Multitek Indonesia: Jurnal Ilmiah
Volume: 13 No. 1, Juli 2019, Hal. 66 - 80
ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)
http://journal.umpo.ac.id/index.php/multitek

Deteksi Dan Prediksi Trajektori Objek Bergerak Dengan Omni-Vision Menggunakan Particle Swarm Optimization-Neural Network (PSO-NN) Dan Interpolasi Polynomial

Nur Alif Mardiyah[1)], Khusnul Hidayat[2)], Novendra Setyawan[3)]

[1), 2),3)] Teknik Elektro Universitas Muhammadiyah Malang
Jl Tlogomas No 246 Malang
Email : nuralif@umm.ac.id

Received: May 10th, 2019. Accepted: July 19th, 2019

ABSTRAK

Pada kompetisi robot sepak bola beroda Indonesia dalam satu tim terdiri dari tiga buah robot, dimana satu buah robot adalah penjaga gawang. Pada kompetisi tersebut pergerakan robot dan bola sangat dinamis. Sehingga dibutuhkan sebuah metode untuk memperediksi pergerakan bola sehingga penjaga gawang dapat mengantisipasi pergerakan bola. Pada penelitin ini perancangan *omnivision* dan pendeteksian bola dilakukan dengan pengolahan citra digital untuk mengenali objek bola dengan *background* yang kemudian akan dihitung posisi bola dalam pixel. Selanjutnya *Neural Network* digunakan sebagai model kalibrasi jarak dalam pixel ke jarak nyata (cm) yang bobotnya dilatih menggunakan *Particle Swarm Optimization*. Selanjutnya untuk memprediksi trajectory pergerakan bola pendekatan interpolasi kurva polynomial digunakan untuk mendapatkan perkiraan model dari data dua dimensi dari posisi bola yang terdeteksi. Hasil penelitian menunjukan bahwa konversi jarak pada pendeteksian objek dengan model PSO-NN didapatkan persentase rata-rata kuadrat *error*(PMSE) pengukuran 0.13% dan rata rata error prediksi sebesar 20%.

Kata kunci: Omni-Vision, Particle Swarm Optimization, Neural Network, Mobile Robot

ABSTRACT

In the Indonesian wheeled soccer robot competition in one team consists of three robots, where one robot is a goalkeeper. In the competition, the movement of robots and balls is very dynamic. So that a method is needed to predict the movement of the ball so that the goalkeeper can anticipate the movement of the ball. Omni-vision is designed, and ball detection is done by digital image processing to recognize spherical objects in a background which would be detected in pixel position in this paper. Furthermore, Neural Network (NN) used as a distance calibration model, which is the weight of NN trained by Particle Swarm Optimization. Furthermore, to predict the trajectory of spherical movement, the interpolation approach of the polynomial curve is used to obtain the approximate model of the two-dimensional data from the detected ball position. The results showed that the distance conversion in object detection with the PSO-NN model obtained a percentage of average squared error (PMSE) measurement of 0.13% and an average prediction error of 20%.

Keywords: Omni-Vision, Particle Swarm Optimization, Neural Network, Mobile Robot

PENDAHULUAN

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)

Kompetisi robot sepakbola adalah tempat yang tepat untuk menguji sistem kontrol [1], perencanaan jalur [2], sensor navigasi dan subjek penelitian sistem *vision* [3]. Dalam dekade terakhir, visi *omnidirectional* atau sistem omni-vision telah menjadi salah satu hal terpenting dalam sistem robot sepakbola. *Omni-vision* memberikan pandangan 360 derajat dari lingkungan sekitarnya robot dalam satu gambar yang dapat digunakan untuk deteksi objek [4], pelacakan [5], dan lokalisasi [6][7].

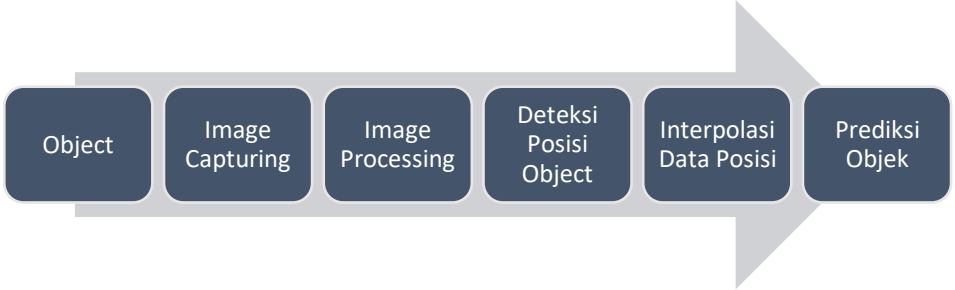
Umumnya, sistem *omni-vision* dapat dibentuk dengan berbagai cara, seperti kamera servo mekanis, lensa bola (*fisheye* panorama), dan cermin hiperbolik. Lensa sferis adalah salah satu cara yang lebih mudah dan efektif untuk memberikan penglihatan-omni, karena strukturnya padat dan kaku daripada menggunakan cermin refleksi yang terdiri dari dua bagian dan rapuh [5]. Terlepas dari keuntungan pandangan lebar dari gambar bulat, distorsi barel membuat pendeteksian objek atau pelacakan lebih rumit. Berbagai metode telah dikembangkan untuk memperbaiki dan mengembalikan menggunakan beberapa teknik pemrosesan gambar [6], yang membuat perhitungan lebih kompleks.

Salah satu teknik pemodelan buatan untuk mengkalibrasi citra bola adalah menggunakan *Neural Network* [8]. Selain NN metode yang berbasis heuristik telah banyak digunakan untuk menyelesaikan banyak masalah optimasi [9], pada penelitian ini PSO digabungkan untuk mengoptimalkan model NN untuk mengkalibrasi dan memodelkan jarak antara objek dan robot dari citra *spherical* dengan beberapa data *training* eksperimental dalam mengembangkan *omni-vision* yang efisien untuk mengenali dan melacak objek bergerak.

Pada prediksi dari sebuah data time series pendekatan interpolasi kurva polynomial sering digunakan untuk mendapatkan perkiraan model dengan mengetahui beberapa data dua dimensi. Pada [10] telah menerapkan pendekatan kurva interpolasi polynomial untuk perkiraan daya maksimum dalam panel surya. Dari penelitian tersebut pendekatan kurva interpolasi polynomial dapat digunakan untuk memperkirakan atau mendapatkan model perkiraan posisi objek yang berada dalam lingkungan kendaraan atau robot. Oleh karena itu dalam penelitian ini PSO-NN digunakan sebagai metode kalibrasi jarak objek dari hasil deteksi menggunakan pengolahan citra. Kemudian interpolasi *polynomial* digunakan sebagai metode prediksi trayektori objek bergerak yang terdeteksi.

