

Deteksi Arah Gerak Bola dengan Metode *Optical Flow* pada Robot Penjaga Gawang

Rendi Nova Handika¹, Budi Rahmani^{2*}

Teknik Informatika, STMIK Banjarbaru, Banjarbaru, Indonesia
 *e-mail corresponding author: budirahmani@gmail.com

Abstract

The goalkeeper robot developed by STMIK Banjarbaru still faces challenges in improving the response to the ball to reduce the possibility of goals being scored by opposing robots. The ability of the robot keeper is currently limited to the starting point of the ball so that when the ball bounces toward the goal, the robot has not been able to predict the end point of the ball, resulting in the movement of the goalkeeper robot, which is sometimes slow in blocking the ball. This research was conducted to develop a detection system of a ball direction using the optical flow method. The Logitech C270 series web camera device is used on a robot with a laptop as a video data processor using Python language. The results show that this system is capable of detecting balls in all five directions or quadrants, including 90°, 135°, 180°, 225°, and 270. The detection results of these directions are represented as position information data, such as a ball in a state of still, right, bottom right, top right, left, top left, bottom left, top, and bottom.

Keywords: *goalkeeper; response; ball direction; robot motion;*

Abstrak

Robot Penjaga gawang yang dikembangkan oleh STMIK Banjarbaru masih menghadapi tantangan dalam meningkatkan respon terhadap bola agar dapat mengurangi kemungkinan terjadinya gol oleh robot lawan. Kemampuan robot keeper saat ini terbatas pada titik awal bola saja, sehingga ketika bola melambung ke arah gawang, robot belum mampu memprediksi titik akhir bola, dan mengakibatkan pergerakan robot goal keeper yang kadang lambat dalam menghalau bola. Penelitian ini dilakukan untuk mengembangkan sistem deteksi arah gerak bola menggunakan metode optical flow. Perangkat kamera web seri Logitech C270 digunakan pada robot dengan sebuah laptop sebagai pengolah data video menggunakan bahasa python. Hasil penelitian menunjukkan bahwa sistem ini mampu mendeteksi bola di kelima arah atau kuadran antara lain 90°, 135°, 180°, 225°, dan 270. Hasil deteksi arah ini direpresentasikan menjadi data keterangan posisi seperti bola pada keadaan 'diam', 'kanan', 'kanan bawah', 'kanan atas', 'kiri', 'kiri atas', 'kiri bawah', 'atas', dan 'bawah'.

Kata kunci: *penjaga gawang; respon; arah bola; gerak robot;*

1. Pendahuluan

Kontes Robot Indonesia (KRI) merupakan wadah yang berharga untuk mempromosikan dan memajukan teknologi robotika di kalangan mahasiswa. KRI merupakan kompetisi robotika bagi tim mahasiswa dari institusi yang terdaftar, baik Perguruan Tinggi Negeri maupun Swasta, di bawah Kementerian Riset, Teknologi, dan Pendidikan Tinggi. Kompetisi ini bertujuan untuk menumbuhkan kreativitas dan pertumbuhan di kalangan mahasiswa perguruan tinggi [1].

Salah satu event dalam KRI adalah Kontes Robot Sepak Bola Indonesia Beroda (KRSBI Beroda), dimana berlangsung pertandingan sepak bola robotik dengan robot yang dilengkapi roda untuk mobilitas [2]. Robot sepak bola berbasis visi ini memanfaatkan kamera dan algoritme komputasi untuk deteksi dan pelacakan objek [3]. Selama pertandingan sepak bola, dua robot penyerang dan satu robot penjaga gawang bersaing untuk mencetak gol dan mencegah lawan melakukannya [4].

Peran utama robot penjaga gawang adalah merespons bola secara efektif dan memblokir tembakan yang masuk untuk meningkatkan peluang kemenangan [5]. Tantangannya terletak pada waktu respons robot penjaga gawang terhadap bola, karena harus memprediksi lintasan

bola secara akurat [6]. Saat ini respon robot terbatas pada posisi awal bola sehingga menyebabkan keterlambatan dalam mengejar pergerakan bola. Untuk mengatasi masalah tersebut, digunakan metode aliran optik (*optical flow*), menganalisis bingkai gambar dari gambar yang terhubung dan membaginya menjadi kuadran untuk mendeteksi arah bola [7].

Dengan menerapkan metode *optical flow* dan membuat model keputusan berdasarkan analisis kuadran, penelitian ini dilakukan untuk meningkatkan respon robot penjaga gawang terhadap pergerakan bola dan memperkecil kemungkinan robot lawan mencetak gol. Lokasi yang tepat dari posisi bola sangat penting untuk pergerakan strategis robot, menjadikannya aspek penting dari kesuksesan robot penjaga gawang dalam kompetisi.

2. Tinjauan Pustaka

Robot penjaga gawang memiliki fungsi untuk menghalangi bola agar tidak masuk ke gawang. Untuk mencapai tujuan ini, robot harus dapat mendeteksi bola dengan akurat dan mengatur pergerakannya berdasarkan informasi tersebut. Beberapa penelitian telah mengembangkan robot penjaga gawang dengan kemampuan mendeteksi arah gerak bola menggunakan metode *optical flow* pada robot beroda [8].

Penerapan metode *optical flow* salah satunya dilakukan menggunakan tools *Visual Studio Code*. Metode *optical flow* ini menghitung aliran medan pada citra dan mengelompokkan sesuai karakteristik *optical flow* dari citra. Dengan metode *optical flow* ini, robot dapat mendapatkan informasi gerakan lengkap dari objek, sehingga objek bergerak dari latar belakang dapat dideteksi dengan akurasi 85%. Metode ini menjadi yang pertama menggunakan intensitas cahaya sebagai dasar untuk mendeteksi objek. *Optical flow* memiliki dua algoritme, yaitu algoritme Horn-Schunck dan algoritme Lucas-Kanade. Algoritme Horn-Schunck dapat menghasilkan kepadatan tinggi pada aliran vektor, tetapi lebih sensitif terhadap suara dibandingkan dengan metode lokal [7], [9], [10].

Terdapat beberapa penelitian terkait yang relevan dengan topik ini. Salah satunya adalah penelitian [11] yang mengimplementasikan pergerakan robot penjaga gawang KRSBI (beroda) dengan metode Fuzzy PID. Metode Fuzzy PID menggunakan sudut kamera dan sensor ultrasonic untuk mengatur pergerakan robot [12]–[15].

