

BAB IV

HASIL DAN PEMBAHASAN

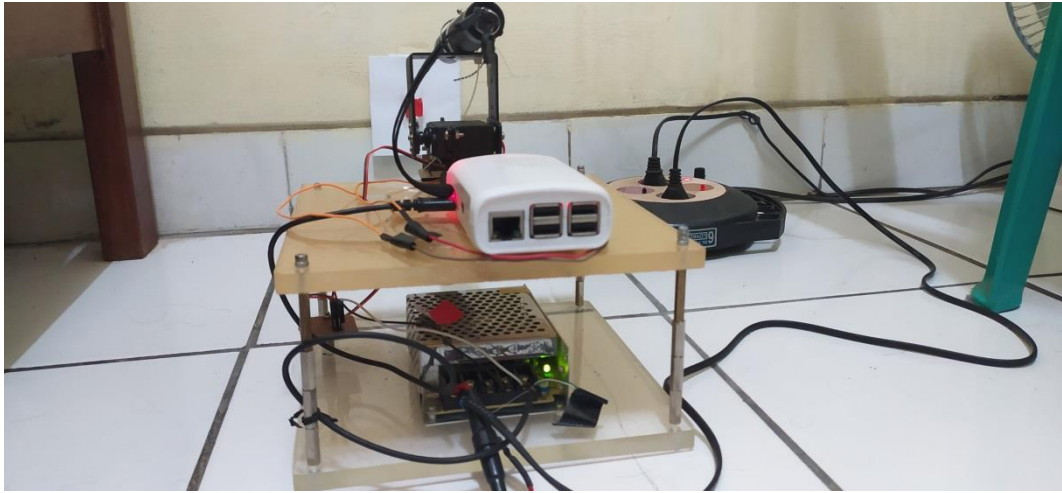
4.1 Pendahuluan

Pada bab ini berisi mengenai hasil dan pembahasan tentang pengujian alat Implementasi Computer Vision Untuk Sistem Deteksi Posisi Laser Menggunakan Raspberry Pi 3 yang telah dibuat. Hasil yang dapat dilihat alat ini yaitu pergerakan sistem Pan-Tilt yang mengikuti objek berdasarkan warna yang dideteksi oleh kamera. Serta hasil objek yang terdeteksi atau tidaknya dapat dilihat pada display monitor yang berdasarkan dari teori serta rancangan yang sudah ditulis pada bab sebelumnya.

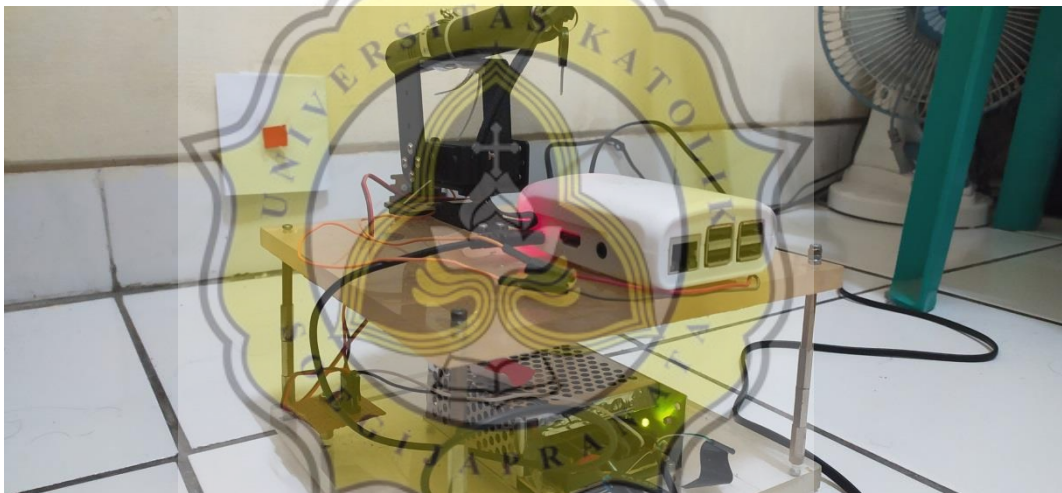
Bab ini akan menampilkan bentuk prototype alat yang telah dibuat sesuai rancangan pada laporan ini.

