PERANCANGAN *OBJECT TRACKING ROBOT* BERBASIS *IMAGE PROCESSING* MENGGUNAKAN RASPBERRY PI

Laksono Budi Prianggodo[1)].RatnasariNur Rohmah[2)]
[1]Teknik Elektro, UniversitasMuhammadiyah Surakarta
email: laksonobudi94@gmail.com
[2]Teknik Elektro, UniversitasMuhammadiyah Surakarta
email: rnr217@ums.ac.id

Abstract

This paper presents the research results on using the object tracking techniques for object recognition and robot's motion control based on object movement. In this study, object tracking used by the robot to recognize and follow the movement of an object. Pattern recognition and robot motion in object tracking controlare done in a real time. Robot developed in this study is a wheeled robot that has the ability to follow the motion of the certain color ball. Pattern recognition algorithms in this study includes: the color space conversion from RGB to HSV, color detection using color filtering, and shape detection using the edge detection technique and the circle hough transform. Image processing and robot"s motion control is done by using a mini computer Raspberry Pi 2 Model B Raspberry Pi and camera. This study use python programming language, which has been embedded in OpenCV library. The performance tests in this research was conducted by analyzing the effect of the image resolution on the camera motion, the robot maximum visibility, the color recognition process, the shape recognition process, and the minimum light intensity requirement. The results show that the best image resolution for the robot to perform ball tracking is 320x240 pixels, the robot maximum visibility is 113 cm, and the minimum of light intensity for robot to recognize the ball is 21.0 lux.

Keywords: *object tracking, patternrecognition, real time, robot"s motion.*

1. PENDAHULUAN

Robot adalah mesin yang dapat melakukan berbagai pekerjaan yang menggantikan satu jenis atau lebih pekerjaan yang biasa dikerjakan oleh manusia.Dalam melakukan pekerjaannya, robot mengikuti suatu pola gerakan yang ditentukan oleh pengguna robot. Sistem kontrol robot merupakan sistem yang digunakan oleh manusia untuk menentukan pola gerak robot.

Terdapat tiga bagian utama yang membangun suatu sistem kontrol, yaitu *input*(masukan), *processor*(pengolah), dan *output*(keluaran). *Input* adalah sekumpulan informasiyang ditangkap oleh robot yang diperoleh dari sensor yang berperan sebagai indra bagi robot. Informasi yang didapat pada bagian *input*tidak dapat langsungdigunakan melainkan harus diproses terlebih dahulu. Pengolahan sinyal input ini dilakukan oleh *processor*yang merupakan otak dari robot. Hasil pengolahan sinyal ini mengendalikanbagian output robot untuk melakukan kerja yang diinginkan oleh perancang robot. *Output*umumnya merupakan bagian mekanik pada robot yang melakukan gerakan tertentu. Dalam menirukan gerakan manusia robot menggunakan piranti-piranti diantaranya dengan *microphone*, *loudspeaker*,*actuator*, kamera, sensor, untuk mendengar, berbicara, bergerak, melihat, dan merasakan. Sedangkan fungsi berfikir pada manusia dilakukan oleh piranti robot yang disebutsebagai*processor*.

Supaya robot dapat melihat objek atau suasana sekitarnya dibutuhkan sebuah indra penglihatan yang dapat terintregasi dengan sistem. Indra penglihatan untuk robot dapat menggunakan kamera. Kamera merupakan perangkat keras yang berfungsi menangkap citra dan mengubahnya ke dalam bentuk citra digital yang dapat dibaca dan diproses oleh komputer. Citra digital ini merupakan sinyal masukan yang akan diproses pada komputer

dapat menghasilkan informasi yang berguna untuk keperluan tertentu pada sistem. Pengolahan citra dengan komputer ini dikenal sebagai pengolahancitradigital.

2. KAJIAN LITERATUR DAN PENGEMBANGAN HIPOTESIS

Pengolahan citra digitalbisa dibagi dalam dua kategori, yaitu pengolahan citra dengan keluaran berupa citra, dan pengolahan citra yang keluarannya berupa suatu keputusan hasilanalisis citra.Kategoripertamadigunakandalam proses perbaikancitraataumanipulasicitrauntukkeperluantertentu.Kategorikeduaseringdisebutsebagaianalisiscitra, yang banyakdigunakandalambeberapabidang, diantaranyadalambidang*artificial intelligence* untuk kecerdasan komputasi (*computational intelligence*), *automatic contro robotics*untuk *computer vision*, *neurobiology* untuk *biological vision* dan *machine learning* untuk *cognitive vision*.

Computer vision adalah salah satu penerapan dari pengolahan citra dalam kategori analisis citra. Tahapanpertama pada pengolahan citra untukkeperluan*computer vision*adalah proses akuisisi citra. Tahapan selanjutnya adalah tahapan pengolahan citra, yang bisa meliputi ekstraksi ciri citra yang dilanjukan dengan analisis citra berdasar ciri citra. Hasil analisis akan merupakan suatu keputusan yang dalam hal *computer vision* untuk robot adalah untuk menentukan gerakan robot. Salah satu kunci untuk melakukan analisis citra adalah kita memerlukan sebuah perangkat komputasi yang berkemampuan tinggi. Karena keperluan ini, maka tidak cukup jika hanya dilakukan dengan *microcontroller* atau *microprocessor*.

Penelitian ini merancang sebuah sistem robotika yang menangkap objek melalui sensor kamera, dapat melakukan proses pengenalan objek menurut bentuk dan warna, serta dapat mengikuti pergerakannya*(object tracking)*secara *real-time.* Peneliti akan melakukan *tracking* terhadap objek bola dengan warna tertentu. Untuk dapat melakukan segala hal tersebut diperlukan sebuah sistem *embeded* yang dapat berperan sebagai komputer untuk melakukan pengolahan citra pada prosesor robot, serta *microcontroller*sebagai pengendali untuk rangkaian elektronik *actuator* pada bagian keluaran robot.

