PERBANDINGAN ALGORITMA *FLOODFILL* DAN *DJIKSTRA'S* PADA MAZE MAPPING UNTUK ROBOT *LINE FOLLOWER*

Ary Sulistyo Utomo[1*], Sri Arttini Dwi Prasetyowati[2], Bustanul Arifin[2]
[1]Jurusan Teknik Elektro Medik, Akademi Teknik Elektro Medik Semarang
Jl. Karangbendo 4-5 Karangrejo Semarang 50234
[2]Jurusan Magister Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Sultan Agung
Jl. Raya Kaligawe Km.4 Semarang 50112.
[*]Email: ary.utomo@gmail.com

Abstrak

Robot line follower (RLF) adalah robot yang dapat berjalan mengikuti suatu maze yang berupa garis secara otomatis.RLF dapat digunakan untuk aplikasi mengantarkan barang dari suatu tempat awal ketempat tujuan dengan tepat dan akurat. Untuk menyelesaikan permasalahan tersebut dibutuhkan suatu algoritma yang digunakan untuk mencari jalur terpendek. Pada penelitian ini digunakan dua algoritma yaitu algoritma djikstra's dan floodfill. Pengujian dilakukan dengan cara menjalankan RLF dari titik start menuju ketitik finish dan sebaliknya dengan jalur terpendek. Input RLF untuk menyusuri garis berupa photodiode berjumlah 8 buah diproses dalam mikrokontroler Atmega16 untuk mengendalikan 2 buah motor. Area yang digunakan berukuran 200 x 200 cm mempunyai tebal garis lintasan 2cm dengan jarak terdekat pada setiap simpangnya adalah 40 cm.. Warna garis adalah putih dan background berwarna hitam.Hasil penelitian menunjukkan bahwa kestabilan RLF menyusuri garis lintasan dicapai pada nilai pengaturan PIDKp=45, Ki=10 dan KD=100. Dengan pengaturan nilai tersebut masing-masing algoritma menghasilkan jarak terdekat yang sama karena maze yang digunakan sama. Tetapi proses pencarian titik finish dengan algoritma floodfill lebih cepat dibandingkan menggunakan algoritma djikstra's. Dengan algoritma floodfill,waktu pencarian titik finish lebih cepat dan jarak tempuh lebihdekat.Persentase rata-rata efisiensi waktu floodfill terhadap djikstra's senilai 52,65 %.

***Kata kunci**: robot line follower, kendali PID, djikstra's, floodfill, maze mapping*

1. PENDAHULUAN

Aplikasi robot yang digunakan untuk menyelesaikan sebuah jaringan jalan yang rumit telah banyak diobservasi serta dipublikasikan bahkan dilombakan, diantaranya adalah *Maze Solving Algorithms for Micro Mouse*yang telah dipublikasikan oleh Swati Mishra pada tahun 2008. Pada permasalahan menyelesaikan jaringan jalan yang rumit diperlukan sebuah algoritma yang tepat dan juga efisien sehingga waktu tempuh robot menjadi lebih cepat karena pada saat perlombaan waktu tempuh yang paling cepat yang akan menjadi juara. Selain algoritma tersebut juga terdapat komponen penyusun robot yang dibuat seminimal mungkin, seringan mungkin dan stabil, sehingga dapat menghasilkan robot yang jalannya sangat cepat. Aplikasi robot tikus ini akan diterapkan pada robot *linefollower* atau robot pengikut garis. Apabila pada robot tikus dinding yang menjadi jaringan jalannya, pada robot *line follower*jaringan jalannya berupa garis.

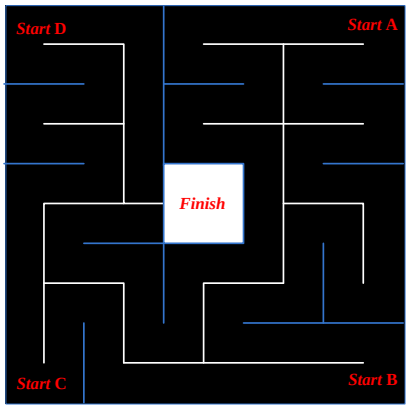
Aplikasi robot *line follower* juga telah diteliti sebelumnya oleh Dr. D. Venkata Vara Prasad ME.Ph.D, DKK dengan menerbitkan jurnal yang berisi tentang *Knowledge based Reinforcement Learning Robot in Maze Environment.* Jurnalnya tersebut mengaplikasikan robot *linefollower* dengan menggunakan algoritma *LSR (Left-Straight-Right)* atau mengikuti jalan kanan atau kiri jika bertemu dengan persimpangan untuk menyelesaikan jaringan jalan yang telah ditentukan.

