

Implementasi Algoritma *Path Planning* dan *Mapping Arena* pada *Mobile Robot*

Daffa Fauzan Adryady¹, Andi Prasetyo², Hafizal Perspicarhanggasidhi Shatyaziamawan³, Ir. Ardy Seto Priambodo, S.T., M.Eng.⁴

^{1,2,3,4} Teknik Elektronika, Universitas Negeri Yogyakarta

Jl. Colombo Yogyakarta No.1, Kabupaten Sleman, Daerah Istimewa Yogyakarta

E-mail: ¹daffafauzan.2019@student.uny.ac.id, ²andiprasetyo.2019@student.uny.ac.id,

³hafizalperspicarhanggasidhi.2019@student.uny.ac.id, ⁴ardyseto@uny.ac.id

Abstrak - Kemajuan teknologi yang pesat di berbagai bidang menyebabkan berkembangnya berbagai aspek kehidupan. Kemajuan dalam bidang robotika adalah contoh berkembangnya berbagai jenis robot. Salah satu jenis robotnya adalah *Mobile Robot*. *Mobile Robot* adalah robot yang memiliki roda untuk menggerakannya. Contoh penerapan dari *Mobile Robot* adalah Algoritma *Path Planning* dan *Mapping Arena* pada *Mobile Robot* yang berfungsi agar robot dapat memetakan sebuah lintasan yang dilaluinya dan setelah itu robot akan menemukan jalan untuk menyelesaikan arena sesuai dengan tujuan yang sudah di *setting*. Untuk dapat memetakan lintasannya, diperlukan sensor ultrasonik dan juga IMU (*Inertial Measurement Unit*). Sensor ultrasonik disini difungsikan untuk mendeteksi adanya dinding atau tidak, sedangkan IMU adalah sensor yang memanfaatkan giroskop, dan akselerometer. Giroskop pada IMU berfungsi untuk mengukur kecepatan sudut dari tiga arah sumbu, yaitu sumbu X, Y dan juga Z. Akselerometer pada IMU berfungsi untuk mengukur percepatan. IMU pada *Mapping Arena Mobile Robot* ini akan difungsikan untuk mengatur derajat perputaran dari robot. Robot ini nantinya akan berputar untuk mengetahui letak dinding disekitarnya dan posisi dinding tersebut akan dipetakan ke dalam bentuk arah mata angin. Robot akan diimplementasikan pada *software* webots. Algoritma *Path Planning* menggunakan modul *deque* yang digunakan dalam *Mobile Robot* bisa menuju ke *finish* yang sudah ditentukan dalam arena labirin dengan waktu yang sangat cepat jika dibandingkan dengan menggunakan algoritma *PID Right Wall Follower*. Algoritma *Path Planning* ini lebih efisien terhadap waktu.

Kata Kunci : *Mobile Robot*, Sensor Ultrasonik, IMU, *Mapping*, *Path Planning*, *Deque*, *Webots*

I. PENDAHULUAN

Perkembangan teknologi sangatlah cepat untuk berkembang. Banyak inovasi yang dapat membantu kehidupan manusia. Kemajuan dalam bidang robotika adalah contoh berkembangnya berbagai jenis robot. Salah satu jenis robotnya adalah *Mobile Robot*. *Mobile Robot* adalah robot yang memiliki roda untuk menggerakannya. *Mobile Robot* banyak diterapkan di beberapa tempat seperti kawasan militer, industri dan juga rumah [1]. Contoh dari *Mobile Robot* adalah robot yang digunakan untuk memetakan suatu lintasan yang berupa labirin untuk memudahkan robot tersebut melewati lintasan.

Mobile Robot merupakan contoh implementasi robot yang dapat membantu manusia. Dengan menggunakan algoritma *Path Planning* dapat digunakan untuk menentukan arah pergerakan dari *Mobile Robot* dan

melakukan perancangan grid lintasan yang sesuai dengan tujuan [2].