4.2 Gambar Prototype Alat

Prototype alat yang dibuat terdiri dari beberapa komponen utama, dan pendukung. Tempat pengujian alat dilakukan di rumah dan Laboratorium Tugas Akhir. Seperti pada (Gambar 4.1) adalah foto dari alat berdasarkan rancangan alat yang sudah dibahas pada bab sebelumnya.



(a)



(b)

Gambar 4.1 Prototype bentuk alat (a) bentuk keseluruhan (b) mekanisme pan-tilt

Pada pengujian ini penulis menggunakan kertas asturo yang berwarna merah seperti pada (Gambar 4.2) yang akan dideteksi kamera. Penulis menggunakan warna merah karena kamera dapat menangkap gambar yang lebih baik jika objek berwarna terang seperti merah, hijau, biru, kuning, dan warna lainnya. Berbeda halnya ketika kamera menangkap warna yang gelap seperti hitam, coklat, dan

warna lainnya, maka kualitas yang diperoleh tidak sebaik saat menangkap objek yang berwarna lebih terang.



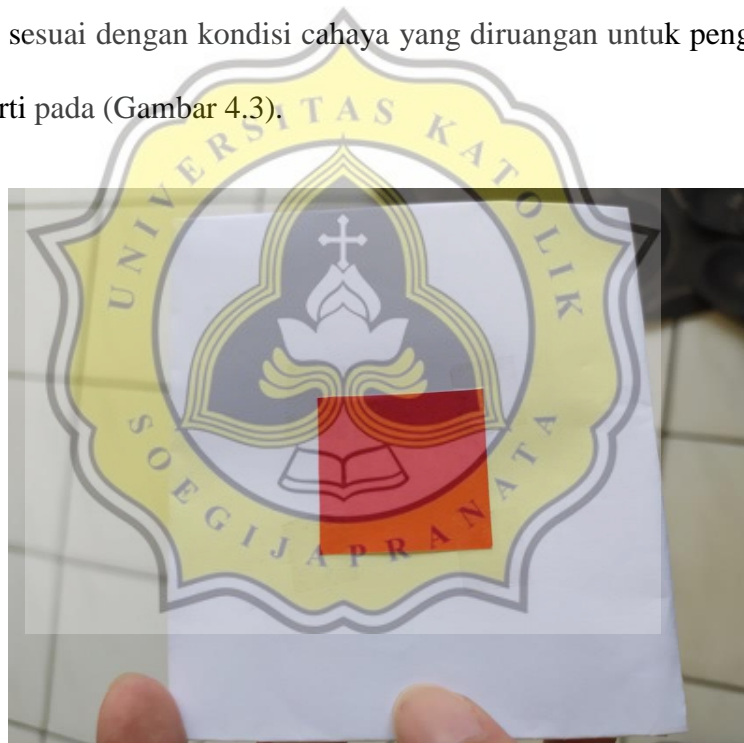
Gambar 4.2 Objek berupa kertas merah yang akan dideteksi

4.3 Hasil Pengujian

Hasil dari data pengujian utama dengan hasil berupa gambar yang ditampilkan dari kamera secara *realtime*. Pada pengujian ini menggunakan data warna pada objek yang dideteksi melalui program konversi RGB ke HSV. Kemudian kamera akan mendeteksi objek melalui data HSV yang dimasukkan ke dalam program. Selanjutnya menampilkan display ketika dalam intensitas cahaya yang cukup maka akan muncul lingkaran merah pada gambar display sebagai penanda bahwa objek terdeteksi. Selanjutnya hasil pengujian kedua yaitu pergerakan sistem Pan-Tilt yang dapat mengikuti objek yang dideteksi menggunakan HSV warna pada objek.