METODE PENELITIAN

Pada penelitian ini terdiri dari beberapa tahapan yang dilakukan untuk memprediksi trajectory dari benda yang bergerak. Proses tahapan itu antaralain, *image capturing / acquisition*, *image processing* dan *object detection*, *distance calculation*, dan prediksi objek seperti yang digambarkan Gambar 1.



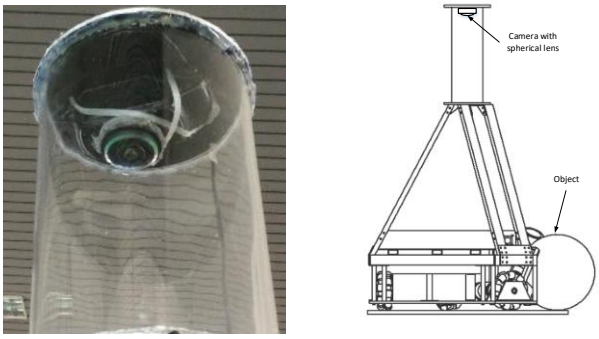
Gambar 1. Diagram system prediksi *trajectory* objek

A. Image Capturing

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)

1. Perancangan kamera *omni directional*

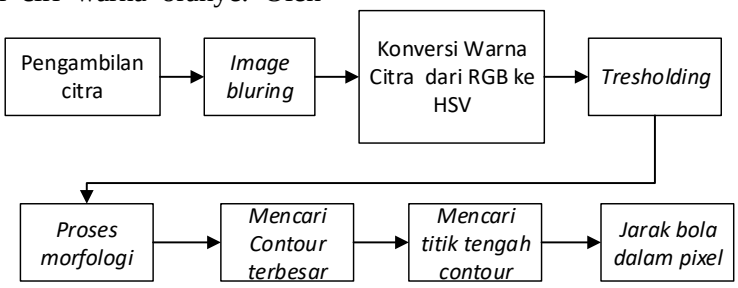
Pada pengambilan citra benda (bola), untuk mendapatkan citra tersebut secara efektif dapat menggunakan kamera *omni directional*. Kamera omnidirectional dirancang agar dapat mengambil citra lingkungan disekeliling robot hanya dalam satu *frame* citra tanpa adanya suatu mekanisme bergerak baik menggunakan *servo* maupun dari pergerakan robot. Untuk mendapatkan citra lingkungan tersebut dapat menggunakan lensa spherical atau biasa disebut dengan lensa *fisheye* dengan sudut kelengkungan 235^0 yang dipasang pada sebuah kamera *web cam* seperti yang ditunjukan pada Gambar 2(a). Kemudian agar mendapatkan citra lingkungan secara keseluruhan di lingkungan robot maka posisi kamera diletakan diatas robot dan menghadap ke bawah robot seperti yang ditunjukan pada **Error! Reference source not found.** Gambar 2 (b).



Gambar 2. Kamera *webcam* dengan lensa spherical (b) Posisi peletakan kamera pada robot

B. *Image Processing* untuk pengenalan benda

Pada penelitian ini objek yang akan dikenali merupakan objek yang memiliki ciri khusus berupa warna maupun bentuk, yaitu sebuah bola yang memiliki ciri warna oranye. Oleh karena itu, dalam pembuatan algoritma pengenalan bola memanfaatkan filter warna dengan pengolahan citra untuk memfilter warna oranye untuk bola. Algoritma pengolahan citra pengenalan bola dilakukan dengan proses seperti yang ditunjukan pada Gambar 3.



Gambar 3 Algoritma pengolahan citra pengenalan bola

1. Pengambilan Citra

Proses pengambilan citra merupakan tahapan awal untuk proses pengolahan citra pengenalan bola. Proses pengambilan citra dilakukan dengan memanfaatkan fitur *ofVideoGrabber* pada *openframework* yang digunakan untuk mengambil data citra yang dihasilkan oleh kamera wabcam. Kemudian data citra tersebut diletakan pada *frame* dengan alokasi ukuran yang diatur dengan ukuran 680x360. Pengaturan ukuran tersebut didasarkan pada kemampuan kamera yang dapat mengambil gambar dengan ukuran maksimal 1024x720. Jika ukuran *frame* diatur maksimal memiliki kelebihan citra yang dihasilkan lebih detil namun pemrosesan citra lebih lambat. Kemudian jika ukuran citra diatur pada ukuran

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)

minimum maka kita akan kehilangan detil citra namun pemrosesan citra lebih cepat. Sehingga ukuran *frame* pada nilai tengah dari ukuran maksimal yang dapat dihasilkan. Setelah itu *frame* citra tersebut dideklarasikan dengan tipe data *Mat* yang merupakan tipe data matriks dari library OpenCv untuk diproses pada tahapan selanjutnya.

2. Pengaburan Citra (*Image Bluring*)

Pada proses pengolahan citra tahapan awal yang dilakukan adalah *image bluring* atau disebut juga *smoothing*. *Image bluring* bertujuan untuk mengurangi *noise* pada pengambilan citra dengan cara mengaburkan (*bluring*) pada citra dengan ukuran pixel tertentu yang dapat diatur. Pada perancangan ini proses *image bluring* dilakukan dengan menggunakan *Gaussian Blur* yang merupakan proses pengaburan citra dengan fungsi *Gaussian*. Penggunaan *Gaussian blur* dilakukan dengan memanggil fungsi Gaussian Blur (*Mat frame in, Mat frame out, int size*) pada library *OpenCv*. Pada fungsi tersebut memiliki 3 parameter yaitu, Frame in yang merupakan sumber citra yang akan dikaburkan, frame out yang merupakan hasil citra yang telah dikaburkan, dan size yang merupakan parameter ukuran pixel yang akan dikaburkan.

