

LAMPIRAN

Program BGR to HSV

Program ini digunakan untuk mengetahui HSV dari warna RGB yang di dapat melalui aplikasi editor foto.

```
import cv2
```

```
import numpy as np
```

perintah import digunakan untuk memanggil library pada open cv.

B G R

```
green = np.uint8([[[35,28,167]]])
```

RGB yang didapat di masukkan ke program di atas tetapi dengan menempatkan menjadi BGR.

```
hsv_green = cv2.cvtColor(green,cv2.COLOR_BGR2HSV)
```

program `hsv_green =` adalah perintah untuk melaksanakan program convert agar menghitung menjadi HSV.

```
print ( hsv_green )
```

lalu untuk mengakhiri menggunakan perintah `print (hsv_green)` otomatis nilai HSV akan muncul.

Program *color tracking*

Program ini untuk mendeteksi warna objek dan mengikuti objek tersebut

Perintah `from` digunakan untuk mengaktifkan video secara realtime

```
from collections import deque
from imutils.video import VideoStream
import numpy as np
import argparse
import cv2
import imutils
import time
import RPi.GPIO as GPIO
```

Perintah import digunakan untuk memanggil library open cv

`objekLower = (168, 100, 100)` untuk memasukkan data HSV objek terendah
`objekUpper = (188, 255, 255)` untuk memasukkan data HSV objek tertinggi

`laserLower = (17, 100, 100)` untuk memasukkan data HSV laser terendah
`laserUpper = (37, 255, 255)` untuk memasukkan data HSV laser tertinggi

`objekMinX = 60` untuk mengatur pembacaan koordinat objek titik paling bawah
`objekCenterX = 90` untuk mengatur pembacaan koordinat objek di titik tengah
`objekMaxX = 120` untuk mengatur pembacaan koordinat objek titik paling atas

`laserMinY = 60` untuk mengatur pembacaan koordinat laser titik paling bawah
`laserCenterY = 90` untuk mengatur pembacaan koordinat laser di titik tengah
`laserMaxY = 120` untuk mengatur pembacaan koordinat laser titik paling atas

`skalaKecepatan = 0.1;` untuk mengatur kecepatan servo bergerak

`pinServo1 = 11` pin yang dipakai servo 1 yaitu yang bergerak ke kanan dan ke kiri
`pinServo2 = 12` pin yang dipakai servo 2 yaitu yang bergerak ke atas dan ke bawah
Perintah GPIO digunakan untuk meminimalkan jitter pada servo saat terhubung langsung ke pin input Raspberry

```
GPIO.setwarnings(False)
```

```
GPIO.setmode(GPIO.BOARD)
```

```
GPIO.setup(pinServo1, GPIO.OUT)
```

```
GPIO.setup(pinServo2, GPIO.OUT)
```

Perintah berikut digunakan untuk mengatur PWM yang di keluarkan di tiap GPIO

```
servo1 = GPIO.PWM(pinServo1, 50)
```

```
servo2 = GPIO.PWM(pinServo2, 50)
```

```
servo1.start(2.5)
```

```
servo2.start(2.5)
```

Perintah berikut adalah hitungan untuk mengeset servo ketika berjalan

```
def setServo(servo, degree):
```

```
    servo.ChangeDutyCycle((degree/180) * 10 + 2.5);
```

`servo1Pos = 90` Perintah ini untuk mengeset posisi servo 1

```
setServo(servo1, servo1Pos)
```

servo2Pos = 45 Perintah ini untuk mengeset posisi servo 2
setServo(servo2, servo2Pos)

Perintah ini untuk menjalankan video secara real time dan kamera secara otomatis menyala.

```
ap = argparse.ArgumentParser()
ap.add_argument("-v", "--video",
                help="path to the (optional) video file")
ap.add_argument("-b", "--buffer", type=int, default=64,
                help="max buffer size")
args = vars(ap.parse_args())
```

```
print("tekan tombol keyboard untuk keluar/selesai")
```

Perintah print digunakan untuk memunculkan pemberitahuan jika selesai maka gunakan tombol untuk mematikan program

Berikut ini adalah perintah untuk kamera agar bekerja ketika benda bergerak otomatis kamera merekam dan mengambil gambar.

```
if not args.get("video", False):
    vs = VideoStream(src=0).start()
else:
    vs = cv2.VideoCapture(args["video"])
time.sleep(2.0)
while True:
    frame = vs.read()
    frame = frame[1] if args.get("video", False) else frame
    if frame is None:
        break
```

