

# Implementasi Metode *Behavior-Based*, *Odometry* Dan *Pid* Pada *Mobile Robot* Dalam Proses *Autodocking* Untuk Mengisi Baterai

Safriudin Rifandi

Program Studi Teknik Elektro Universitas Hang Tuah Surabaya

Jl, Arief Rahman Hakim 150 Keputih Sukolilo Surabaya

(031)5945864, 50945894

E-mail: safriudin.rifandi@hangtuah.ac.id

**Abstrak**—Kendali *behavior-based* merupakan salah satu metode yang relatif banyak digunakan dalam kendali *mobile robot*. *Behavior-based* sendiri merupakan salah satu metode yang didasarkan pada perilaku makhluk hidup yang mengacu pada masukan yang didapat dari lingkungan sekitar. Dengan menggunakan metode *behavior-based*, *mobile robot* dapat menghindari halangan yang ada di sekitar robot. Selain menggunakan metode *behavior-based*, dalam penelitian ini juga digunakan kendali *PID* yang berfungsi untuk menstabilkan pergerakan *mobile robot* saat menuju ke titik tujuan yang telah ditentukan. Dengan menggunakan kendali *PID*, *heading error mobil robot* terhadap titik tujuan diatur agar selalu sama dengan nol atau mendekati nol. *Heading error* inilah yang menjadi masukan bagi sistem gerak *mobile robot* untuk melakukan pergerakan sehingga *mobile robot* dapat menuju ke titik tujuan. Implementasi metode *behavior-base* dan *PID* ini cukup bagus diterapkan pada kendali *mobile robot*. *Mobile robot* dapat menghindari halangan dan tetap dapat menuju ke titik tujuan yang ditentukan dengan *error* yang tidak terlalu besar. Aplikasi penelitian ini nantinya dapat diterapkan pada *mobile robot* dalam proses *autodocking* untuk melakukan pengisian baterai secara otomatis.

**Keywords**—*behavior-based*, kendali *PID*, *heading error*, motor *DC*, *autodocking*.

## I. PENDAHULUAN

Keunggulan dalam teknologi robotika tak dapat dipungkiri telah lama dijadikan ikon kebanggaan Negara-negara maju di dunia [1]. Banyak sekali robot yang telah dibuat dan beberapa dari robot tersebut sudah diproduksi secara massal untuk membantu kebutuhan manusia. Di Indonesia sendiri perkembangan robotika juga sudah mulai meningkat meskipun belum seperti negara-negara maju. Jenis robot yang dibuatpun bermacam-macam antara lain *non-Mobile robot*, *mobile robot*, robot terbang, robot bawah air dan robot *Humanoid* [2]. Salah satu jenis robot yang ada di Indonesia adalah *mobile robot*. *Mobile robot* adalah jenis robot yang dapat berpindah tempat dari satu titik ke titik yang lain dengan menggunakan alat gerak berupa roda. Salah satu hal yang terpenting pada sebuah *mobile robot* adalah masalah pergerakan *mobile robot*. Dalam melakukan pergerakan, *mobile robot* harus dapat menghindari halangan yang ada di lingkungan. Selain itu, robot juga harus tetap dapat menuju ke titik tujuan yang telah ditentukan [3].

Pada penelitian ini, telah dibuat sebuah sistem navigasi robot yang dapat menghindari halangan serta robot masih tetap dapat menuju ke titik tujuan yang telah ditentukan. Sistem pergerakan yang digunakan adalah menggunakan metode *behavior-based*. *Behavior-based* sendiri merupakan salah satu metode yang didasarkan pada perilaku makhluk hidup yang mengacu pada masukan yang didapat dari

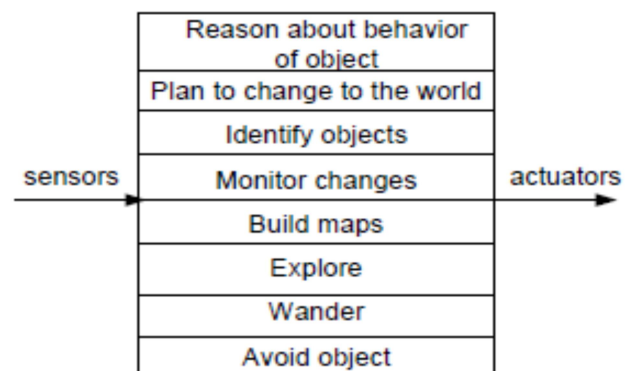
lingkungan sekitar [3]. Selain itu, pada penelitian ini juga digunakan metode odometri untuk mengetahui posisi robot serta kendali *PID* yang berfungsi untuk menstabilkan pergerakan *mobile robot* saat menuju ke titik tujuan yang telah ditentukan. Dengan menggunakan kendali *PID*, *heading error mobil robot* terhadap titik tujuan diatur agar selalu sama dengan nol atau mendekati nol. *Heading error* inilah yang menjadi masukan bagi sistem gerak *mobile robot* dalam hal ini motor *DC* kiri dan motor *DC* kanan untuk melakukan pergerakan sehingga *mobile robot* dapat menuju ke titik tujuan. *Mobile robot* ini dilengkapi dengan sensor ultrasonik untuk mendeteksi adanya halangan serta sensor *IMU* untuk mengetahui arah hadap *mobile robot*.