4.3.1 Pengujian langkah pertama mengambil data warna

Pada langkah pertama diperlukan data warna objek karena sebelum pembuatan program *color tracking*, dibutuhkan hsv warna objek terlebih dahulu maka kamera dapat mendeteksi objek. Data warna dapat ditemukan menggunakan format *RGB*(*Red, Green, Blue*) kemudian format *RGB* harus dikonversikan menjadi format *HSV*(*Hue, Saturation, Value*) yang sudah ditentukan dalam library *OpenCV*. Cara untuk mencari data *RGB* yaitu dengan memfoto objek yang akan kita deteksi sesuai dengan kondisi cahaya yang diruangan untuk pengujian. Dapat dilihat seperti pada (Gambar 4.3).



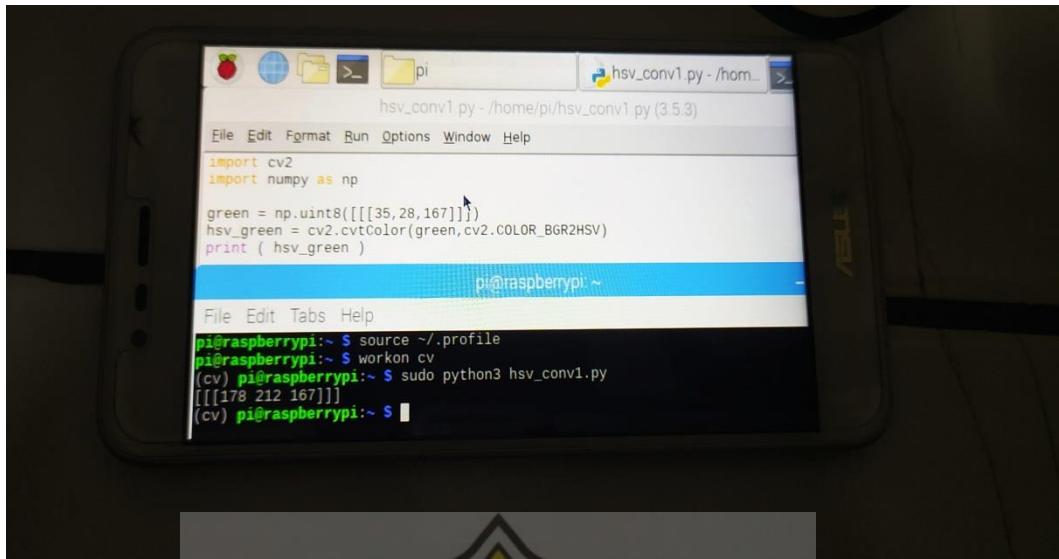
Gambar 4.3 Kertas warna merah yang akan dicari data *RGB* nya

Pencarian data warna *RGB* pada objek dapat dicari menggunakan aplikasi *Pics Art* atau aplikasi editor foto yang lainnya. Dapat dilihat seperti pada (Gambar 4.4).



Gambar 4.4 Data warna RGB pada kertas warna merah

Setelah menemukan data warna RGB, kemudian dikonversikan menjadi HSV dengan program OpenCV yang sudah dibuat seperti pada (Gambar 4.5). Titik terendah dan titik teratas dari warna tersebut dicari menggunakan program konverter *BGR to HSV* yang sudah ada pada website resmi OpenCV. Setelah itu didapat hasil angka (178, 212, 167) yang dikonversi menjadi titik bawah yaitu (178-10,100,100) dan (178+10,255,255) hasilnya yang terendah (168, 100, 100) dan yang tertinggi adalah (188, 255, 255).



```
hsv_conv1.py - /home/pi/hsv_conv1.py (3.5.3)
File Edit Format Run Options Window Help
import cv2
import numpy as np

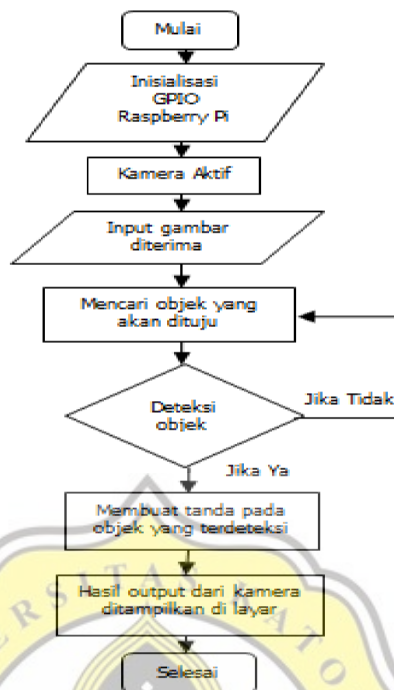
green = np.uint8([[35,28,167]])
hsv_green = cv2.cvtColor(green,cv2.COLOR_BGR2HSV)
print ( hsv_green )

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ source ~/.profile
pi@raspberrypi:~ $ workon cv
(cv) pi@raspberrypi:~ $ sudo python3 hsv_conv1.py
[[[178 212 167]]]
(cv) pi@raspberrypi:~ $
```

