

## Algoritma Tracking dan Estimasi Jarak Bola Pada Robot Soccer Humanoid

Syaiful Amri<sup>1)</sup>

Jurusan Teknik ELEKTRO,  
Politeknik Negeri Bengkalis  
email: syaifulamri@polbeng.ac.id

### Abstract

*Humanoid soccer robot is a robot that is designed to play ball as humans play soccer, one of the algorithms owned by humanoid soccer robot to be able to see the ball is a scanning-tracking algorithm. Actuators used for scanning-tracking are two dynamixel AX12 servo motors. The task of the actuator is to make horizontal and vertical directions with the control method used in the tracking process, namely proportional control. After the ball is obtained from the scanning results, the tracking process is carried out which is then calculated the distance between the robot's foot and the ball. by knowing the distance of the ball, the robot can perform actions such as kicking, dribbling and stepping. The method used to get the distance of the ball with a robot is the trigonometry equation method based on the dynamixel servo motor angle. The results obtained after testing with ball spacing varies from 5cm to 500 cm, the process of scanning and tracking by finding the midpoint values of coordinates (position) of the ball object can be done at 160x120 pixel image frame resolution with the maximum distance that can be identified is 350 cm . while the ball spacing based on the actuator angle  $\gamma$  has an average estimated error of 15.71%.*

**Keywords:** Robot Humanoid, scanning, tracking, kontrol proporsional

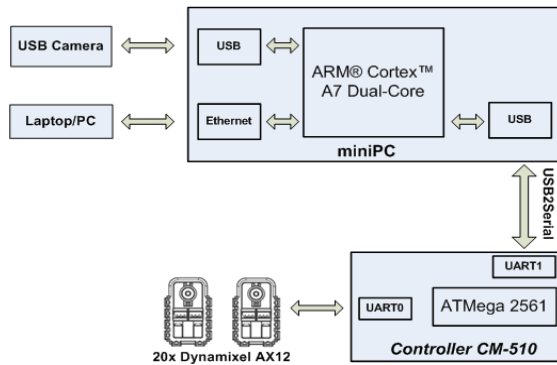
### PENDAHULUAN

Robot humanoid soccer adalah salah satu divisi dari Kontes Robot Indonesia (KRI). Tugas robot ini adalah bermain bola sesuai dengan rule standar internasional yang telah ditetapkan organisasi RoboCup. Sensor-sensor yang digunakan pada robot soccer ini harus merupakan representasi dari indera yang dimiliki oleh manusia. Sensor kamera adalah solusi yang paling tepat untuk memenuhi kebutuhan vision ini, dimana sensor kamera ini juga bisa mengenali objek yang lebih spesifik, diantaranya bisa berfungsi untuk mengidentifikasi objek (warna dan bentuk), dan mengukur jarak objek yang merupakan fitur yang penting pada robot soccer. Objek yang dimaksud pada robot soccer adalah objek-objek yang ada pada lapangan pertandingan robot soccer seperti bola, gawang, garis putih, warna lapangan dan robot lawan.

Proses scanning dan Tracking adalah algoritma yang harus dimiliki robot soccer untuk bermain bola. Proses scanning adalah suatu algoritma untuk menemukan objek yang belum dideteksi sedangkan proses Tracking yaitu suatu proses mengikuti jejak dari pergerakan bola dan proses untuk mempertahankan posisi bola berada pada posisi center kamera. Algoritma Scanning dan Tracking adalah salah satu algoritma yang dimiliki robot soccer sebagai pendeteksi untuk memperebutkan bola dan menendang bola ke gawang lawan.

### METODE PENELITIAN

Pada penelitian ini secara umum ada dua perancangan, yaitu perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*).



Gambar 1. Interkoneksi perangkat keras sistem Tracking dan scanning

Interkoneksi masing-masing perangkat keras dari satu perangkat ke perangkat lainnya seperti yang ditunjukkan pada Gambar 1, perangkat kerasnya meliputi : usb webcam, miniPC (mini personal computer), controller CM510, aktuator dynamixel AX12 dan laptop.