## II. BAHAN DAN METODE

Pada bagian ini, akan dijelaskan metode dan algoritma yang digunakan. Metode dan algoritma yang digunakan adalah kendali *behavior-based*, odometri dan kendali *PID*.

### A. Kendali *Behavior-Based*

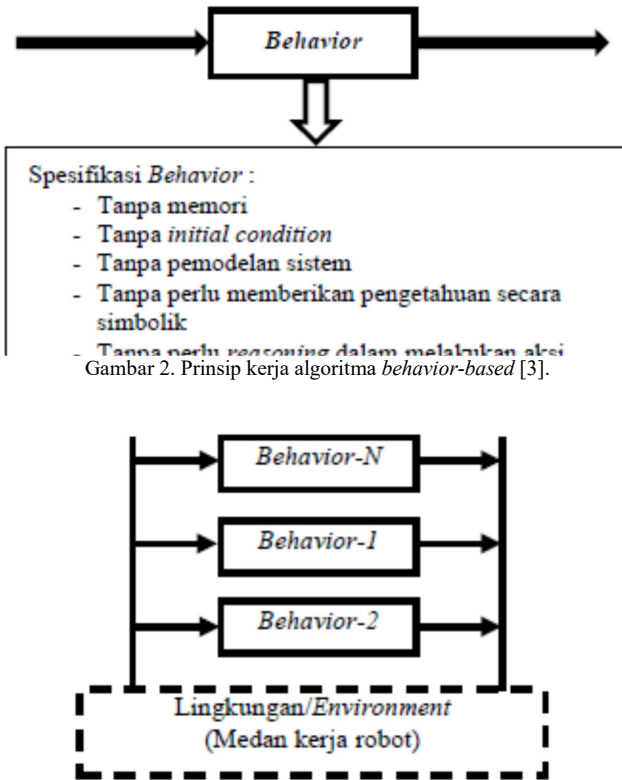
*Behavior-based Control* mendesain sistem kontrol robot menggunakan pendekatan *task achieving behaviors* (perilaku menunaikan tugas) sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Dekomposisi sistem kontrol *mobile robot* dengan *task achieving behavior* [4].

Pendekatan *behavior-based* menggunakan sekumpulan *behavior* yang tersusun secara paralel yang bereaksi terhadap lingkungan yang dihadapi dan menghasilkan respon yang sesuai dengan tujuan *behavior* tersebut [4]. Robot yang menggunakan algoritma *high-level control* berbasis *behavior* (tabiat, kelakuan) biasa disebut sebagai *behavior-based robot*. Metode *behavior-based* memiliki kelebihan antara lain mampu bekerja dengan baik dalam lingkungan yang dinamis. Prinsip kerja algoritma *behavior-based* ditunjukkan pada Gambar 2. Sedangkan struktur algoritma *behavior-based*

yang dikenal sebagai arsitektur *subsumption* (*subsumption architecture*) dapat diilustrasikan seperti pada Gambar 3 [3].



Gambar 3. Arsitektur *subsumption* behavior-based control [3].

Gambar 3. menjelaskan bahwa *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 bentuk “tabiat dasar makhluk” jika menjumpai suatu permasalahan [3].