Pada proses akuisisi citra,peletakan kamera pada sebuah sistem sangat berpengaruh pada hasil yang diperoleh. Terdapat 2 macam peletakan kamera pada sistem yaitu *eye in hand* dan *stand alone*. Kragic dan Christensen (2011) menyatakan bahwa peletakanunit kamera untuk sebuah sistem dapat secara *eye in hand*atau *stand alone*. *Eye in hand* merupakan metode peletakkan kamera pada *end reflector*(bagian ujung) dari robot dan *stand alone* merupakan metode peletakkan kamera terpisah dari bagian tubuh robot. Dengan menggunakanmetode *eye in hand* sistem akan langsung melihat objek yang akan diikuti secara langsung sedangkan metode *stand alone* sistem akan dapat melihat posisi objek serta robot dari jarak jauh.

Marchand (2007) menyatakan bahwa peletakkan kamera sangat mempengaruhi hasilcitra yang diperoleh, pencahayaan juga dapat mempengaruhi citra yang diperoleh dan akan diproses. Pada penelitiantersebut, terdapat berbagai cara untuk memperoleh citrayang bagus sepertimemaksimalkan intensitas cahayadan meningkatkan kontras antara objek dengan latar, dan teknik pengambilanyang tepat.

Pada penelitian ini obyek yang akan dikenali oleh robot adalah obyek berbentuk bola.Obyek bola pada penelitian ini dikenali sebagai bentuk lingkaran. Ikwuagwu(2010) menyatakan bahwa sebuahfungsi pada Open CV yang disebut dengantransformasi*Hough circles* biasa digunakan untuk mendeteksi seluruh bentuk lingkaran pada sebuah citra2D.

Selain pengenalan bentuk bola, pada penelitian ini robot dirancang untuk dapat mengenali warna. Ekstraksi ciri warna oleh robot dilakukan dengan*colour filtering.*Pada penelitian sebelumnya terdapat beberapa peneliti yang pernah melakukan percobaan mengenai perbandingan dari penggunaan

color filtering pada ruang warna HSV dan RGB. Beberapa penelitian memperlihatkanhasil yang didapat pada *color filering* pada ruang warna HSV lebih bagus dibandingkan dengan RGB.

Sural, Qiandan Pramanik(2002) menganalisis sifat dari ruang warna HSV dengan penekanan pada variasi persepsi visual dalam nilai *hue, saturation* dan *value* dari suatu piksel *image.*Penelitian ini mengekstraksi fitur piksel dengan memilih *hue* atau *value* sebagai nilai dominan berdasarkan nilai saturasi piksel. Segmentasi ruang warna menggunakan metode ini memberikan identifikasi yang lebih baik terhadap objek dalam sebuah *image* dibandingkan dengan yang dihasilkan menggunakan ruang warna RGB. Sedangkan Yustinus(2012) menyatakan bahwa metode *color filtering* RGB untuk segmentasi warna masih belum dapat memahami objek secara akurat disebabkan oleh faktor dari intensitascahaya yang belum diperhitungkan.

3. METODE PENELITIAN

3.1 Alat dan Bahan

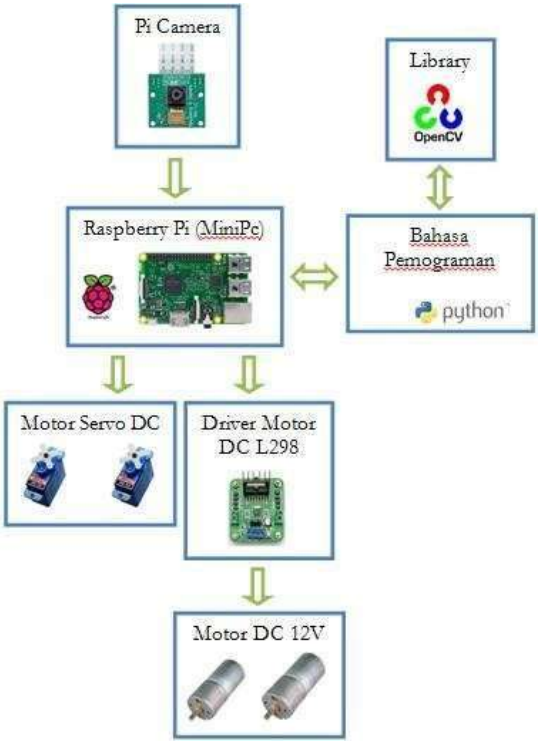
Peralatan dan komponen elektronika yang akan digunakan dalam perancangan ini meliputi :

a. Raspberry Pi 2 Model B.
b. Modul Kamera Raspberry Pi.
c. Baterai Lippo 3Cell2200mA.
d. Konektor HDMI to VGA.
e. Kabel LAN.
f. Motor DC 12Volt.
g. Motor Servo Micro.
h. IC L298N,Resistor, Dioda, LED.
i. Kabel,PCB, Solder, Timah,Atraktor.
j. Obeng, AVO Meter, Tang Potong.
k. Aklirik.
l. Roda Tank Mainan.
m. Komputer/laptop.

3.2 Perancangan

Dalam perancangan *object tracking robot* berbasis *image processing* menggunakan Raspberry Pi terdapat 2 tahap perancangan yaitu *hardware* dan *software*. Pada robot ini terpasang Pi *camera* yang bekerja melakukan pengambilan citrasecara terus menerus dengan selang waktu tertentu *(streaming)*.

Pada penelitian ini, Raspberry Pi menjadi sebuah unit komputer serta *microcontroller* yang dapat melakukan proses pengolahan citra sekaligus dapat mengendalikan rangkaian elektronik lain untuk melakukan pergerakan pada robot.Citra *streaming* dari kamera akan dianalisa oleh Raspberry Pi dengan library Open CV menggunakan bahasa pemograman python. Berdasarkan hasil analisis citraini, robot akan bergerak mengikuti pergerakan posisi bola dengan menggunakan motor DC 12 Volt. Motor servo digunakan agar kamera bisa leluasa mengikuti perpindahan bola secara *real-time*.