Penelitian lainnya, seperti [16], menggunakan metode *Color Tracking* untuk mengontrol robot sepak bola beroda. Robot ini melakukan tracking objek bola dengan warna tertentu dan menggunakan lapangan sepak bola dengan ukuran tertentu. Pengolahan citra dilakukan untuk mengenali objek yang harus diikuti berdasarkan data kamera [17], [18].

Penelitian [19] juga relevan dengan menggunakan sistem deteksi bola berdasarkan warna bola dan latar lapangan pada robot Bareleng FC. Proses deteksi objek dilakukan dengan segmentasi HSV warna bola dan warna latar lapangan [20][21]. Fitur auto dari kamera dinonaktifkan untuk mengurangi pengaruh perubahan intensitas cahaya dan digunakan library *cvtColor* untuk mengukur ukuran bola.

Selanjutnya, penelitian [20] menggunakan metode *Hough Circle* untuk mendeteksi bola pada robot penjaga gawang [22]. Metode ini mengatasi permasalahan pendeteksian bola berwarna oranye pada robot KRSBI Beroda dengan *EmguCV* sebagai library [20], [23]. Penggunaan metode ini membantu mengurangi kesalahan deteksi bola dan arah datang bola.

Penelitian [24] menggunakan metode *template matching* dalam mendeteksi bola pada bidang datar dengan bantuan Labview. Proses *template matching* memudahkan pengguna dalam mendeteksi objek dengan cepat dan tidak kompleks. Penelitian yang telah dilakukan terkait deteksi arah gerak bola menggunakan *optical flow* pada robot beroda yang melibatkan perhitungan shift vector. Hal ini dilakukan agar piksel yang merupakan pembeda pada perpindahan objek antara dua gambar yang berurutan. Dengan menggunakan metode *optical flow* ini, robot dapat menentukan arah pergerakan bola yang menuju gawang yang dijaganya berdasarkan informasi pergerakan objek tersebut [25].

Penelitian ini menguji kinerja metode deteksi pergerakan bola yang menuju ke robot menggunakan kamera web. Adapun bola yang menggelinding berjarak lebih dari 100 cm. Sedangkan penelitian [26] juga menguji coba proses deteksi bola dalam jarak 100 cm hingga 400 cm, namun dengan metode dan algoritme berbeda. Penelitian [20] berusaha mendeteksi bola hingga jarak 600 cm.

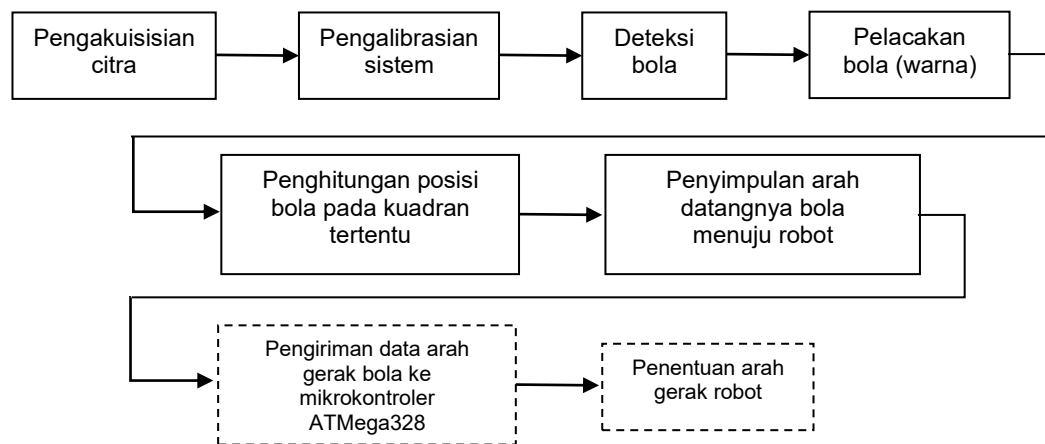
Dari sisi skenario pengujian, penelitian ini menguji hasil deteksi sistem terhadap bola yang bergerak. Dalam uji coba, ada enam skenario berbeda, yaitu pergerakan bola ke arah depan, belakang, kiri, kanan, serta arah diagonal seperti depan kiri, depan kanan, belakang kiri,

dan belakang kanan. Masing-masing skenario diulang sebanyak 30 kali. Penelitian-penelitian sebelumnya menggunakan skenario yang serupa namun tidak sama jumlah pengulangannya. Begitu pula metode dan algoritme yang penelitian sebelumnya gunakan, beberapa diantaranya berbeda.

3. Metodologi

Secara umum sistem yang dibangun dapat mendeteksi bola dan memperkirakan arah gerakan bola yang menuju ke gawang. Adapun urutan proses yang dilakukan ditunjukkan pada blok diagram sistem (Gambar 1). Namun blok atau bagian yang berada dalam kotak bergaris putus-putus tidak dibahas pada penelitian ini. Secara umum metode optical flow digambarkan sejak proses pelacakan warna bola, penghitungan posisi bola pada kuadran tertentu hingga penyimpulan arah datangnya bola menuju robot. Dalam artian proses komputasi dilakukan untuk melihat arah pergerakan warna dari objek bola yang diujicobakan dalam penelitian ini.

Jika dirunut kembali proses yang terjadi adalah dimulai pada proses akuisisi citra menggunakan kamera web (webcam logitech C270) dan dilanjutkan dengan proses kalibrasi oleh sistem. Proses kalibrasi dilanjutkan dengan proses filterisasi warna guna pendeteksian bola. Jika warna bola telah terdeteksi, barulah kemudian proses *tracking* atau pelacakan dilakukan selama masih terlihat oleh kamera yang digunakan.