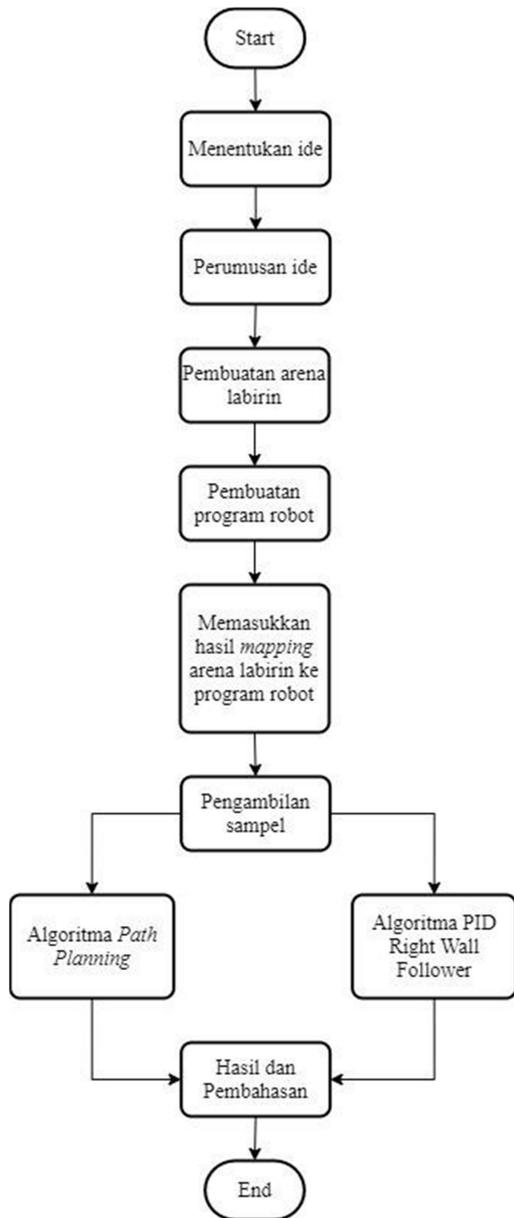
Adapun Faela Shofa dalam metode *simple maze* pada robot wall follower untuk menyelesaikan jalur dalam menelusuri sebuah labirin. Sensor yang digunakan adalah sensor ultrasonik dan juga photodiode. Metoda yang digunakan Faela Shofa adalah *simple maze (Maze Mapping)* yaitu dengan berjalan mengikuti dinding kanan atau kiri dan juga robot akan menggambarkan peta jalan keluar dari labirin [3].

Dari pembahasan jurnal diatas, pemilihan algoritma *Path Planning* dan *Mapping Arena* pada *Mobile Robot* merupakan pemilihan yang tepat untuk memudahkan robot menyelesaikan arena labirin. Karena metode pemetaannya sederhana dengan cara menerapkan arah mata angin untuk mendeteksi adanya dinding, sehingga robot akan menemukan sendiri jalan menuju ke *finish* yang sudah ditentukan berdasarkan *mapping* arena yang sudah ada.

II. METODE PENELITIAN

A. Metode Penelitian

Metode penelitian yang digunakan adalah pendekatan kuantitatif. Sehingga menekankan pembahasan objek yang lebih spesifik, terencana dan tersistematis. Metode kuantitatif merupakan metode penelitian yang berbasis pada filsafat positivisme, yang mana digunakan untuk meneliti populasi atau sampel tertentu, yang umumnya pengambilan sampelnya dilakukan secara random, dan data dikumpulkan menggunakan instrumen penelitian, lalu dianalisis secara kuantitatif/statistik dengan tujuan menguji hipotesis yang telah ditetapkan [4]. Pendekatan kuantitatif ini menguji bagaimana pengaruh objek terhadap variabel lainnya yang diubah. Dalam penelitian ini variabel yang diubah adalah metode robot untuk mencapai *finish*.



Gbr. 1 Bagan Alir Metode Penelitian

B. Target/Subjek Penelitian

Variabel yang diubah dalam penelitian ini adalah metode yang digunakan oleh robot untuk mencapai *finish*. Ada dua metode yang dibandingkan yaitu menggunakan Metode PID *Wall Follower (Right Wall)* dan Metode *path planning*. Yang akan dibandingkan disini adalah waktu untuk mencapai *finish*. Untuk kedua metode tersebut akan dibahas di bagian pembahasan dan hasil penelitian.