3. Konversi Format warna RGB ke HSV

Pada tahapan ini format warna citra RGB (*red, green, blue*) disusun ulang kedalam format warna HSV (*hue, saturation, value*). Konversi warna dilakukan dikarenakan format warna RGB tidak memisahkan antara informasi warna dan informasi kecerahan. Sedangkan pada format warna HSV informasi disajikan dengan informasi warna (*hue)* dan derajat kecerahan *(saturation, value)*. Proses konversi warna dilakukan dengan memanggil fungsi *cvtColor(Mat frame in, Mat frame out,int code, dstCn*=0). Pada fungsi ini frame in adalah sumber data citra yang akan dikonversi, kemudian frame out adalah data citra hasil konversi, sedangkan code merupakan kode jenis konversi yang akan dilakukan. Pada konversi dari data citra dengan format warna RGB ke format warna HSV dilakukan dengan menggunakan code CV_BGR2HSV.

4. *Thresholding*

Proses *Tresholding* menghasilkan citra biner, yang mana objek yang berada di batasan warna tertentu akan berwarna putih. Sedangkan pada *background* yang berada diluar Batasan warna akan menghasilkan citra berwarna hitam. Dengan menggunakan proses *thresholding* objek berwarna dapat dipisahkan dari *background* yang berada disekitar objek.

Pada *OpenCV* proses *thresholding* dinyatakan dengan fungsi *inRange* (*InputArray src, InputArray lowerb, InputArray upperb, OutputArray* dst). Fungsi tersebut terdiri dari 4 parameter. Parameter pertama yaitu *InputArray* src adalah *array input* yang ingin dilakukan proses *thresholding*. Parameter kedua dan ketiga adalah *InputArray lowerb* dan *InputArray upperb* yang merupakan batas threshold bawah dan atas dari masing-masing kanal, parameter terakhir yaitu OutputArray dst yang merupakan array output yang akan berisi citra biner dari proses *thresholding*.

5. Proses Morfologi

Proses morfologi yang diberikan pada citra berupa proses *morfologopening*. Proses morfologi opening terdiri dari dua proses morfologi dasar yaitu *erotion* dan dilanjutkan dengan *dilation*. Sesuai dengan namanya proses *erosion* adalah proses mengikis atau mengerosi objek pada bagian terluarnya. Dalam hal ini daerah berwarna putih berkurang luasannya bahkan hilang untuk objek-objek berukuran kecil. Proses selanjutnya adalah *dilation* yaitu proses pelapisan objek pada bagian terluar. Proses ini menyebabkan luasan objek bertambah. Kedua proses tersebut menyebabkan objek-objek kecil yang biasanya berupa noise menghilang. Dengan proses ini diharapkan pengukuran yang diperoleh lebih akurat.

Di *openCV* telah disediakan fungsi untuk melakukan proses *morfologi erotion* dan *dilation* yaitu erode(InputArray src,

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)

OutputArray dst, InputArray kernel, Point anchor= Point(-1,-1), int iterations = 1, int borderType = BORDER_CONSTANT, const Scalar& borderValue =morphologyDefaultBorderValue()) dan dilate(InputArray src, OutputArray dst, InputArray kernel, Point anchor = Point(-1,-1), int iterations = 1, int borderType = BORDER_CONSTANT, const Scalar& borderValue = morphologyDefaultBorderValue()). Kedua fungsi tersebut memiliki parameter yang sama. Dari beberapa parameter yang tersedia, di penelitian ini hanya memanfaatkan 3 parameter pertama dan membiarkan parameter yang lain pada nilai defaultnya. Parameter pertama adalah InputArray src yaitu array input berupa citra yang ingin diberikan proses morfologi. Selanjutnya adalah OutputArray dst yaitu array output dari proses morfologi. Parameter selanjutnya adalah InputArray kernel yaitu kernel yang digunakan pada proses morfologi.

6. Pencarian titik tengah dan ukuran terbesar contour

Pendeteksian objek dilakukan dengan mencari objek bola terbesar dan melihat apakah disekitar objek tersebut terdapat warna hijau atau tidak. Jika disekirar objek tersebut terdapat warna hijau maka objek tersebut adalah bola dan ditandai sebagai bola sedangkan jika disekitar objek bola tersebut tidak terdapat warna hijau objek tersebut diabaikan dan tidak ada bola terdeteksi.

Sebelum melakukan seleksi terlebih dahulu dilakukan proses menandai setiap objek berdasarkan letaknya pada citra dan radiusnya. Untuk itu digunakan fungsi dari openCV yaitu minEnclosingCircle((Mat)contours[i], center[i], radius[i]). Fungsi tersebut bekerja untuk membuat lingkaran terkecil yang mampu melingkupi seluruh bagian objek (untuk satu objek). Data yang dikeluarkan daari fungsi tersebut berupa radius lingkaran yang melingkari objek yang akan disimpan pada variabel radius dan titik tengah dari lingkaran tersebut pada citra yang akan disimpan pada variabel center.

Setelah didapatkan beberapa lingkaran yang melingkari objek beserta informasi posisi dan radiusnya, proses selanjutnya adalah melakukan proses seleksi untuk menentukan objek bola. Objek bola adalah objek terbesar yang berada diatas lapangan berwarna hijau. Jika objek tersebut tidak berada diatas lapangan hijau maka objek tersebut bukan bola.

C. Perancangan Model Konversi Jarak Objek (*Distance Calculation*)

Pada proses pengolahan citra didapatkan data berupa kordinat posisi bola $(x_o\ , y_o)$ dalam satuan pixel. Untuk lebih mudah dalam memodelkan konversi kordinat posisi bola dari satuan pixel ke dalam jarak real dengan satuan centimeter, kordinat bola dirubah dari koordinat kartesian kedalam koordinat polar dengan perhitungan data jarak menggunakan persamaan sebagai berikut:

$$r_o = \sqrt{(x_o - x_c)^2 + (y_o - y_c)^2}$$

dan perhitungan sudut objek dengan persamaan sebagai berikut:

$$\theta_o = \text{atan}\left(\frac{(y_o - y_c)}{(x_o - x_c)}\right)$$

dimana:

r_o adalah jarak objek terhadat titik pusat koordinat robot

θ_o adalah sudut objek terhadat titik pusat koordinat robot

$(x_o\ , y_o)$ adalah koordinat posisi objek dalam pixel

$(x_c\ , y_c)$ adalah titik tengan koordinat robot dalam pixel

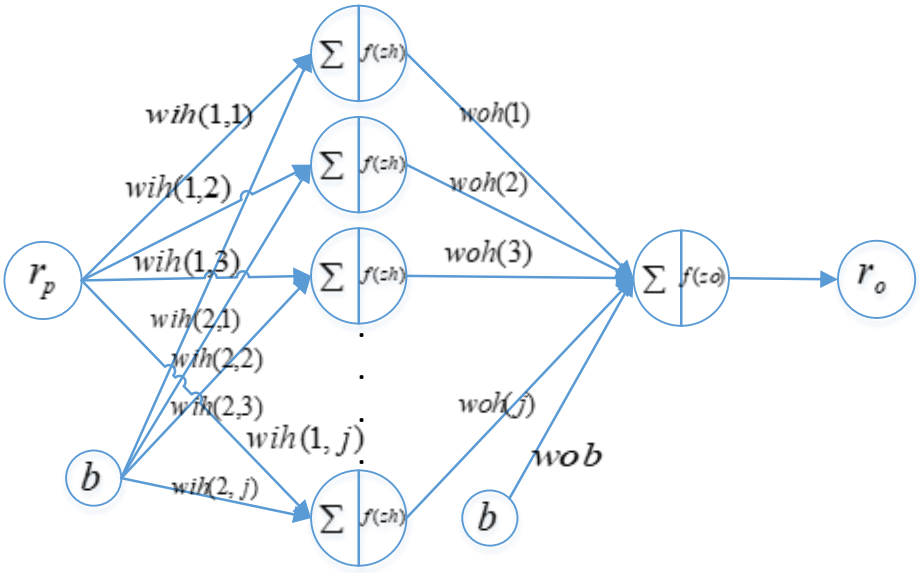
1. Training Model Konversi Jarak Objek Dengan PSO-NN

a. Model NN

Setelah proses konversi koordinat diakukan untuk mendapatkan data jarak real dalam centimeter model konversi dibuat menggunakan model neural network dengan

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)

satu input dan satu output dengan satu hidden layer seperti dinyatan pada gambar 4.



Gambar 4. Model Neural Network untuk Model Konversi Jarak

dengan persamaan model input ke hidden layer dijabarkan pada persamaan 1 dan 2.

$$zh(i,j) = \sum_{i=1}^{jml\_data} \sum_{j=1}^{10} wih(1,j).rp(i) + wih(2,j).b \quad (1)$$

$$yh(i,j) = f\big(zh(i,j)\big) \quad (2)$$

dan persamaan model hidden layer ke output dijabarkan pada persamaan 3 dan 4.

$$zo(i) = \sum_{j=1}^{10} woh(j).yh(i,j) + wob.b \quad (3)$$

$$r_o(i) = f\big(zo(i)\big) \quad (4)$$

dengan fungsi aktifasi pada $f\big(zh(i,j)\big)$ dan $f\big(zo(i)\big)$ menggunakan fungsi aktifasi bipolar sigmoid atau $\tanh(x)$.

b. Training Model NN dengan PSO

Untuk mendapatkan akurasi konversi, model NN dilakukan proses *training* menggunakan PSO. Pada proses *training* menggunakan PSO dengan menginisialisasikan bobot bobot pada model NN sebagai partikel pada PSO dengan inisialisasi sebagai berikut:

$$x(j,d) = [wih(1,1) \quad \dots \quad wih(1,j) \quad wih(2,1) \quad \dots \quad wih(2,j) \quad woh(1) \quad \dots \quad woh(j) \quad wob]$$

dengan fungsi objektif berupa presentasi rata rata kuadrat *error* yang dinyatakan dalam persamaan 5.

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)

$$pmse = \frac{\sum_{i=1}^{jml\_data}\left(r_t(i) - r_o(i)\right)^2}{jml\_data} * 100 \quad (5)$$

Proses training model-NN dengan menggunakan PSO dapat dinyatakan dengan *pseudocode* pada Gambar 5

Start
Initialize particle $x_{i,d}$ as a wih(1,j), wih(2,j), woh(1,j), and wob;
For i=1 to n
$P_i = x_i$;
End
While (termination condition is false)
SC=0;
For i=1 to number of particle
For d=1 to n
Update velocity of particle;
Update position of particle;
Evaluate particle with objective function;
End
End
Update P_{id} and G_d;
Update Inertia parameter;
End
$(wih(1,j), wih(2,j), woh(1,j), and\ wob) = G_d$;
End

Gambar 5. Pseudo Code PSO-NN

7. *Pendekatan prediksi objek dengan Interpolasi Polynomial*

a. Model Interpolasi Polynomial

Karena informasi mengenai objek yang didapat dari pengolahan citra hanya posisi sedangkan dinamika halangan tidak bisa diketahui secara akurat maka pergerakan objek didekati dengan persamaan polynomial dengan orde 2, seperti pada persamaan 6

$$x(t) = a_0 + a_1 t + a_2 t^2$$
$$y(t) = b_0 + b_1 t + b_2 t^2 \quad (6)$$

dimana $a_{0,1,2}$ dan $b_{0,1,2}$ adalah parameter yang dicari dari tiga kali sampling data pengukuran posisi.

b. Perhitungan parameter model interpolasi

Setelah dipilih model dengan jumlah orde 2, maka untuk interpolasi/ekstrapolasi data diperlukan perhitungan data parameter model dengan menggunakan persamaan 7.

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)

$$y_o(t) = a_0 + a_1(kT) + a_2(kT)^2$$
$$y_o(t-t) = a_0 + a_1\big((k-1)T\big) + a_2\big((k-1)T\big)^2 \quad (7)$$
$$y_o(t-2t) = a_0 + a_1\big((k-2)T\big) + a_2\big((k-2)T\big)^2$$

Persamaan (7) dapat dibentuk menjadi persamaan (8)