Salah satu aplikasi dari *embeded system* adalah robot yang memiliki kecerdasan tertentu sehingga dapat menyelesaikan tugasnya dengan cepat dan tepat. Thomas Bräunl mengeluarkan buku yang berjudul "*Mobile Robot Design and Applications with Embedded Systems*". Buku tersebut membahas bagaimana cara membuat desain mobile robot, penerapan sensor, kendali pada gerak robot, menggunakan aktuator dan memberikan suatu kecerdasan pada robot dengan algoritma tertentu.

ISBN 978-602-99334-4-4

2. METODOLOGI PENELITIAN

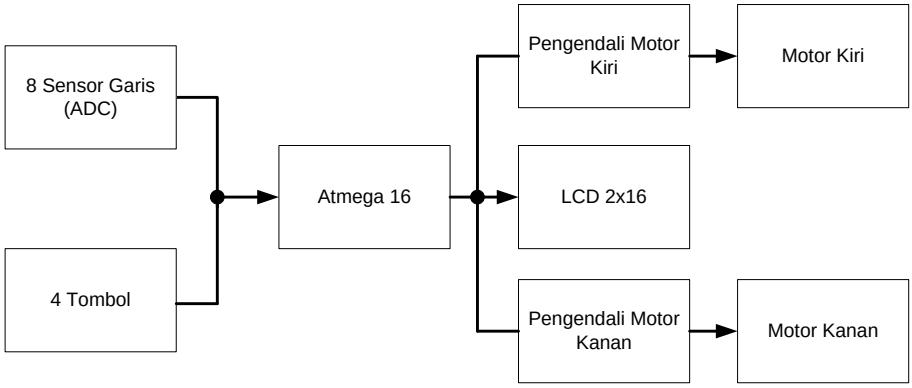
Robot *linefollower* akan menyelesaikan jaringan jalan yang telah ditentunkan. Jaringan jalan ditunjukan pada Gambar 1. Robot *linefollower* berjalan mulai dari titik start yang berada di ujung garis, setelah itu mencarii titik *finish*. Langkah pertama robot *linefollower* akan mencari posisi titik finish dengan melewati jaringan jalan. Setelah sampai ketitik *finish* robot akan kembali ke titik *start* dengan melewati jalan terpendek yang telah ditempuh. Algoritma yang digunakan untuk menentukan jalan terpendek menggunakan dua buah algoritma yaitu *djikstra's* dan *floodfill*. Garis yang bewarna putih merupakan jalan yang akan dilalui. Garis berwarna biru diibaratkan dinding. Kotak putih adalah titik finish.



Gambar 1. Lintasan untuk robot maze

Rancangan line maze yang dibuat adalah tampak seperti pada Gambar 1. Lapangan yang digunakan berukuran 200 x 200 cm. Tebal garis adalah 2cm dengan jarak terdekat pada setiap simpangnya adalah 40 cm. Warna garis adalah putih dan *background* berwarna hitam. Untuk posisi start diletakkan pada ujung garis yaitu startA, startB,startC,dan startD. Sedangkan posisi *finish* ditandai dengan sebidang kotak berwarna putih berukuran 20 x 20cm. Berdasarkan bentuk dari *line maze* tersebut, maka hanya ada satu jalan keluar saja dari *start* menuju *finish* sehingga robot harus menemukan jalan keluar tersebut dengan cara mencari (*search mode*) terlebih dahulu. Jika sudah, berikutnya robot akan kembali dari *finish* menuju *start* dengan jalur terpendek (*return mode*).

