

PERANCANGAN SISTEM DETEKSI DAN PELACAKAN BOLA MENGGUNAKAN OPENCV PADA ROBOT PENJAGA GAWANG

Felix Marcello Vioneidy¹, Ocsa Bimantara¹, Teuku Ahmad Rizal Hekmatyar¹, Suci Dwijayanti¹, Djulil Amri¹,
Dwirina Yuniarti¹, dan Bhakti Yudho Suprapto^{1*}

¹Teknik Elektro, Universitas Sriwijaya, Palembang

*Corresponding author e-mail: bhakti@ft.unsri.ac.id

ABSTRAK: Pengontrolan robot tidak lepas dari suatu sistem kendali yang dapat mengolah sinyal yang diterima agar sebuah robot dapat bekerja dengan baik, hal ini tentunya di dukung oleh kemampuan komunikasi antara sebuah kamera yang mampu mengambil gambar dengan jernih (noiseless) dengan mikrokontroler robot. Agar mikrokontroler dapat mengenali informasi yang di dapat oleh kamera, di gunakanlah Python untuk mengolah informasi yang di dapatkan oleh kamera agar dapat di digunakan oleh mikrokontroler pada robot. Library OpenCV digunakan untuk meningkatkan efisiensi komputasi serta fokus terhadap aplikasi secara realtime. Image Processing tak hanya di aplikasikan untuk industry semata, namun bisa juga di gunakan pada hal lain dalam kehidupan sehari hari.

Kata Kunci: Robot, Image Procesing, Deteksi Objek, Penjaga Gawang, Bola

ABSTRACT: Robot control cannot be separated from a control system that can process the received signals so that a robot can work well, this is certainly supported by the communication capability between a camera that is capable of taking clear (noiseless) images with the robot microcontroller. So that the microcontroller can recognize the information obtained by the camera, Python is used to process the information obtained by the camera so that it can be used by the microcontroller on the robot. The OpenCV library is used to increase computational efficiency and focus on real-time applications. Image Processing is not only applied to industry alone, but can also be used in other things in everyday life.

Keywords: Robot, Image Procesing, Object Detection, Goalkeeper, Ball

1 Pendahuluan

Teknologi Image Processing telah banyak digunakan di berbagai bidang salah satunya adalah robotika[1][2]. Kontes Robot Indonesia di adakan untuk memicu semangat bersaing antar mahasiswa di bidang robotika agar dapat memenuhi kebutuhan robotik yang terus berkembang di Indonesia[3][4]. Agar sebuah robot dapat bekerja dengan baik, hal ini tentunya di dukung oleh kemampuan komunikasi antara sebuah kamera dengan mikrokontroler robot. Agar mikrokontroler dapat mengenali informasi yang di dapat oleh kamera, di gunakanlah Python untuk mengolah informasi yang di dapatkan oleh kamera agar dapat di digunakan oleh mikrokontroler pada robot[5]. Library OpenCV digunakan untuk meningkatkan efisiensi komputasi serta fokus terhadap aplikasi secara realtime[6].

Untuk mencapai efisiensi dalam sebuah aplikasi image processing diperlukan keselarasan antara kompleksitas printah dengan jumlah algoritma yang digunakan[7]. Tentunya keterbatasan ini tidak luput dari spesifikasi hardware yang nantinya akan digunakan. Selain itu aplikasi tersebut tentunya harus lebih mudah digunakan, agar masalah yang nantinya muncul akan lebih mudah diperbaikannya. Untuk aplikasi yang digunakan oleh robot penjaga gawang tentunya aplikasi tersebut di tuntut dengan delay seminimal mungkin dikarenakan robot tersebut harus segera merspon bola yang datang, aplikasi tersebut tentunya harus segera mencerna informasi yang diterima, memprosesnya dan memberi langkah yang harus dilakukan pada robot[4].

Berdasarkan permasalahan tersebut penelitian ini dibuat berdasarkan pada sebuah prototype pengontrol robot penjaga gawang soccer beroda yang dibuat oleh tim RIMAU Universitas Sriwijaya untuk mengikuti Kontes Robot Sepak Bola Indonesia Beroda. Pengontrolannya robotnya disesuaikan menurut peraturan RoboCup Middle Size League (MSL), yaitu dengan cara membuat komputer mini yang dapat mengendalikan robot untuk menghadang bola secara otomatis, serta dilengkapi sebuah kamera yang berfungsi sebagai sensor untuk mendeteksi bola yang cakupannya dapat menangkap video dari keseluruhan lapangan permainan sepak bola[8].