### B. Odometri

*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 *millimeter* [5]. Gambar 4. menunjukkan ilustrasi *rotary encoder*. Sedangkan Gambar 5. menunjukkan contoh desain robot yang menggunakan odometri dan parameter-parameter yang digunakan pada odometri. Untuk mendapatkan jumlah pulsa setiap satu kali putaran roda digunakan persamaan (1) dan persamaan (2) sebagai berikut:

$$K \text{ roda} = 2\pi r \quad (1)$$

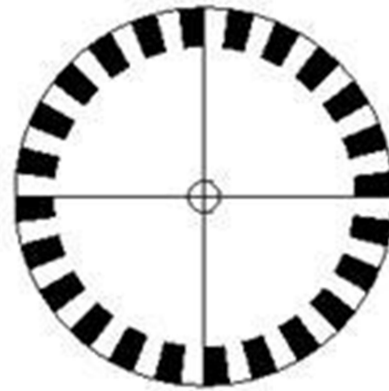
$$\text{pulsa\_per\_mm} = \text{resolusi\_enc} / K \text{ roda} \quad (2)$$

Pada sistem penggerak diferensial 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*)

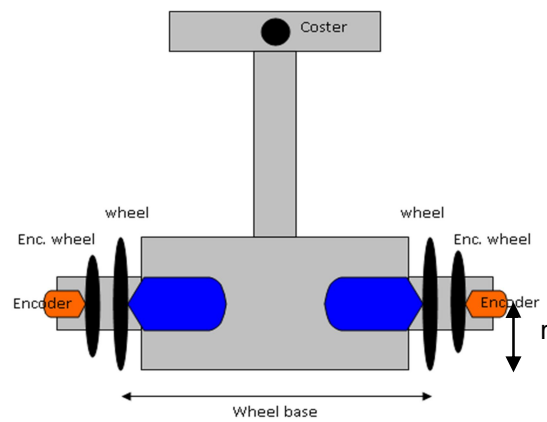
dan sudut orientasi ( $\theta$ ) dalam satuan radian dengan persamaan (3) dan persamaan (4) berikut [5].

$$\text{distance} = (\text{left\_enc} + \text{right\_enc}) / 2 \quad (3)$$

$$\theta = (\text{left\_enc} - \text{right\_enc}) / \text{wheel\_base} \quad (4)$$

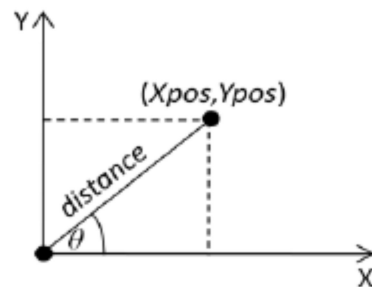


Gambar 4. Ilustrasi piringan *rotary encoder* [7].



Gambar 5. Contoh desain robot [5].

Dari persamaan (3) dan persamaan (4) 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  $\theta$ ) maka kita dapat mengetahui koordinat X dan koordinat Y dengan persamaan trigonometri seperti ilustrasi pada Gambar 6 [5].



Gambar 6. Ilustrasi pada sumbu *Cartesian* [5].

Dari ilustrasi pada Gambar 6. maka koordinat dari robot dapat kita ketahui dengan persamaan (5) dan persamaan (6) berikut [5].

$$X_{pos} = \text{distance} \sin(\theta) \quad (5)$$

$$Y_{pos} = \text{distance} \cos(\theta) \quad (6)$$

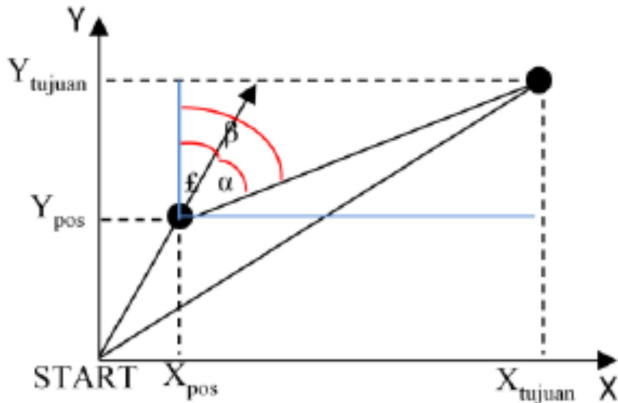
Untuk menentukan *error* arah hadap dari robot terhadap titik tujuan maka digunakan teorema *pythagoras* yang akan menghasilkan posisi (*path*) saat ini dan jarak terhadap titik tujuan seperti persamaan (7) sampai persamaan (9) berikut [5].

$$x = X_{tujuan} - X_{pos} \quad (7)$$

$$y = Y_{tujuan} - Y_{pos} \quad (8)$$

$$target\_distance = \sqrt{x^2 + y^2} \quad (9)$$

*Heading* dari robot yang telah diketahui sehingga kita dapat menghitung *error* arah hadap (*heading error*) robot terhadap titik tujuan dengan persamaan (11) berikut serta ilustrasi yang ditunjukkan pada Gambar 7 [5].