Gambar 4.5 Program konversi data warna RGB menjadi HSV

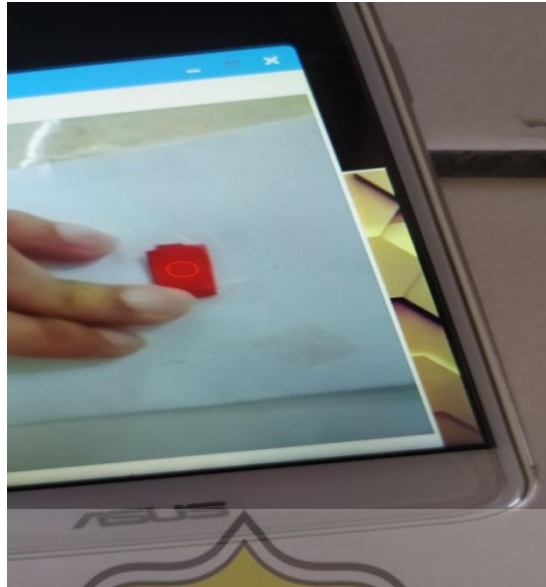
4.3.2 Pengujian langkah kedua kamera mendeteksi warna

Pada langkah kedua ini menjelaskan tentang cara kerja program yang telah dibuat agar dapat mendeteksi objek melalui data warna HSV yang sudah dimasukkan ke dalam program dan sistem Pan-Tilt dapat mengikuti arah objek yang ditangkap kamera. Kemudian kamera memberi penanda berupa lingkaran merah pada objek yang di deteksi, dan ditampilkan pada monitor. Setelah kamera dapat mendeteksi objek pada cahaya terang maka dilakukan pengujian jika cahaya gelap. Pada (Gambar 4.6) adalah gambar *flowchart* dari program yang dibuat oleh penulis.



Gambar 4.6 Flowchart cara kerja pada program

Pada flowchart di atas, program deteksi warna ini bekerja dengan resolusi gambar 640x480p pada 50 FPS (Frame Rate per Second), supaya program dapat bekerja dengan optimal karena disesuaikan dengan spesifikasi dari Raspberry Pi. Jika resolusi lebih besar maka FPS otomatis lebih rendah dan objek akan sulit dilacak oleh kamera karena performa yang lambat. Seperti (Gambar 4.7) adalah tampilan ketika program dijalankan pada saat cahaya terang maka objek dapat dideteksi, tetapi jika cahaya redup maka objek tidak terdeteksi.



(b)

Gambar 4.7 Objek (a) terdeteksi, dan objek (b) tidak terdeteksi

Setelah melakukan pengujian pada cahaya yang terang dan redup seperti pada gambar diatas. Kesimpulannya adalah cahaya sangat berpengaruh pada warna objek ketika akan dideteksi. Jika pada cahaya yang redup maka kamera tidak mendeteksi adanya suatu objek tersebut. Sebaliknya jika objek berada pada cahaya yang terang maka objek akan terdeteksi. Begitu juga jika pengujian

menggunakan cahaya lampu di ruangan dengan cahaya matahari pada siang hari maka hasilnya akan berbeda. Maka membutuhkan perubahan data warna pada program untuk menyesuaikan dengan foto pada saat diambil dalam kondisi cahaya pada saat itu.

