

Navigasi *Mobile Robot* Untuk Mencapai *Docking Station* Dalam Proses *Autodocking*

Safriudin Rifandi

Universitas Hang Tuah Surabaya
safriudin.rifandi@hangtuah.ac.id

Received: Mei 2020; Accepted: Juni 2020; Published: Juli 2020

DOI: <http://dx.doi.org/10.30649/j-eltrik.v2i1.54>

Abstrak

Proses *autodocking* pada *mobile robot* merupakan sebuah kemampuan tambahan yang dapat dimiliki oleh sebuah robot. Untuk dapat melakukan proses tersebut, robot harus dapat mencapai area lokasi *docking station* dengan tepat. Dengan menggunakan metode *odometry*, robot dapat mengetahui posisi robot dari waktu ke waktu dengan membaca kecepatan dari masing-masing roda *mobile robot*. Dengan data kecepatan dari masing-masing roda, maka dengan perhitungan metode *odometry*, robot dapat mengetahui posisi terakhir dari robot. Selain itu, dengan menggunakan metode *behavior-based*, robot juga dapat menghindari halangan yang terdapat di area percobaan. Sedangkan untuk menuju ke arah *docking station* dengan tepat, digunakan kendali PI untuk menjaga *error* arah hadap *mobile robot* atau disebut *heading error mobile robot* sama dengan atau mendekati nol. Dari hasil percobaan, *mobile robot* dapat menghindari halangan yang ada dengan baik menggunakan metode *behavior-based*. Kendali PI dapat bekerja dengan baik untuk menjaga *heading error* mendekati nol sehingga *mobile robot* dapat menuju titik *docking station* dengan benar. Sedangkan metode *odometry* mampu membantu *mobile robot* untuk dapat mengetahui posisi robot dari waktu ke waktu sehingga robot dapat menuju *docking station* dengan tepat.

Kata kunci: *Autodocking, Odometry, Docking station, Behavior-based, Kendali PI*

Abstract

Autodocking process on a mobile robot is an additional capability that can be possessed by a robot. To be able to do this process, the robot must be able to reach the docking station location area properly. By using the odometry method, robot can find out the position of the robot from time to time by reading the speed of each wheel of the mobile robot. With the speed data from each wheel, the calculation of the odometry method, robot can determine the final position of the robot. In addition, by using behavior-based methods, robots can also avoid obstacles that are in the experiment area. Meanwhile, to get to the docking station appropriately, PI controls are used to keep the direction of the mobile robot's error or the so-called mobile robot's heading error equal to or close to zero. From the results of the experiment, the mobile robot can avoid obstacles well using behavior-based methods. PI controls can work well to keep heading errors close to zero so the mobile robot can get to the docking station point correctly. While the odometry

method is able to help the mobile robot to be able to know the position of the robot from time to time so that the robot can go to the docking station appropriately.

Key words: *Autodocking, Odometry, Docking station, Behavior-based, PI controls*

I. PENDAHULUAN

Perkembangan robotika di dunia saat ini sangat pesat. Banyak sekali robot yang telah dibuat dan beberapa dari robot tersebut sudah diproduksi secara massal untuk membantu kebutuhan manusia. Robot-robot tersebut membutuhkan sumber tegangan agar dapat bekerja. Sumber tegangan dapat berupa sumber tegangan AC atau sumber tegangan DC [1]. Khusus pada jenis *mobile robot*, robot tersebut biasanya menggunakan sumber tegangan DC atau biasa disebut baterai. Baterai memiliki kelemahan yaitu energi pada baterai akan berkurang saat digunakan. Oleh karena itu, pada level baterai tertentu, robot harus segera melakukan pengisian energi pada baterai agar dapat terus bekerja. Agar lebih efektif, maka robot harus dapat melakukan pengisian baterai secara mandiri atau otomatis. Pengisian baterai secara otomatis ini biasa disebut dengan *autodocking*.