C. IMU dan Sensor Ultrasonik

IMU (*Inertial Measurement Unit*) dalam sensor tersebut terdapat gyroscope dan juga akselerometer. Sensor tersebut mampu mendeteksi perubahan sudut atau posisi pada 3 dimensi [5].

Sensor ultrasonik merupakan sensor yang memanfaatkan sumber gelombang suara untuk mengukur jarak. Teknik pengukurannya dengan menggunakan metode

pulsa, pancaran pulsa dikirim melalui media transmisi dan akan dipantulkan oleh objek yang ada pada jarak tertentu [6].

D. Algoritma Path Planning

Path Planning adalah metode/algoritma untuk menemukan jalur yang akan menjadi sebuah tujuan dari sebuah *mobile robot* tersebut, *path planning* juga mampu menemukan jalur terpendek atau optimal antara dua titik [2].

Path yang optimal belum tentu yang terpendek, ada hal yang perlu dipertimbangkan seperti jumlah belok dan berhenti untuk *mobile robot* itu melakukan pergerakan.

Path Planning merupakan suatu metode yang digunakan untuk memetakan atau membuat jalur pada suatu perjalanan dari titik Start ke titik Finish. Asumsi dasar *path planning* pada robot ialah:

- 1) Robot mengenali daerah sekitarnya seperti indera yang terdapat pada manusia (sensor ultrasonik).
- 2) Adanya obstacle (hambatan) dapat mengganggu indera pengenalan pada robot (dinding),
- 3) Robot mengetahui koordinat dan orientasinya (seperti menggunakan gps).

E. Proses Konversi Mapping dari Arena Labirin (World Webots) ke Program Robot

Observasi Arena Labirin

Hal pertama yang perlu dilakukan untuk mapping arena labirin ialah melakukan observasi posisi/arah *obstacle* (dinding) pada setiap sel arena.

Sebagai contoh pada arena labirin yang digunakan pada penelitian ini. Pada sel ke-1 terdapat dinding di arah barat dan utara, sedangkan pada sel ke-2 terdapat dinding di arah utara dan selatan, dan seterusnya hingga sel ke-16. Berikut hasil observasi *obstacle* pada arena labirin yang digunakan pada penelitian ini.

✓	✓	✓	✓	✓	✓					
✓	1	2	3	✓	✓		4	✓		
✓	5	✓	6	✓	✓		7	8	✓	
✓	✓	✓	✓	✓	✓		✓	✓	✓	
✓	9	✓	10	✓	✓	11	✓	✓	12	✓
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
✓	13	✓	✓	14	✓	15	✓	16	✓	✓

Gbr. 2 Hasil observasi obstacle pada arena labirin

Penerapan Hasil Observasi lokasi Obstacle ke Program Robot

Langkah kedua ialah, memindahkan hasil observasi *obstacle* ke program robot. Untuk format penulisan inialisasi *obstacle* pada setiap sel dapat mengikuti format sebagai berikut: [0, Barat, Utara, Timur, Selatan]. Ketika salah satu dari arah mata angin bernilai "1", maka pada sel

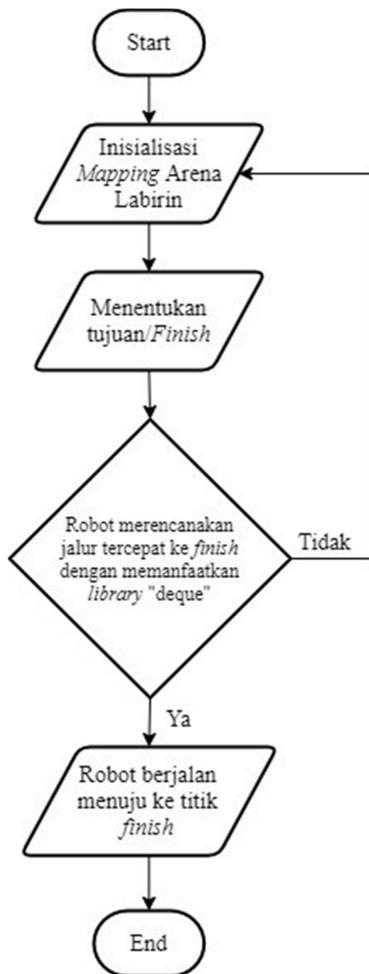
dan arah tersebut terdapat *obstacle* atau dinding. Berikut hasil *mapping* arena pada program robot.

```
grid_maze = [[0, 1, 1, 0, 0], [0, 0, 1, 0, 1], [0, 0, 1, 1, 0],
[0, 1, 1, 1, 0], [0, 1, 0, 0, 1], [0, 0, 1, 1, 1], [0, 1, 0, 0, 0],
[0, 0, 0, 1, 1], [0, 1, 1, 0, 0], [0, 0, 1, 0, 1], [0, 0, 0, 1, 0],
[0, 1, 1, 1, 0], [0, 1, 0, 1, 1], [0, 1, 1, 0, 1], [0, 0, 0, 0, 1],
[0, 0, 0, 1, 1]]
```