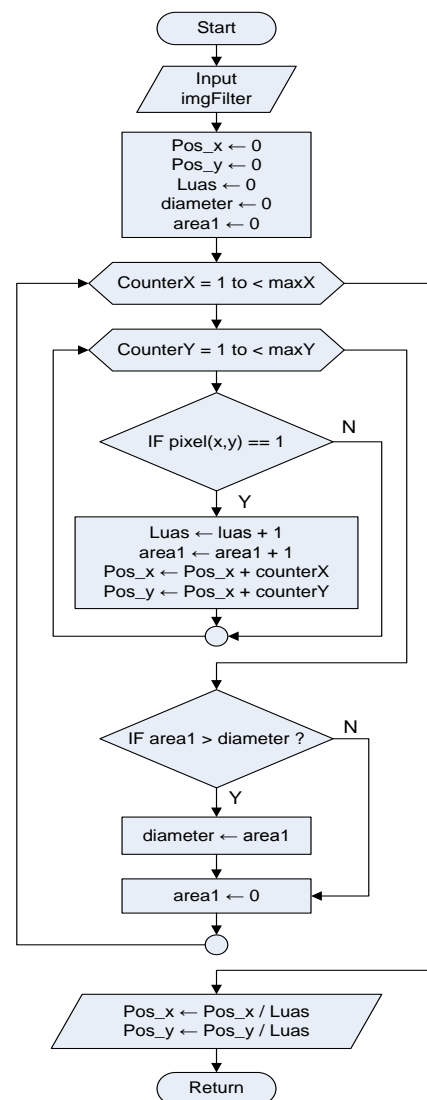
Adapun penjelasan interkoneksi masing-masing perangkat keras adalah sebagai berikut :

- USB Camera berfungsi untuk mengambil video yang kemudian dikonversi menjadi citra digital. Citra Digital kemudian akan di proses oleh miniPC untuk melakukan proses pengolahan citra dengan bantuan bahasa pemrograman C/C++ dan openCV library. Citra yang dihasilkan dari webcam merupakan citra digital yang selanjutnya akan diproses.
- Untuk menghubungkan laptop ke miniPC yaitu dengan menggunakan kabel jaringan atau yang biasa dikenal dengan kabel UTP.
- Controller CM-510 yang digunakan pada penelitian ini adalah berbasis ATMega2561, dimana pada controller ini terdapat dua jalur komunikasi UART. Pada UART1 digunakan untuk jalur komunikasi controller dengan miniPC. Dimana pada port USB miniPC ditambah dengan USB2Serial converter agar bisa terhubung secara serial dengan UART1 controller CM-510.

- Hasil identifikasi objek yang diproses dalam miniPC akan ditransfer yaitu berupa data nilai posisi dan jarak objek yang dibutuhkan oleh controller CM-510. Sementara pada UART0 digunakan untuk jalur komunikasi controller dengan aktuator dynamixel AX12 dengan mode half duplex.

## ALGORITMA Mencari LUAS DAN MOMENT

Setelah kontur di-filter, proses berikutnya adalah mencari posisi (titik koordinat) objek yaitu dengan mencari pusat massa atau centroid dengan menggunakan nilai rerata koordinat setiap pixel yang menyusun objek.



Gambar 2. Algoritma mencari posisi koordinat (moment) objek

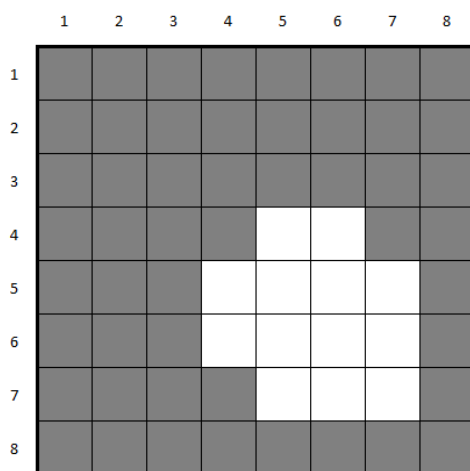
Citra biner yang mana axy akan bernilai 0 atau 1, moment tingkat ke-0 ( $m_{00}$ ) adalah sama dengan area dari objek. Adapun algoritma untuk mencari moment objek ditunjukkan pada flowchart Gambar 1.