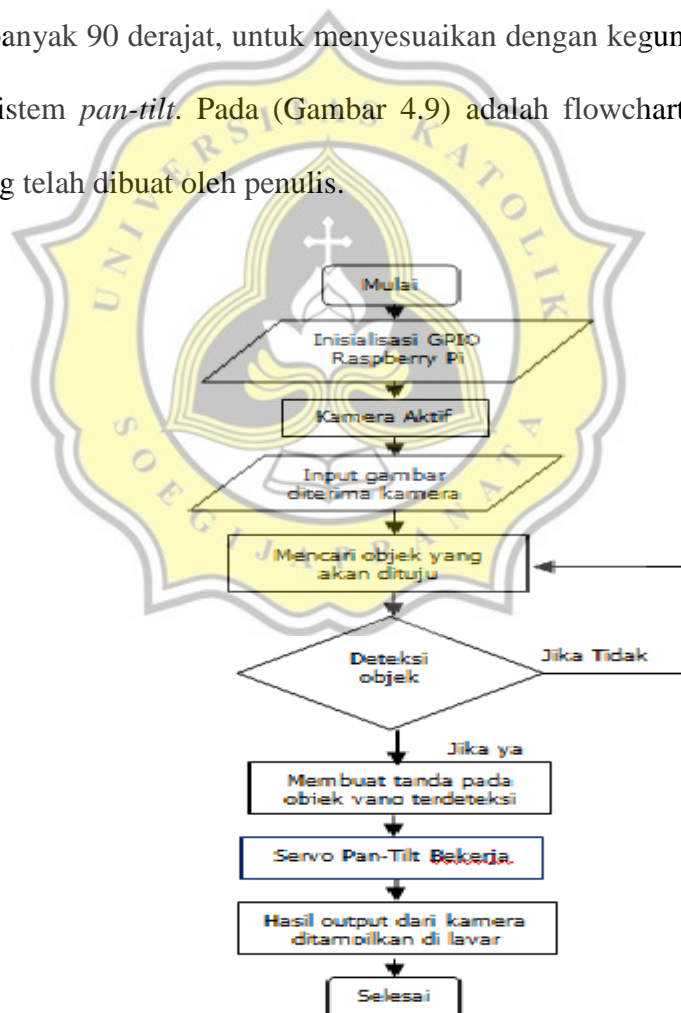
Pada (Gambar 4.8) adalah gambar pelacakan objek.



Gambar 4.8 Pengujian program color tracking pada kertas warna merah

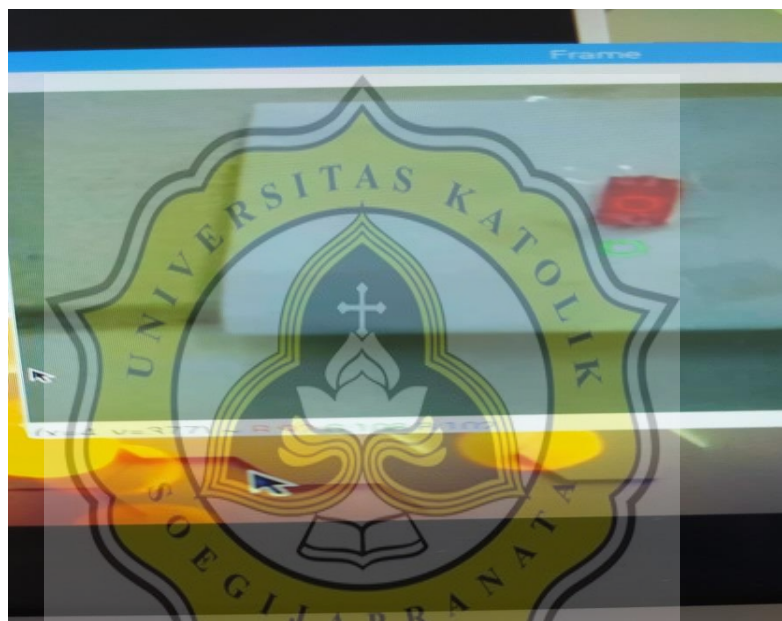
4.3.3 Pengujian langkah ketiga *color tracking* pada sistem *pan-tilt laser*