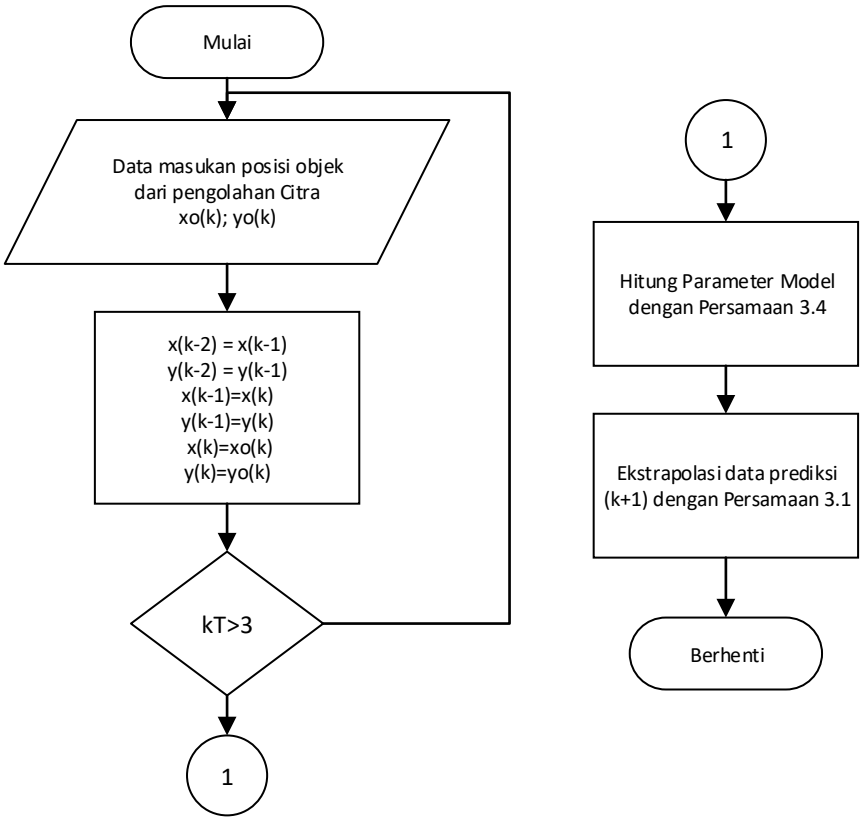
$$\begin{bmatrix} y_o(t) \\ y_o(t-ts) \\ y_o(t-2ts) \end{bmatrix} = \mathbf{H}\Big(kT, \big((k-1)T\big), \big((k-2)T\big)\Big) \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix};\ dimana$$
$$\mathbf{H}\Big(kT, \big((k-1)T\big), \big((k-2)T\big)\Big) = \begin{bmatrix} 1 & (kT) & (kT)^2 \\ 1 & \big((k-1)T\big) & \big((k-1)T\big)^2 \\ 1 & \big((k-2)T\big) & \big((k-2)T\big)^2 \end{bmatrix} \quad (8)$$

dimana $\mathbf{H}\Big(kT, \big((k-1)T\big), \big((k-2)T\big)\Big)$ adalah bukan matrik singular maka parameter $a_{0,1,2}$ dan $b_{0,1,2}$ dicari dengan persamaan (9)

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & kT & (kT)^2 \\ 1 & (k-1)T & ((k-1)T)^2 \\ 1 & (k-2)T & ((k-2)T)^2 \end{bmatrix}^{-1} \begin{bmatrix} x(k) \\ x(k-1) \\ x(k-2) \end{bmatrix}$$
$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 & kT & (kT)^2 \\ 1 & (k-1)T & ((k-1)T)^2 \\ 1 & (k-2)T & ((k-2)T)^2 \end{bmatrix}^{-1} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \end{bmatrix} \quad (9)$$

Dari persamaan diatas tahapan dariprediksi objek dapat diuraikan pada flowchar pada Gambar 6

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)



Gambar 6. Diagram alir untuk prediksi data kedepan dengan interpolasi linear

HASIL DAN PEMBAHASAN

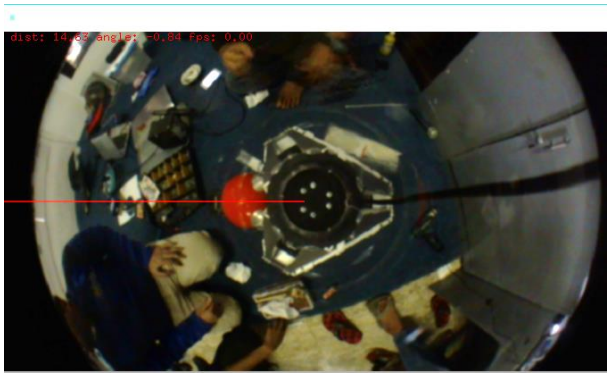
Pada peneltian ini hasil yang sementara didapatkan hanya pada proses *image capturing* dan *image processing.*

1. Pengujian Kamera *Omnidirectional* dan Pengambilan Citra

Pada pengujian ini bertujuan untuk mendapatkan ukuran frame yang tepat. Pengujian dilakukan dengan mengatur ukuran frame pada library ofVideoGrabber, dengan mengatur ukuran lebar dan tinggi *frame.*

Pada pengujian pengambilan citra, ukuran frame yang dapat mengambil citra lingkungan secara keseluruhan adalah 640x360 pixel, seperti yang ditunjukan Gambar 7. Pemilihan perbandingan ukuran tersebut dikarenakan kemampuan maksimal kamera dapat mengambil gambar adalah pada ukuran 1280x720 pixel yang memiliki perbandingan 16:9 antara lebar dan tinggi. Ketika pengambilan citra tidak memenuhi perbandingan tersebut hasil citra tidak akan mendaptkan citra lingkungan secara keseluruhan.

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)

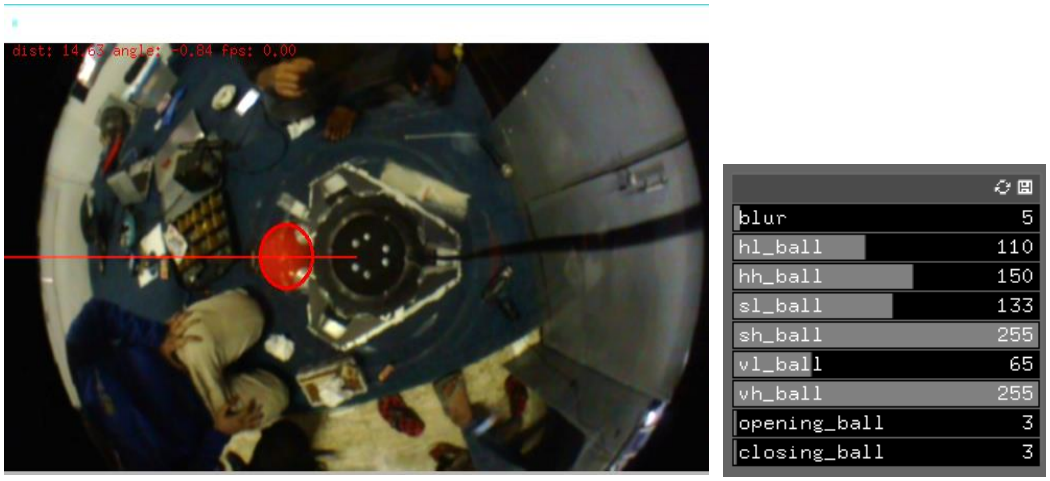


Gambar 7. Hasil Citra dengan Ukuran frame 640x360

2. Pengujian Pendeteksian Benda (Bola)

Pengujian ini dilakukan dengan melakukan pengaturan parameter pengolahan citra yang meliputi ukuran blur, batas atas dan bawah pada parameter warna untuk *hue, satruration,* dan *value*. Pengujian dilakukan pertamakali dengan mengatur ukuran *blur* yang sesuai agar noise dapat direduksi.