- counterX = variabel iterasi X (horizontal/kolom).
- counterY = variabel iterasi Y (vertikal/baris).
- maxX = nilai maksimal X (jumlah maksimal kolom).
- maxY = nilai maksimal Y (jumlah maksimal baris).
- Pos\_x = variabel untuk  $m_{10}$ .
- Pos\_y = variabel untuk  $m_{01}$ .
- Luas = variabel untuk menyimpan luas area objek ( $m_{00}$ ).

Pusat dari area atau massa (centroid) adalah parameter untuk menyatakan lokasi dari objek. Pusat area dari suatu objek didefinisikan sebagai berikut.

$$x' = \frac{m_{10}}{m_{00}} \text{ dan } y' = \frac{m_{01}}{m_{00}} \quad (1)$$

Dimana ( $x'$ ,  $y'$ ) merupakan pusat koordinat dari objek. Pada Gambar 4.30 adalah contoh hasil dari threshold yang akan dicari pusat koordinat sehingga akan diketahui lokasi objek pada citra.



Gambar 3. Contoh objek hasil dari segmentasi

Dari Gambar 3 dapat dicari luas area, dan koordinat dengan rerata setiap pixel yang menyusun objek pada citra tersebut. Berikut adalah tabel titik koordinat (x,y) yang didapatkan dari hasil

segmentasi yang ditunjukkan pada Gambar3.

Tabel 1. Ilustrasi Menentukan Nilai Koordinat

Luas/Area	Koordinat pixel x	Koordinat pixel y
1	5	4
2	6	4
3	4	5
4	5	5
5	6	5
6	7	5
7	4	6
8	5	6
9	6	6
10	7	6
11	5	7
12	6	7
13	7	7
Jumlah	73	73

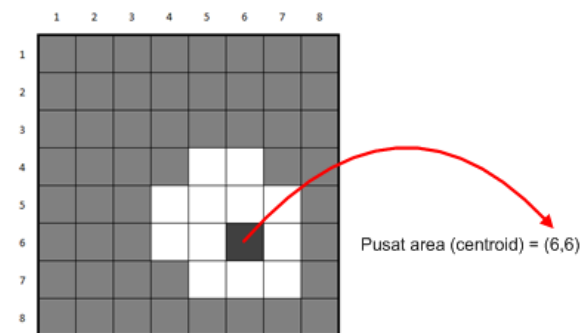
Dengan menggunakan persamaan 1, maka didapatkan :

$$x' = \frac{m_{10}}{m_{00}} \text{ dan } y' = \frac{m_{01}}{m_{00}}$$

$$x' = \frac{43}{73} = 5,6 \rightarrow 6$$

$$y' = \frac{43}{73} = 5,6 \rightarrow 6$$

Sehingga posisi objek pada koordinat pixel (6,6) yang diperlihatkan pada Gambar 4.31.