Pengujian ini merupakan pengujian pemrograman *color tracking* yang diterapkan pada gerakan sistem *pan-tilt laser*. Cara kerja program *color tracking* sama seperti langkah sebelumnya ketika kamera mendeteksi warna pada objek. Tetapi pada pengujian kali ini ditambahkan dengan program outputnya yaitu dua servo pada sistem *pan-tilt* untuk memposisikan laser. Pada servo pertama bergerak ke kanan dan kekiri sebanyak 180 derajat. Serta servo kedua bergerak ke atas dan kebawah sebanyak 90 derajat, untuk menyesuaikan dengan kegunaan serta rangka konstruksi sistem *pan-tilt*. Pada (Gambar 4.9) adalah flowchart cara kerja dari program yang telah dibuat oleh penulis.



Gambar 4.9 Flowchart cara kerja program pan-tilt color tracking

Pada flowchart cara kerja program di halaman sebelumnya, menjelaskan bahwa titik koordinat x dan y pada objek saat terdeteksi oleh kamera dapat mengatur pergerakan servo. Pada pengujian program deteksi warna dan saat tracking menggunakan resolusi gambar 640x480p dan diatur pada 50FPS(*Frame Rate per Second*), seperti pada (Gambar 4.10).



Gambar 4.10 Display objek yang ditangkap kamera dengan resolusi 640x480p

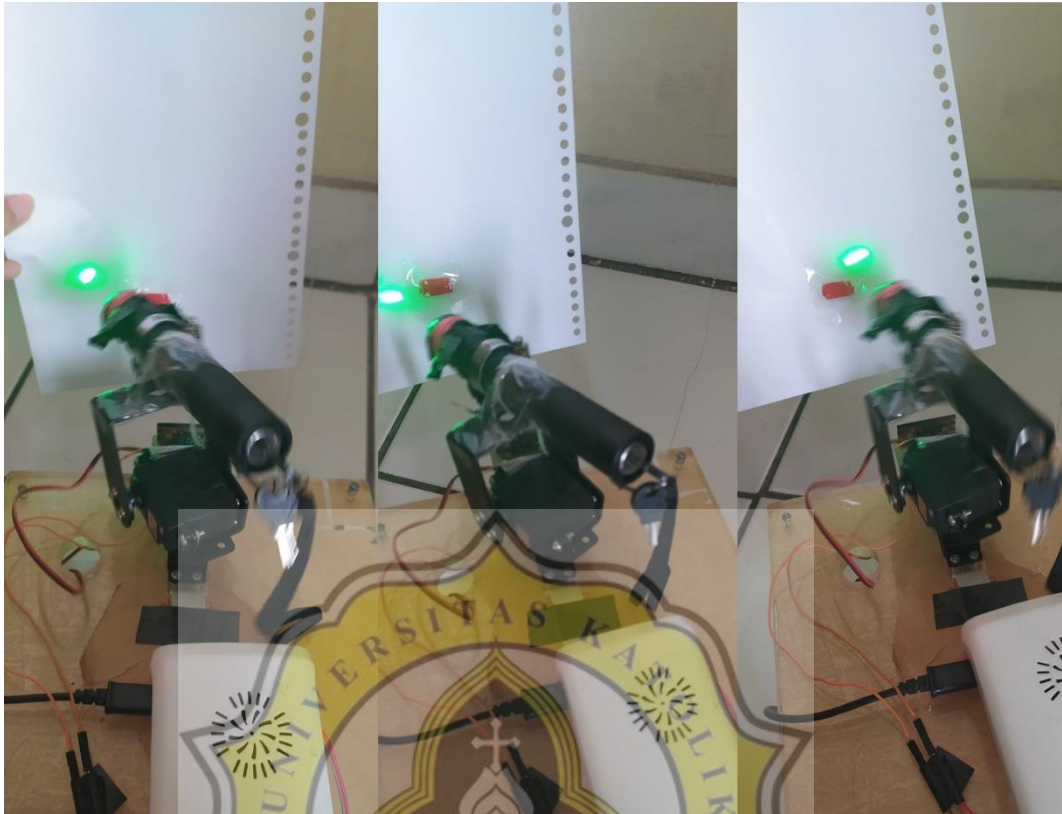
Pada gambar diatas menjelaskan bahwa objek terdeteksi di koordinat ($x=4$ dan $y=377$) seperti yang tertulis pada gambar. Ketika posisi objek di ketahui maka kedua servo akan bergerak sesuai titik koordinat yang di dapat oleh kamera. Begitu juga ketika objek bergerak maka koordinat x dan y akan berubah dengan sendirinya. Ketika kamera mengetahui objek bergerak maka akan memberikan koordinat pada saat itu ke kedua servo yang telah terpasang seperti pada (Gambar 4.10).