Gambar 7. Ilustrasi sudut  $\alpha$ ,  $\beta$ , dan  $\epsilon$  [5].

Pada Gambar 7. menunjukkan ilustrasi untuk mencari *heading error* ( $\alpha$ ) dimana  $\beta$  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  $\beta$ , maka digunakan persamaan (10) sebagai berikut [5].

$$\beta = \arctan \frac{(Y_{tujuan} - Y_{pos})}{(X_{tujuan} - X_{pos})} \quad (10)$$

Sehingga *heading error* dapat dicari dengan menggunakan persamaan (11) berikut [5].

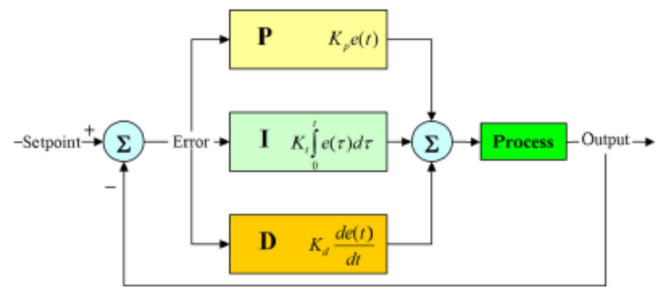
$$\alpha = \beta - \epsilon \quad (11)$$

### C. Kontrol PID (*Proporsional, Integral, Derivative*)

Kontrol *Proportional, Integral, Derivative* (*PID*) adalah sistem pengendali yang umum digunakan di industri maupun militer. Sekitar 90% dari peralatan industri menggunakan pengendali *PID* karena mudah digunakan dan paling sederhana [6]. Gambar 8. menunjukkan blok diagram kontrol *PID* yang umum digunakan di industri. Sedangkan persamaan nilai keluaran dari kontrol *PID* adalah seperti persamaan (12) berikut [6].

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) dt + K_d \frac{d}{dt} e(t) \quad (12)$$

Persamaan (12) 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 [6].



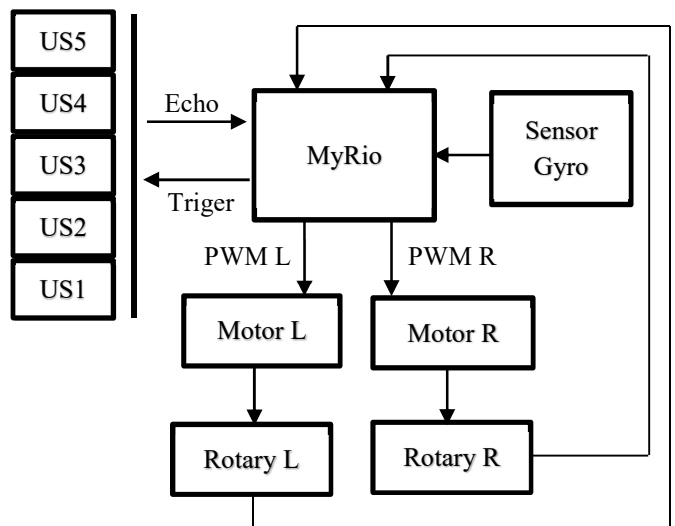
Gambar 8. Blok diagram kontrol *PID* [6].

## III. PERANCANGAN SISTEM

Pada bagian ini akan dijelaskan bagaimana perancangan sistem dari *mobile robot* yang digunakan serta desain dari *mobile robot*.

### A. Blok Diagram Sistem

Secara keseluruhan diagram blok sistem dari dari *mobile robot* ini ditunjukkan pada Gambar 9.

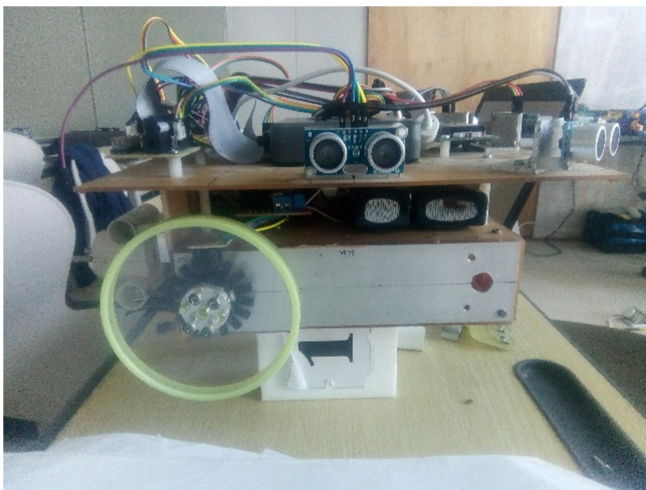


Gambar 9. Diagram blok sistem secara keseluruhan.