Pada *autodocking* terdapat beberapa cara yang sering digunakan untuk mengisi baterai. Cara-cara tersebut antara lain adalah *docking station* [2], *catenary*, rel ketiga [3], *wireless*, dan penggantian baterai [4]. Dari cara-cara tersebut memiliki beberapa keunggulan dan kelemahan antara lain pada cara *docking station* memiliki keunggulan biayanya murah, bisa ditempatkan dimana saja dan membutuhkan area yang kecil. Sedangkan kelemahannya antara lain robot harus berhenti bekerja pada saat mengisi baterai [2]. Kemudian untuk *catenary* memiliki keunggulan yaitu robot dapat tetap berjalan hanya di wilayah tertentu yang terdapat saluran kabel listrik di atas robot. Untuk kelemahan *catenary* antara lain kabel

listrik yang digunakan untuk mengisi baterai rawan terkena manusia yang dapat menyebabkan sengatan listrik. Sedangkan pada cara rel ketiga keunggulan dan kelemahan sama dengan *catenary*. Perbedaan hanya terletak pada material yang digunakan untuk mengalirkan listrik saat mengisi baterai yaitu berupa sebuah rel yang berada di samping robot [3].

Kemudian untuk cara *wireless* memiliki keunggulan antara lain tidak menggunakan kabel dalam proses pengisian baterai. Sedangkan kelemahannya antara lain membutuhkan biaya yang cukup besar untuk mengaplikasikannya dan sistem pengisian dengan *wireless* ini cukup rumit. Untuk cara penggantian baterai memiliki keunggulan antara lain membutuhkan waktu yang singkat untuk melakukan proses ini karena baterai langsung diganti dengan yang masih penuh. Sedangkan kelemahannya antara lain baterai tidak dapat diisi kembali [4].

Dari keunggulan dan kelemahan yang dimiliki oleh cara-cara tersebut maka dipilihlah cara *docking station* untuk melakukan proses *autodocking* pada robot karena biayanya murah, bisa ditempatkan dimana saja dan membutuhkan area yang kecil. Agar dapat melakukan proses *autodocking* dengan cara *docking station*, robot harus dapat mencapai area lokasi *docking station* dengan tepat. Penelitian tentang *autodocking* ini telah dikembangkan oleh beberapa peneliti antara lain yang dilakukan oleh [5] menggunakan *scanning* area pemancar infra-merah untuk menentukan posisi *docking* dan membaca kode infra-merah yang dipancarkan *docking* untuk menentukan posisi tengah dari *docking station*. Kemudian penelitian oleh [6] juga menggunakan Sonar dan infra-

merah untuk menemukan tempat *docking station* serta menggunakan kompensasi kesalahan untuk menentukan posisi tengah *docking station*. Demikian juga penelitian yang dilakukan oleh [7] juga menggunakan metode *scanning* dengan *photodiode* dan *infrared* untuk menemukan lokasi *docking station* dan penelitian oleh [8] menggunakan dua buah *infrared* sebagai petunjuk bagi robot untuk menuju tempat *docking*.

Dari penelitian-penelitian sebelumnya terdapat beberapa kekurangan yaitu salah satunya menggunakan sistem yang cukup rumit dalam proses menuju lokasi *docking* serta hanya menggunakan satu buah *docking station*. Dari kekurangan tersebut maka perlu dibuat sebuah cara atau sistem untuk *autodocking* yang lebih efektif dan efisien dengan menggunakan metode yang lebih mudah serta menggunakan lebih dari satu *docking station*.

II. METODE PENELITIAN

Dalam penelitian ini diperlukan beberapa metode agar robot dapat men-capai area *docking station* dengan tepat. Metode yang digunakan antara lain kendali *behavior-based*, kendali PID dan metode *odometry*.

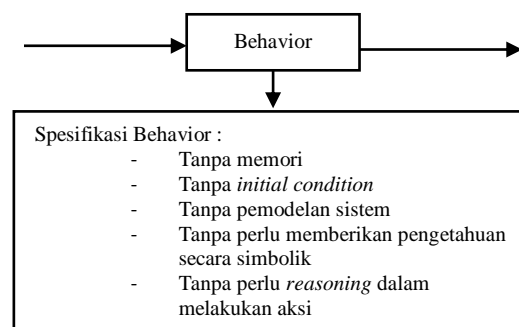
A. Kendali *Behavior-Based*

Robot yang menggunakan algoritma *high-level control* berbasis *behavior* (tabiat, kelakuan) biasa disebut sebagai *behavior-based robot*. Algoritma ini adalah salah satu metode dalam *high-level control* yang relatif banyak digunakan dan diuji coba dewasa ini. Metode *behavior-based* memiliki kelebihan antara lain mampu bekerja dengan baik dalam lingkungan yang dinamis. Misalnya untuk pengejaran objek pada ajang kontes *Robo-soccer* (Robot Sepak Bola) [9].