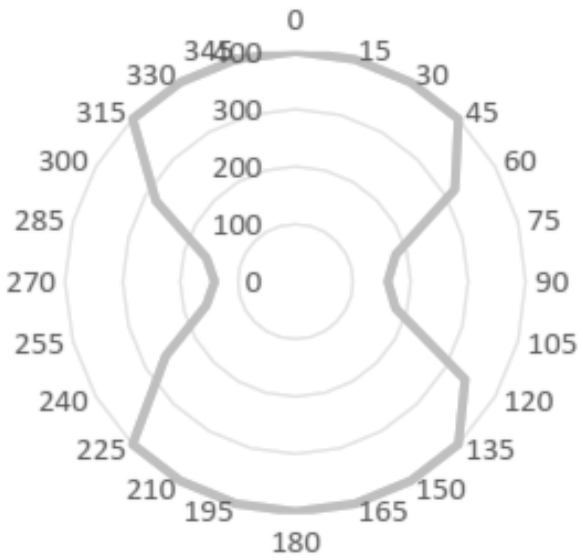
Setelah pengujian blur selesai, selanjutnya adalah pengaturan parameter batas atas dan bawah pada parameter warna untuk *hue, satruration,* dan *value.* Hasil terbaik diperoleh untuk mendeteksi bola adalah dengan ukuran blur 5pixel, *hue* 110-150, *saturation* 133-255, dan *value* 65-255 seperti yang ditunjukan Gambar 8.



Gambar 8. Hasil citra dalam mendeteksi Bola dan Nilai Parameter untuk dapat mendeteksi bola berwarna oranye.

Pengujian selanjutnya dilakukan untuk menguji seberapa luas jangkauan kamera *omnidirectional* untuk menjangkau bola seperti yang ditunjukan Gambar 9. Pengujian dilakukan dengan meletakan bola mulai dari jarak 23 cm hingga 414 cm dengan interfal 23 cm dan juga diuji dengan rentang sudut 0 hingga 180 dengan interfal 15 derajat.

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)



Gambar 9. Hasil uji jangkauan pendetreksian bola

Dari hasil pengujian menunjukan bahwa kamera dapat mendeteksi bola dengan jangkauan 23 cm hingga 400 cm. Namun pada sudut tertentu kamera tidak dapat mendeteksi bola karena kamera yang digunakan memiliki perbandingan 16:9, sehingga pada axis horizontal dari kamera terbatas.

3. Training model jarak dengan PSO-NN

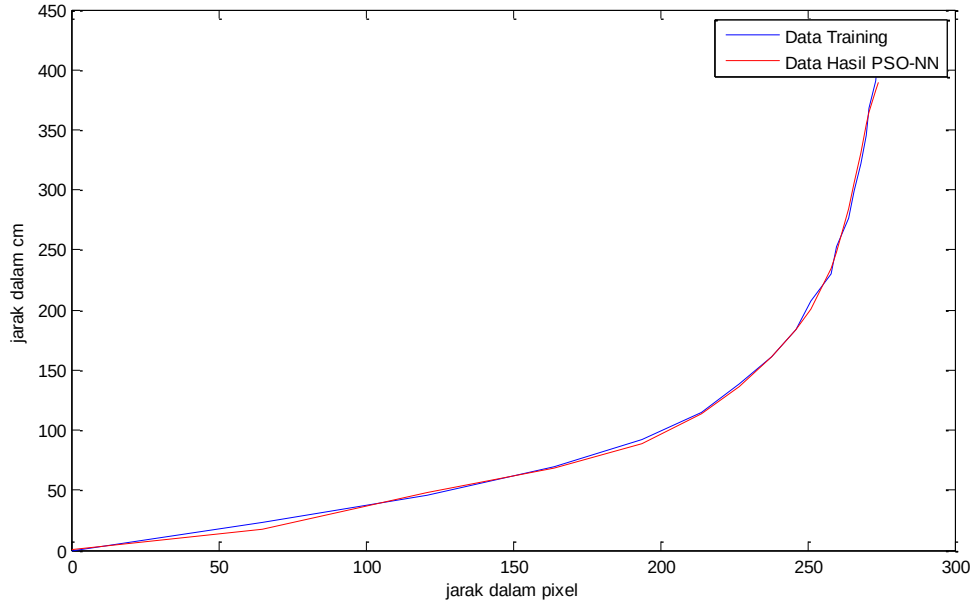
Setelah didapatkan data training yang dijabarkan pada sub bab **Error! Reference source not found.**, selanjutnya data tersebut ditraining menggunakan PSO untuk mendapatkan bobot bobot pada model NN. Proses training dengan PSO menggunakan parameter yang ditunjukan Tabel 1.

Tabel 1. Parameter PSO

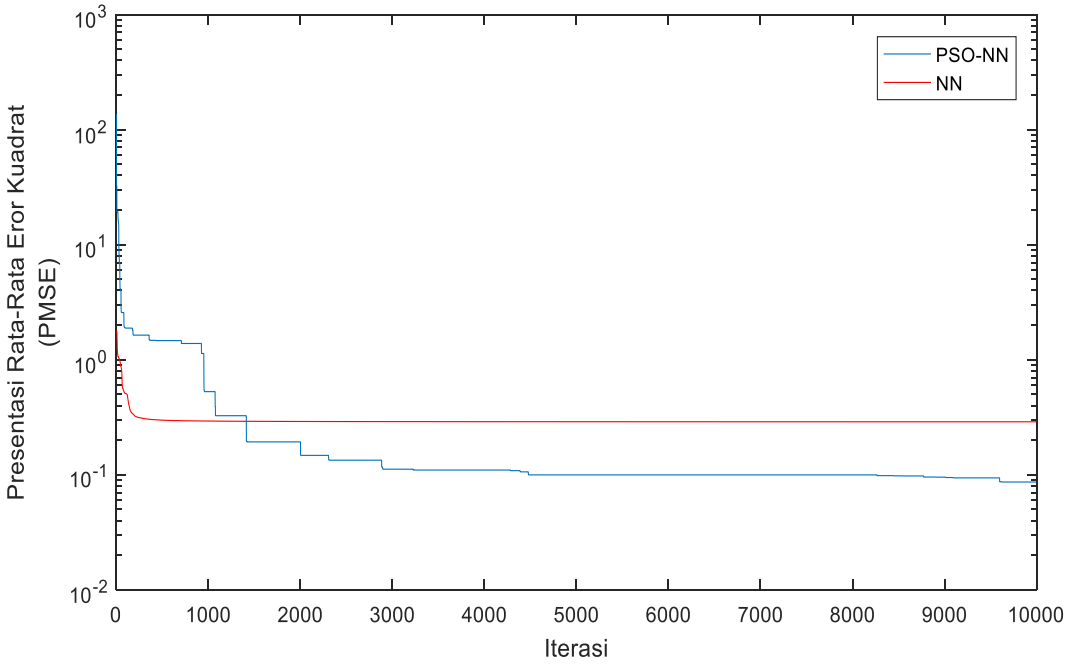
No	Parameter	Nilai
1	jumlah partikel	30
2	Rentang pencarian	0-10
3	Jumlah Iterasi	1000
4	w min	0.1
5	w max	1
6	c1	2
7	c2	2