Gambar 1. Blok diagram sistem

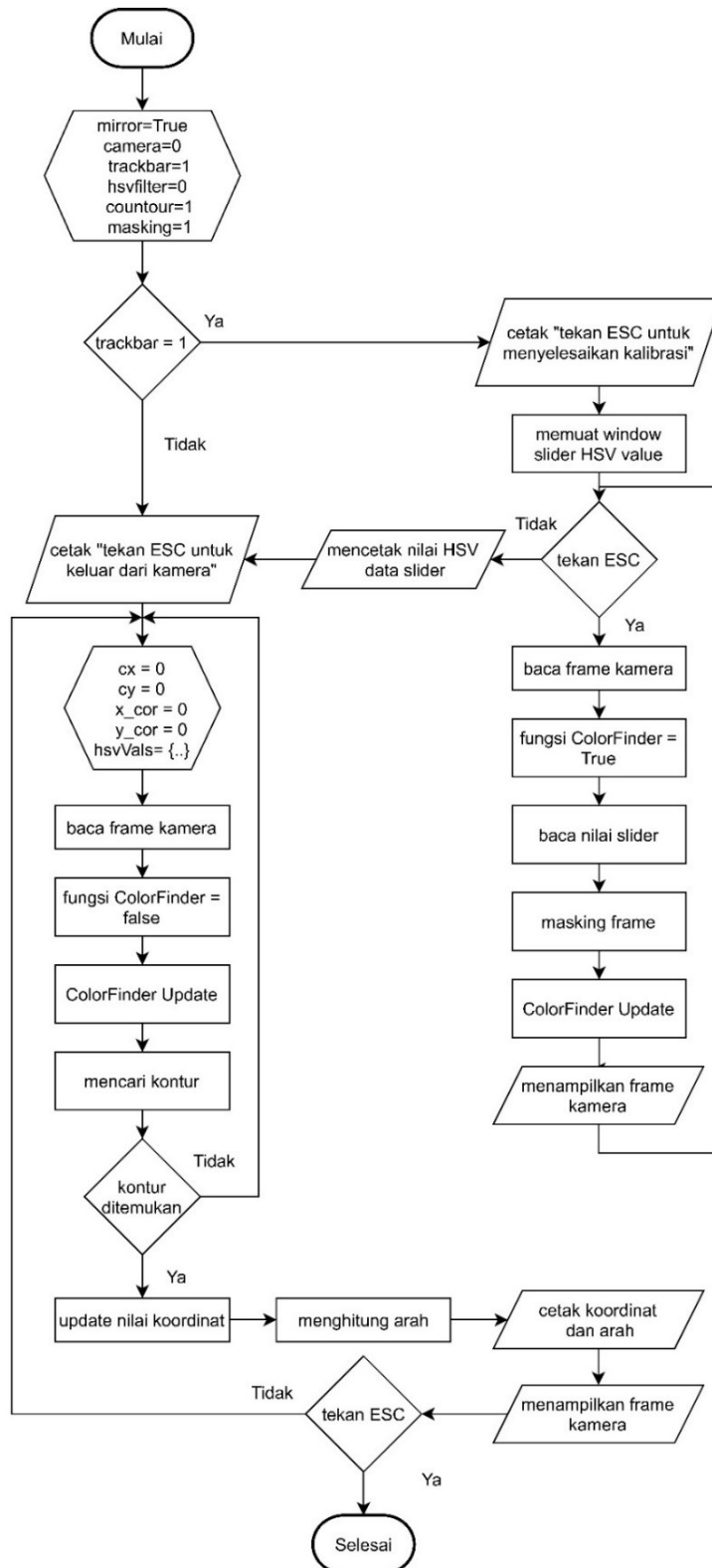
Gambar 2 memperlihatkan lebih detail proses yang telah dijelaskan secara umum pada

Gambar 1. Program diawali dengan membaca fungsi *main* yang berisi variabel-variabel seperti *mirror*, *camera*, *trackbar*, *hsvfilter*, *countour*, dan *masking*. Program memiliki instruksi untuk memeriksa nilai variabel-variabel tersebut. Jika nilai variabel *trackbar* adalah 1, maka sistem akan memasuki fungsi kalibrasi. Tujuan dari kalibrasi ini adalah untuk mendapatkan nilai HSV dari objek yang akan dilacak. Selanjutnya, program akan mencetak perintah untuk pengguna agar menekan tombol ESC untuk mengakhiri proses kalibrasi. Jika tombol ESC ditekan, maka proses kalibrasi akan berhenti, dan sistem akan menampilkan window slider untuk nilai HSV. Jika tombol ESC tidak ditekan, proses kalibrasi akan terus berlanjut.

Selanjutnya, program akan membaca frame dari kamera dan memasuki proses fungsi *colorfinder* dengan mengatur nilai variabel *colorfinder* menjadi true, karena kita ingin mengambil nilai HSV. Program kemudian akan membaca nilai dari slider dan melakukan *masking* antara nilai *lower* dan *upper* HSV. Fungsi *colorfinder update* akan digunakan untuk mengupdate nilai HSV berdasarkan hasil *slider*. Selanjutnya, program akan menampilkan *frame* dari kamera, dan jika tombol ESC ditekan, proses kalibrasi akan selesai, dan nilai HSV dari *slider* akan dicetak.

Setelah berhasil mendapatkan nilai *preset* HSV, *trackbar* yang sebelumnya bernilai 1 akan diubah menjadi 0 (nol) untuk melanjutkan proses pelacakan bola. Jika *trackbar* bernilai 0, program akan masuk ke proses pelacakan bola. Program akan mencetak perintah "Tekan ESC untuk keluar dari kamera" dan kemudian membaca variabel *cx*, *cy*, *x_cor*, *y_cor* yang berfungsi untuk menentukan koordinat. Program akan membaca *frame* dari kamera, dan nilai variabel

colorfinder akan diubah menjadi *false* karena kita sudah mendapatkan nilai preset HSV dari proses kalibrasi sebelumnya. Program kemudian akan masuk lagi ke fungsi *colorfinder update* untuk mengupdate warna berdasarkan nilai HSV hasil kalibrasi.