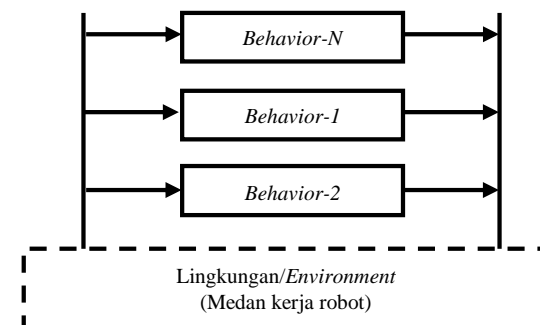
Algoritma *behavior-based* diturunkan dari sifat-sifat dasar makhluk hidup yang

bertingkah laku sesuai dengan keadaan lingkungannya. Berdasarkan informasi dari panca indera tertentu, syaraf yang berhubungan dengan otot gerakan badan terkait akan mendapat stimuli dari otak sehingga akan memberikan suatu respon yang khas. Seseorang akan dengan reflek menarik tangannya bila menyentuh bara api. Kita akan secara reflek menoleh bila seseorang memanggil nama kita. Termasuk, jika kita takut, kita akan mengambil langkah seribu jika kita dikejar anjing. Prinsip kerja algoritma *behavior-based* ditunjukkan oleh Gambar 1 [9].

Sedangkan struktur algoritma *behavior-based* yang dikenal sebagai arsitektur *subsumption* (*subsumption architecture*) dapat diilustrasikan seperti dalam Gambar 2 [9].



Gambar 1. Prinsip kerja algoritma *behavior-based* [9]



Gambar 2. Arsitektur subsumption *behavior-based control* [9]

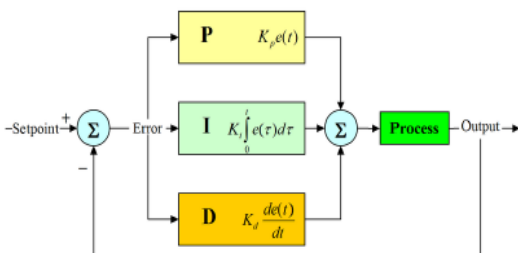
Behavior (1,2,3,...,N) didefinisikan sebagai tingkah laku robot “alami”. Misalnya seperti jalan maju hingga menemukan halangan, belok kanan, belok kiri, mencari obyek, menuju dan menguasai obyek, mundur dan belok bila melanggar dinding, dan segala “aksi” yang dapat kita definisikan sebagai sebagai bentuk “tabiat dasar makhluk” jika menjumpai suatu permasalahan [9].

B. Kendali PID (*Proportional Integral Derivative*)

Kendali *Proportional, Integral, Derivative* (PID) adalah sistem pengendali yang umum digunakan di industri maupun militer [10]. Sekitar 90% dari peralatan industri menggunakan pengendali PID karena mudah digunakan dan paling sederhana. Gambar 3 menunjukkan blok diagram kendali PID yang umum digunakan di industri. Kendali *Proportional* (P), *Integral* (I), dan *Derivative* (D) dapat digunakan bersamaan secara paralel ataupun digunakan terpisah dengan tidak menggunakan salah satu komponen P, I atau D [11].

Persamaan nilai keluaran dari kendali PID, dirumuskan seperti pada Persamaan (1) [11].

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) dt + K_d \frac{d}{dt} e(t) \quad (1)$$



Gambar 3. Blok diagram kendali PID [11]

Persamaan (1) menjelaskan bahwa nilai keluaran $u(t)$, merupakan jumlah dari *gain proportional* (K_p), *gain integral* (K_i), dan *gain derivative* (K_d) yang masing-

masing dipengaruhi oleh *error* (e) dalam selang waktu (t) tertentu [11].