Sistem deteksi bola menggunakan ruang warna HSV dan metode segmentasi warna telah terbukti mampu memisahkan objek bola dari latar belakang dengan baik serta mengatasi variasi pencahayaan yang ada di lapangan[9]. Selain itu, algoritma contour detection menjadikan proses identifikasi bola lebih robust dengan mengandalkan fitur bentuk fisik bola sebagai lingkaran yang presisi. Penggunaan kamera webcam Logitech 720p menjanjikan kualitas gambar yang optimal dengan tingkat resolusi yang cukup untuk aplikasi visi komputer dalam robotika olahraga[10].

Robot sepak bola beroda juga harus dapat bereaksi secara cepat berdasarkan data posisi bola yang terdeteksi agar dapat menyesuaikan gerakannya dalam menjaga gawang. Integrasi sistem deteksi bola secara real-time dengan modul kontrol robot menjadi kunci keberhasilan dalam implementasi robot penjaga gawang yang efektif dan efisien di lapangan[11].

Pada robot ini, dilengkapi dengan mikrokontroler dengan spesifikasi yang digunakan adalah Arduino mega 2560[12] yang nantinya akan mengontrol robot secara otomatis serta dilengkapi pula dengan laptop yang fungsi utamanya adalah untuk mengakses webcam yang digunakan robot untuk mengambil video untuk menetukan koordinat bola yang akan menjadi sasaran[13]. Robot ini di harapkan mampu menghadang bola dari sisi kiri, kanan, maupun tengah.

2 Metode

Pada bagian ini akan dijelaskan langkah-langkah pada proses penelitian dalam perancangan sistem deteksi bola menggunakan OpenCV dengan menggunakan kamera webcam Logitech 720p pada robot penjaga gawang soccer beroda. Pada penelitian ini digunakan aplikasi Bahasa python, kemudian aplikasi python tersebut akan dilengkapi dengan dengan kebutuhan pendukung seperti numpy dan lain-lain. Hal ini

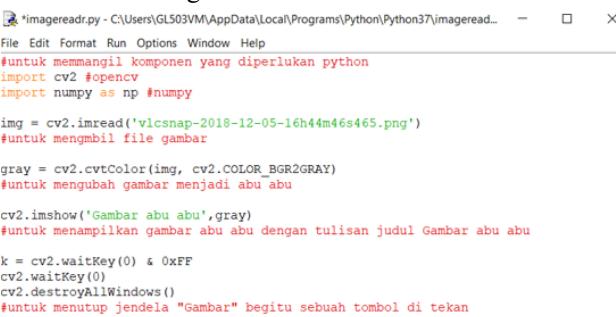
dimaksudkan agar aplikasi python mampu menggunakan library OpenCV yang fungsinya untuk menjalankan algoritma dengan menggunakan berbagai macam code yang tersimpan di dalam library OpenCV, setelah itu barulah aplikasi python dapat digunakan untuk merancang susunan algoritma untuk mendeteksi bola dengan dasar algoritma metode canny.

Metode ini adalah sistem pendeksi dengan dasar pengenalan bentuk dari gambar yang di tangkap yang nantinya akan digunakan untuk mendeksi posisi bola dari robot penjaga gawang. Metode canny ini sendiri merupakan method dimana gambar akan dibuat kabur dengan menggunakan filter gaussian, yang memiliki karakteristik tidak memiliki overshoot ke input step function sambil meminimalkan waktu naik dan turun. Karakteristik ini dinilai ideal dikarenakan memiliki kemungkinan waktu delay tersingkat, hal ini penting karena robot harus bertindak cepat karena harus menghadang bola yang datang secara real time.

Kemudian dicarilah garis horizontal, vertical dan diagonal pada gambar pada gambar yang telah dibuat kabur dengan menggunakan operator edge detection. Kemudian digunakan teknik Non-maximum suppression atau teknik penipis garis, karena masih ada sisa akibat variasi warna dan noise pada gambar, maka hasilnya akan di perjelas lagi dengan melakukan double threshold dan blob analysis.