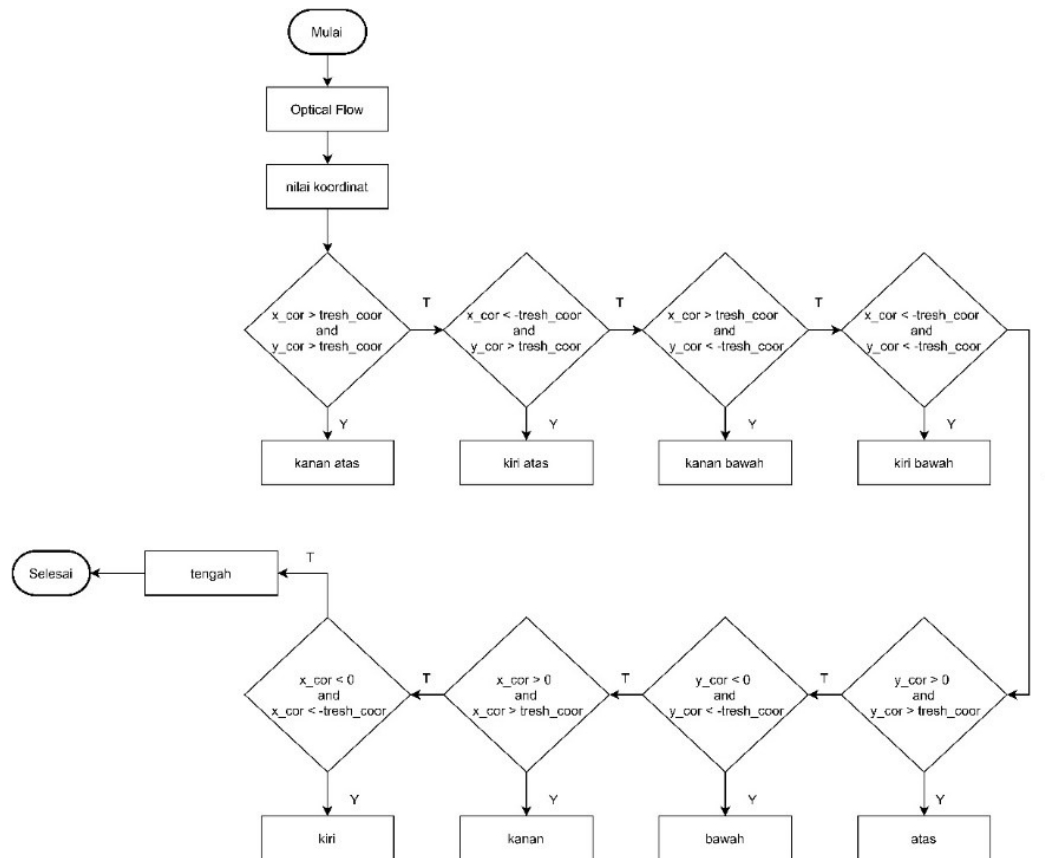


Gambar 2. Detail diagram alir sistem

Selanjutnya, program akan mencari kontur dari *masking* dan *imgcolor*. Jika kontur tidak ditemukan, maka program akan kembali ke proses membaca variabel koordinat. Jika kontur ditemukan, nilai koordinat akan diupdate, kemudian program akan menghitung arah dan koordinat, dan mencetaknya. Selanjutnya, program akan menampilkan frame dari kamera. Jika tombol ESC ditekan, sistem akan berhenti, dan jika tidak, program akan terus melanjutkan proses pelacakan bola.

Gambar 3 memperlihatkan diagram alir penentuan arah gerak bola yang dideteksi. Dalam mencari arah gerak bola, sistem akan mencari titik tengah bola terlebih dahulu. Kemudian, sistem akan memeriksa posisi koordinat x_cor dan y_cor terhadap nilai $tresh_coor$. Jika x_cor lebih besar dari $tresh_coor$ dan y_cor lebih besar dari $tresh_coor$, bola akan dianggap berada di kanan atas. Jika x_cor lebih kecil dari $min\ tresh_coor$ dan y_cor lebih besar dari $tresh_coor$, bola dianggap berada di kiri atas. Jika x_cor lebih besar dari $tresh_coor$ dan y_cor lebih kecil dari $min\ tresh_coor$, bola dianggap berada di kanan bawah. Jika x_cor lebih kecil dari $min\ tresh_coor$ dan y_cor lebih kecil dari $min\ tresh_coor$, bola dianggap berada di kiri bawah.

Selain itu, jika y_cor lebih besar dari 0 (nol) dan y_cor lebih besar dari $tresh_coor$, bola dianggap berada di atas. Jika y_cor lebih kecil dari 0 (nol) dan y_cor lebih kecil dari $min\ tresh_coor$, bola dianggap berada di bawah. Jika x_cor lebih besar dari 0 (nol) dan x_cor lebih besar dari $tresh_coor$, bola dianggap berada di kanan. Jika x_cor lebih kecil dari 0 (nol) dan x_cor lebih kecil dari $min\ tresh_coor$, bola dianggap berada di kiri. Jika tidak memenuhi syarat-syarat tersebut, maka bola dianggap berada di tengah.



Gambar 3. Diagram alir penentuan arah gerak bola yang dideteksi

Dalam rangka proses kendali berikutnya yang melibatkan mikrokontroler menggunakan sistem minimum arduino, maka peneliti juga mengumpulkan sinyal via kabel USB. Jadi mikrokontroler Atmega328 juga diprogram untuk memberikan indikasi berupa kedipan LED yang terpasang di papan Arduino. Indikasi ini merupakan indikasi atas gerakan yang akan dilakukan oleh robot beroda (penjaga gawang) untuk mengantisipasi datangnya bola ke arah robot. Dan indikasi arah yang diberikan dari komputer ke mikrokontroler antara lain: jika bola

berada di kiri, LED akan berkedip tiga kali. Jika bola berada di kanan, LED akan berkedip dua kali. Perintah dan atau indikasi yang diberikan ke mikrokontroler pada prinsipnya bisa beragam dan cukup banyak, tergantung keperluan. Namun karena fokus dari penelitian yang telah dilakukan adalah pada proses pengolahan citra dan atau video, maka bahasan ke arah data yang dikirim ke mikrokontroler tidaklah secara detail disampaikan.

Berikutnya *pseudocode* dari sistem yang dikembangkan diantaranya adalah:

1. Tracking_bar

```
cap = cv2.VideoCapture(camera)
myColorFinder = ColorFinder(True)
hsvVals="red"
while True:
    success, img = cap.read()
    img = cv2.flip(img,1)
    imgColor, mask = myColorFinder.update(img, hsvVals)
    img = cv2.resize(img, (0, 0), None, 0.7, 0.7)
    cv2.imshow("Image", img)
    cv2.imshow("ImageColor", imgColor)
    k = cv2.waitKey(5) & 0xFF
    if k == 27:
        cap.release()
        cv2.destroyAllWindows()
        break
    print("tekan ESC untuk keluar dari kamera")
    show_webcam(mirror=mirror,camera=camera,hsvfilter=hsvfilter,masking=
        masking,)
    input("tekan ENTER untuk keluar")
```