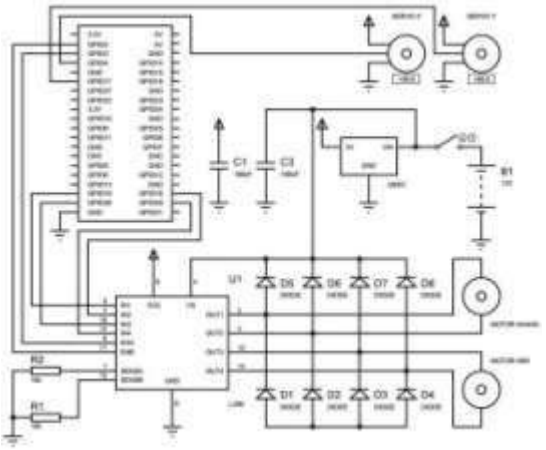


Gambar 1. Diagram blok sistem.

3.2.1 Perancangan *Hardware*

Perancangan *hardware* dilakukan dengan melakukan desain mekanik serta elektronik robot. Skema rangkaian dapat dilihat pada Gambar 2. GPIO pin pada Raspberry Pi mengatur secara langsung pergerakan 2 motor servo dengan menyalurkan pulsa PWM *(Pulse Width Modulation).*Pengendalian arah

dan kecepatan motor DC oleh Raspberry PI harus dilakukan melalui perantara IC L298N, karena Raspberry Pi bekerja pada tegangan 0-3.3V sedangkan motor bekerja pada tegangan 12V.Sumber daya baterai 12 V untuk motor DC, dan UBEC 5V 3A untuk memberikan catu kepada Raspberry Pi, motor servo dan IC l298N.

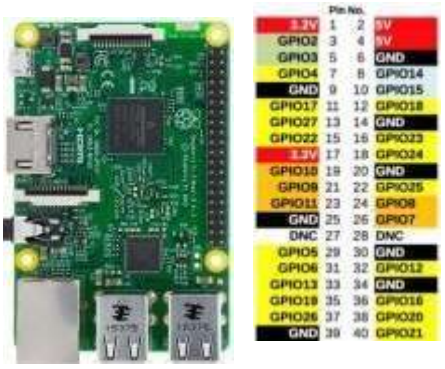


Gambar 2. Skema rangkaian.

Pemilihan minikomputer Raspberry Pi pada penelitian ini adalah karena fitur yang disediakan sesuai dengan kebutuhan untuk pembuatan sistem robot ini. Desain Raspberry Pi didasarkan pada SoC (*System-on-a-Chip*) Broadcom BCM2835.Padakonfigurasi Raspberry Pi(Pi 2 model B)telah tertanam*prosesor* QuadCore ARM dengan kecepatan *clock* sebesar 900 MHz, VideoCore IV GPU, dan 1 Gigabyte RAM. Penyimpanan data didisain tidak untuk menggunakan *hard disk* atau *solid-state drive*, melainkan mengandalkan kartu SD (SD *memory card*) untuk *booting* dan penyimpanan jangkapanjang.

Sistem operasi utama Raspberry Pi menggunakan Debian GNU/Linux dan bahasa pemrograman python. Komputer ini dilengkapi dengan 4 USB versi 2.0 Port yang dapat dihubungkan dengan perangkat yang menggunakan port USB apapun. Komputer dilengkapi juga denganethernet port untuk sambungan LAN (Local Area Network), HDMI port untuk disambungkan dengan perangkat layar LCD/LED dengan kualitas HD (*High Definition*), *connector* 3,5mm untuk disambungkan ke perangkat *speaker*atau *headset*, dan USB *micro* untuk *power supply*. Selain itu disediakan juga 2 Port khusus untuk kamera dan LCD Raspberry Pi.Komputer ini menyediakan 40 pin GPIO (*General Purpose Input Output*) agar Raspberry Pi dapat digunakan sebagai pengendali rangkaian elektronik.

Kamera yang digunakan untuk melakukan proses akusisi citrapada penelitian iniadalah Pi *Camera* yang merupakan kamera khusus yang didesain untuk *minicomputer* Raspberry Pi. Dengan ukuran kecil, modul kamera Raspberry Pi dapat digunakan untuk mengambil citradengan kualitas *high definition* memiliki kualitas 5 MP mendukung resolusi video 1080p 30fps,720p 60 fps dan VGA90,dapat bekerja pada semua model Raspberry Pi yang terhubung pada port CSI.



Gambar 3. Detil konfigurasi pin GPIO Raspberry Pi.

Gambar 4. Bentuk fisik modul kamera Raspberry Pi.

Sebagai penggerak untuk robot agar dapat berjalan digunakan sepasang motor DC 12 Volt.Keuntungan utama motor DC adalah dalam hal pengendalian kecepatan serta arah putaran motor DC tersebut sangat mudah. Motor ini dapat dikendalikan dengan

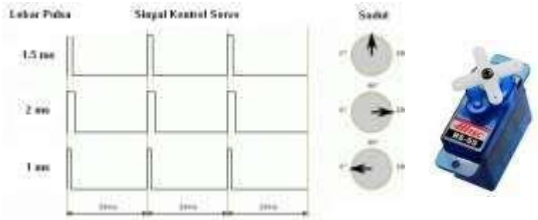
mengatur tegangan sumber atau arus yang mengalir padamotor DC. Dengan meningkatkan tegangansumber motor DC akan menambah kecepatan putaran sedangkan menambah arus yang mengalir pada motor DC akan menurunkan kecepatan.

Gambar 5. Bentuk fisik motor DC.

Supaya Raspberry Pi dapat melakukan *controlling* terhadap motor DC digunakan *driver* motor. Kelebihan dari *driver* motor ini adalah cukup presisi dalam mengendalikan motor DC dan mudah dikendalikan. Untuk mengendalikan driver L298N ini dibutuhkan 6 pin. Pada prinsipnya rangkaian driver motor L298N inihanya mengatur tegangan dan arus sehingga kecepatan dan arah motor dapat disusaikan dengankeinginan.