2.1 Pengolahan citra menggunakan python terhadap file gambar

Python dapat digunakan untuk mengolah gambar, tapi sebelum python dapat mengolah gambar terlebih dahulu harus mengambil gambar yang diinginkan untuk diolah, untuk itu digunakan printah untuk mengambil dan menampilkan gambar dengan printah cv2.imshow, dan cv2.imread[14]. Selain dengan menggunakan python, warna juga dapat diubah menjadi abu-abu dengan menggunakan printah cv2.COLOR_BGR2GRAY yang dicontohkan sebagai berikut :



```
*imagereadr.py - C:\Users\GL503VM\AppData\Local\Programs\Python\Python37\imagereadr.py
File Edit Format Run Options Window Help
#untuk memanggil komponen yang diperlukan python
import cv2 #opencv
import numpy as np #numpy

img = cv2.imread('vlcsnap-2018-12-05-16h44m46s465.png')
#untuk mengambil file gambar

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
#untuk mengubah gambar menjadi abu-abu

cv2.imshow('Gambar abu abu',gray)
#untuk menampilkan gambar abu abu dengan tulisan judul Gambar abu abu

k = cv2.waitKey(0) & 0xFF
cv2.waitKey(0)
cv2.destroyAllWindows()
#untuk menutup jendela "Gambar" begitu sebuah tombol di tekan
```

(a)



(b)

Gambar 1.(a) tampilan python IDLE shell, (b) Gambar yang ditampilkan oleh python

2.2 Pengolahan gambar menggunakan HSV

Metode HSV menggunakan nilai dari warna, kepekatan, nilai (hue, saturation, value) berikut merupakan contoh pendekripsi warna kuning dengan menggunakan metode HSV.

```
File Edit Format Run Options Window Help
Untuk memanggil komponen yang diperlukan python
import numpy as np #numpy
import cv2 #openCV

#Untuk memanggil Gambar
img = cv2.imread('gambar4-warna.png')

#untuk mengubah warna dari RGB menjadi HSV
hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

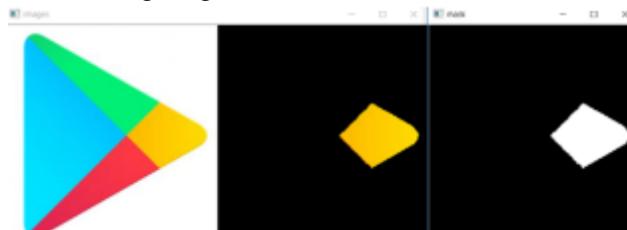
#batasan warna Kuning
lower_range = np.array([10, 215, 135], dtype=np.uint8)
upper_range = np.array([65, 255, 255], dtype=np.uint8)

#untuk menetap warna selain warna yang di deteksi
mask = cv2.inRange(hsv, lower_range, upper_range)
output = cv2.bitwise_and(img, img, mask = mask)

#untuk menampilkan gambar hasil
cv2.imshow('mask',mask)
cv2.imshow('Image', img)
cv2.imshow('Images', np.hstack([img, output]))

while(1):
    k = cv2.waitKey(0)
    if(k == 27):
        break
cv2.destroyAllWindows()
```

Gambar 2. Perintah yang digunakan untuk mendekripsi warna kuning dengan metode HSV



Gambar 3. Hasil pendekripsi warna kuning dengan metode HSV

2.3 Pengolahan citra terhadap video

Kamera yang digunakan adalah kamera webcam C525, dimana cameranya sendiri telah dilengkapi dengan