Berikut pada (Gambar 4.11) merupakan pergerakan servo ke kanan ketika mengikuti koordinat dari objek yang terdeteksi.



Gambar 4.11 Servo bergerak ke kanan

Setelah mencoba pergerakan servo ke kanan selanjutnya pergerakan servo ke kiri. Ketika objek bergerak ke arah kanan, kiri, maupun ke atas, dan bawah maka kedua servo akan mengikuti posisi x dan y pada objek ketika bergerak ke kanan jika objek ke kanan seperti pada (Gambar 4.11). Berikut (Gambar 4.12) merupakan pergerakan servo ke kiri ketika sedang melacak objek yang bergerak ke kiri.



Gambar 4.12 Servo bergerak ke kiri

4.4 Pembahasan

Berdasarkan hasil pengujian yang telah didapat, dapat dilihat bahwa untuk mendeteksi suatu objek menggunakan warna dari objek tersebut menggunakan library OpenCV mempunyai kelebihan dan kekurangan. Kelebihan dari deteksi warna yaitu ketika mendeteksi objek tidak bergantung dari besarnya ukuran suatu objek. Karena kamera hanya membutuhkan warna pada objek untuk dapat dikenali sebagai objek yang di deteksi. Setelah itu akan muncul penanda jika objek terdeteksi oleh kamera berbentuk lingkaran atau kotak dari program yang di buat. Kelebihan selanjutnya adalah kamera tidak tergantung jarak sekalipun objek jauh bisa terdeteksi apabila kamera yang terpasang sanggup untuk mendeteksi

objek tersebut. Kekurangan dari deteksi warna yaitu ketika mengandalkan warna saat mendeteksi suatu objek membutuhkan cahaya yang sesuai agar warna tidak pudar atau berubah. Selanjutnya ketika ada objek lain yang berwarna sama mendekati objek yang akan dideteksi, maka kamera akan mendeteksi semua objek yang warnanya sama. Sedangkan kekurangan dari library OpenCV adalah, pendeteksian warna yang kurang akurat ketika akan mendeteksi objek berwarna merah, maka objek dengan warna lain yang mendekati warna merah sebagai contoh warna merah muda ikut terdeteksi seperti pada (Gambar 4.13).



(b)

Gambar 4.13 Hasil Pengujian terhadap objek dengan warna (a) merah dan (b) merah muda

Deteksi warna juga berpengaruh terhadap intensitas cahaya juga berpengaruh terhadap RGB yang diperoleh dari warna objek seperti pada (Gambar 4.7) maka saat kita mencari data RGB dari warna suatu objek, membutuhkan intensitas cahaya yang cukup pada suatu ruangan atau tempat yang sama untuk digunakan pada saat pengujian deteksi warna. Setelah itu kita dapat mengambil gambar objek yang akan dideteksi untuk mendapatkan RGB yang sesuai seperti (Gambar 4.3).

Pengujian sistem pan-tilt untuk mengerakan laser, membuktikan bahwa pergerakan servo berdasarkan koordinat x dan y suatu objek yang di dapat dari kamera dapat bekerja dengan akurat karena servo terus bergerak mengikuti x dan y yang selalu berubah saat objek bergerak yang ada pada display seperti pada (Gambar 4.10). Dalam pengujian ini memiliki kendala dimana kedua servo memiliki *jitter* maka dibutuhkan library atau software *Pigpio* pada (Gambar 3.8) untuk dapat mengatasi error jika Pin PWM pada servo langsung dicolokkan ke Pin GPIO yang ada di Raspberry Pi.