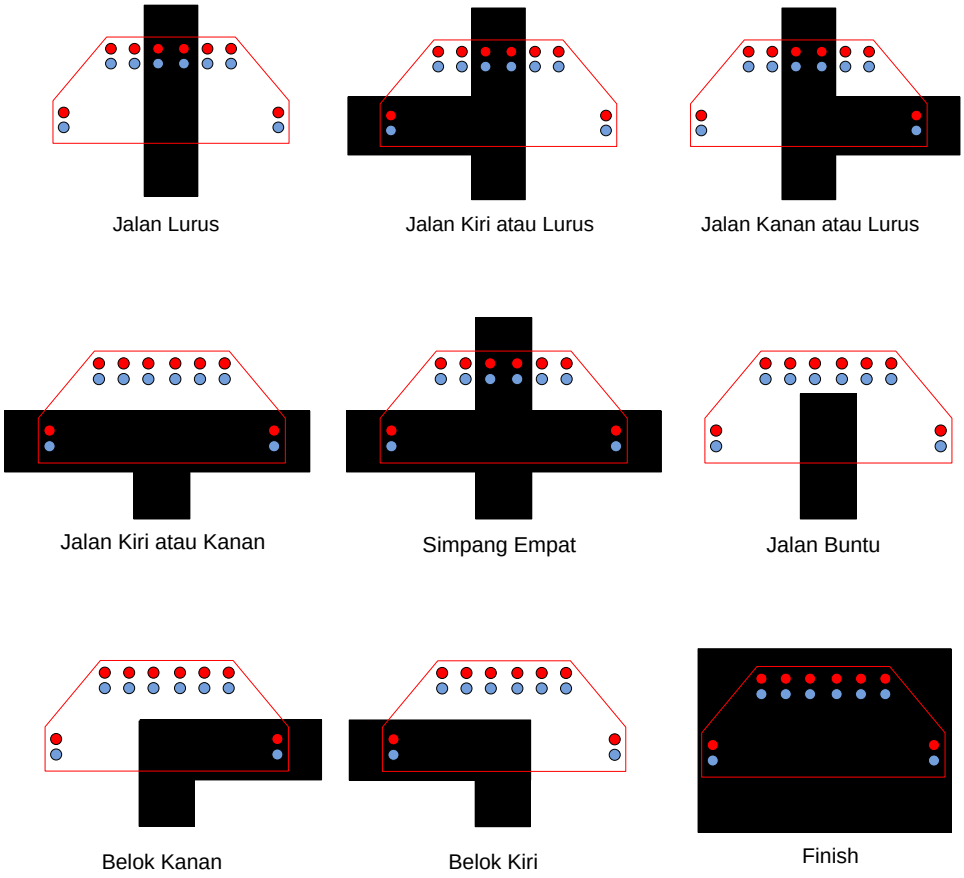
Perancangan hardware dalam penelitian ini terdapat beberapa bagian yaitu sensor, mikrokontroler, dan aktuator yang ditunjukan pada Gambar 2 diagram blok sistem. Sensor yang digunakan yaitu photodioda yang dapat mengeluarkan tegangan berbeda sesuai dengan intensitas cahaya yang diterima. Mikrokontroller digunakan sebagai otak dari robot, dengan memasukkan algoritma ke dalam bentuk program yang akan dijalankan oleh mikrokontroller. Aktuator robot yaitu menggunakan motor DC yang dapat menggerakan robot sesuai dengan perintah.



Gambar 2. Diagram blok sistem

Pada penelitian ini menggunakan delapan buah sensor *photodiode* sebagai sensor garis. Dengan penempatan yaitu, enam buah diletakkan di tengah dan dua buah lagi di sebelah kiri dan

kanan namun dengan posisi lebih ke belakang.Jarak antara sensor yang tengah dengan yang disebelah kiri dan kanan adalah 2,5 cm. Konfigurasi seperti ini dilakukan dengan tujuan untuk membedakan antara pembacaan persimpangan dan *finish.* Keterangan lebih jelas dalam penempatan dan cara pembacaan simpang ada pada Gambar 3.