autoplay untuk windows 8 ke atas, software yang diperlukan akan di download secara otomatis sehingga camera bisa langsung dipergunakan. apabila masih menggunakan windows 7 atau windows vista, maka software webcam Logitech perlu di download, apabila menggunakan windows 8 keatas, camera dapat diakses dengan membuka aplikasi camera bawaan windows, kemudian apabila terdapat lebih dari satu kamera maka ganti display kamera ke kamera Logitech C525. Apabila camera sudah bisa digunakan maka selanjutnya adalah menguji aplikasi python dengan printah yang sederhana, berikut ini merupakan printah sederhana untuk menampilkan gambar pada python. Untuk mengambil video dengan warna abu abu maka digunakanlah printah sebagai berikut:

```
File Edit Format Run Options Window Help
Untuk memanggil komponen yang diperlukan python
import cv2 #openCV
import numpy as np #numpy

cap = cv2.VideoCapture(0)
#Untuk menanipol camera, angka pada () menentakan kamera yang di gunakan
#dimana 0 adalah camera lokal

while(True):
    ret, frame = cap.read()
    #Untuk menginformasikan bahwa frame adalah printah untuk membaca cap
    #dimana cap adalah printah untuk memanggil kamera yang telah di atur pada printah
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    #cv2.cvtColor(input_image, flag) adalah operasi warna, dimana flag adalah operasi
    #cv2.imshow('frame',gray)
    if cv2.waitKey(160) & 0xFF == ord('q'):
        break
    #Wait key untuk mengatur delay pada frame
    #print('') untuk mengatur tempo yang di gunakan untuk menghentikan program
    cap.release()
    cv2.destroyAllWindows()
#Untuk menutup semua jendela ketika printah di hentikan
```

Gambar 4. Perintah yang digunakan python untuk mengambil video abu abu

Setelah printah dimasukan, tekan Run>Run module atau F5 untuk menjalankan program, apabila printah dimasukan dengan benar maka python akan menampilkan gambar dari kamera yang telah di ubah menjadi abu abu.

2.4 Pengolahan video terhadap warna tertentu menggunakan HSV

Kebanyakan aplikasi video deteksi warna modern, lebih banyak yang menggunakan HSV dibandingkan RGB, hal ini disebabkan oleh karena nilai yang dihasilkan dari warna RGB sensitif dengan cahaya, sehingga akan sulit dalam menentukan batasan pada warna tertentu. Untuk mendekripsi gambar HSV, terlebih dahulu gambar di ubah dalam bentuk HSV dengan printah cv2.COLOR_BGR2HSV, seperti pada contoh berikut ini:

```

beckelipy - OpenCV (0.72)
File Edit Famar Run Options Window Help
# untuk memulakan baslik tangkapkan gambar dari webcam laptop ini
import cv2
import cv
import numpy as np
import threading

cap = cv2.VideoCapture(0) #untuk fungsi memanggil camera angka dalam (0) untuk menentukan kamera mana yang digunakan
kernel = np.ones((5,5),np.uint8)
while True:
    # Capture frame-by-frame
    ret, frame = cap.read() # menangkap frame adalah mencapai file dari cap ( dimana cap adalah printah yang tel
    #cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY, frame) #mengubah frame adalah operasi warna, dimana flag adalah operasi yang dilakukan.
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    lower_orange = np.array([5, 96, 30])
    upper_orange = np.array([16, 255, 255])

    lower_kuning = np.array([14, 100, 100])
    upper_kuning = np.array([25, 255, 255])

    orange_mask = cv2.inRange(hsv, lower_orange, upper_orange)
    kuning_mask = cv2.inRange(hsv, lower_kuning, upper_kuning)

    mask = orange_mask + kuning_mask

    res = cv2.bitwise_and(frame, frame, mask=mask)
    #dilatasi = cv2.Canny(mask, 100, 200)
    closing = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)

    contours = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[-2]
    center = None

    if len(contours) > 0:
        c = max(contours, key=cv2.contourArea)
        ((x, y), radius) = cv2.minEnclosingCircle(c)
        M = cv2.moments(c)
        center = (int(M["m01"] / M["m00"]), int(M["m10"] / M["m00"]))

        if radius > 10:
            cv2.circle(frame, (int(x), int(y)), int(radius), (0, 0, 255), 1)
            cv2.circle(frame, (int(x), int(y)), 3, (0, 255, 255), 1)
            cv2.putText(frame, "Kuning", (int(x) + 10, int(y) + 10), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0, 255, 255), 1)
            cv2.putText(frame, "Orange", (int(x) + 10, int(y) + 30), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0, 255, 255), 1)
            x0,y0,x1,y1 = cv2.boundingRect(c)
            x0Line = x0+10
            y0Line = y0+10
            x1Line = x1+10
            y1Line = y1+10
            print("x0Line", x0Line)
            y0Line
            print("y0Line", y0Line)
            print("x1Line", x1Line)
            y1Line

        cv2.imshow('frame', frame) # perintah untuk menampilkan fungsi frame
        cv2.imshow('mask', mask)
        cv2.imshow('res', res)
        cv2.imshow('Edges', edges)

    if cv2.waitKey(1) & 0xFF == ord('q'): # .waitkey meminta seseorang delay frame yang di tampilkan, sementara ord 'esok'
        cv2.destroyAllWindows() # destroy all windows