B. Odometry

Odometry adalah penggunaan data dari pergerakan aktuator untuk memperkirakan perubahan posisi dari waktu ke waktu. *Odometry* digunakan untuk memperkirakan posisi relatif terhadap posisi awal. Untuk memperkirakan posisi relatif robot, digunakan perhitungan jumlah pulsa yang dihasilkan oleh sensor *rotary encoder* setiap satuan ukuran yang kemudian dikonversi menjadi satuan *milimeter*. Untuk mendapatkan jumlah pulsa setiap satu kali putaran roda digunakan rumus pada Persamaan (2) dan Persamaan (3) [12].

$$K_{roda} = 2 * \pi * r \quad (2)$$

$$Pulsa_per_mm = \frac{Resolusi\ encoder}{Keliling\ roda} \quad (3)$$

Pada sistem penggerak *differensial* terdapat dua roda, yaitu roda kanan dan roda kiri dan dimisalkan jumlah pulsa_per_mm untuk roda kanan adalah *right_encoder* dan roda kiri adalah *left_encoder* dan jarak antara dua roda adalah *wheel_base*, maka didapatkan jarak tempuh (*distance*) dan sudut orientasi (θ). Persamaan yang digunakan seperti pada Persamaan (4) dan (5) [12].

$$distance = \frac{(left_encoder+right_encoder)}{2} \quad (4)$$

$$\theta = \frac{(left_encoder-right_encoder)}{wheel_base} \quad (5)$$

Karena θ adalah sudut dalam radian, maka untuk mengetahui sudut dalam derajat (*heading*) digunakan Persamaan (6) [12].

$$heading = \theta * \frac{180}{\pi} \quad (6)$$

Dari Persamaan (9) didapatkan bahwa nilai *heading* akan bernilai negatif (-) ketika robot berputar melawan arah jarum jam dan akan bernilai positif (+) ketika robot berputar searah dengan jarum jam. Dengan mengetahui jarak dan sudut (*distance* dan θ), maka kita dapat mengetahui koordinat X dan koordinat Y dengan persamaan trigonometri seperti ilustrasi pada Gambar 4. Dari ilustrasi pada Gambar 4, maka koordinat dari robot dapat kita ketahui dengan menggunakan Persamaan (7) dan Persamaan (8) [12].

$$X_{pos} = distance * \sin(\theta) \quad (7)$$

$$Y_{pos} = distance * \cos(\theta) \quad (8)$$

Untuk menentukan *error* arah hadap dari robot terhadap titik tujuan, maka digunakan teorema *pythagoras* yang akan menghasilkan posisi (*path*) saat ini dan jarak terhadap titik tujuan dengan menggunakan Persamaan (9), (10) dan Persamaan (11) [12].

$$x = X_{tujuan} - X_{pos} \quad (9)$$

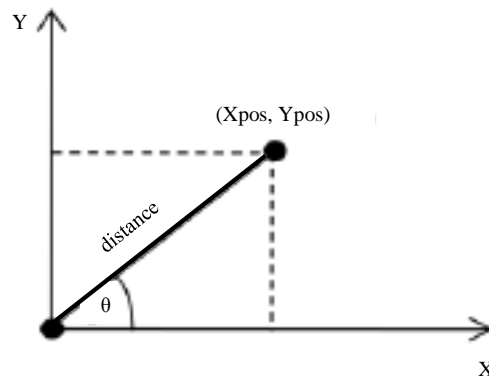
$$y = Y_{tujuan} - Y_{pos} \quad (10)$$

$$target_distance = \sqrt{x^2 + y^2} \quad (11)$$

Heading dari robot yang telah diketahui sehingga kita dapat menghitung *error* arah hadap (*heading error*) robot terhadap titik tujuan seperti ilustrasi pada Gambar 5 [12].

Pada Gambar 5 menunjukkan ilustrasi untuk mencari *heading error* (α) dimana β adalah *target bearing* yaitu sudut antara posisi robot saat ini terhadap titik tujuan. Sedangkan garis berwarna biru adalah garis bantu yang masing-masing sejajar dengan sumbu X dan sumbu Y. Untuk mendapat nilai dari β dan

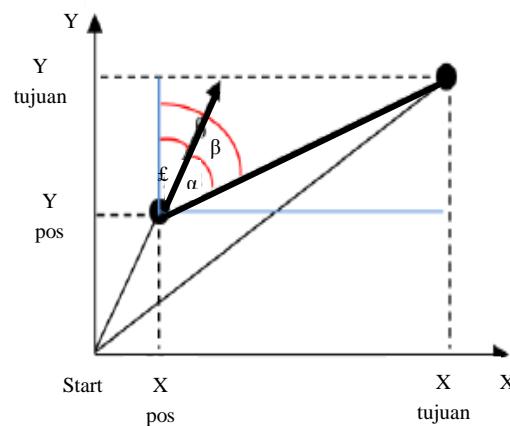
α , maka digunakan Persamaan (12) dan Persamaan (13) [12].