Pada Gambar 9. Dapat dilihat bahwa *mobile robot* ini memiliki beberapa sensor yang berfungsi sebagai masukan pada controller. Sensor-sensor tersebut adalah sensor ultrasonik, sensor *gyro*, dan sensor *rotary*. *Mobile robot* ini menggunakan MyRio sebagai controller utama. Sedangkan untuk mendeteksi adanya halangan, *mobile robot* ini menggunakan lima (5) buah sensor ultrasonik. Untuk pembacaan sensor ultrasonik dilakukan secara acak dengan menghindari pembacaan sensor secara berurutan pada letak sensor yang bersebelahan. Selain itu, untuk mengetahui arah hadap robot, *mobile robot* ini menggunakan sensor *gyro*. Sedangkan untuk mengetahui posisi relatif robot, digunakan dua (2) buah *rotary encoder* untuk membaca perubahan pergerakan pada roda kanan dan roda kiri sehingga posisi relatif dapat dihitung. Seluruh data-data sensor tersebut diolah oleh MyRio dengan menggunakan metode *behavior-base* dan metode *odometry* sehingga *mobile robot* dapat

mengambil tindakan sesuai masukan dari sensor ultrasonik, sensor *gyro*, dan sensor *rotary*.

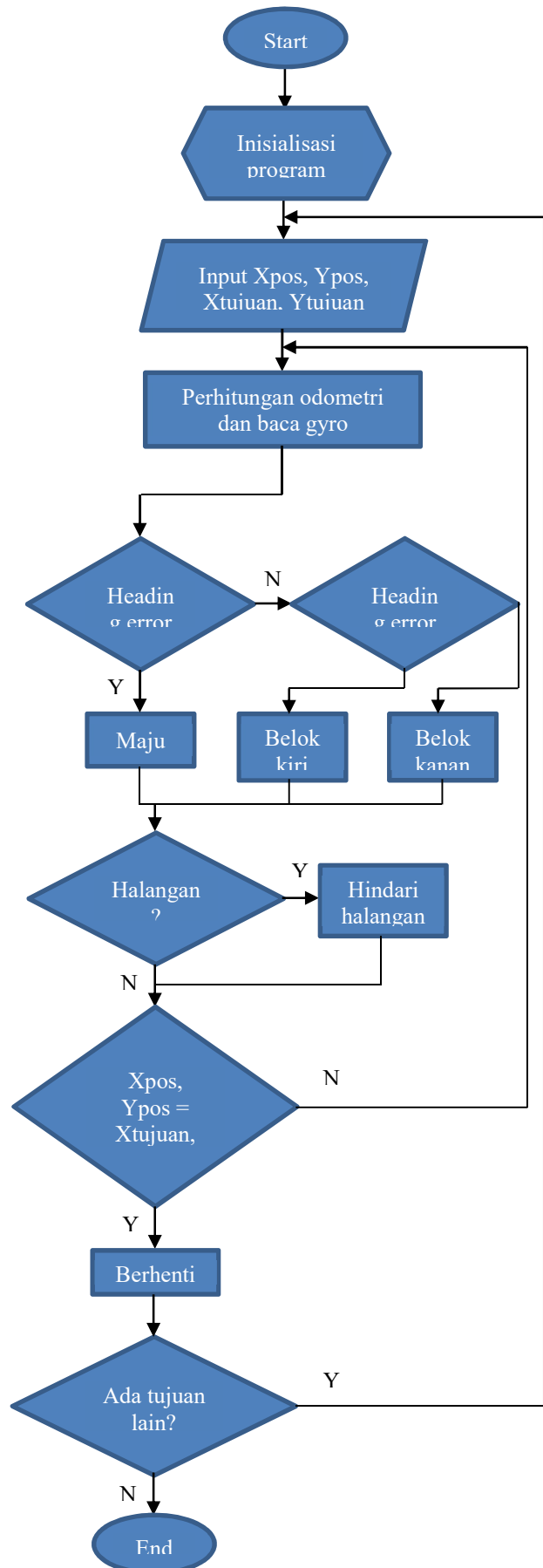
Untuk sistem pergerakan pada *mobile robot* ini akan mengacu pada metode *odometry*. Metode *odometry* ini digunakan untuk menentukan posisi relatif *mobile robot* terhadap posisi awal *mobile robot*. Faktor slip pada roda harus dapat dihindari agar tidak terjadi kesalahan pada perhitungan pulsa dari *rotary encoder*. Sedangkan sensor *gyro* akan digunakan untuk mengetahui arah belok dari *mobile robot* terhadap titik tujuan. Penggunaan dua sensor ini (sensor *rotary* dan sensor *gyro*) diharapkan dapat mengimplementasikan metode *odometry* yang digunakan sehingga *mobile robot* dapat menuju ke titik tujuan yang telah ditentukan. Untuk menuju ke titik tujuan yang telah ditentukan, *mobile robot* menggunakan referensi dari kesalahan arah hadap (*heading error*) *mobile robot* terhadap titik tujuan. Sehingga *heading error* *mobile robot* diharapkan bernilai nol atau mendekati nol agar *mobile robot* dapat menuju ke titik tujuan dengan benar. Agar *heading error* *mobile robot* selalu bernilai nol atau mendekati nol, maka digunakan kendali PD (*Proportional Derivative*). Kendali PD ini akan mengendalikan kecepatan motor kanan dan motor kiri sehingga *heading error* *mobile robot* dapat bernilai nol atau mendekati nol. Bentuk mekanik dari *mobile robot* yang digunakan dapat dilihat pada Gambar 10.