```

Gambar 5. perintah yang digunakan untuk mendeteksi warna dengan metode HSV

Perintah tersebut digunakan pada robot penjaga gawang yang menggunakan metode HSV untuk mendeteksi warna kuning dan orange pada bola, agar python dapat mengenal warna kuning ketika gambar di ubah dalam bentuk HSV. Data 8bit dari HSV warna kuning orange, dimana HSV sendiri terdiri atas Hue,yang merupakan atribut suatu warna yang secara visual mirip dengan suatu atau berbagai warna. Kemudian Saturation, merupakan ukuran seberapa cerah terhadap warnanya sendiri. Sedangkan Value, atau yang biasa disebut juga lightnes merupakan ukuran seberapa terang warna dibandingkan warna putih terang. Setelah itu python dapat menangkap warna yang telah ditentukan seperti pada gambar seperti output berikut:



Gambar 6. hasil deteksi bola dengan dengan metode HSV

2.5 Pengolahan video terhadap pergerakan tertentu dengan Optical Flow

Optical flow adalah polah pada pergerakan sebuah objek diantara frame yang bersambungan yang diakibatkan oleh pergerakan benda atau kamera itu sendiri. Optical Flow sendiri menggunakan asumsi bahwa intensitas piksel pada objek antara kedua frame yang berhubungan tidak berubah, dan piksel di sebelahnya memiliki pergerakan serupa, printah liblary yang digunakan untuk menggunakan optical follow adalah cv2.calcOpticalFlowPyrLK seperti pada contoh berikut

```

# untuk memasukkan library pada printah
import cv2
import numpy as np

cap = cv2.VideoCapture(0) #untuk fungsi memanggil camera angka dalam (0) untuk menentukan kamera mana yang digunakan, 0 = lokal
_, frame = cap.read() # menangkap frame yang di dapat dari cap
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #mengubah frame adalah operasi warna, dalam hal ini, dari warna menjadi bitan putih

lk_params = dict(winSize = (15, 15),
                 maxLevel = 4,
                 criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_COUNT, 10, 0.03))

# Mouse function (untuk memilih titik pada objek dengan menggunakan mouse)
def select_point(event, x, y, flags, param):
    global point, point_selected
    point = (x, y)
    point_selected = True
    old_point = np.array([1,1])
    if event == cv2.EVENT_LBUTTONDOWN:
        point = (x, y)
        point_selected = True
        old_point = np.array([x, y], dtype=np.float32)
    cv2.namedWindow("frame")
    cv2.setMouseCallback("frame", select_point)

# parameter yang dipunakan ketika titik belum terpasang
point_selected = False
point = (0,0)
old_point = np.array([1,1])
# parameter yang dipunakan ketika titik sudah terpasang
# membuat frame baru sebagai pembanding arah pergerakan objek
while True:
    _, frame = cap.read()
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    if point_selected == True:
        cv2.circle(frame, point, 5, (0, 0, 255), 2)

    #penerapan operasi optical flow dengan membandingkan frame lama dan frame baru
    new_points, status, err = cv2.calcOpticalFlowPyrLK(old_gray, gray_frame, old_points, None, **lk_params)
    old_gray = gray_frame.copy()
    old_points = new_points

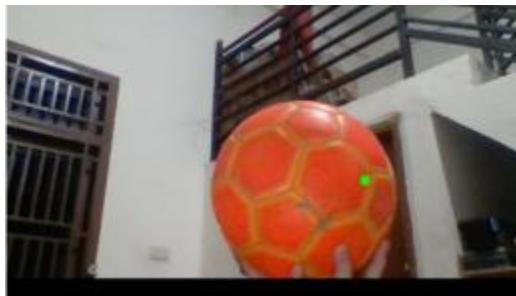
    x, y = new_points.ravel()
    cv2.circle(frame, (x, y), 5, (0, 255, 0), -1)
    print("x", x, "y", y)

    cv2.imshow("frame", frame)

```

Gambar 7. Perintah yang digunakan untuk mendeteksi benda dengan flow motion

Perintah tersebut digunakan untuk mendeteksi pergerakan benda dengan menentukan titik pada objek yang akan di track dan kemudian secara otomatis akan memantau perubahan possisi objek dengan membandingkan perpindahan vector dari titik yang dilacak.