Gambar 6. Bentuk fisik IC L298N

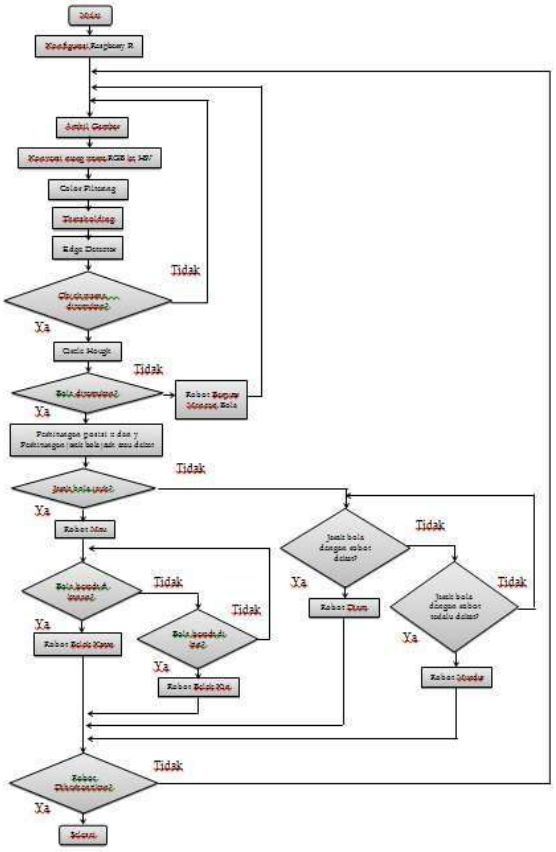
Agar kamera dapat melakukan *tracking* terhadap objek dibuat sistem mekanik yang dapat membuat kamera leluasa bergerak dengan sumbu sejajar x dan y, untuk melakukan hal tersebut *actuator* yang cocok ialah motor servo. Untuk mengendalikan putaran motor servo dilakukan dengan mengirimkan pulsa kontrol dengan frekuensi 50 hz dengan periode 20ms dan *duty cycle* yang berbeda.



Gambar 7. Sinyal pulsa motor servo.

Perancangan *Software*

Pada perancangan software dilakukan dengan pembuatan program pada Python Idle yang merupakan *compiler* bawaaan pada Raspberry Pi. Dalam merancang suatu program untuk menjalankan fungsi hardware pada Raspberry Pi, perlu dilakukan beberapa pengaturan terlebih dahulu.Pengaturan yang diperlukan pada pembuatan robot ini diantaranyaadalah pengaturan pin GPIO mana saja yang digunakandanpengaturan resolusi citradari kamera.*Default* resolusi kamera yang menjadi pengaturan awal pada sistem ini adalah 320x240 piksel. Algoritma pemograman yang dibuat untuk sistem robot dapat dilihat pada Gambar 8.



Gambar 8. *Flowchart* program.

Pengenalan warna dilakukan dengan konversi dari RGB ke HSV. Selanjutnya penetapan warna obyek yang dikenali adalah

dengan membandingkan ciri warna obyek tersebut dengan ciri warna yang ada pada basis data. Pada penelitian ini ciri yang digunakan adalah nilai Hue min, Hue max, Sat min, Sat max, Value min dan Value max, dari citra HSV.

Pengenalan bentuk dilakukan dengan proses pengubahan citra menjadi citra biner dengan *thresholding*, deteksi tepi atas citra biner, dan deteksi bentuk lingkaran dengan transformasi *Hough circle*. Program untuk proses ini diperlihatkan pada Gambar 9.

```
.
.
# Convert RGB to HSV
image_HSV = cv2.cvtColor(gambar,cv2.COLOR_BGR2HSV)
Hue,Sat,Val = cv2.split(image_HSV)
# Color Filtering and Thresholding
Hthresh = cv2.inRange(np.array(Hue),np.array(Hmin),
    np.array(Hmax))
Sthresh = cv2.inRange(np.array(Sat),np.array(Smin),
    np.array(Smax))
Vthresh = cv2.inRange(np.array(Val),np.array(Vmin),
    np.array(Vmax))
tracking = cv2.bitwise_and(Hthresh,
    cv2.bitwise_and(Sthresh,Vthresh))
.
.
# Edge Detection and Circle Detection
circles = cv2.HoughCircles(closing,
    method=cv2.HOUGH_GRADIENT,dp=1,minDist=400,
    param1=50,param2=13,minRadius=3,maxRadius=240)
if circles is not None:
        for i in circles[0,:]:
                Cx = int(round(i[0]))
                Cy = int(round(i[1]))
                radius = int(round(i[2]))
.
```

Gambar 9.*Script* program untukpengolahancitra.

Beberapa parameter yang terdapatpadafungsitransformasi*Houghcircle*y aitudp, minDist, param1, param2, minRadiusdanmaxRadius. Dpadalahrasio invers dariresolusiakumulatoruntukresolusigambar. Misalnya, jikadp = 1, akumulatormemilikiresolusi yang samadengancitra input. Jikadp = 2, akumulatormemilikisetengahlebarbesardantin ggi. MinDistadalahjarak minimum antarapusat-pusatlingkaran yang terdeteksi.Jika parameter terlalukecil, beberapalingkaran yang terdeteksimungkinpalsu.Jikaterlalubesar, beberapalingkaranmungkinterlewatkan. Param1 adalahnilaiambangatas*threshold*untuk*internal Canny edge detector*. Param2 adalahnilai*threshold*untukdeteksipusatlingkar an.MinRadiusadalahnilai minimaldarijari-jarilingkaran yang dideteksi.MaxRadiusadalahnilaimaksimaldari jari-jarilingkaran yang dideteksi.