Gambar 4. Ilustrasi pada sumbu cartesian [12]

$$\beta = \arctan \frac{(Y_{tujuan} - Y_{pos})}{(X_{tujuan} - X_{pos})} \quad (12)$$

$$\alpha = \beta - \epsilon \quad (13)$$



Gambar 5. Sudut α , β dan ϵ [12]

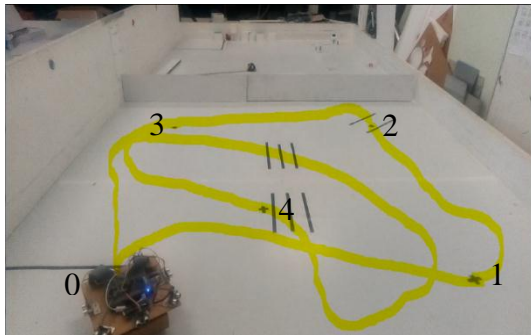
III. HASIL DAN PEMBAHASAN

Pada bagian ini, dibahas tentang hasil dari percobaan pencapaian *docking station* dalam proses *autodocking*. Percobaan dilakukan pada area percobaan *mobile robot* dalam beberapa skenario atau model pengujian antara lain

memerintah *mobile robot* untuk menuju ke titik tujuan yang telah ditentukan tanpa halangan, dengan 1 halangan, dengan jumlah halangan lebih dari 1, serta memerintah *mobile robot* untuk menuju ke titik tujuan yang telah ditentukan secara acak.

A. Pengujian Dengan Tanpa Menggunakan Halangan

Pada percobaan ini *mobile robot* diperintahkan untuk menuju ke 5 titik tujuan yang telah ditentukan secara berurutan seperti pada Gambar 6. Titik tujuan tersebut adalah (145, 0), (145, 145), (0, 145), (75, 75), (0, 0). Titik tujuan dianggap sebagai *docking station* dan pada Gambar 6 ditandai dengan tanda titik atau x dan diberi nomor 0-4.



Gambar 6. Pencapaian 5 titik *docking station* dengan tanpa halangan.

Pengujian pada Gambar 6 bertujuan untuk mengetahui kestabilan *mobile robot* mencapai *docking station* tanpa adanya halangan. Dari hasil pengujian didapatkan kesalahan yang tidak terlalu besar. Tabel pengujian pencapaian *docking station* tanpa halangan dapat dilihat pada Tabel 1. Dari hasil percobaan pada Tabel 1 dapat diketahui bahwa salah satu penyebab terjadinya kesalahan adalah slip pada roda. Slip yang terjadi pada roda tidak terbaca oleh *rotary encoder*. Selain itu, penyebab terjadinya kesalahan dapat juga disebabkan oleh pembatasan minimal posisi robot berhenti terhadap *docking station*. Robot akan berhenti dan menganggap lokasi *docking station* tercapai jika jarak robot dengan *docking station* (*target distance*) kurang dari delapan (8 cm). Selain itu, total *error* atau kumulatif *error* untuk percobaan *odometry* tanpa halangan cukup besar. Nilai total *error* tersebut adalah 38,26 cm.

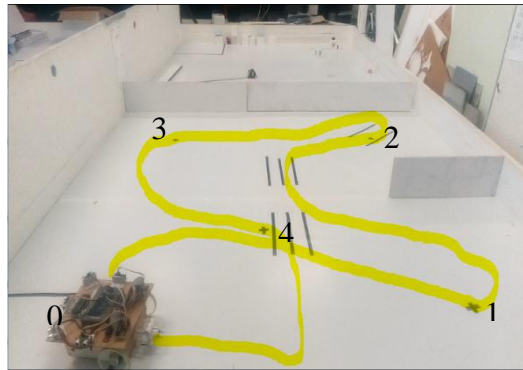
Sedangkan untuk rata-rata *error* memiliki nilai 7,65 cm. Nilai rata-rata dari kumulatif *error* tersebut dapat ditoleransi karena masih lebih kecil dibandingkan ukuran robot yang memiliki panjang 19,7 cm dan lebar 17 cm. Selain itu, saat berhenti di titik *docking station*, titik *docking station* selalu berada di bawah robot.