Gambar 8. Pendekatan benda dengan menggunakan flowmotion

Optical Flow sendiri merupakan pola pada objek atau permukaan secara visual yang diakibatkan oleh pergerakan relatif antara pengamat dan yang di amati.

3 Hasil dan Pembahasan

3.1 Pengujian Objek

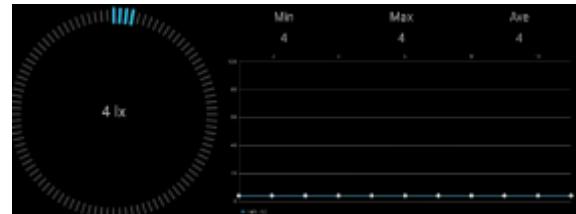
Pengujian objek akan dilakukan dengan beberapa objek dengan bentuk dan ukuran dari objek yang berbeda. Pengujian akan dilakukan dengan membandingkan berbagai benda dengan pendekatan objek.

Table 1 Pengujian objek terhadap pendekatan kamera

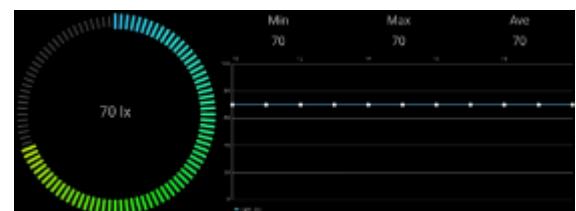
Objek	Ukuran Objek	Objek pada koordinat awal	Objek pada koordinat akhir	Koordinat objek awal	Koordinat objek Akhir	Lumen
Bola Kecil	Diameter: 8 cm			Koordinat X: 154 Koordinat Y: 221	Koordinat X: 424 Koordinat Y: 210	58 lx
Bola Besar	Diameter: 20 cm			Koordinat X: 110 Koordinat Y: 185	Koordinat X: 490 Koordinat Y: 191	59 lx
Kotak	Sisi: 6 cm			Koordinat X: 28 Koordinat Y: 233	Koordinat X: 477 Koordinat Y: 213	56 lx

3.2 Pengujian pada lumen meter

Pengujian pada lumen meter dilakukan dengan menggunakan aplikasi smart phone, dan dilakukan dengan cara menangkap cahaya selama ± 10 detik.



Gambar 9. Hasil deteksi cahaya pada lumen meter pada ruangan tanpa penerangan



Gambar 10. Hasil deteksi cahaya pada lumen meter pada ruangan dengan lampu



Gambar 11. Hasil deteksi cahaya pada lumen meter di luar ruangan

3.3 Hasil Analisa Video Ketika Berada pada ruangan dengan nilai intensitas cahaya lumen 50 lx dan lumen 80 lx

Table 2. Hasil Deteksi Objek dengan HSV

Jarak (CM)	Koordinat yang terbaca	80 lx Dengan Metode Optical Flow Tracking	50 lx Dengan Metode Optical Flow Tracking	60 lx Dengan Metode HSV Object Tracking
150	Koordinat X: 238 Koordinat Y: 185			
90	Koordinat X: 316 Koordinat Y: 169			
50	Koordinat X: 318 Koordinat Y: 186			

Berdasarkan pengujian yang telah dilakukan, penelitian ini dapat dikatakan berhasil. Adapun keberhasilannya adalah mampu mendeteksi objek dari disekitar kamera seperti bola kecil, bola besar dan kotak seperti pada tabel 1, meskipun pembacaan akhir nampak bergeser objeknya. Untuk pengujian terhadap pencahayaan, deteksi objek tidak terlalu terganggu seperti pada gambar 9, 10 dan 11. Untuk deteksi objek dengan algoritma HSV berhasil baik dengan jarak 50 cm, 90 cm dan 150 cm namun kelemahan deteksi dengan algoritma ini jika latar belakang objek yang dideteksi memiliki warna yang sama sehingga perlu ditambahkan algoritma lain.