2. Show_webcam

```
font = cv2.FONT_HERSHEY_SIMPLEX
location = (10,20)
fontScale = .5
fontColor = (0,255,0)
lineType = 2
cam = cv2.VideoCapture(camera)
myColorFinder = ColorFinder(False)
cx = 0
cy = 0
x_cor = 0
y_cor = 0
hsvVals= {'hmin': 1, 'smin': 28, 'vmin': 0, 'hmax': 14, 'smax': 255, 'vmax': 255}
while True:
    _, frameori = cam.read()
    if mirror:
        frameori = cv2.flip(frameori, 1)
    frame = frameori.copy()
    hsv = cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
    imgColor, mask = myColorFinder.update(frame, hsvVals)
    imgContours, contours = cvzone.findContours(frame, mask, minArea=500)
    if contours:
        cx,cy = contours[0]['center']
        cv2.circle(imgContours, (cx,cy),5,(0,255,0),cv2.FILLED)
        cv2.circle(frame, (cx,cy),5,(0,255,0),cv2.FILLED)
        x_cor = cx - 320
        y_cor = 240 - cy
        tresh_coor = 30
        if( x_cor > tresh_coor and y_cor > tresh_coor):
            print("kanan atas")
            arah="kanan atas"
```

```

elif( x_cor < -tresh_coor and y_cor > tresh_coor):
print("kiri atas")
arah="kiri atas"
elif( x_cor > tresh_coor and y_cor < -tresh_coor):
print("kanan bawah")
arah="kanan bawah"
elif( x_cor < -tresh_coor and y_cor < -tresh_coor):
print("kiri bawah")
arah="kiri bawah"
elif( y_cor > 0 and y_cor > tresh_coor):
print("atas")
arah="atas"
elif( y_cor < 0 and y_cor < -tresh_coor):
print("bawah")
arah="bawah"
else :
if( x_cor > 0 and x_cor > tresh_coor ):
print("kanan")
arah="kanan"
elif( x_cor < 0 and x_cor < -tresh_coor):
print("kiri")
arah="kiri"
else:
arah="tengah"
print("tengah")
frame=cv2.line(frame,((320-tresh_coor),0),((320-tresh_coor),480),(255,0,0),5)
frame=cv2.line(frame,((320+tresh_coor),0),((320+tresh_coor),480),(255,0,0),5)
frame=cv2.line(frame,(0,(240 - tresh_coor)),(640,(240 - tresh_coor)),(255,0,0),5)
frame=cv2.line(frame,(0,(240+tresh_coor)),(640,(240+ tresh_coor)),(255,0,0),5)
tulis(frame,(10,20),'Koordinat')
tulis(frame,(10,40),'%d,%d'%(x_cor,y_cor))
tulis(frame,(10,60),f'arah : {arah}')
frame = cv2.resize(frame, (640, 480))
if masking:
cv2.imshow('masking', mask)
if hsvfilter:
cv2.imshow('filter', imgContours)
if countour:
cv2.imshow('contour', imgContours)
cv2.imshow('kamera', frame)
if cv2.waitKey(1) == 27:
break
cv2.destroyAllWindows()

```

3. Main

```

def main():
mirror=True
camera=0
trackbar=0
hsvfilter=0
countour=1
masking=1
if trackbar:
print("tekan ESC untuk menyelesaikan kalibrasi")
track_bar(mirror=mirror,camera=camera,hsvfilter=hsvfilter,masking=masking)
else:
show_webcam(mirror=mirror,camera=camera,hsvfilter=hsvfilter,masking=masking,countour=countour)
print("tekan ESC untuk keluar")
if __name__ == '__main__':
main()

```

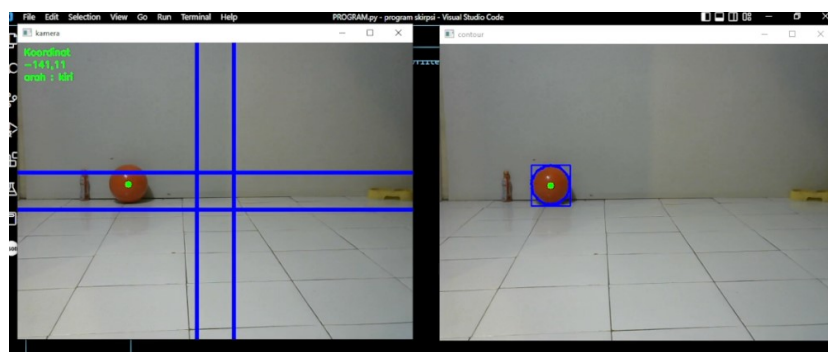
Pengujian sistem yang dikembangkan terdiri dari enam skenario yang masing-masing dilakukan 30 kali percobaan. Berikut adalah detail penentuan pada setiap skenario:

- Skenario pertama adalah menggelindingkan bola dari kanan ke kiri (dalam posisi tengah layar). Untuk memenuhi kriteria, nilai koordinat deteksi bagian kanan harus berada dalam rentang kurang dari 320 hingga 30, dan koordinat deteksi bagian kiri harus berada dalam rentang lebih dari -30 hingga -320.
- Skenario kedua adalah menggelindingkan bola dari kiri ke kanan (dalam posisi tengah layar). Pada skenario ini, syarat yang harus dipenuhi adalah nilai koordinat deteksi bagian kiri harus berada dalam rentang lebih dari -30 hingga -320, dan koordinat deteksi bagian kanan harus berada dalam rentang kurang dari 320 hingga 30.
- Skenario ketiga adalah menggelindingkan bola dari bawah kiri ke atas kiri. Untuk memenuhi kriteria, nilai koordinat deteksi bagian kiri bawah harus berada dalam rentang lebih dari 30 hingga 240, dan koordinat deteksi bagian kiri atas harus berada dalam rentang kurang dari -30 hingga -240.
- Skenario keempat adalah menggelindingkan bola dari depan kamera ke arah kamera. Pada skenario ini, syarat yang harus dipenuhi adalah nilai koordinat deteksi bagian atas (depan) harus berada dalam rentang kurang dari -30 hingga -240, dan koordinat deteksi bagian bawah harus berada dalam rentang lebih dari 30 hingga 290.
- Skenario kelima adalah menggelindingkan bola dari kanan atas kamera ke arah kiri bawah kamera. Untuk memenuhi kriteria, nilai koordinat deteksi bagian kanan atas harus berada dalam rentang lebih dari 30 hingga 290, dan koordinat deteksi bagian kiri bawah harus berada dalam rentang lebih dari -30 hingga -290.
- Skenario keenam adalah menggelindingkan bola dari kiri atas kamera ke arah kanan bawah kamera. Pada skenario ini, syarat yang harus dipenuhi adalah nilai koordinat deteksi bagian kiri atas harus berada dalam rentang kurang dari -30 hingga -240, dan koordinat deteksi bagian kanan bawah harus berada dalam rentang lebih dari 30 hingga 320.

