 10:
        cy = '0'+ '0'+str(cy)
    elif cy < 100:
        cy = '0'+str(cy)

```

```

##### Fungsi untuk Deteksi Objek
#####

```

```

def track_objects(img, imgContour):
    global cx, cy, x_target, y_target, x_laser, y_laser, area, tx, ty, lx, ly,
11, 12, data1, data2, A, B

```

```

    contours, hierarchy = cv2.findContours(img, cv2.RETR_EXTERNAL,

```

```

cv2.CHAIN_APPROX_NONE)
for cnt in contours:
    area = cv2.contourArea(cnt)
    #areaMin = cv2.getTrackbarPos("Area", "Parameters")

    cv2.drawContours(imgContour, cnt, -1, (255, 0, 255), 1)
    peri = cv2.arcLength(cnt, True)
    approx = cv2.approxPolyDP(cnt, 0.02 * peri, True)
    x , y , w , h = cv2.boundingRect(approx)
    cv2.rectangle(imgContour, (x , y ), (x + w , y + h ), (0, 255, 0),
1)

M = cv2.moments(cnt)
if M['m00'] != 0:
    cx,cy = int(M['m10']/M['m00']),int(M['m01']/M['m00'])
    cv2.circle(imgContour, (cx,cy), 1, (0, 255, 0), -1)
    cv2.putText(imgContour, ("+str(cx)+","+str(cy)+"),
((cx,cy)[0]+1,(cx,cy)[1]+20), cv2.FONT_HERSHEY_SIMPLEX, 0.6,(0, 125,
255),2)

format_data()
#global x_target, y_target,x_laser,y_laser
areaTarget= int(teks4.get())
areaLaser = int(teks5.get())
area = area - 50
if area > areaTarget:
    tx, ty = cx, cy
    x_target, y_target = tx, ty
    l1= ("+str(x_target)+str(y_target)+")"
    data1 = l1

cv2.putText(imgContour, 'Target', (int(x_target),int(y_target)),
cv2.FONT_HERSHEY_SIMPLEX, 0.6,(0, 125, 255),2)
    #cv2.putText(imgContour, 'Target', (int(cx),int(cy)),
cv2.FONT_HERSHEY_SIMPLEX, 1,(0, 125, 255),2)

elif area < areaLaser:
    lx, ly = cx, cy
    x_laser, y_laser = lx, ly
    #l2=str(x_laser)+str(y_laser)
    l2= ("+str(x_laser)+str(y_laser)+")"
    data2 = l2
    cv2.putText(imgContour, 'Laser', (int(x_laser),int(y_laser)),
cv2.FONT_HERSHEY_SIMPLEX, 0.6,(0, 125, 255),2)
    #cv2.putText(imgContour, 'Laser', (int(cx),int(cy)),
cv2.FONT_HERSHEY_SIMPLEX, 1,(0, 125, 255),2)
    ##atas##

A = 59
B = 123

if int(x_laser) > int(x_target):
    ServoA = A
    setServoAngle(ServoA)

    ServoB = B
    setServoAngle2(ServoB)
    break
elif int(x_laser) == int(x_target):
    ServoA = A
    setServoAngle(ServoA)

```

```

        ServoB = B
        setServoAngle2(ServoB)
        break

    else:
        ServoA = A
        setServoAngle(ServoA)

        ServoB = B
        setServoAngle2(ServoB)
        break
#
lx, ly = cx, cy
x_laser, y_laser = lx, ly
l2="("+str(x_laser)+str(y_laser)+")"
data2 = l2

print ((data1),",", (data2))

data3= data1+" , "+data2

save (data3)
root.update()

##### Fungsi untuk Set Derajat Servo
#####

def set_servo():
    global ServoA, ServoB
    ServoA = int(teks6.get())
    ServoB = int(teks7.get())
    setServoAngle(ServoA)
    setServoAngle2(ServoB)

##### Fungsi untuk Persiapan Deteksi Objek
#####

def setting():
    global a, img
    a = 1

    while (a==1):
        img = cap.read()
        cv2.rectangle(img, (365,230), (285,125), (0,0,255),0)
        crop_img = img[125:230 , 285:365]
        imgContour =crop_img.copy()
        imgBlur = cv2.GaussianBlur(crop_img, (5, 5), 1)
        imgGray = cv2.cvtColor(imgBlur, cv2.COLOR_BGR2GRAY)
        threshold1 = w1.get()
        threshold2 = w2.get()
        th1 =threshold1
        th2 =threshold1
        imgCanny = cv2.Canny(imgGray,threshold1,threshold2)
        kernel = np.ones((5, 5))
        imgDil = cv2.dilate(imgCanny, kernel, iterations=1)
        getContours(imgDil,imgContour)
        imgStack
stackImages(0.8, ([img,imgCanny],[imgDil,imgContour]))
cv2.imshow('Tampilan pada Lapangan',imgStack)

```