Gambar 3.Konfigurasi sensor pada robot

3. HASIL DAN ANALISA

Hasil pengukuran dan pengujian yang didapatkan meliputi pengujian kendali robot, pengujian algoritma *Djikstra's* dan pengujian algoritma *floodfill*.

3.1Kendali Robot

Pengujian pada kendai robot PID, dimana percobaan dilakukan pada suatu sistem dengan Kd = 2, ki = 0 , dan pengaturan Kp . Gain Proposional dinaikkan secara perlahan sampai batas kestabilan, dimana sistem mulai mengalami osilasi. Berdasarkan data yang diperoleh sistem mulai stabil pada nilai Kp antara 20 s/d 50. Nilai dari kontrol P semakin besar nilainya maka semakin cepat sistem mengejar nilai set point dan nilai terlalu kecil semakin lambat mengejar set poin, tetapi jika nilai Kp terlalu besar dapat mengakibatkan tidak stabilnya sistem. Apabila diperlukan gerak robot yang halus maka nilai dari Kd diperbesar dan penambahan sedikit Ki. Nilai Kd semakin besar semakin memperhalus jalannya robot, tetapi jika terlalu besar maka gerak robot semakin lambat.Dari hasil percobaan robot didapatkan Kp=44, Kd=100, dan Ki=10. Pada pengaturan tersebut respon stabil dihasilkan 0.4 detik.

3.2 Pengujian Algoritma Djikstra's

Dalam pengujian algoritma ini, robot *linefollower* dicoba untuk melakukan tugas yang sebenarnya sesuai dengan maze yang telah ditentukan. Tujuannya adalah apakah algoritma yang digunakan bisa diterapkan atau tidak. Percobaan dilakukan pada titik startA sampai startD menuju

ISBN 978-602-99334-4-4

ketitik finish dan kembali lagi ke titik start. Apabila robot bertemu dengan persimpangan robot mengambil belok kiri.

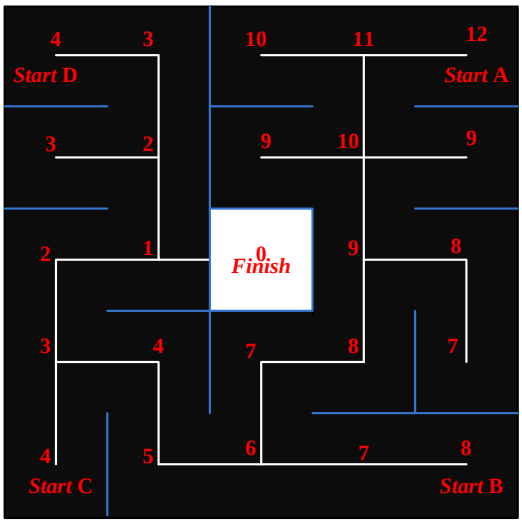
Pada pengujian start di titik D diperoleh hasil waktu tempuh pencarian sama dengan waktu tempuh kembali dikarenakan rute robot yang dilalui pada saat pencarian dan kembali adalah sama. Waktu tercepat yang dapat dihasilkan adalah 7 detik. Waktu rata-rata pencarian sebesar 7,9 detik dan kembali 7,9 detik.

Pada pengujian start di titik C diperoleh hasil waktu tempuh pencarian tercepat 27detik dan waktu tempuh kembali 7 detik.Waktu rata-rata pencarian sebesar 29,2 detik dan kembali 7,2 detik. Jarak pada saat pencarian yaitu 480 cm sedangkan jarak terpendek diperoleh 160 cm.

Pada pengujian start di titik B diperoleh hasil waktu tempuh pencarian tercepat 44 detik dan waktu tempuh kembali 18 detik. Waktu rata-rata pencarian sebesar 46,3detik dan kembali 19,4 detik. Jarak pada saat pencarian yaitu 720 cm sedangkan jarak terpendek diperoleh 320 cm.

Pada pengujian start di titik A diperoleh hasil waktu tempuh pencarian tercepat 97 detik dan waktu tempuh kembali 32 detik. Waktu rata-rata pencarian sebesar 101,1 detik dan kembali 36 detik. Jarak pada saat pencarian yaitu 1240 cm sedangkan jarak terpendek diperoleh 480 cm.