4. Hasil dan Pembahasan

Berdasarkan rancangan pengujian yang telah ditentukan maka didapati hasil pengujian dari masing-masing skenario. Berikut adalah gambaran dari keenam skenario pengujian yang telah dilakukan. Adapun masing-masing skenario dilakukan sebanyak 30 kali.

Gambar 4 menunjukkan hasil pengujian skenario pertama yaitu penggelindingan bola dari kanan ke kiri (dalam posisi tengah layar). Pada pengujian ini sistem telah mendeteksi keberadaan akhir dari bola adalah di posisi sebelah kiri namun masih di tengah tampilan layar.



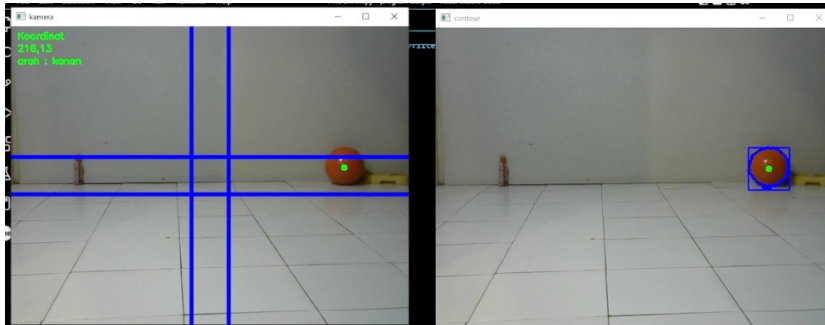
Gambar 4. Hasil deteksi bola pada arah kiri (skenario pertama)

Gambar 5 menunjukkan hasil pengujian skenario kedua yaitu penggelindingan bola dari kiri ke kanan (dalam posisi tengah layar). Pada pengujian ini sistem telah mendeteksi keberadaan akhir dari bola adalah di posisi/bagian sebelah kanan namun masih di tengah tampilan layar.

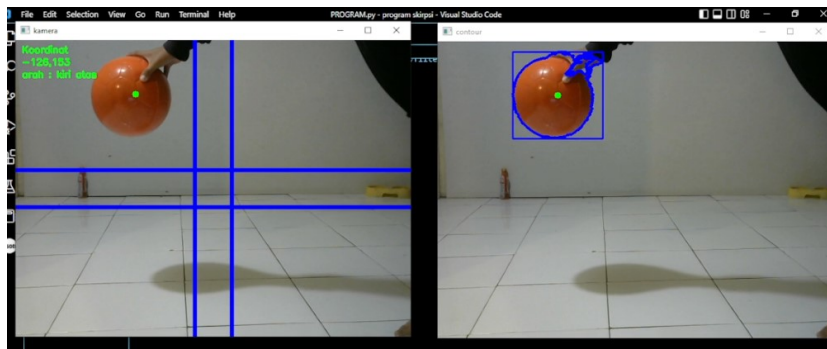
Gambar 6 menunjukkan hasil pengujian skenario ketiga yaitu penggelindingan bola dari bawah kiri ke atas kiri. Pada pengujian ini sistem telah mendeteksi keberadaan akhir dari bola adalah di posisi/bagian sebelah kiri atas.

Gambar 7 menunjukkan hasil pengujian skenario keempat yaitu penggelindingan bola dari depan kamera ke arah kamera. Pada pengujian ini sistem telah mendeteksi keberadaan akhir dari bola adalah di posisi/bagian bawah, namun secara vertikal masih berada di tengah

layar. *Gambar 8* menunjukkan hasil pengujian posisi bola pada bagian atas sebelum digelindingkan ke arah bawah secara vertikal.



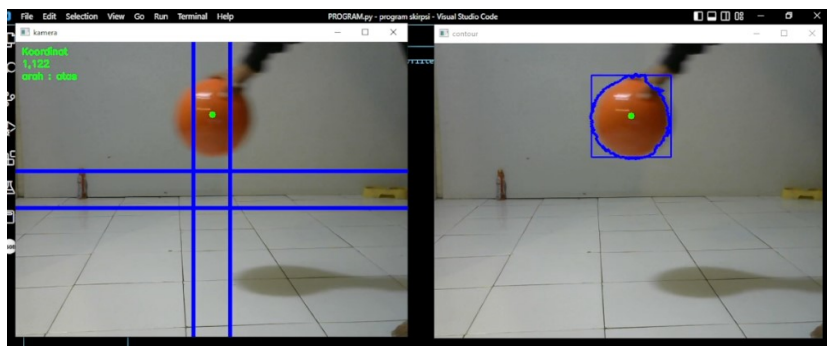
Gambar 5. Hasil deteksi bola pada arah kanan (skenario kedua)



Gambar 6. Hasil deteksi bola pada arah kiri atas (skenario ketiga)



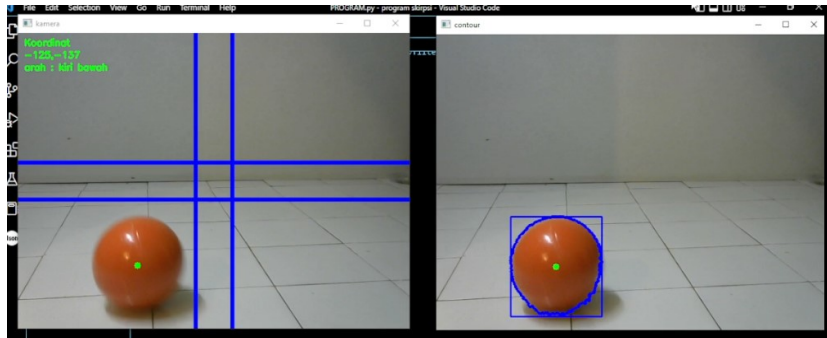
Gambar 7. Hasil deteksi bola pada arah bawah (skenario keempat)



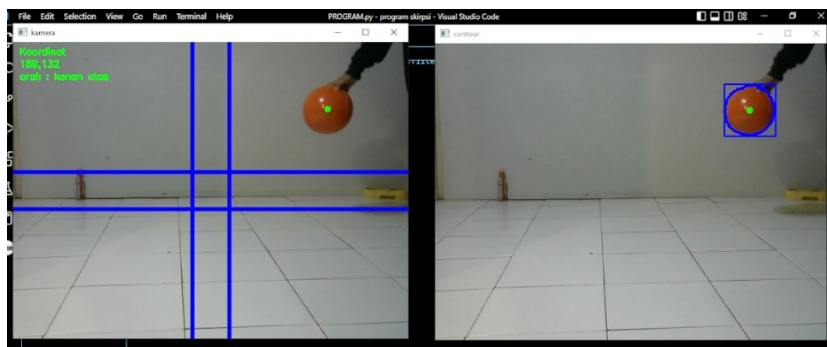
Gambar 8. Hasil deteksi bola pada arah atas

Gambar 9 menunjukkan hasil pengujian skenario kelima yaitu penggelindingan bola dari kanan atas kamera ke arah kiri bawah kamera. Pada pengujian ini sistem telah mendeteksi

keberadaan akhir dari bola adalah di posisi/bagian kiri bawah. Dan *Gambar 10* menunjukkan posisi awal sebelum bola digelindingkan dari posisi kanan atas.