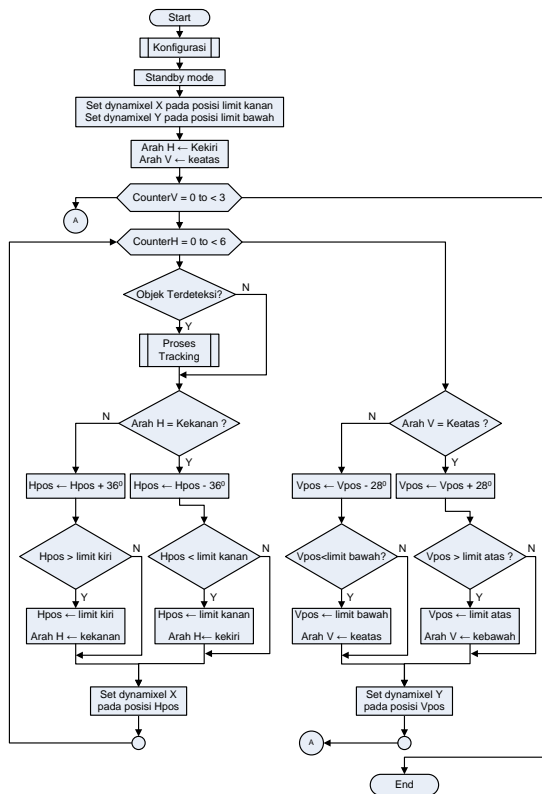


Gambar 4. Contoh titik koordinat objek

### ALGORITMA SCANNING DAN TRACKING

Perancangan algoritma yang akan ditanamkan di controller CM-510 dengan fungsi untuk menggerakkan aktuator dynamixel AX12 berdasarkan data posisi objek yang didapatkan dari proses identifikasi objek yang dilakukan di miniPC yang dikirim melalui jalur komunikasi serial. Terdapat dua poses yang dilakukan, yaitu proses scanning (suatu algoritma untuk menemukan objek yang belum dideteksi) seperti yang

ditunjukkan pada Gambar 5 dan proses Tracking yang ditunjukkan pada Gambar 6. Proses pertama yang dilakukan adalah proses scanning, kemudian proses Tracking yaitu suatu proses mengikuti jejak dari pergerakan bola dan proses untuk mempertahankan posisi bola berada pada posisi center kamera. Dimana kedua aktuator ini adalah bagian leher dari robot.



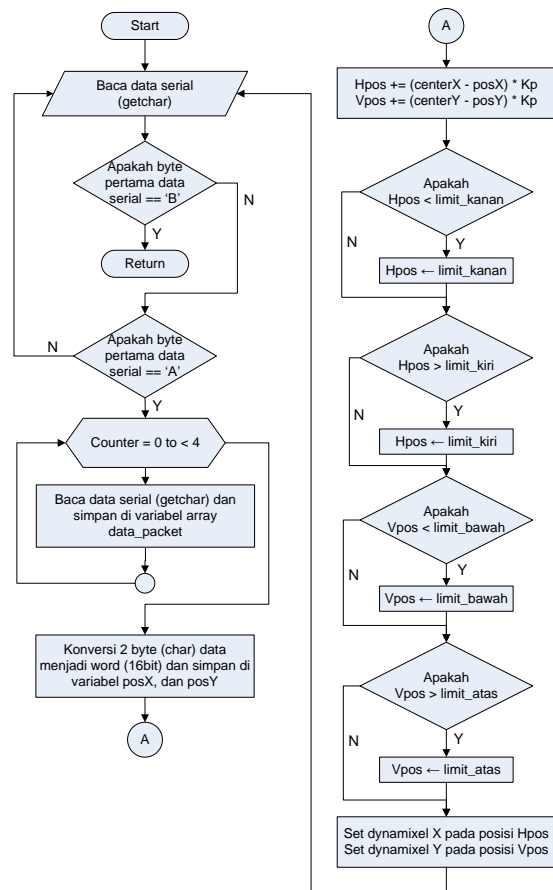
Gambar 5. Algoritma scanning untuk controller

Pada saat awal start, aktuator akan bergerak untuk melakukan proses scanning untuk mendeteksi objek, setelah objek terdeteksi, membaca data serial yaitu informasi posisi objek (x,y). Setelah informasi didapatkan, kemudian membandingkan untuk mencari titik tengah posisi objek.

- Center = nilai posisi yang diinginkan (posisi tengah).
- centerX = selisih nilai center dengan nilai X (posisi horizontal) yang dibaca (nilai aktual).

- centerY = selisih nilai center dengan nilai Y (posisi vertikal) yang dibaca (nilai aktual).