Gambar 10. Bentuk mekanik mobile robot.

### B. Flowchart sistem

Secara keseluruhan, *flowchart* sistem dari *mobile robot* ini dapat dilihat pada Gambar 11.



Gambar 11. Flowchart sistem secara keseluruhan

## IV. HASIL DAN PEMBAHASAN

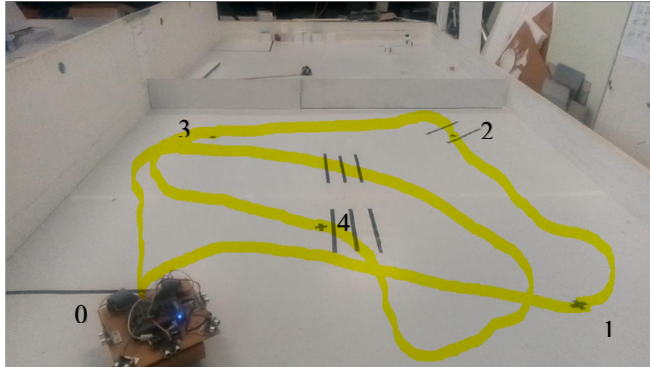
Pada bagian ini akan dibahas hasil percobaan yang dilakukan pada area percobaan *mobile robot*. Percobaan



yang dilakukan antara lain dengan 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. Untuk bentuk halangan yang digunakan berupa papan kayu yang memiliki beberapa ukuran panjang sedangkan tinggi dan lebarnya sama.

#### A. Pengujian dengan tanpa menggunakan halangan

Pada percobaan ini *mobile robot* diperintahkan untuk menuju ke 5 titik tujuan yang telah ditentukan secara berurutan. Titik tujuan tersebut adalah (145, 0), (145, 145), (0, 145), (75, 75), (0, 0). Gambar 12. menunjukkan hasil pencapaian target serta penghindaran halangan oleh *mobile robot*.



Gambar 12. Pencapaian target 5 titik tujuan dengan tanpa halangan.

Pengujian pada Gambar 12. bertujuan untuk mengetahui kestabilan *mobile robot* mencapai target tanpa adanya halangan. Dari hasil pengujian didapatkan kesalahan yang tidak terlalu besar. Tabel pengujian pencapaian target tanpa halangan dapat dilihat pada TABEL I.

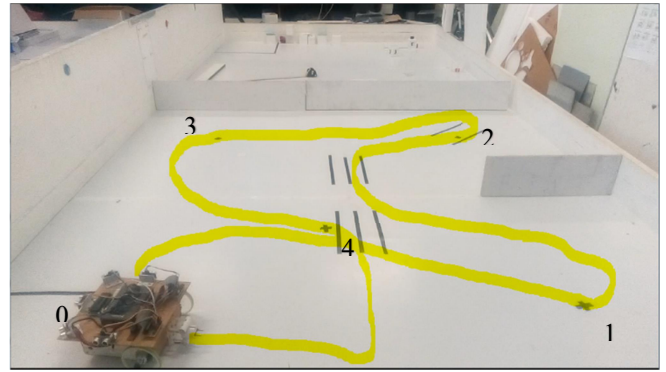
TABEL I. Pengujian pencapaian target tanpa halangan.