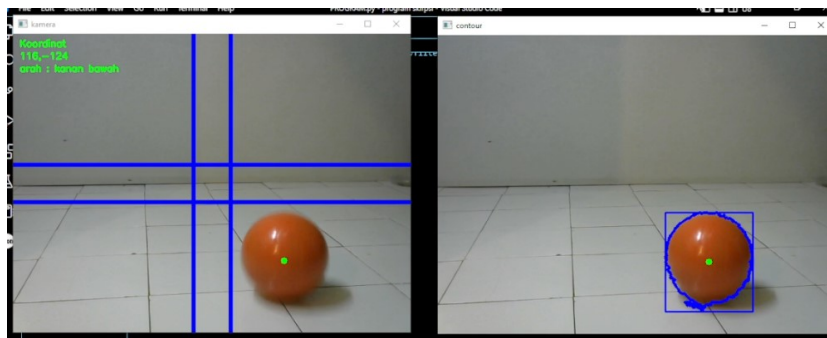


Gambar 9. Hasil deteksi bola pada arah kiri bawah (skenario kelima)



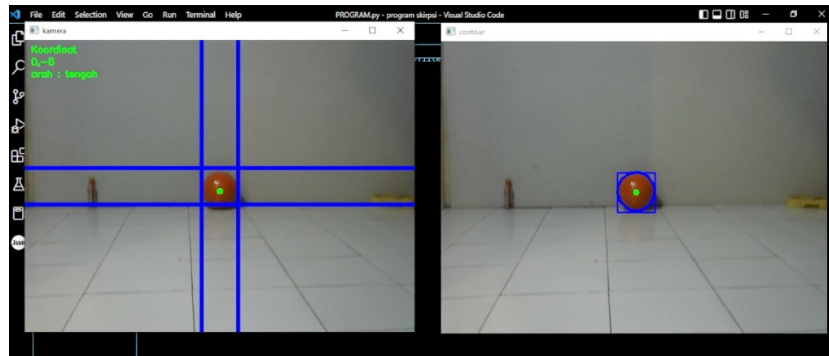
Gambar 10. Hasil deteksi bola pada arah kanan atas

Gambar 11 menunjukkan hasil pengujian skenario keenam yaitu penggelindingan bola dari kiri atas kamera ke arah kanan bawah kamera. Pada pengujian ini sistem telah mendeteksi keberadaan akhir dari bola adalah di posisi/bagian kanan bawah



Gambar 11. Hasil deteksi bola pada arah kanan bawah (skenario kedua)

Gambar 12 menunjukkan hasil pengujian lainnya untuk memastikan bahwa posisi bola pada saat berada tepat di tengah layar baik secara vertikal maupun horizontal terdeteksi oleh sistem. Hasil pengujian keenam skenario dengan pengulangan sebanyak 30 per skenario menunjukkan tingkat keberhasilan 100%.



Gambar 12. Hasil deteksi bola pada arah tengah



Gambar 13. Pengujian sistem dengan latar belakang berwarna mirip dengan bola

Berdasarkan pengujian terhadap keenam skenario tersebut maka didapati bahwa Dalam eksperimen ini, kamera bola digunakan untuk mendeteksi pergerakan bola yang digelindingkan pada jarak antara 5 cm hingga 100 cm. Hasilnya menunjukkan bahwa deteksi berada pada tingkat akurasi yang tinggi, baik saat bola digelindingkan ke arah depan, belakang, kiri, maupun kanan. Bahkan ketika bola digelindingkan ke arah diagonal seperti depan kiri, depan kanan, belakang kiri, dan belakang kanan, deteksi tetap akurat. Jarak 5 cm sampai dengan 100 ini memang tidak sebaik jika dibandingkan dengan penelitian [20] yang mampu mendeteksi bola hingga jarak 600 cm. Namun keduanya menggunakan metode dan atau algoritme yang berbeda. Kemudian penelitian [26], jarak deteksi bola yang dapat diakomodasi adalah 100 cm hingga 400 cm, dan tentu saja juga dengan metode dan algoritme yang berbeda.

5. Simpulan

Berdasarkan hasil pengujian yang telah dilakukan, maka intensitas cahaya tidak terlalu mempengaruhi proses pendeteksian bola. Hanya saja metode *optical flow* tetap rentan terpengaruh jika warna latar belakang objek yang serupa dengan warna objek itu sendiri (Gambar 13). dan latar memiliki kesamaan. Objek pada pendeteksian *optical flow* harus diberi tanda sebelum dapat dilacak. Penggunaan metode *optical flow* menggunakan kamera logitech C270 dapat meningkatkan akurasi dan kecepatan dalam menentukan titik akhir bola saat bola melambung menuju gawang. Penelitian ini menunjukkan bahwa penerapan metode *optical flow* pada kamera C270 menghasilkan tingkat akurasi yang tinggi dalam menentukan arah gerak bola, sehingga robot penjaga gawang dapat merespons bola dengan cepat dan efisien. Disarankan untuk mengimplementasikan metode ini langsung pada robot KRSBI beroda kiper sebagai sensor pelacakan objek. Penelitian selanjutnya dapat meningkatkan mekanisme pembacaan objek dengan menerapkan sistem kamera Omnidirectional. Pertimbangkan penggunaan kamera dengan kualitas lebih tinggi daripada jenis webcam untuk meningkatkan tingkat deteksi arah bola secara keseluruhan.