Apabila objek berada di kanan ( $centerX > 0$ ), maka aktuatorX akan bergerak kekanan. Sebaliknya, apabila objek berada di posisi kekiri ( $centerX < 0$ ), maka aktuatorX akan bergerak kekiri dan akan berhenti apa bila telah berada di posisi tengah ( $X = 0$ ). Begitu juga pada posisi vertikal, apabila objek berada di posisi atas ( $centerY > 0$ ), maka aktuatorY akan bergerak keatas. Sebaliknya apabila objek berada di posisi bawah ( $centerY < 0$ ), maka aktuatorY akan bergerak kebawah dan aktuatorY akan berhenti jika sudah berada di posisi tengah vertikal ( $centerY = 0$ ). Jika objek teridentifikasi maka proses Tracking ini akan berlangsung terus menerus (endless loop) sampai waktu pertandingan selesai.



Gambar 6. Algoritma tracking untuk controller

Algoritma scanning yang ditunjukkan pada Gambar 5 terdapat beberapa

variabel dan konstanta, adapun penjelasannya adalah :

- Hpos = variabel posisi horizontal
- Vpos = variabel posisi vertikal.
- Limit\_kanan = konstanta batas gerak kanan untuk aktuatorX (60°).
- Limit\_kiri = konstanta batas gerak kiri untuk aktuatorX (240°).
- Limit\_atas = konstanta batas gerak atas untuk aktuatorY (75°).
- Limit\_bawah = konstanta batas gerak bawah untuk aktuatorY (142°).

Terdapat 2 variabel yang dibutuhkan untuk melakukan proses scanning dan tracking adalah Hpos dan Vpos. Sementara untuk menentukan nilai Hpos dan Vpos menggunakan metode kontrol proportional yaitu [10] :

$$pos = Kp \times error \quad (2)$$

$$error = SP - PV \quad (3)$$

Dimana,

- pos = Perubahan posisi
- Kp = Konstanta proportional
- SP = Set Point
- PV = Proses Variabel

Error adalah deviasi atau simpangan antara variabel terukur (PV) dengan nilai acuan (SP). PV yang dimaksud adalah posisi objek yang terukur, pada kasus ini adalah variabel posX dan posY yang didapatkan dari proses identifikasi dan pencarian titik pusat (centroid) objek. Sedangkan SP adalah nilai yang diinginkan, yaitu posisi tengah citra yaitu centerX dan centerY. Pada frame 160x120 pixel, centerX = 80 dan centerY = 60 Sehingga didapatkan :

$$errorX = (centerX - posX)$$

$$Hpos += Kp \times errorX$$

$$errorY = (centerY - posY)$$

$$Vpos += Kp \times errorY$$

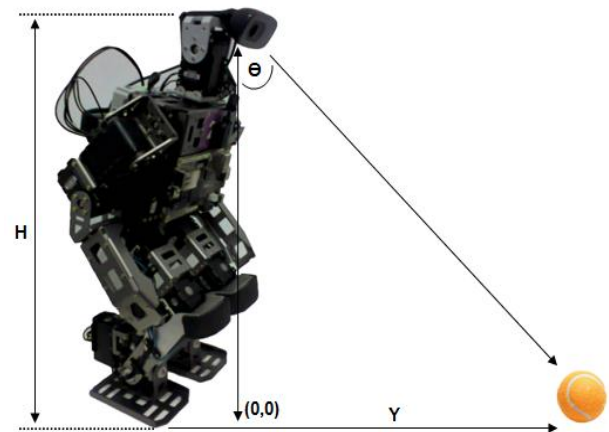
Dimana,

- errorX = error pada posisi X (horizontal).
- Vpos = update posisi horizontal untuk aktuatorX (goal\_position aktuatorX).
- errorY = error pada posisi Y (vertikal).

Hpos = update posisi horizontal untuk aktuatorX(goal\_position aktuatorX).

### ESTIMASI JARAK BOLA

Ada dua cara yang bisa dilakukan untuk mengestimasi jarak bola yaitu berdasarkan sudut aktuatorY, dan berdasarkan luas area atau diameter bola yang dihasilkan dari data percobaan.