Tabel 1. Pengujian pencapaian *docking station* tanpa halangan

Titik ke-	Koordinat asli (x, y)	Koordinat robot (x, y)	Kesalahan koordinat (cm)		Kesalahan kumulatif (cm)
			x	y	
1	(145, 0)	(138, 7)	-7	7	9,90
2	(145, 145)	(138, 138)	-7	-7	9,90
3	(0, 145)	(3, 142)	3	-3	4,24
4	(75, 75)	(72, 79)	-3	4	5,00
5	(0, 0)	(7, 6)	7	6	9,22
Total <i>error</i>					38,26
Rata-rata <i>error</i>					7,65

B. Pengujian Dengan Menggunakan 1 Halangan

Pada percobaan ini *mobile robot* diperintahkan untuk menuju ke 5 titik tujuan yang telah ditentukan dengan terdapat satu buah halangan pada area percobaan secara berurutan. Titik tujuan tersebut adalah (145, 0), (145, 145), (0, 145), (75, 75), (0, 0). Titik tujuan dianggap sebagai *docking station* dan ditandai dengan tanda titik atau x serta diberi nomor 0-4. Gambar 7 menunjukkan hasil pencapaian *docking station* serta penghindaran halangan oleh *mobile robot*.



Gambar 7. Pencapaian 5 titik *docking station* dengan 1 halangan.

Pengujian pada Gambar 7 bertujuan untuk mengetahui kestabilan *mobile robot* mencapai *docking station* dengan adanya 1 halangan.

Tabel pengujian pencapaian *docking station* dengan 1 halangan dapat dilihat pada Tabel 2.

Tabel 2. Pengujian pencapaian *docking station* dengan 1 halangan

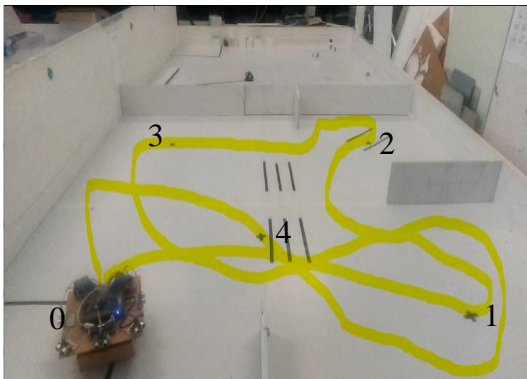
Titik ke-	Koordinat asli (x, y)	Koordinat robot (x, y)	Kesalahan (<i>Error</i>) (cm)		Kesalahan kumulatif (cm)
			x	y	
1	(145, 0)	(142, 6)	-3	6	6,71
2	(145, 145)	(139, 139)	-6	-6	8,49
3	(0, 145)	(11, 143)	11	-2	11,18
4	(75, 75)	(63, 77)	-12	2	12,17
5	(0, 0)	(11, -2)	11	-2	11,18
Total <i>error</i>					49,72
Rata-rata <i>error</i>					9,94

Dari hasil percobaan pada Tabel 2 dapat diketahui bahwa terjadi kesalahan yang cukup besar di bandingkan saat tanpa halangan. Selain itu, total *error* atau kumulatif *error* untuk percobaan *odometry* dengan satu halangan cukup besar. Nilai total *error* tersebut adalah 49,72 cm. Sedangkan untuk rata-rata *error* memiliki nilai 9,94 cm. Nilai rata-rata dari kumulatif *error* tersebut dapat ditoleransi karena masih lebih kecil dibandingkan ukuran robot yang memiliki panjang 19,7 cm dan lebar 17 cm. Selain itu, posisi robot saat mencapai *docking station* selalu berada di atas titik *docking station*.

C. Pengujian Dengan Jumlah Halangan Lebih Dari 1

Pada percobaan ini *mobile robot* diperintahkan untuk menuju ke 5 tujuan yang telah ditentukan secara berurutan dengan beberapa buah halangan. Titik tujuan tersebut adalah (145, 0), (145, 145), (0, 145), (75, 75), (0, 0). Titik tujuan dianggap sebagai *docking station* dan ditandai dengan tanda titik atau x serta diberi nomor 0-4. Gambar 8 menunjukkan hasil pencapaian *docking station* serta penghindaran halangan oleh *mobile robot*. Pengujian pada Gambar 8 bertujuan untuk mengetahui kestabilan *mobile robot*

mencapai *docking station* dengan adanya banyak halangan.