Titik ke-	Koordinat asli (x, y)	Koordinat robot (x, y)	Kesalahan ( <i>Error</i> ) (cm)	
			x	y
1	(145, 0)	(138, 7)	-7	7
2	(145, 145)	(138, 138)	-7	-7
3	(0, 145)	(3, 142)	3	-3
4	(75, 75)	(72, 79)	-3	4
5	(0, 0)	(7, 6)	7	6

Dari hasil percobaan pada TABEL I 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 target. Robot akan berhenti dan menganggap target tercapai jika jarak robot dengan target (*target distance*) kurang dari delapan (8 cm).

#### 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). Gambar 13. menunjukkan hasil pencapaian target serta penghindaran halangan oleh *mobile robot*.



Gambar 13. Pencapaian target 5 titik tujuan dengan 1 halangan.

Pengujian pada Gambar 13. bertujuan untuk mengetahui kestabilan *mobile robot* mencapai target dengan adanya 1 halangan. Tabel pengujian pencapaian target dengan 1 halangan dapat dilihat pada TABEL II.

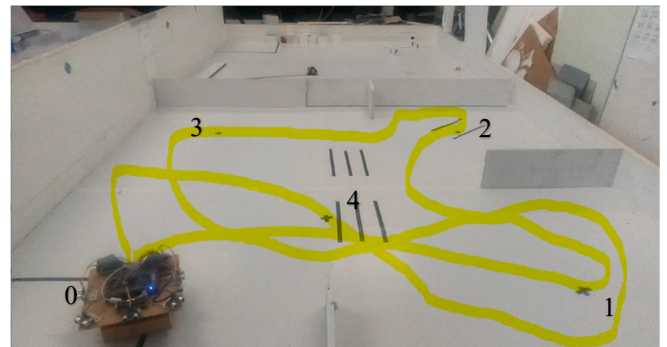
TABEL II. Pengujian pencapaian target dengan 1 halangan.

Titik ke-	Koordinat asli (x, y)	Koordinat robot (x, y)	Kesalahan ( <i>Error</i> ) (cm)	
			x	y
1	(145, 0)	(142, 6)	-3	6
2	(145, 145)	(139, 139)	-6	-6
3	(0, 145)	(11, 143)	11	-2
4	(75, 75)	(63, 77)	-12	2
5	(0, 0)	(11, -2)	11	-2

Dari hasil percobaan pada TABEL II dapat diketahui bahwa terjadi kesalahan yang cukup besar di bandingkan saat tanpa halangan. Akan tetapi, kesalahan tersebut dapat ditoleransi karena posisi robot saat mencapai target selalu berada di atas titik tujuan.

#### C. Pengujian dengan jumlah halangan lebih dari 1

Pada percobaan ini *mobile robot* diperintahkan untuk menuju ke 5 tujuan yang telah ditentukan dengan beberapa buah halangan pada area percobaan secara berurutan. Titik tujuan tersebut adalah (145, 0), (145, 145), (0, 145), (75, 75), (0, 0). Gambar 14. menunjukkan hasil pencapaian target serta penghindaran halangan oleh *mobile robot*.



Gambar 14. Pencapaian target 5 titik tujuan dengan banyak halangan.

kestabilan *mobile robot* mencapai target dengan adanya banyak halangan. Tabel pengujian pencapaian target dengan banyak halangan dapat dilihat pada TABEL III.

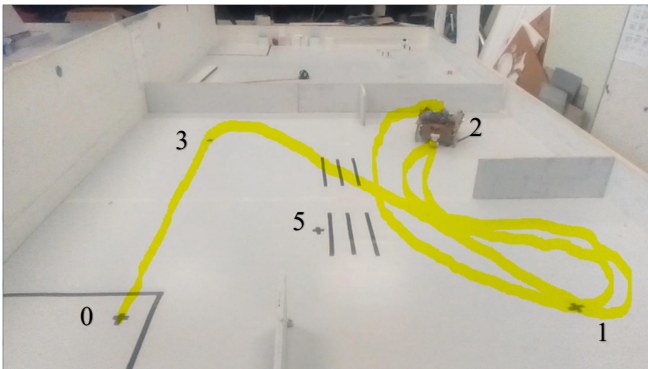
TABEL III. Pengujian pencapaian target dengan banyak halangan.