Gambar 7. Hubungan sudut aktuatorY terhadap jarak bola

Dengan mengetahui adanya perubahan sudut aktuatorY, maka jarak bola dapat diestimasi seperti yang diperlihatkan pada Gambar 7.

Dengan menggunakan persamaan *trigonometry*, sehingga didapatkan.

$$Y = H \tan \theta \quad (4)$$

- Y = Jarak antara kaki robot terhadap bola
- H = Tinggi robot (36 cm)
- θ = Sudut pandang robot terhadap bola (sudut aktuatorY).

Dengan berubahnya jarak bola, maka akan berubah pula sudut pandang robot (aktuatorY). Untuk mengkonversi nilai data aktuatorY menjadi sudut, langkah pertama adalah dengan mengasumsi jarak pengukuran pertama yaitu jarak aktual (40 cm), kemudian didapatkan sudut aktuatorY 480 yang di representasikan dengan nilai data 353, sehingga inilah menjadi asumsi pertama untuk mendapatkan sudut aktuatorY yaitu dengan diketahui Y=40cm dan

H=36cm sehingga didapatkan sudut acuan pertama yaitu :

$$\tan^{-1} \theta = \frac{Y}{H} \quad (5)$$

Setelah sudut acuan didapatkan, berikutnya pada jarak yang berbeda bisa dicari dengan persamaan.

$$\theta = \frac{\text{Selisih data}}{kd} + \text{sudut acuan} \quad (6)$$

$\theta$  = sudut hasil konversi nilai data aktuatorY yang terbaca.

Kd = konstanta data aktuatorY yaitu (3,41/derjat).

Selisih data = nilai data aktuatorY yang terbaca dikurangi data acuan (353).

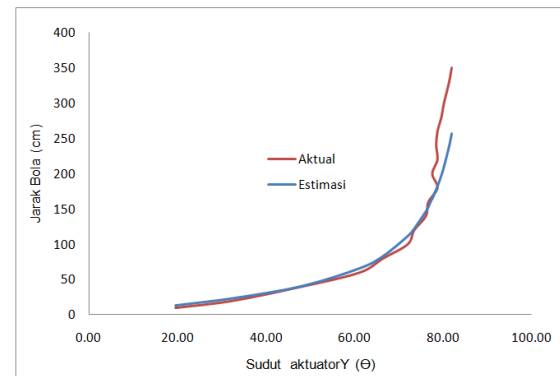
Tabel 1. Hasil pengujian jarak bola berdasarkan sudut aktuatorY

Jarak Aktual (cm)	Data AktuatorY	$\theta$	Jarak Estimasi (cm)	Error (%)
350	469	82,02	256,73	26,65
330	467	81,43	238,92	27,60
300	463	80,26	209,68	30,11
280	461	79,67	197,54	29,45
260	458	78,79	181,68	30,12
240	457	78,50	176,92	26,28
220	458	78,79	181,68	17,42
200	454	77,62	163,99	18,00
180	458	78,79	181,68	0,93
160	451	76,74	152,76	4,53
140	449	76,15	146,04	4,32
120	440	73,51	121,64	1,36
100	435	72,05	111,11	11,11
80	416	66,48	82,70	3,37
60	398	61,20	65,47	9,12
40	353	48,00	39,98	0,04
20	300	32,46	22,90	14,48
10	256	19,55	12,79	27,87
Rata-rata :				15,71

Dari data pengujian seperti yang ditunjukkan pada Tabel 3, didapatkan sudut aktuatorY, maka didapatkan jarak estimasi yang kemudian didapatkan error estimasi jarak terhadap jarak aktual. Dari hasil pengujian didapatkan rata-rata error adalah 15,71 %. Pada Gambar 8 diperlihatkan hubungan sudut aktuatorY terhadap jarak estimasi dan jarak aktual.

Kurva pada Gambar 8 dapat disimpulkan bahwa terdapat error estimasi pada jarak lebih besar 180 cm dikarenakan resolusi data sudut

aktuatorY pada jarak lebih besar 180 cm sangat kecil seperti pada Tabel 3, bahkan jika dikonversi menjadi sudut tidak melebihi dari 1 derajat, karena 1 derajat sama dengan 3,41 data aktuatorY.