Gambar 8. Pencapaian 5 titik *docking station* dengan banyak halangan.

Tabel pengujian pencapaian *docking station* dengan banyak halangan dapat dilihat pada Tabel 3. Dari hasil percobaan pada Tabel 3 dapat diketahui bahwa terjadi kesalahan yang hampir mirip dengan saat tanpa halangan. Selain itu, total *error* atau kumulatif *error* untuk percobaan *odometry* dengan banyak halangan cukup besar. Nilai total *error* tersebut adalah 30,15 cm.

Sedangkan untuk rata-rata *error* memiliki nilai 6,03 cm. Nilai rata-rata dari kumulatif *error* tersebut dapat ditoleransi karena masih lebih kecil dibandingkan ukuran robot yang memiliki panjang 19,7 cm dan lebar 17 cm. Selain itu, posisi robot saat mencapai *docking station* selalu berada di atas titik *docking station*.

Tabel 3. Pengujian pencapaian *docking station* dengan banyak halangan

Titik ke-	Koordinat asli (x, y)	Koordinat robot (x, y)	Kesalahan (<i>Error</i>) (cm)		Kesalahan kumulatif (cm)
			x	y	
1	(145, 0)	(142, 2)	-3	7	7,62
2	(145, 145)	(143, 141)	-2	-4	4,47
3	(0, 145)	(1, 142)	1	-3	3,16
4	(75, 75)	(71, 72)	-3	-4	5,00
5	(0, 0)	(7, -7)	7	-7	9,90
Total <i>error</i>					30,15
Rata-rata <i>error</i>					6,03

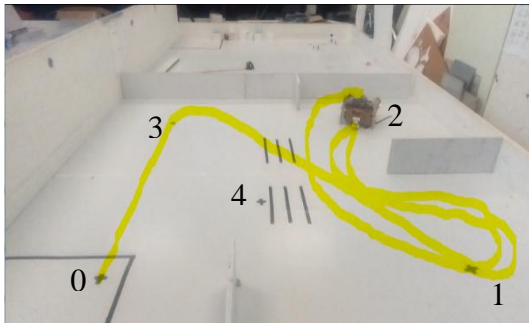
D. Pengujian 5 Titik *docking station* Yang Ditentukan Secara Acak

Pada percobaan ini *mobile robot* diperintahkan untuk menuju ke 5 titik tujuan yang telah ditentukan dengan beberapa buah halangan pada area percobaan. Titik tujuan tersebut adalah (145, 0), (145, 145), (0, 145), (75, 75), (0, 0). Titik tujuan dianggap sebagai *docking station* dan ditandai dengan tanda titik atau x serta diberi nomor 0-4. Akan tetapi robot akan memilih secara acak titik mana yg akan dituju oleh *mobile robot*. Gambar 9 menunjukkan hasil pencapaian *docking*

station serta penghindaran halangan oleh *mobile robot*.

Pengujian pada Gambar 9 bertujuan untuk mengetahui kestabilan *mobile robot* mencapai *docking station* secara acak dengan adanya banyak halangan. Tabel pengujian pencapaian *docking station* secara acak dengan banyak halangan dapat dilihat pada Tabel 4. Dari hasil percobaan pada Tabel 4 dapat diketahui bahwa terjadi kesalahan yang hampir mirip dengan saat tanpa halangan. Selain itu, total *error* atau kumulatif *error* untuk percobaan *odometry* dengan banyak halangan dan pemilihan *docking station* secara acak

cukup besar. Nilai total *error* tersebut adalah 30,21 cm.



Gambar 9. Pencapaian 5 titik *docking station* secara acak dengan banyak halangan.

Nilai rata-rata dari kumulatif *error* tersebut dapat ditoleransi karena masih lebih kecil dibandingkan ukuran robot yang memiliki panjang 19,7 cm dan lebar 17 cm. Selain itu, posisi robot saat mencapai *docking station* selalu berada di atas titik *docking station*.

IV. SIMPULAN

Setelah dilakukan percobaan pencapaian *docking station* dengan menggunakan 5 titik *docking station* dan beberapa halangan serta dalam mode acak ataupun berurutan, maka dapat diambil beberapa kesimpulan. Kesimpulan tersebut adalah *mobile robot* dapat menghindari halangan yang ada dengan baik menggunakan metode *behavior-based*.