4 Kesimpulan

Sistem deteksi bola yang dikembangkan pada robot penjaga gawang soccer beroda menggunakan pustaka OpenCV dan kamera webcam Logitech 720p berhasil mencapai tujuan utama berupa pengenalan dan pelacakan bola secara real-time. Implementasi metode pengolahan citra berbasis pustaka OpenCV memungkinkan ekstraksi fitur visual bola menggunakan teknik segmentasi warna berbasis ruang warna HSV (Hue, Saturation, Value). Pemilihan ruang warna HSV terbukti efektif dalam memisahkan objek bola dengan latar belakang, terutama dalam kondisi pencahayaan yang bervariasi. Pada deteksi bola dengan menggunakan metode optical flow seringkali titik focus bergeser jauh dari pusat bola. Meski pengaruh cahaya terhadap pendekslsian bola tidak terganggu, pendekslsian dengan menggunakan optical flow masih bisa terganggu apabila warna objek dengan warna latar sama. Pada pendekslsian optical flow objek masih harus di tandai terlebih dahulu sebelum bisa di lacak. Objek yang dapat di lacak oleh optical flow tidak terbatas hanya benda Bulat.

5 Daftar Pustaka

- [1] D. K. Prasad, “Survey of the problem of object detection in real images,” *Int. J. Image Process.*, vol. 6, no. 6, p. 441, 2012.
- [2] M. Vashisht and B. Kumar, “A survey paper on object detection methods in image processing,” in *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, IEEE, 2020, pp. 1–4.
- [3] R. A. Fatekha, B. S. B. Dewantara, and H. Oktavianto, “Sistem Deteksi Bola pada Robot Kiper Pemain Sepakbola Beroda,” *J. Integr.*, vol. 13, no. 2, pp. 127–134, 2021.
- [4] B. R. Wijaya and M. Khaycal, “Robot Penjaga Gawang Pada Kontes Robot Sepak Bola Indonesia Jenis Beroda 2019,” *Autocracy J. Otomasi, Kendali, dan Apl. Ind.*, vol. 7, no. 2, pp. 60–65, 2020.
- [5] A. Mordvintsev and K. Abid, “Opencv-python tutorials documentation,” *Obtenido https://media.readthedocs.org/pdf/opencv-python-tutorials/latest/opencv-python-tutorials.pdf*, 2014.
- [6] A. Rosebrock, “Practical python and opencv,” *Miami: pyimagesearch*, 2016.
- [7] S. Uchida, “Image processing and recognition for biological images,” *Dev. Growth Differ.*, vol. 55, no. 4, pp. 523–549, 2013.
- [8] W. Yao, W. Dai, J. Xiao, H. Lu, and Z. Zheng, “A simulation system based on ros and gazebo for robocup middle size league,” in *2015 IEEE international conference on robotics and biomimetics (ROBIO)*, IEEE, 2015, pp. 54–59.
- [9] C. D. M. T. Purnama and G. E. Setyawan, “Deteksi Bola pada Robot Humanoid Sepak Bola dengan Menggunakan Haar Cascade dan HSV,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 1, 2025.
- [10] M. R. Setiawan, “Penerapan Computer Vision pada Sistem Pengejaran Benda Bergerak: Studi Kasus Robot KRSBI (Kontes Robot Sepak Bola Indonesia) Beroda,” 2024, *Politeknik Manufaktur Bandung*.
- [11] A. F. Qhofari, A. Habibi, and A. Arman, “Pengembangan Sistem Navigasi Dan Pengendalian Robot Sepak Bola Beroda Untuk Menjejak Menggunakan Algoritma Yolo,” *Sci. ELECTRO*, vol. 19, no. 1, 2025.
- [12] A. A. GUNTAMA, “Prototype Robot Line Follower Pengangkut Sampah Dengan Memanfaatkan Solar Cell Berbasis Mikrokontroler Arduino Mega 2560,” 2017, *Universitas Negeri Jakarta*.
- [13] G. R. Anandya and S. T. Fajar Budiman, “Rancang Bangun Lengan Robot Penjepit PCB 3 DOF Berbasis Arduino Untuk Proses Etching PCB Otomatis,” *Surabaya Inst. Teknol. Sepuluh Nop.*, 2017.
- [14] D. P. S. A. Diandra, “Monitoring Lahan Parkir Menggunakan Python, CV2, CVZONE DAN NUMPY,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 1, pp. 157–161, 2025.