Hasildari proses transformasi*Hough circle*adalahkoordinatpusatlingkarandan radius lingkaran. Parameter tersebutdijadikansebagaimasukandarisebuahp erhitunganuntukpengendaliansistemgerak robot.Perhitungan yang dilakukandidapatdarihasilpercobaan. Output dariperhitunganakanmenghasilkannilai PWM (*Pulse Width Modulation*)untukderajat motor servo dankecepatanmotor DCsehinggaakanmenyebabkanperubahangera kdari*body* robot. Program untuk proses ini diperlihatkan pada Gambar 10 dan 11.

```
.
.
# Tracking Object
if Cx >= (width/2):
        Error_X = ((Cx-(width/2))/(width/2.0))*100
        Move_X = (10*Error_X)/100.0
        Derajat_X = Derajat_X + Move_X
        if Derajat_X >= 180:
                Derajat_X = 180
if Cx > 0 and Cx < (width/2):
        Error_X = (((width/2)-Cx)/(width/2.0))*100
        Move_X = (10*Error_X)/100.0
        Derajat_X = Derajat_X - Move_X
        if Derajat_X <= 0:
                Derajat_X = 0
if Cy >= (height/2):
        Error_Y = ((Cy-(height/2))/(height/2.0))*100
        Move_Y = (10*Error_Y)/100.0
        Derajat_Y = Derajat_Y + Move_Y
        if Derajat_Y >= 180:
                Derajat_Y = 180
if Cy > 0 and Cy < (height/2):
        Error_Y = (((height/2)-Cy)/(height/2.0))*100
        Move_Y = (10*Error_Y)/100.0
        Derajat_Y = Derajat_Y - Move_Y
        if Derajat_Y <= 0:
                Derajat_Y = 0
# Drive Servo
Servo_X = Derajat_X /18.0 + 2.5
Servo_Y = Derajat_Y /18.0 + 2.5
Servo_X = round (Servo_X,2)
Servo_Y = round (Servo_Y,2)
Right_or_Left.ChangeDutyCycle(Servo_X)
Up_or_Down.ChangeDutyCycle(Servo_Y)
.
.
```

Gambar 10.*Script* program untukpengendaliangerakmekanik*pan tilt*.

```
.
.
# Drive Motor
if radius > 0 and radius <= 45:
    Speed_Max = int(80-(radius*2))
    if Cx >(height/2):
        Speed_Right = Speed_Max - (Derajat_X - 90)
        Speed_Left = Speed_Max + (Derajat_X - 90)
        Speed_Left = int(Speed_Left/2)
        Speed_Right = int(Speed_Right/2)
    if Cx <(height/2):
        Speed_Right = Speed_Max - (90 - Derajat_X)
        Speed_Left = Speed_Max + (90 - Derajat_X)
        Speed_Left = int(Speed_Left/2)
        Speed_Right = int(Speed_Right/2)
    if Speed_Right <= 10:
        Speed_Right = 0
    if Speed_Right >= 70:
        Speed_Right = 70
    if Speed_Left <= 10:
        Speed_Left = 0
    if Speed_Left >= 70:
        Speed_Left = 70
    Speed_Right = round (Speed_Right,0)
    Speed_Left = round (Speed_Left,0)
    motor_forward(Speed_Right,Speed_Left)
if radius >= 48:
    Speed_Mundur = radius
    if Speed_Mundur >= 49:
        Speed_Mundur = 49
    if Speed_Mundur <= 10:
        Speed_Mundur = 0
    motor_backward(Speed_Mundur)
if radius > 45 and radius < 48:
    motor_stop()
.
.
```

Gambar 11.*Script* program untukpengendaliangerak motor DC.

Setelah objek dikenali, maka robot akan terus-menerus mengukur jarak bola, melakukan pergerakan mendekati atau menjauhi bola sesuai gerakan bola, dan akan berhenti pada jarak yang telah ditentukan.Posisi bola menjauhdiperlihatkandenganadanyaperubahanukuran diameter mengecildansebaliknya.Robot akanbergerakmendekatsaat bola menjauh dan pada saat terlalu dekat akan mundur. Selain memperhitungkan jarak, robot juga memperhitungkan posisi bola. Jika bola bergerak ke arah kanan dari koordinat awal. Makagerakan maju robot akan cenderung kekanan, dansebaliknya.

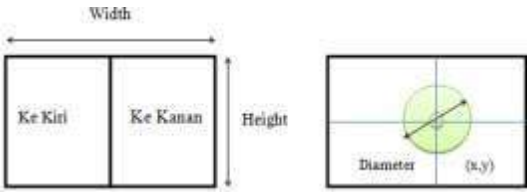
Segala keperluan proses pengolahan citra dapat tersedia dengan penggunaan library Open CV.OpenCV (*Open Computer Vision*) merupakan sebuah API (*Application Programming Interface*) *library* yang sudah sangat familiar pada pengolahan citra *computer vision.* OpenCV adalah library *open source* untuk *computer vision* yang didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk *image*/*video*.

4. HASIL DAN PEMBAHASAN

Sistem Pergerakan Robot

Pada hasil percobaan robot bergerak mengikuti perubahan posisi bola. Ketika posisi bola berada dikanan dari *center*resolusi citramaka robot akan bergerak cenderung kekanan dan ketika posisi bola berada di kiri dari *center* resolusi citramaka robot akan bergerak ke kiri. Jarak robot dengan bola didapat dari diameter bola. Posisi robot akan selalu menjaga jaraknya dengan posisi bola. Lebih jelasnya dapat dilihat pada Gambar 13. Pada percobaan ini menggunakan bola dengan diameter 6.5 cm.

Gambar 12. Bentuk fisik robot.



Gambar 13. Koordinat serta diameter gambar bola untuk sistem gerak robot.

Pengaruh Resolusi CitraDenganSistem Kerja Robot

Dengan mengatur resolusi dari citrayang ditangkap melalui kamera mempengaruhi kualitas citrayang ditangkap, jarak pandang robot terhadap objek serta kecepatan dalam

melakukan pergerakan untuk mengikuti perpindahan posisi objek karena memerlukan banyak waktu untuk proses pengolahan citra. Jarak robot dengan objek bola dihitung dari posisi kamera dengan pusat bola, sedangkan waktu *tracking* objek diambil saat robot bergerak dari posisi awal hingga posisi akhir hingga robot berhenti bergerak karena telah mengenali bola.