Setelah dilakukan beberapa kali proses training hasil terbaik disajikan pada Gambar 10. Hasil tersebut menunjukan keakurasian pengukuran jarak bola yang akurat. Pada hasil training dengan PSO-NN didapatkan dengan presentasi rata rata kuadrat error (PMSE) yaitu 0.13 %. Hasil tersebut didapatkan dengan lama waktu iterasi untuk konvergen yaitu 400 iterasi. Jika dibandingkan dengan NN konvensional, lama konvergensi PSO-NN lebih lambat, namun hasil yang didapatkan lebih optimal seperti yang ditunjukan pada Gambar 11.

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)



Gambar 10. Grafik hasil perbandingan antara data training dan data hasil PSO-NN



Gambar 11. Grafik perbandingan nilai error pada PSO-NN dan NN konvensional

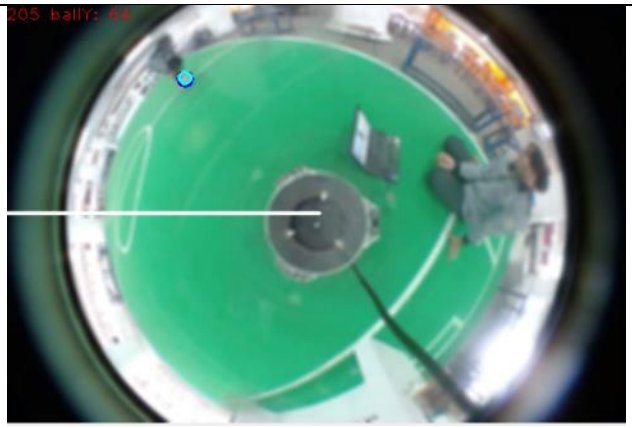
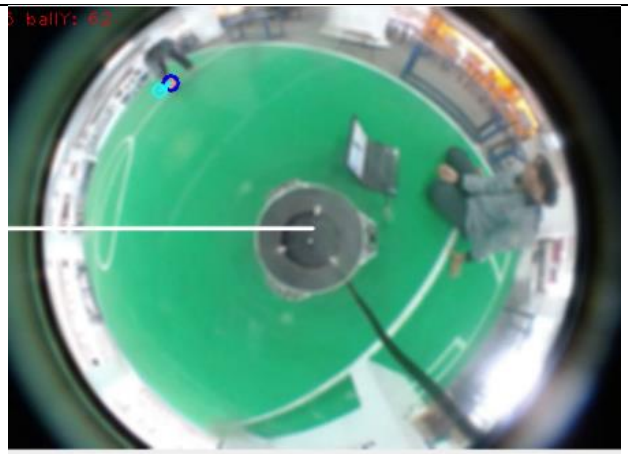
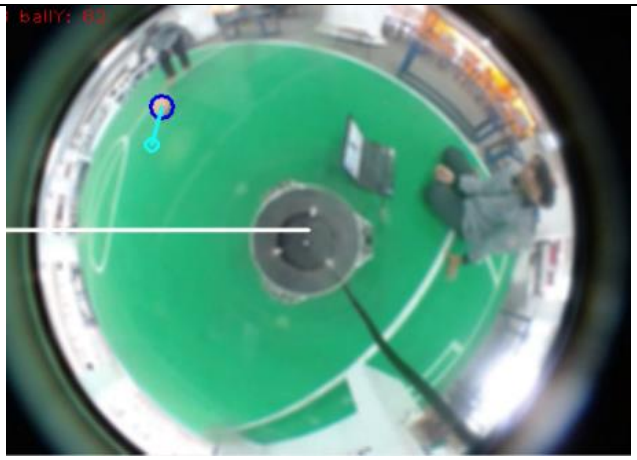
4. Pengujian Prediksi Objek Bergerak dengan Metode Interpolasi

Proses interpolasi dilakukan setelah pengujian pendeteksian benda dan pengujian pengukuran jaran benda dilakukan. Dari hasil yang didapatkan diketahui bahwa error rata-rata pendeteksian objek dan perhitungan jarak objek adalah 0.13%. Dengan hasil tersebut diharapkan pada prediksi objek tidak terjadi kesalahan prediksi yang besar dikarenakan perhitungan atau pengukuran jarak objek terdapat error yang besar. Pada pengujian prediksi dilakukan dengan menggerakan objek bergerak yang kemudian membandingkan hasil prediksi pada waktu t dibandingkan dengan posisi real pada waktu $t+1$ yang ditunjukan pada