Daftar Referensi

- [1] Kementerian Pendidikan dan Kebudayaan, "Petunjuk Pelaksanaan Kontes Robot Indonesia (Kri)," *Pus. Prestasi Nas. Kementeri. Pendidik. dan Kebud.*, pp. 6–8, 2020.
- [2] B. W. Sanjaya, H. Priyatman, J. Teknik, E. Fakultas, T. Universitas, and K. Capek, "Implementasi Logika Fuzzy Pada Robot Beroda Penghindar Halangan Berbasis Arduino Uno R3 dan pengembangan robot diberbagai bidang , robot juga sudah digunakan di," *Jurnal Electr. Eng. Energy, Inf. Technol.*, vol. 1, no. 1, pp. 1–9, 2017.
- [3] U. B. Gohatre, "Estimation of velocity and distance measurement for projectile trajectory prediction of 2D image and 3D graph in real time system," *2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput.*, pp. 2543–2546, 2017.
- [4] 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, doi: 10.30871/ji.v13i2.3133.
- [5] M. I. Moha *et al.*, "Implementasi Kamera 360 Derajat Untuk Mendeteksi Objek Pada Robot Sepak Bola Beroda," *J. Tek. Inform.*, vol. 14, no. 3, pp. 321–328, 2019, doi: 10.35793/jti.14.3.2019.27123.
- [6] A. Abid, M. T. Khan, and C. W. De Silva, "Fault Detection in Mobile Robots using Sensor Fusion," in *The 10th International Conference on Computer Science & Education (ICCSE 2015)*, 2015, no. Iccse, pp. 8–13.
- [7] H. Ponce, J. Brieva, and E. Moya-Albor, "Distance Estimation Using a Bio-Inspired Optical Flow Strategy Applied to Neuro-Robotics," in *Proceedings of the International Joint Conference on Neural Networks*, 2018, vol. 2018-July, pp. 1–7, doi: 10.1109/IJCNN.2018.8489597.
- [8] Muliady and G. Arisandy, "Implementasi Sistem Gerak Holonomic Pada Robot Krsbi Beroda 2017 Implementation Of Holonomic Motion In Indonesian Soccer Wheeled Robot Contest 2017 Pada Kontes Robot Indonesia 2017 divisi Kontes Robot Sepak Bola Indonesia," *J. Tek. dan Ilmu Komput.*, vol. 7, no. 25, pp. 9–25, 2017.
- [9] A. Aguilar-González, M. Arias-Estrada, and F. Berry, "Depth from a motion algorithm and a hardware architecture for smart cameras," *Sensors*, vol. 19, no. 1, pp. 1–20, 2019, doi: 10.3390/s19010053.
- [10] A. Solichin, A. Harjoko, and A. E. Putra, "Grid-based Histogram of Oriented Optical Flow for analyzing movements on video data," in *Proceedings of 2015 International Conference on Data and Software Engineering, ICODSE 2015*, 2016, no. NOVEMBER, pp. 114–119, doi: 10.1109/ICODSE.2015.7436982.
- [11] C. M. Adi, "Implementasi Pergerakan Robot Penjaga Gawang Krsbi (Beroda) Dengan Metode Fuzzy Pid," Universitas Jember, 2020.
- [12] C. Wang and S. Zhao, "Research for Control System of Soccer Robot Based on DSP," vol. 2, pp. 91–95.
- [13] F. Abdessemed and K. Benmahammed, "A Two-Layer Robot Controller Design Using Evolutionary Algorithms," *J. Intell. Robot. Syst.*, vol. 30, pp. 73–94, 2001.
- [14] F. Control, "Navigation of an Autonomous Wheeled Robot in Unknown Environments Based on Evolutionary Fuzzy Control," *inventions*, vol. 3, no. 3, pp. 1–14, 2018, doi: 10.3390/inventions3010003.
- [15] C. M. Adi, "Implementasi Pergerakan Robot Penjaga Gawang KRBSI (Beroda) dengan Metode Fuzzy Pid," Universitas Jember, 2020.
- [16] F. X. W. Artanto, "Pengontrol Robot Sepak Bola Beroda Wheeled Soccer Robot Controller By," 2018.
- [17] B. Rahmani, A. Harjoko, T. K. Priyambodo, and H. Aprilianto, "Research of smart real-time robot navigation system," *AIP Conf. Proc.*, vol. 1707, 2016, doi: 10.1063/1.4940854.
- [18] B. Rahmani, H. Aprilianto, H. Ismanto, and H. Hamdani, "Distance estimation based on color-block: A simple big-O analysis," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 4, pp. 2169–2175, 2017, doi: 10.11591/ijece.v7i4.pp2169-2175.
- [19] H. Mandala and E. Rudiawan, "Sistem Deteksi Bola Berdasarkan Warna Bola Dan Background Warna Lapangan Pada Robot Bareleng FC," *Semin. Nas. Apl. Teknol. Inf.*, vol. 2016, no. snati, pp. 14–20, 2016.
- [20] A. W. Pradana and D. Irmawati, "Pendeteksi Warna dan Bentuk Bola Pada Robot Penjaga Gawang Menggunakan EmguCV," *Elinvo (Electronics, Informatics, Vocat. Educ.)*, vol. 5, no. 1, pp. 21–31, 2020, doi: 10.21831/elinvo.v5i1.20794.

- [21] Y. Li and Z. Wang, "RGB Line Pattern-Based Stereo Vision Matching for Single-Shot 3-D Measurement," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, doi: 10.1109/TIM.2020.3041086.
- [22] M. L. Bukhori, A. Triwiyatno, and R. R. Isnanto, "The Design of Object Tracking System in Robot Vision Using Circle Hough Transform and CAMSHIFT Methods," in *ICCSET 2018*, 2018, pp. 149–156, doi: 10.4108/eai.24-10-2018.2280634.
- [23] A. W. Pradana and D. Irmawati, "Pendeteksi Bola Pada Robot Penjaga Gawang Menggunakan Metode Hough Circle," *Elinvo (Electronics, Informatics, Vocat. Educ.*, vol. 5, no. 1, pp. 21–31, 2020, doi: 10.21831/elinvo.v5i1.20794.
- [24] S. N. Larasati, "Rancang Bangun Sistem Deteksi Posisi Bola pada Bidang Datar dengan Metode Template Matching Menggunakan Labview," Universitas Telkom, 2016.
- [25] K. Watanabe, T. Kageyu, S. Maeyama, and I. Nagai, "An Obstacle Avoidance Method by Combing Image-based Visual Servoing and Optical Flow," in *SICE Annual Conference 2015*, 2015, pp. 2–7.
- [26] D. Daniyah, B. Arifin, and I. M. I. Subroto, "Prediksi Arah Datang Bola Menggunakan Kalman Filter Pada Robot Kiper Sepakbola," *Transmisi*, vol. 22, no. 2, pp. 56–61, 2020, doi: 10.14710/transmisi.22.2.56-61.