F. Modul Deque

Double-ended queue atau Modul *Deque*, merupakan sebuah modul yang memiliki fitur untuk menambah dan juga menghapus elemen. Modul *deque* merupakan bagian dari *library Collections* dalam bahasa pemrograman python. Cara kerja dari modul *deque* ini adalah dengan menambahkan sebuah *list* yang sudah ada, dalam paper ini *list* diasumsikan sebagai sebuah jalur yang akan dilewati oleh *mobile robot* dan jika robot sudah melewati *list* tersebut maka fungsi dari modul *deque* akan mengurangi atau menghapus *list* tersebut [7].

III. HASIL PENELITIAN DAN PEMBAHASAN

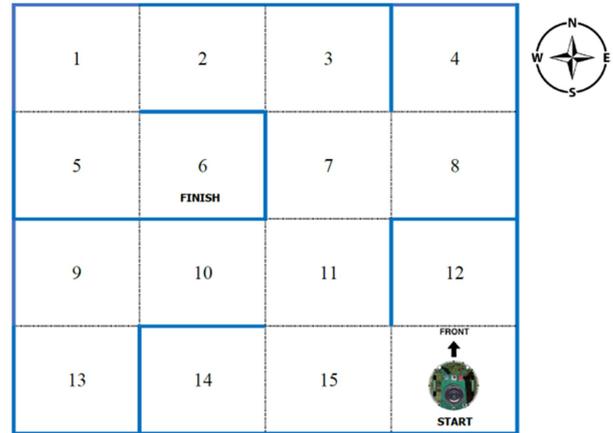


Gbr. 3 Bagan Alir Cara kerja Robot

A. Perbandingan antara Metode *PID Wall Follower (Right)* dengan *Path Planning*

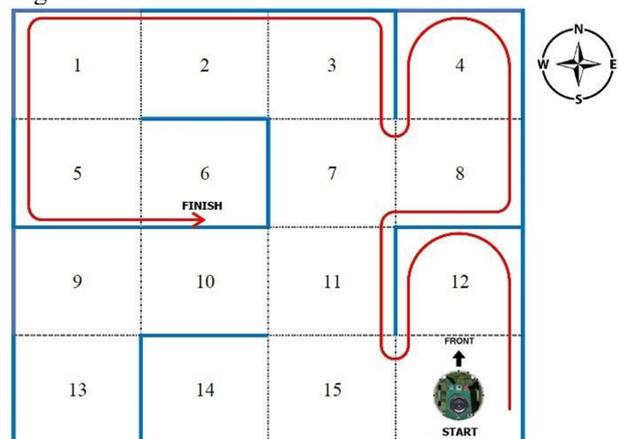
Contoh kasus yaitu terdapat sebuah arena berukuran 4 x 4 dengan hambatan berupa tembok yang tersusun seperti

gambar dibawah ini, sehingga membentuk suatu arena labirin.



Gbr. 4 Arena Labirin

Tugas dari robot ini ialah menuju dari titik *Start* ke titik *Finish* tanpa menabrak dinding. Dengan menggunakan metode *right wall follow*, robot akan menelusuri tembok yang berada di kanan robot dari titik *Start* ke titik *Finish* tanpa menabraknya dengan menggunakan bantuan sensor ultrasonik. Sehingga didapatkan pola pergerakan robot sebagai berikut.