3.2 Pengujian Algoritma Foodfill



Pada pengujian start di titik D diperoleh hasil waktu tempuh pencarian tercepat 7 detik dan waktu tempuh kembali 7 detik. Waktu rata-rata pencarian sebesar 7,3 detik dan kembali 7,2 detik. Jarak pada saat pencarian yaitu 160 cm sedangkan jarak terpendek diperoleh 160 cm. Karena jarak percarian dan jarak kembali sama maka waktu keduanya juga hampir sama.

Pada pengujian start di titik C diperoleh hasil waktu tempuh pencarian tercepat 5detik dan waktu tempuh kembali 5 detik. Waktu rata-rata pencarian sebesar 6,4 detik dan kembali 6 detik. Jarak pada saat pencarian yaitu 160 cm sedangkan jarak terpendek diperoleh 160 cm. Karena jarak percarian dan jarak kembali sama maka waktu keduanya juga hampir sama.

Pada pengujian start di titik B diperoleh hasil waktu tempuh pencarian tercepat 16 detik dan waktu tempuh kembali 17 detik. Waktu rata-rata pencarian sebesar 18,4 detik dan kembali 18 detik. Jarak pada saat pencarian yaitu 320 cm sedangkan jarak terpendek diperoleh 320 cm. Karena jarak percarian dan jarak kembali sama maka waktu keduanya juga hampir sama.

Pada pengujian start di titik A diperoleh hasil waktu tempuh pencarian tercepat 33 detik dan waktu tempuh kembali 30 detik. Waktu rata-rata pencarian sebesar 35,7 detik dan kembali 31,5 detik. Jarak pada saat pencarian yaitu 560 cm sedangkan jarak terpendek diperoleh 480 cm. Perbedaan waktu pencarian dan kembali sedikit dikarenakan jarak keduanya hanya sedikit perbedaannya yaitu 80cm.

4. KESIMPULAN

Pada saat melakukan pengujian algoritma *djikstra's* memerlukan waktu pencarian titik *finish* lebih lama daripada algoritma *flood fill*. Pada algoritma *djikstra's* setiap RLF bertemu dengan persimpangan maka RLF akan mengutamakan belok kiri.Presentasi efisien waktu algoritma *flood fill* terhadap *djikstra's* pada saat pencarian titik *finish* yaitu titik A sebesar 64,6%, titik B sebesar 60,2%, titik C sebesar 78,08% dan titik D sebesar 7,5%. Rata-rata efisiensi 52,65 %.

Nilai keluaran pembacaan ADC pada *background*hitam rata-rata adalah sebesar 4,8, sedangkan pada garis putih rata-rata sebesar 32,25 sehingga dapat menghasilkan rata-rata nilai tengah (*Threshold*) sebesar 18,5. Dapat disimpulkan bahwa kinerja dari sensor telah bekerja dengan baik pada arena yang telah ditentukan.

Nilai Kp=45, Ki=10 dan KD=100 merupakan nilai pengaturan untuk pengendalian robot *linefollower* agar dapat berjalan dengan stabil mengikuti jalur yang telah ditentukan. Waktu stabil didapatkan 0.4 detik.

DAFTAR PUSTAKA

Budiharto, W.,(2006), Membuat Robot Cerdas. Jakarta : PT Elex Media Komputindo.

Braunl, T.,(2006),*EMBEDDED ROBOTICS*., New York : Springer.

Fawaz, Y. Annaz., (2012).,*A Mobile Robot Solving a Virtual Maze Environment*., IJECCT 2012, Vol. 2 (2).

Hagglund.T and Astrom.K., (1995), *PID Controllers: Theory, Design, and Tuning* 2nd Ed. Instrument Society of America 67 Alexander

Venkata, D.V.P.,(2011),*Knowledge based Reinforcement Learning Robot in Maze Environment*. International Journal of Computer Applications (0975 – 8887)Volume 14– No.7, February 2011.

Yadav,S ;& Verma,K.K ;& Mahanta,S. 2012. *The Maze Problem Solved by Micro mouse*. International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-4, April 2012

ISBN 978-602-99334-4-4