Kendali PI dapat bekerja dengan baik untuk menjaga *heading error* mendekati nol sehingga *mobile robot* dapat menuju titik *docking station* dengan benar. Selain karena slip pada roda, kesalahan yang terjadi dalam pencapaian *docking station* dengan metode *odometry* dapat juga disebabkan oleh pembatasan minimal posisi robot berhenti terhadap *docking station*.

Tabel 4. Pengujian pencapaian *docking station* secara acak dengan banyak halangan

Titik ke-	Koordinat asli (x, y)	Koordinat robot (x, y)	Kesalahan (<i>Error</i>) (cm)		Kesalahan kumulatif (cm)
			x	y	
1	(0, 145)	(1, 140)	1	-5	5,10
2	(145, 145)	(139, 144)	-6	-1	6,08
3	(145, 0)	(147, -2)	2	-2	2,83
4	(145, 145)	(144, 132)	-1	-13	13,04
5	(145, 0)	(146, 3)	1	3	3,16
Total <i>error</i>					30,21
Rata-rata <i>error</i>					6,04

V. RUJUKAN

- [1] S. Rifandi, I. A. Sulistijono, S. Kuswadi, "Pencarian Jalur Terpendek Dalam Proses *Autodocking* Pada *Mobile Robot* Menggunakan *Artificial Bee Colony*", *Jurnal Inovtek Polbeng*, Vol. 9, 2019, pp. 204-213.
- [2] C. Ricaud and P. Vollet, "Connection Method For Charging Systems A Key Element For Electric Vehicle", *Schneider Electric Conference, France*, pp. 1-11, 2010.

- [3] H. Maghfiroh. (view Maret. 2020). "Dunia kereta-sistem propulsi KRL"[Online]. Available: <http://www.keretalistrik.com/2016/07/dunia-kereta-sistem-propulsi-krl.html>.
- [4] Sham Shui Po, Hong Kong, "Typical Electric Vehicle Charging Methods", CLP Power, 2013.
- [5] K. H. Kim, H. D. Choi, S. Yoon, K. W. Lee, H. S. Ryu, C. K. Woo, and Y. K. Kwak, "Development Of Docking System For Mobile Robots Using Cheap Infrared Sensors", *1st International Conference on Sensing Technology, New Zealand*, pp. 287-291, 2005.
- [6] S. Roh, J. H. Park, Y. H. Lee, Y. K. Song, K. W. Yang, M. Choi, H. S. Kim, H. Lee, and H. R. Choi, "Flexible Docking Mechanism With Error Compensation Capability For Auto Recharging System Of Mobile Robot", *International Journal of Control, Automation, and Systems*, Vol. 6, pp. 731-739, 2008.
- [7] R. Purwanto and D. Purwanto, "Rancang Bangun Sistem Autodocking Untuk Pengisian Baterai Pada Mobile Robot", *Surabaya, Indonesia : Institut Teknologi Sepuluh Nopember Surabaya*, 2009.
- [8] K. Varun Raj, K. Patil, D. V. K. Kariappa, and A. M. Jakati, "A Beacon-Based Docking System For An Autonomous Mobile Robot", *National Conference on Mechanisms and Machines (NaCoMM07), India*, pp. 1-7, 2007.
- [9] M. Ashary, I. A. Sulistijono, Z. Darojah, "Proyek Mobile Robot Penghindar Halangan: Sistem Navigasi Pada Mobile Robot Menggunakan Metode Behavior-Based", *Jurnal Mekanika dan Energi PENS*, Vol. 3, pp. 1-10, 2014.
- [10] Katsuhiko O., *Modern Control Engineering*, Pearson Education, 5th Edition, 2010.
- [11] D. W. Wardhana, A. Wahyudi, dan H. Nurhadi, "Perancangan Sistem Kontrol PID Untuk Pengendali Sumbu Azimuth Turret Pada Turret-Gun Kaliber 20 Mm", *Jurnal Teknik ITS*, Vol. 5, pp. 512-516. 2016.
- [12] B. S. Marta, F. Ardilla, dan A. R. A. Besari, "Path Tracking Pada Mobile Robot Dengan Umpan Balik Odometry", *Proceedings of IES 2011 – Emerging Technology For Better Human Life, Indonesia*, pp. 1-8, 2011.