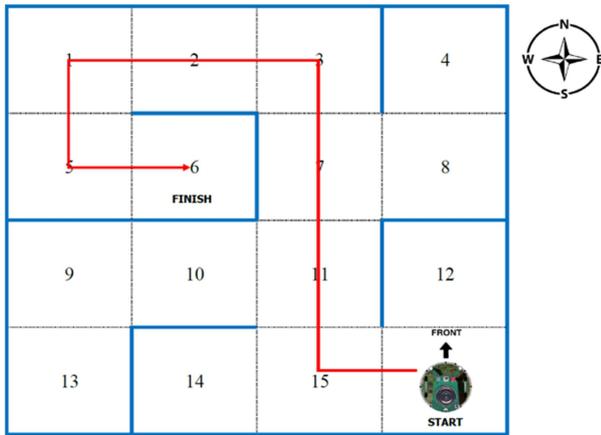
Gbr. 5 Pola Pergerakan Robot metode *wall follower (Right)*

Tabel. I Hasil pengujian arena labirin menggunakan metode *right wall following*.

Pengujian ke - <i>i</i>	Tujuan/Finish (sel)	Waktu (detik)
1	6	78,5
2	6	78,5
3	13	118,7
4	13	118,7
5	4	36,2
6	4	36,2
7	10	107,5
8	10	107,5
9	14	143,6
10	14	143,6

Apabila menggunakan Algoritma *path planning*, robot akan terlebih dahulu menginisialisasi arena labirin terlebih dahulu (misalnya: lokasi hambatan dinding dari

setiap sel). Kemudian robot akan merencanakan jalur terpendek untuk menuju ke titik tujuan (*Finish*) dengan cara memberikan tanda berupa penomoran/indeks yang urut pada setiap sel arena yang akan dilewati nantinya. Sehingga didapatkan pola pergerakan robot sebagai berikut.



Gbr. 6 Pola Pergerakan Robot metode metode *path planning*

Tabel. II Hasil pengujian arena labirin menggunakan metode *path planning*.

Pengujian ke - i	Tujuan/Finish (sel, arah)	Waktu (detik)
1	(6, West)	40,7
2	(6, West)	40,7
3	(13, East)	29,8
4	(13, East)	29,8
5	(4, West)	20,7
6	(4, West)	20,7
7	(10, North)	13,9
8	(10, North)	13,9
9	(14, East)	12,8
10	(14, East)	12,8

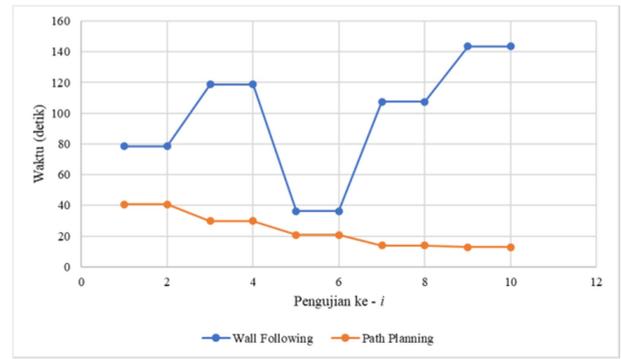
Gambar dibawah ini merupakan *output* pada *console* simulasi ketika penyusunan *path planning* yang dilakukan oleh robot sebelum robot bergerak ke lokasi tujuan. Penyusunan *path planning* ini menggunakan library Python yang bernama “*deque*”.

```

Time: 0.41
plan of moves: deque([('left', 10, 15), ('forward', 9, 14), ('forward', 8, 10)])
Time: 0.41000000000000004
plan of moves: deque([('left', 10, 15), ('forward', 9, 14), ('forward', 8, 10), ('forward', 7, 6)])
Time: 0.448
plan of moves: deque([('left', 10, 15), ('forward', 9, 14), ('forward', 8, 10), ('forward', 7, 6), ('left', 6, 2)])
Time: 0.48
plan of moves: deque([('left', 10, 15), ('forward', 9, 14), ('forward', 8, 10), ('forward', 7, 6), ('left', 6, 2), ('left', 5, 1)])
Time: 0.512
plan of moves: deque([('left', 10, 15), ('forward', 9, 14), ('forward', 8, 10), ('forward', 7, 6), ('left', 6, 2), ('left', 5, 1), ('down', 4, 0)])
Time: 0.544
plan of moves: deque([('left', 10, 15), ('forward', 9, 14), ('forward', 8, 10), ('forward', 7, 6), ('left', 6, 2), ('left', 5, 1), ('down', 4, 0), ('right', 3, 0)])
camera.pane(0,0)-#14008
    
```

Gbr. 7 *Output* pada *console* simulasi *path planning*

Gambar diatas berarti bahwa robot akan berencana bergerak ke kiri, maju, maju, maju, kiri, kiri, bawah/mundur, dan kanan. Dan setiap sel yang akan dilewati sudah diberikan penomoran (10, 9, 8, ... , 3).