Gambar 14. Pengukuran jarak pandang maksimum.

Tabel 1. Efek perubahan resolusi citrakamera.

Resolusi citra	**Jarak Maksimal**	**Waktu *Tracking* Objek**
240x160	102 cm	4 detik
320x240	113 cm	5 detik
480x320	135 cm	6 detik
640x480	172 cm	error
720x560	180 cm	error
800x600	190 cm	error

Pada resolusi citra640x480 hingga 800x600 terjadi error yang membuat *streaming*citrapada kamera terjadi *lag* serta pergerakan motor servo dalam mencari objek memiliki *delay* yang lama sehingga menyebabkan robot terlalu lama dalam mengenali objek bola.

Pemilihan Objek Berdasarkan Warna

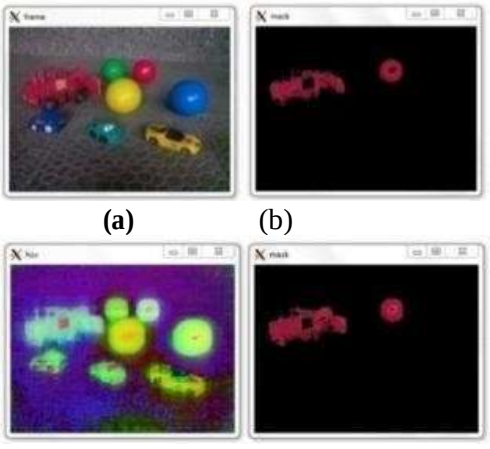
Citradengan ruang warna RGB dan HSV sangat berbeda, citradengan ruang warna RGB didapat dari kamera, pengubahan ruang warna menjadi HSV dilakukan menggunakan library OpenCV yang berguna mempermudah proses *color filtering.*Para peneliti sebelumnya menyatakan bahwa penggunakan ruang warna HSV untuk melakukan proses *color filtering*lebih bagus. Perbedaan hasil *color filtering*RGB dan HSV dapat dilihat pada Gambar 15 hingga 18.

Tabel 2. RGB*color range filter*.

Warna	**Nilai R (*red*)**		**NilaiG (*green*)**		**Nilai B (*blue*)**	
	Min	Max	Min	Max	Min	Max
Merah	90	255	0	40	20	100
Hijau	10	80	50	150	10	80
Biru	0	30	30	170	100	230
Kuning	170	240	130	220	0	60

Tabel 3. HSV *color range filter*.

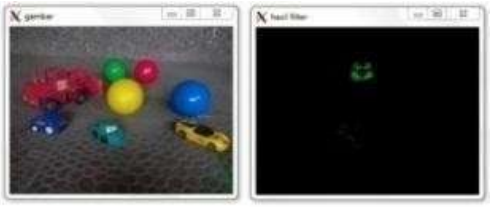
Warna	**Nilai H *(hue)***		**Nilai S *(saturation)***		**Nilai V *(value)***	
	Min	Max	Min	Max	Min	Max
Merah	50	79	50	50	0	55
Hijau	0	100	133	255	30	255
Biru	102	139	140	255	30	255
Kuning	0	35	90	255	70	255



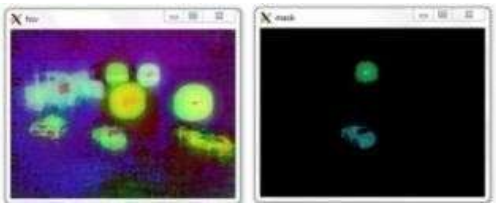
(a) (b)

(c) (d)

Gambar 15. Hasil *color filtering* gambar warna merah dengan ruang warna RGB dan HSV.(a) citra RGB (b) *color filtering* RGB (c) citra HSV (d) *color filtering* HSV.

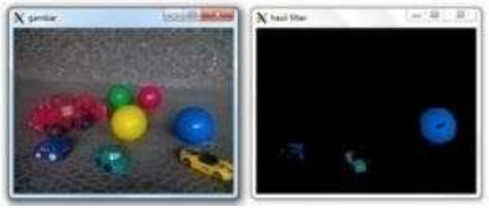


(a) (b)

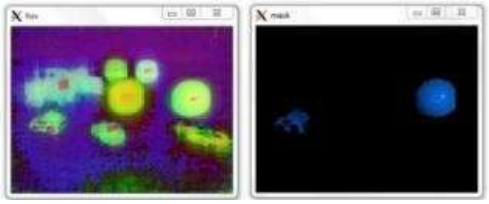


(c) (d)

Gambar 16. Hasil *color filtering* gambar warna hijau dengan ruang warna RGB dan HSV.(a) citra RGB (b) *color filtering* RGB (c) citra HSV (d) *color filtering* HSV.

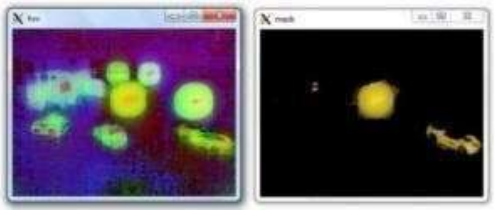


(a) (b)

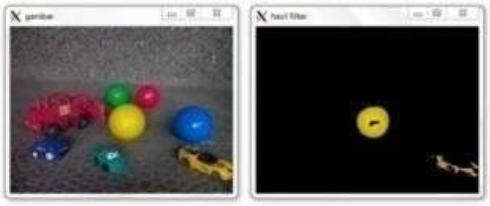


(c) (d)

Gambar 17. Hasil *color filtering*citrawarna biru dengan ruang warna RGB dan HSV.(a) citra RGB (b) *color filtering* RGB (c) citra HSV (d) *color filtering* HSV.