Gambar 8. Hubungan jarak bola terhadap sudut aktuatorY dan error estimasi jarak

## HASIL DAN PEMBAHASAN

Sesuai dengan aturan pertandingan robot soccer bahwa gerakan leher (arah pandangan) robot tidak boleh melebihi dari 180° arah horizontal, sehingga pada pengujian Tracking ini, dilakukan didalam wilayah sudut 140°.

Tabel 2. Hasil pengujian scanning dan Tracking bola pada jarak 5cm- 100cm

No Uji	Jarak		
	5 cm (7,91°)	50 cm (54,24°)	100 cm (72,05°)
1	Berhasil	Berhasil	Berhasil
2	Berhasil	Berhasil	Gagal
3	Gagal	Berhasil	Berhasil
4	Gagal	Berhasil	Berhasil
5	Berhasil	Berhasil	Berhasil

Hasil pengujian diperlihatkan pada Tabel 1 dan 2, dimana jarak minimal 5 cm dan maksimal 350 cm (jarak di ukur dari kaki robot ke bola) yang masing-masing jarak di uji sebanyak 5 kali. Dari hasil pengujian seperti pada Tabel 1 dan 2, bahwa terjadi banyak kegagalan pada jarak 5 cm dan 350 cm. Penyebab kegagalan pada jarak 5 cm ini dikarenakan citra yang diambil dari kamera terhalang oleh bodi robot.

Sedangkan kegagalan pada jarak 350 cm, ini dikarenakan resolusi frame citra 160x120 pixel belum bisa mendeteksi pixel objek (bola) pada jarak jauh, pada pengujian ini memiliki banyak noise pada jarak lebih besar dari 350 cm.

Tabel 3. Hasil pengujian scanning dan Tracking bola pada jarak 200cm - 350cm

No Uji	Jarak		
	200 cm (77,62 <sup>o</sup> )	300 cm (80,26 <sup>o</sup> )	350 cm (82,02 <sup>o</sup> )
1	Berhasil	Berhasil	Berhasil
2	Berhasil	Berhasil	Gagal
3	Gagal	Berhasil	Berhasil
4	Gagal	Berhasil	Berhasil
5	Berhasil	Berhasil	Berhasil

- [5] Amri. S. Algoritma Tracking Dan Scanning Pada Robot Soccer. Prosiding Seminar Nasional Teknik Elektro Volume 2, ISSN 2580-1988, 2017.
- [6] Amri. S. Perbandingan Estimasi Jarak Bola Berdasarkan Luas Area Bola Dan Sudut Pada Robot Soccer. Prosiding Seminar Nasional Teknik Elektro Volume 2, ISSN 2580-1988, 2017.

## KESIMPULAN

Algoritma tracking bola dapat dengan kontrol proporsional dengan nilai Kp 0.25 pada resolusi frame citra 160x120 pixel, sedangkan jarak maksimal yang bisa diidentifikasi adalah 350 cm. Penentuan jarak bola berdasarkan sudut aktuatorY mempunyai rata-rata error estimasi 15,71%.

## REFERENSI

- [1] M. N. A. Wahab., N. Sivadev and K. Sundaraj. Target Distance Estimation Using Monocular Vision System for Mobile Robot. IEEE Conference on Open System (ICOS2011), 2011.
- [2] Anon. Buku Panduan buku panduan KRCI RoboSoccer Humanoid League 2013 versi 1.0, Jakarta, 2013.
- [3] Amri. S., Suyono. H., Setyawaty. O. Identifikasi Warna pada Robot Soccer dengan Color Filtering, k-NN dan Naive Bayes. Jurnal EECIS Vol.8, No.2, 2014.
- [4] Jatmiko. Wisnu Dkk. Robotika Teori dan Aplikasi. Perpustakaan Nasional Fakultas Ilmu Komputer UI. Depok, 2012.