Gbr. 8 Grafik hasil pengujian pada kedua metode terhadap waktu

IV. KESIMPULAN

Algoritma *Path Planning* pada robot adalah metode yang digunakan untuk memetakan atau membuat jalur pada suatu perjalanan dari titik Start ke titik Finish. Apabila menggunakan Algoritma *path planning*, robot akan terlebih dahulu menginisialisasi arena labirin terlebih dahulu (misalnya: lokasi hambatan dinding dari setiap sel). Kemudian robot akan merencanakan jalur terpendek untuk menuju ke titik tujuan (*Finish*) dengan cara memberikan tanda berupa penomoran/indeks yang urut pada setiap sel arena yang akan dilewati nantinya.

Jika dibandingkan dengan Algoritma *Right Wall Follower*, *Path planning* memiliki waktu tempuh yang lebih cepat untuk menuju ke finish yang sudah ditentukan. Sebagai contoh untuk menuju ke finish dengan index penomoran 6, *path planning* hanya membutuhkan waktu 40,7 detik. Sedangkan *Right Wall Follower* membutuhkan waktu 78,5 detik.

Kekurangan dari *path planning* yang menggunakan modul *deque* ini adalah harus memasukkan hasil *mapping* dari arena labirin yang akan dilalui robot terlebih dahulu. Hal ini kurang efisien karena robot tidak bisa memilih jalan untuk menuju ke *finish* dengan sendirinya jika tanpa *mapping* dari arena labirin yang akan dilaluinya. Jadi ketika ingin mengubah arena labirin, maka harus mengubah *mapping* pada program terlebih dahulu.

DAFTAR PUSTAKA

- [1] H. Y. Zhang, W. M. Lin, and A. X. Chen, “Path planning for the mobile robot: A review,” *Symmetry (Basel)*, vol. 10, no. 10, 2018, doi: 10.3390/sym10100450.
- [2] R. Amirullah, A. Rusdina, and D. Darlis, “Implementasi Sistem Path Planning dan Routing untuk Mobile Robot Berbasis Visible Light Communication Implementation of Path Planning and Routing System Based-on Visible Light Communication for Mobile Robot,” vol. 8, no. 5, pp. 4283–4291, 2021.
- [3] F. Shofa, “Simple Maze Pada Robot Wall Follower Untuk Menyelesaikan Jalur Dalam Menelusuri Sebuah Labirin,” 2015.
- [4] M. G. Saragih, L. Saragih, J. W. P. Purba, P. D. Panjaitan, and A. Karim, *Metode Penelitian Kuantitatif: Dasar - Dasar Memulai Penelitian*. Yayasan Kita Menulis, 2021.
- [5] A. H. Kurniawan and M. Rivai, “Sistem Stabilisasi Nampun Menggunakan IMU Sensor Dan Arduino Nano,” *J. Tek. ITS*, vol. 7,

no. 2, 2018, doi: 10.12962/j23373539.v7i2.31043.

- [6] F.- Puspasari, I.- Fahrurrozi, T. P. Satya, G.- Setyawan, M. R. Al Fauzan, and E. M. D. Admoko, "Sensor Ultrasonik HCSR04 Berbasis Arduino Due Untuk Sistem Monitoring Ketinggian," *J. Fis. dan Apl.*, vol. 15, no. 2, p. 36, 2019, doi: 10.12962/j24604682.v15i2.4393.
- [7] M. Graichen, J. Izraelevitz, and M. L. Scott, "An Unbounded Nonblocking Double-Ended Queue," *Proc. Int. Conf. Parallel Process.*, vol. 2016-September, pp. 217–226, 2016, doi: 10.1109/ICPP.2016.32