(a) (b)



(c) (d)

Gambar 18. Hasil *color filtering*citrawarna kuning dengan ruang warna RGB dan HSV. (a) citra RGB (b) *color filtering* RGB (c) citra HSV (d) *color filtering* HSV.

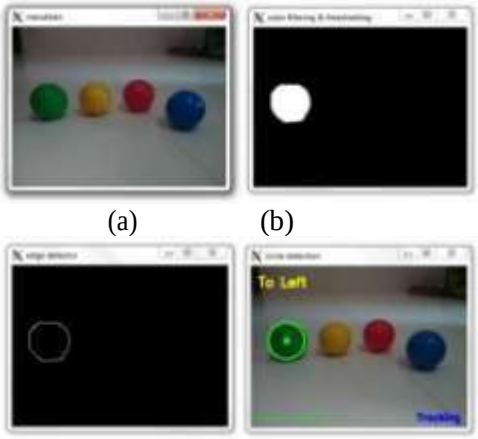
Agar dapat mengenali warna tertentu di berbagai kondisi pencahayaan menggunakan proses *trial and error*,sehingga didapatkan nilai *range* warna seperti pada tabel 2 dan 3. Kekurangan dari proses *color filtering* dengan ruang warna RGB yaitu proses *trial and error* sulit diperoleh nilai yang sesuai, pada saat percobaan di kondisi pencahayaan yang baik hasil yang diperoleh cukup baik namun di kondisi intensitas cahaya yang berbeda, maka hasilnya pun akan kurang sesuai yang di inginkan.Dengan pengaturan nilai range HSV maka didapatkan hasil seperti pada Gambar 19 hingga 22.Saat robot dihadapkan beberapa bola yang memiliki warna berbeda robot dapat memilih bola dengan warna sesuai perintah yang dibuat.

Jika dibandingkan dengan penggunaan ruang warna RGB, ruang warna HSV dapat menyeleksi warna dengan lebih teliti. Seperti terlihat pada Gambar 16 saat mengenali warna hijau. Pada sampel citra, terdapat dua benda berwarna hijau, namun dengan tingkat ‘kehijauan‘ yang tidak sama. Dengan ruang warna RGB, hanya dikenali satu benda berwarna hijau saja, sedang dalam ruang warna HSV, kedua benda berwarna hijau tersebut dapat dikenali.

(a)(b)

(c) (d)

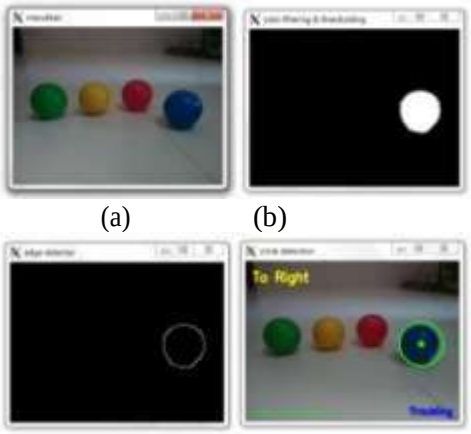
Gambar 19. Hasil pengolahan citra bola merah.(a) citramasukkan (b) *color filtering&thresholding* (c) *edge detector* (d) *circle detection.*



(a) (b)

(c) (d)

Gambar 20. Hasil pengolahan citra bola hijau.(a) citramasukkan (b) *color filtering&thresholding* (c) *edge detector* (d) *circle detection.*

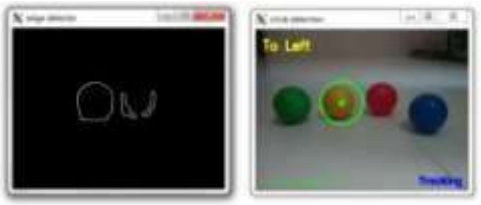


(a) (b)

(c) (d)

Gambar 21. Hasil pengolahan citra bola biru.(a) citramasukkan (b) *color filtering&thresholding* (c) *edge detector* (d) *circle detection.*

(a)(b)

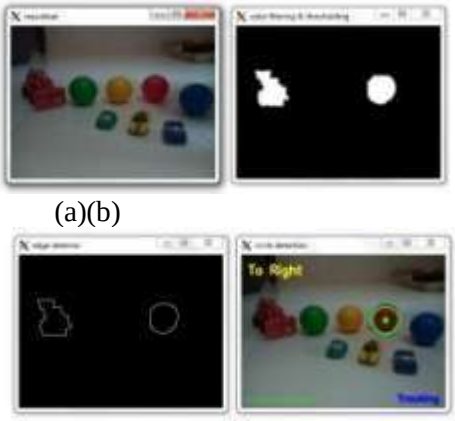


(c) (d)

Gambar 22. Hasil pengolahan citra bola kuning.(a) citramasukkan (b) *color filtering&thresholding* (c) *edge detector* (d) *circle detection.*

Pemilihan Objek Berdasarkan Bentuk

Saat dihadapkan dengan objek yang memiliki warna sama namun berbeda bentuk, robot dapat memilih objek mana yang memiliki bentuk lingkaran melalui proses transformasi*Hough circle.* Robot dapat mengenali bentuk lingkaran walaupun hasil proses *edge detector* menghasilkan bentuk lingkaran yang tidak sempurna, dapat dilihat pada Gambar 23 hingga 26.