```

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
        root.update()

##### Fungsi untuk Mulai Deteksi Objek
#####

def mulai():
    global b, th1, th2, threshold1, threshold2
    b = 1

    tampilan()

    th1= int(teks2.get())
    th2 = int(teks3.get())

    while (b==1):
        img = cap.read()
        cv2.rectangle(img, (365,230), (285,125), (0,0,255),0)
        crop_img = img[125:230 , 285:365]
        imgContour = crop_img.copy()
        imgBlur = cv2.GaussianBlur(crop_img, (5, 5), 1)
        imgGray = cv2.cvtColor(imgBlur, cv2.COLOR_BGR2GRAY)
        threshold1 = th1
        threshold2 = th2
        imgCanny = cv2.Canny(imgGray,threshold1,threshold2)
        kernel = np.ones((5, 5))
        imgDil = cv2.dilate(imgCanny, kernel, iterations=1)
        track_objects(imgDil,imgContour)
        imgStack =
stackImages(0.8, ([img,imgCanny],[imgDil,imgContour]))
        cv2.imshow('Tampilan pada Lapangan',imgStack)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
        root.update()

def berhenti():
    global b
    b = 2
    while (b==2):
        b=0
        root.update()

##### Fungsi Tampilan
#####
root = Tk()
root.title("JUDUL")

root.t1 = IntVar()

cnt = Label(root, text='JUDUL', font=('Arial', 20))
cnt.pack()

frm1 = LabelFrame(root, text="Mode yang dipilih")
frm1.pack()

frm2 = Frame(frm1)
frm2.pack(side=LEFT)

cnt1 = LabelFrame(frm2, text='Setting', font=('Arial', 20))

```



```

cnt1.pack(side=LEFT)

cnt12 = LabelFrame(frm2, text='Threshold1')
cnt12.pack(side=TOP)

w1 = Scale(cnt12, from_=0, to=255, orient=HORIZONTAL)
w1.pack()

cnt13 = LabelFrame(frm2, text='Threshold2')
cnt13.pack(side=TOP)

w2 = Scale(cnt13, from_=0, to=255, orient=HORIZONTAL)
w2.pack()

cnt14 = LabelFrame(frm2, text='Area')
cnt14.pack(side=TOP)

w3 = Scale(cnt14, from_=0, to=30000, orient=HORIZONTAL)
w3.pack()

frm3 = Frame(frm1)
frm3.pack(side=LEFT)

cnt11 = LabelFrame(frm3, text='Set Servo', font=('Arial', 20))
cnt11.pack(side=TOP)

cnt9 = LabelFrame(frm3, text="Servo Atas")
cnt9.pack(side=LEFT)

teks6 = Entry(cnt9, bd =5, width=15)
teks6.pack(side=RIGHT)

cnt10 = LabelFrame(frm3, text="Servo Bawah")
cnt10.pack(side=LEFT)

teks7 = Entry(cnt10, bd =5, width=15)
teks7.pack(side=RIGHT)

frm4 = Frame(frm1)
frm4.pack(side=LEFT)

cnt2 = LabelFrame(frm4, text='Mulai', font=('Arial', 20))
cnt2.pack(side=TOP)

cnt4 = LabelFrame(frm4, text="Data Ke-")
cnt4.pack(side=TOP)

teks1 = Entry(cnt4, bd =5, width=15,)
teks1.pack(side=BOTTOM)

frm5 = Frame(frm1)
frm5.pack(side=LEFT)

frm6 = Frame(frm1)
frm6.pack(side=LEFT)

cnt5 = LabelFrame(frm5, text="Threshold1")
cnt5.pack(side=TOP)

teks2 = Entry(cnt5, bd =5, width=15)

```

```

teks2.pack(side=BOTTOM)

cnt6 = LabelFrame(frm6, text="Threshold2")
cnt6.pack(side=TOP)

teks3 = Entry(cnt6, bd =5, width=15)
teks3.pack()

cnt7 = LabelFrame(frm5, text="Area Laser")
cnt7.pack(side=TOP)