Perintah frame digunakan untuk mengatur besarnya frame

```
frame = imutils.resize(frame, width=600)
blurred = cv2.GaussianBlur(frame, (11, 11), 0)
```

Perintah hsv digunakan untuk membaca hsv yang dideteksi kamera

```
hsv = cv2.cvtColor(blurred, cv2.COLOR_BGR2HSV)
```

Selanjutnya perintah untuk mendeteksi objek dan gerakan laser melalui servo
Jika tidak mendeteksi objek maka kamera akan terus mencari hsv yang sesuai

```
maskObjek = cv2.inRange(hsv, objekLower, objekUpper)
maskObjek = cv2.erode(maskObjek, None, iterations=2)
maskObjek = cv2.dilate(maskObjek, None, iterations=2)
maskLaser = cv2.inRange(hsv, laserLower, laserUpper)
maskLaser = cv2.erode(maskLaser, None, iterations=2)
maskLaser = cv2.dilate(maskLaser, None, iterations=2)
```

Perintah ketika objek terdeteksi maka kamera akan menghitung x dan y suatu objek

```

cntsObjek = cv2.findContours(maskObjek.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
cntsObjek = imutils.grab_contours(cntsObjek)

```

```

cntsLaser = cv2.findContours(maskLaser.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
cntsLaser = imutils.grab_contours(cntsLaser)

```

Perintah berikut digunakan untuk mengatur jarak objek dan laser pada waktu program berjalan.

```

radiusObjek = 0
radiusLaser = 0

centerObjek = (0,0);
centerLaser = (0,0);

```

Perintah ini digunakan untuk mencari objek di kamera lalu menampilkan di frame display muncul lingkaran pada objek tersebut.

```

if len(cntsObjek) > 0:
    c = max(cntsObjek, key=cv2.contourArea)
    ((xObjek, yObjek), radiusObjek) = cv2.minEnclosingCircle(c)
    M = cv2.moments(c)
    centerObjek = (int(M["m10"] / M["m00"]), int(M["m01"] /
M["m00"]))

```

Perintah ini digunakan untuk mencari koordinat laser dengan menggunakan lingkaran hijau di tengah.

```

if len(cntsLaser) > 0:
    c = max(cntsLaser, key=cv2.contourArea)
    ((xLaser, yLaser), radiusLaser) = cv2.minEnclosingCircle(c)
    M = cv2.moments(c)
    centerLaser = (int(M["m10"] / M["m00"]), int(M["m01"] /
M["m00"]))

```

```

h, w, c = frame.shape
centerLaser = (int(w/2),int(h/2))
radiusLaser = 5

```

Selanjutnya jika jarak objek dan laser sudah sesuai maka servo dari bergerak akan berhenti di jarak tersebut.

```

if radiusObjek > 3 :
    cv2.circle(frame, centerObjek, 10, (0, 0, 255), 1)

if radiusLaser > 3:
    cv2.circle(frame, centerLaser, 10, (0, 255, 0), 1)

if radiusObjek > 3 and radiusLaser > 3:

    cv2.circle(frame, centerLaser, 10, (0, 255, 0), 1)

if centerObjek[0] > centerLaser[0]:
    servo1Pos += (centerObjek[0] - centerLaser[0]) *

```

Program ini digunakan untuk mengatur kecepatan ketika servo berjalan mengikuti objek.
skalaKecepatan;

```
    if servo1Pos > objekMaxX:
        servo1Pos = objekMaxX
        setServo(servo1, 180 - servo1Pos)
    else:
        servo1Pos -= (centerLaser[0] - centerObjek[0]) *
skalaKecepatan;
    if servo1Pos < objekMinX:
        servo1Pos = objekMinX
        setServo(servo1, 180 - servo1Pos)

    if centerObjek[1] > centerLaser[1]:
        servo2Pos += (centerObjek[1] - centerLaser[1]) *
skalaKecepatan;
    if servo2Pos > laserMaxY:
        servo2Pos = laserMaxY
        setServo(servo2, servo2Pos)
    else:
        servo2Pos -= (centerLaser[1] - centerObjek[1]) *
skalaKecepatan;
    if servo2Pos < laserMinY:
        servo2Pos = laserMinY
        setServo(servo2, servo2Pos)
```