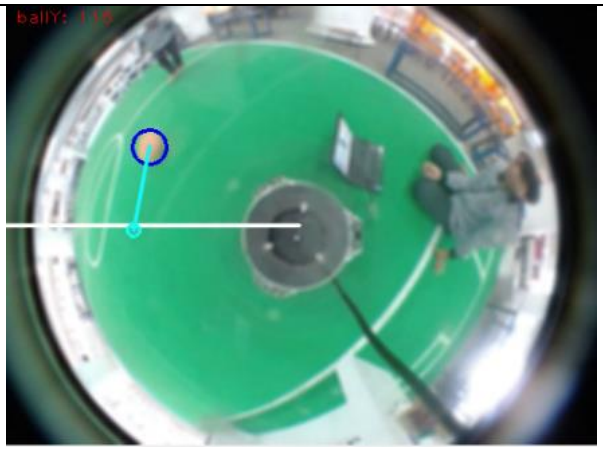
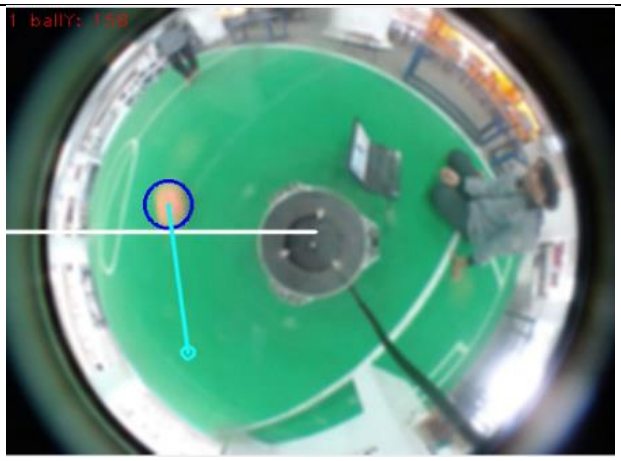
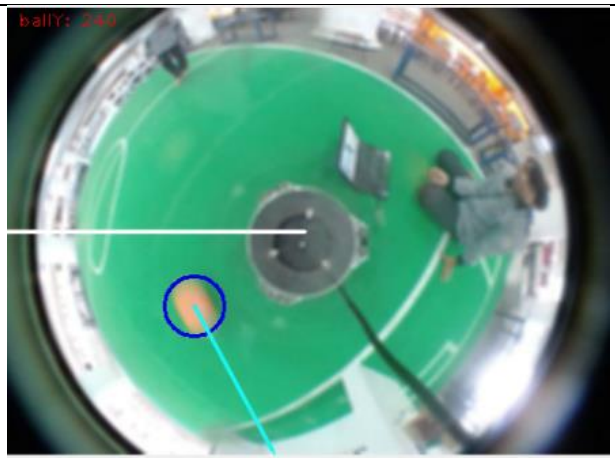
ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)

Tabel **2.**

Tabel 2. Hasil Prediksi Bola untuk waktu prediksi T+1

t	Gambar Posisi dan hasil Prediksi	Posisi (cm)	Prediksi (cm)
0		x=205 y=64	x=205 y=64
1		X=205 Y=62	X=205 Y=66
2		X=203 Y=82	X=195 Y=113

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)

3		X=198 Y=115	X=206 Y=160
4		X=201 Y=158	X=243 Y=250
5		X=231 Y=240	X=298 Y=317

Pada beberapa pengujian diperoleh nilai error rata-rata adalah 20% yang merupakan nilai error yang cukup untuk prediksi. Namun pada beberapa saat yaitu saat kecepatan objek tinggi prediksi memiliki nilai error yang cukup besar. Hal tersebut dikaernakan proses interpolasi yang bergantung pada sampling 3 data sebelumnya..

KESIMPULAN

Berdasarkan beberapa pengujian pada penelitian ini dapat disimpulkan bahwa Pada pengukuran jarak bola terhadap robot dengan menggunakan pengolahan citra dan PSO-NN didapatkan akurasi yang sangat baik dengan persentase rata rata kesalahan kuadrat pengukuran sebesar 0.13 %. Pada system prediksi dengan interpolasi polynomial akurasi prediksi didapatkan 20 % yang membutuhkan peningkatan system metode interpolasi agar akurasi meningkat.

ISSN: 1907-6223 (Print)
ISSN: 2579-3497 (Online)

UCAPAN TERIMA KASIH

Ucapan terimakasih dihaturkan kepada Direktorat Penelitian dan Pengabdian Masyarakat (DPPM) universitas Muhammadiyah Malang yang telah memberikan dukungan dan pendanaan pada penelitian ini. Terimakasih juga disampaikan kepada Tim Workshop Robotika UMM yang telah memberikan dukungan berupa sarana, prasarana, dan sumbangsih pemikiran terhadap penelitian ini

DAFTAR PUSTAKA

[1] R. Comasolivas, J. Quevedo, T. Escobet, A. Escobet, and J. Romera, "Low-level control of an omnidirectional mobile robot *," pp. 1160–1166, 2015.

[2] E. Engineering, I. Teknologi, and S. Nopember, "Adaptive Gaussian Parameter Particle Swarm Optimization And Its Implementation in Mobile Robot Path Planning," pp. 238–243, 2017.

[3] S. Barone, M. Carulli, P. Neri, A. Paoli, and A. Razionale, "An Omnidirectional Vision Sensor Based on a Spherical Mirror Catadioptric System," *Sensors*, vol. 18, no. 2, p. 408, 2018.

[4] D. B. Kusumawardhana and K. Mutijarsa, "Object recognition using multidirectional vision system on soccer robot," *2017 Int. Conf. Inf. Technol. Syst. Innov. ICITSI 2017 - Proc.*, vol. 2018-Janua, pp. 183–187, 2018.

[5] F. Weijia, L. Yuli, and C. Zuoliang, "Omnidirectional vision tracking and positioning for vehicles," *Proc. - 4th Int. Conf. Nat. Comput. ICNC 2008*, vol. 6, pp. 183–187, 2008.

[6] A. J. R. Neves, A. J. Pinho, D. A. Martins, and B. Cunha, "An efficient omnidirectional vision system for soccer robots: From calibration to object detection," *Mechatronics*, vol. 21, no. 2, pp. 399–410, 2011.

[7] P. Heinemann, J. Haase, and A. Zell, "A combined Monte-Carlo localization and tracking algorithm for RoboCup," *IEEE International Conference on Intelligent Robots and Systems*. pp. 1535–1540, 2006.

[8] Setiawardhana, R. Dikairono, T. A. Sardjono, and D. Purwanto, "Visual ball tracking and prediction with unique segmented area on soccer robot," *2017 Int. Semin. Intell. Technol. Its Appl. Strength. Link Between Univ. Res. Ind. to Support ASEAN Energy Sect. ISITIA 2017 - Proceeding*, vol. 2017-Janua, pp. 362–367, 2017.

[9] J. G. B and A. P. Engelbrecht, "Swarm Optimization Algorithms," vol. 1, pp. 350–357, 2016.

[10] Huanhai Xin, Deqiang Gan, Yun Liu, Liyuan Chen, and Lin Chen, "A Newton quadratic interpolation based control strategy for photovoltaic system," pp. 62–62, 2013.