Titik ke-	Koordinat asli (x, y)	Koordinat robot (x, y)	Kesalahan ( <i>Error</i> ) (cm)	
			x	y
1	(145, 0)	(142, 2)	-3	7
2	(145, 145)	(143, 141)	-2	-4
3	(0, 145)	(1, 142)	1	-3
4	(75, 75)	(71, 72)	-3	-4
5	(0, 0)	(7, -7)	7	-7

Dari hasil percobaan pada TABEL III dapat diketahui bahwa terjadi kesalahan yang hampir mirip dengan saat tanpa halangan. Akan tetapi, kesalahan tersebut dapat ditoleransi karena posisi robot saat mencapai target selalu berada di atas titik tujuan.

#### D. Pengujian 5 titik tujuan 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). Akan tetapi robot akan memilih secara acak titik mana yg akan di tuju oleh *mobile robot*. Gambar 15. menunjukkan hasil pencapaian target serta penghindaran halangan oleh *mobile robot*.



Gambar 15. Pencapaian target 5 titik tujuan secara acak dengan banyak halangan.

Pengujian pada Gambar 15. bertujuan untuk mengetahui kestabilan *mobile robot* mencapai target secara acak dengan adanya banyak halangan. Tabel pengujian pencapaian target secara acak dengan banyak halangan dapat dilihat pada TABEL IV.

Dari hasil percobaan pada TABEL IV dapat diketahui bahwa terjadi kesalahan yang hampir mirip dengan saat tanpa halangan. Akan tetapi, kesalahan tersebut dapat ditoleransi karena posisi robot saat mencapai target selalu berada di atas titik tujuan.

TABEL IV. Pengujian pencapaian target secara acak dengan banyak halangan.

Titik ke-	Koordinat asli (x, y)	Koordinat robot (x, y)	Kesalahan ( <i>Error</i> ) (cm)	
			x	y
1	(0, 145)	(1, 140)	1	-5
2	(145, 145)	(139, 144)	-6	-1
3	(145, 0)	(147, -2)	2	-2
4	(145, 145)	(144, 132)	-1	-13
5	(145, 0)	(146, 3)	1	3

## V. KESIMPULAN

Setelah dilakukan percobaan dengan menggunakan 5 titik tujuan dan beberapa halangan serta dalam mode acak ataupun berurutan, maka dapat diambil beberapa kesimpulan sebagai berikut :

- *Mobile robot* dapat menghindari halangan yang ada dengan baik menggunakan metode *behavior-based*.
- Kendali PD dapat bekerja dengan baik untuk menjaga *heading error* mendekati nol sehingga *mobile robot* dapat menuju titik tujuan dengan benar.
- Selain karena slip pada roda, kesalahan yang terjadi dalam pencapaian target dengan metode *odometry* dapat juga disebabkan oleh pembatasan minimal posisi robot berhenti terhadap target.
- Meskipun terjadi kesalahan pada titik posisi robot saat mencapai target, kesalahan tersebut dapat ditoleransi selama robot dapat berhenti di atas titik tujuan.

## DAFTAR PUSTAKA

- [1] E. Pitowarno, Buku Robotika: Desain, Kontrol, Dan Kecerdasan Buatan. Yogyakarta, Penerbit : Andi, 2006.
- [2] R. Supriyanto, Hustinawati, R. W. Nugraini, A. B. Kurniawan, Y. Permadi, dan A. Sa'ad, Robotika. Depok, Penerbit : Universitas Gunadarma, 2010.
- [3] M. Ashary, I. A. Sulistijono, dan Z. Darojah, Proyek *Mobile Robot* Penghindar Halangan : Sistem Navigasi Pada *Mobile Robot* Menggunakan Metode *Behavior-Based*. Surabaya, Penerbit : PENS, 2014.
- [4] A. Yulianto dan H. P. Handoyo, "Penerapan *behavior-based control* dan *fuzzy logic controller* pada sistem navigasi robot soccer," *Civil and Electrical Engineering Journal*, Vol.7, pp. 16-23, 2012.
- [5] 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*, pp. 1-8, Surabaya. 2011.
- [6] D. Wisnu, A. Wahjudi, dan H. Nurhadi, "Perancangan sistem kontrol PID untuk pengendali sumbu azimuth turret pada turret-gun kaliber 20 mm," *Jurnal Teknik ITS*, Vol. 5, pp. 512-5, 2016.
- [7] \_\_\_\_\_, Interface Rotary Encoder. <https://denumon.wordpress.com/research/interface-rotary-encoder>, diakses pada tanggal 12 April 2019.