Perintah berikut digunakan untuk mengeset dimana servo terkunci agar tidak merusak laser dan raspberry.

```
    else:
        servo2Pos = 45
        setServo(servo2, servo2Pos)
        servo1Pos = 0
        setServo(servo1, servo1Pos)
```

Perintah print dibawah ini digunakan untuk menampilkan x dan y objek dan digunakan untuk membaca jarak anatar objek dengan laser.

```
    print("centerObjek = " + str(centerObjek[0]) + ", " +
str(centerObjek[1]))
    print("centerLaser = " + str(centerLaser[0]) + ", " +
str(centerLaser[1]))
```

```
    print("radius objek = " + str(radiusObjek))
    print("radius laser = " + str(radiusLaser))
```

```
    print("servo x = " + str(servo1Pos))
    print("servo y = " + str(servo2Pos))
```

```
    print("")
```

```
    cv2.imshow("Frame", frame)
```

Perintah cv2.imshow digunakan untuk menampilkan pada display frame

```
    if cv2.waitKey(1) >= 0:
        break
```

Perintah selanjutnya jika video di stop dengan menekan tombol maka kamera akan mati, dan program akan di tutup

```
if not args.get("video", False):  
    vs.stop()
```

```
else:  
    vs.release()
```

```
cv2.destroyAllWindows()
```

Program destroy all window akan memerintahkan jendela frame ditutup

```
servo1.stop()  
servo2.stop()  
GPIO.cleanup()
```

Perintah diatas Servo 1 dan 2 akan berhenti bergerak, dan GPIO tidak aktif





Sertifikat

NOMOR : 020/SNETO2019/PEM/EL/Itenas/XII/2019

RIKI ANDREAS PEMAKALAH

Seminar Nasional Energi Telekomunikasi Dan Otomasi (SNETO 2019)

Ketua Jurusan Teknik Elektro



DR. WALUYO

Ketua Pelaksana



DINI FAUZIAH, S.PD., M.T.




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 : 00125/B.8.8/ST.FT/11/2019

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

N a m a : Riki Andreas (NIM : 15.F2.0003)
S t a t u s : Mahasiswa Program Studi Teknik Elektro Fakultas Teknik Universitas Katolik Soegijapranata Semarang
T u g a s : Mengadakan Kegiatan One Day Short Course sebagai kegiatan pembelajaran dan Ketrampilan Robotik di SMA Krista Mitra Semarang
W a k t u : Kamis,- Jumat / 14-15 November 2019
T e m p a t : Ruang Kelas / Aula SMA Krista Mitra Semarang
Lain-lain : Harap dilaksanakan dengan sebaik-baiknya dengan penuh tanggung jawab dan setelah selesai harap memberikan laporan.

Semarang, 12 November 2019
Dekan

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

Telah melaksanakan tugas,


(Riki Andreas)



8.48% PLAGIARISM
APPROXIMATELY

Report #10429264

BAB IPENDAHULUAN 1.1 LATAR BELAKANG Teknologi Computer Vision banyak digunakan untuk berbagai macam dalam sistem visual, seperti Scanner, deteksi wajah, deteksi plat nomor kendaraan, dan sebagainya. Teknologi ini menggunakan kamera untuk menangkap, dan mendeteksi objek. Dengan menggunakan kamera yang sudah terkomputasi, gambar yang dihasilkan oleh kamera kemudian diolah melalui Raspberry Pi, yang sudah terprogram untuk mendeteksi suatu objek. Teknologi ini banyak digunakan untuk sensor warna, sensor jarak, sensor gerak dan sebagainya ADDIN [1]. Penelitian ini penulis akan mendeteksi objek berdasarkan warna pada objek untuk menggerakkan Pan-Tilt Laser otomatis yang telah di program melalui Raspberry Pi. Program untuk penelitian ini menggunakan bahasa pemrograman library OpenCV, dan Python. Program tersebut berguna untuk menangkap suatu objek berdasarkan warna objek tersebut melalui RGB (Red, Green, Blue). Setelah diketahui RGB dari objek yang akan dideteksi, kemudian diubah menjadi HSV (Hue, Saturation, Value) untuk menentukan batas atas dan bawah suatu warna yang akan dideteksi menggunakan library OpenCV. Sesudah warna objek terdeteksi dengan baik, maka kamera memberikan informasi koordinat pada objek yang terdeteksi dan Pan-Tilt diprogram mengikuti objek sesuai dengan koordinat yang diterima