teks4 = Entry(cnt7, bd =5, width=15)
teks4.pack()

cnt8 = LabelFrame(frm6, text="Area Target")
cnt8.pack(side=BOTTOM)

teks5 = Entry(cnt8, bd =5, width=15)
teks5.pack()

frm7 = Frame(frm1)
frm7.pack(side=LEFT)

cnt3= LabelFrame(frm7, text='Berhenti', font=('Arial', 20))
cnt3.pack(side=LEFT)

mode = IntVar()
btn1 = Button(cnt1, text="Setting", padx=30, pady=40, command=setting)
btn1.pack(side=LEFT)
btn4 = Button(cnt11, text="Set Servo", padx=100, pady=10,
command=set_servo)
btn4.pack(side=LEFT)
btn2 = Button(cnt2, text="Mulai", padx=40, pady=10, command=mulai)
btn2.pack(side=LEFT)
btn3 = Button(cnt3, text="Berhenti", padx=30, pady=40, command=berhenti)
btn3.pack(side=LEFT)

root.mainloop()

```



FAKULTAS TEKNIK

Jl. Pawiyatan Luhur IV/1 Bendan Duwur Semarang 50234
Telp : (024) 8441555 (hunting) Fax : (024) 8415429 – 8445265
Email : elektro@unika.ac.id




SURAT-TUGAS

Nomor : 00174/B.8.8/ST.FT/11/2020

Dekan Fakultas Teknik Universitas Katolik Soegijapranata Semarang, dengan ini memberikan tugas kepada Dosen dan Mahasiswa Program Studi Teknik Elektro Fakultas Teknik sebagai berikut :

- N a m a** : Dr. Leonardus Heru Pratomo, ST, MT
Dr. Florentinus Budi Setiawan, ST, MT
Prof. Dr. Ir. Slamet Riyadi, MT
Stephanus Yudha Setyawan (NIM : 16.F1.0010)
- S t a t u s** : Dosen dan Mahasiswa Program Studi Teknik Elektro Fakultas Teknik Universitas Katolik Soegijapranata Semarang
- T u g a s** : Pembicara pada kegiatan Electrical Week Handsanitizer Otomatis Berbasis Arduino Mega 2560 sebagai kegiatan pembelajaran dan ketrampilan Robotik dengan judul "Sensor dan Akuator yang Digunakan pada Alat Handsanitizer Otomatis".
- W a k t u** : Jumat, 20 November 2020
- T e m p a t** : Youtube Channel HMJTE Unika Soegijapranata
- Lain-lain** : Harap dilaksanakan dengan sebaik-baiknya dengan penuh tanggung jawab dan setelah selesai harap memberikan laporan.

Semarang, 19 November 2020
Dekan,

Prof. Dr. Ir. Slamet Riyadi, MT
NPP: 058.1.1992.110



8.78% PLAGIARISM
APPROXIMATELY

Report #12347589

BAB I PENDAHULUAN Latar Belakang Bencana tanah longsor merupakan suatu ancaman serius bagi pemukiman di lereng pegunungan. 6 Tanah longsor merupakan suatu peristiwa alam, fenomena alam ini akan berubah menjadi bencana tanah longsor manakala tanah longsor tersebut menimbulkan kerusakan secara langsung seperti rusaknya fasilitas umum, lahan pertanian, ataupun adanya korban manusia, akan tetapi juga kerusakan secara tidak langsung yang melumpuhkan kegiatan pembangunan dan aktivitas ekonomi di daerah bencana dan sekitarnya. Intensitas curah hujan yang tinggi dan kejadian gempa yang sering muncul, secara alami akan dapat memicu terjadinya bencana alam tanah longsor (Supriyadi, 2016). Teknologi seperti kamera pada awalnya hanya digunakan untuk menangkap gambar saja. Namun seiring perkembangan teknologi, kamera dapat difungsikan sebagai sensor untuk mendeteksi apapun gambar yang ditangkap oleh kamera. Gambar yang dihasilkan kemudian diolah oleh perangkat komputer yang sudah terprogram untuk tujuan tersebut. 1 Teknologi itu dinamakan Computer Vision. Teknologi ini banyak digunakan untuk tujuan tertentu. 1 Dikarenakan hanya mengandalkan kamera yang sudah terkomputasi, teknologi ini dapat menggantikan fungsi dari banyak sensor seperti sensor warna, jarak, gerak dan sebagainya Computer Vision merupakan