(a)(b)

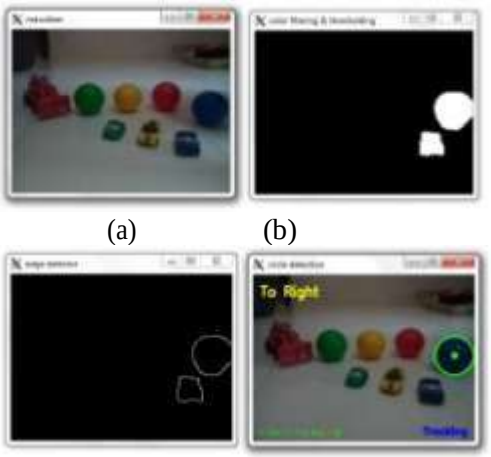
(c) (d)

Gambar 23. Hasil pengolahan citra membandingkan bentuk objek berwarna merah.(a)citramasukkan(b) *color filtering&thresholding* (c) *edge detector* (d) *circle detection.*

(a) (b)

(c) (d)

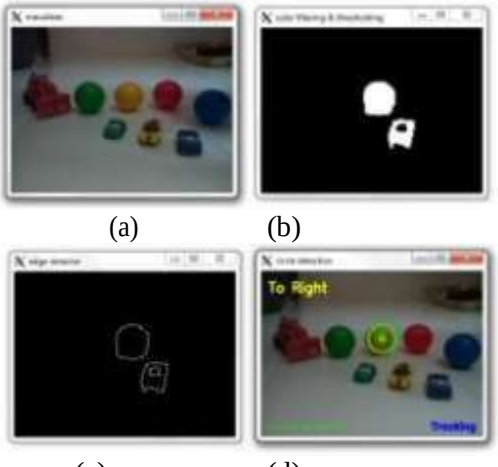
Gambar 24. Hasil pengolahan citra membandingkan bentuk objek berwarna hijau. (a) citramasukkan (b) *color filtering&thresholding* (c) *edge detector* (d) *circle detection.*



(a) (b)

(c) (d)

Gambar 25. Hasil pengolahan citra membandingkan bentuk objek berwarna biru. (a) citramasukkan (b) *color filtering&thresholding* (c) *edge detector* (d) *circle detection.*



(a) (b)

(c) (d)

Gambar 26Hasilpengolahancitramembandingkanbentukobjekberwarnakuning.(a) citramasukkan *(b) color filtering&thresholding* (c) *edge detector* (d) *circle detection.*

Pengaruh Intensitas Cahaya Terhadap Sistem Kerja Robot

Saat penentuan nilai range warna dari proses *color filter*padacitradengan ruang warna HSV,nilai dari V (*value*) mempengaruhi nilai maksimum dan minimum tingkat kecerahan warna. Dapat diamati pada Gambar 27 hingga 30 bahwa walaupun dalam keadaan gelap robot masih dapat mengenali warna dengan baik

Tabel 4. Intensitas cahaya minimum robot dapat mengenali bola.

Warna	Intensitas CahayaMinimum
Merah	22.7 lux
Hijau	21.2 lux
Biru	21.1 lux
Kuning	23.6 lux

Gambar 27. Intensitas cahaya paling gelap dalam mengenali bola warna merah.

Gambar 28. Intensitas cahaya paling gelap dalam mengenali bola warna biru.

Gambar 29. Intensitas cahaya paling gelap dalam mengenali bola warna hijau.

Gambar 30. Intensitas cahaya paling gelap dalam mengenali bola warna kuning.

5. SIMPULAN

Pada hasil penelitian dapat disimpulkan bahwa penggunaan resolusi kamera mempengaruhi jarak objek yang dapat dikenali.Semakin besar resolusi kamera maka akan semakin jauh jarak objek dapat dikenalioleh robot. Meskipun demikian, besarnya resolusi ini akan menyebabkan kerja *processor* dalam proses pengolahan citra yang semakinberat.

Penggunaan format citradengan ruang warna HSV untuk proses *color filtering* menghasilkan performa yang baik dalam mengenali objek dengan warna yang diinginkan. Pada kondisi pencahayaan yang bagus yaitu tidak terlalu terang dan tidak terlalu gelap didapatkan proses *color filtering*yang baik.Bentuk objekdapat diamati dengan baik melalui proses *edge detector*. Dengan menggunakan proses *circle hough*, kamera dapat mengenali objek yang berbentuk lingkaran dengan baik.

Robot dapat melakukan *tracking* terhadap boladenganmenggunakan resolusi citraterbaikpadaukuran320x240 piksel, memiliki jarak pandang maksimum 113 cm, kecepatan maksimalmengikuti perpindahan objek 22,6 cm/detik danmampu mengenali bola pada intensitas cahaya minimum 21,0 lux.

Saran untuk penelitian selanjutnya baiknya digunakan penggerak motor *omnidirectional* agar robot dapat bergerak bebas ke berbagai arah. Perluditambahkan sensor pendeteksihalangan agar tidakterjadi robot menabrakdindingataubenda lain. Penggunaan motor servo baiknya menggunakan motor servo jenis digital agar pergerakannya lebih halus serta derajatnya lebih teliti dan dapat bergerak 0˚ hingga 360˚. Proses pengolahan citra diperlukan algoritma yang lebih kompleks lagi untuk mengurangi *error* dalam pengenalan objek.

6. REFERENSI

Sural, S., Qian, G., Pramanik, S. 2002. Segmentation and histogram generation using the HSV color space for image retrieval.*Proceedings of IEEE International Conference on Image Processing*. 589-592.

Marchand, E. 2007. Control Camera and Light Source Positions using Image Gradient
Information. *IEEE Int. Conf. on Robotics and Automation*. 417 – 422.

Evan, Y. 2010. *Thresholding citra*, kuliahinformatika.wordpress.com

Sigit, H., & Agung, T. 2010. *Image processing dasar*, kliktedy.wordpress.com

Kragic, D., Christensen, H.I. 2011. Survey on Visual Servoing for Manipulation: Centre forAutonomous Systems. *Numerical Analysis and ComputerScience*.

Ikwuagu, E. 2011. Design Of An Image Processing Algorithm ForBall Detection. *Computing Research Association.*

Brigida, A.2012. *Transformasi hough*, informatika.web.id.

Yustinus, P. 2012. Rancang Bangun Aplikasi Pendeteksi Bentuk Dan Warna Benda Pada Mobile Robot Berbasis Webcam. *Academia*.

Agus, K. 2015. *Convert RGB To HSV for color tracking in Raspberry and openCV*, ilmu-otomasi.blogspot.com.