

Penyelesaian Jalur Terpendek dengan menggunakan Algoritma Maze Mapping Pada Line Maze

Akhmad Hendriawan, Reesa Akbar
 Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya
 Kampus PENS-ITS Sukolilo, Surabaya
 Email: hendri@eepis-its.edu, reesa@eepis-its.edu

Abstrak—Maze Pada umumnya algoritma maze digunakan untuk mencari jalan keluar yang terbentuk atas lorong-lorong yang dibatasi oleh dinding tanpa atas yang biasa dibuat labirin. Tetapi algoritma ini belum banyak digunakan pada line maze yang biasa digunakan pada perlombaaan linetracer. Oleh karena pada paper ini mencoba untuk menyelesaikan permasalahan maze dengan menggunakan algoritma maze mapping. Algoritma ini terdiri dari dua mode, yaitu mode search dan mode return. Pada mode search, robot melakukan perjalanan dari posisi start menuju finish dengan aturan bahwa robot akan mengutamakan belok kiri bila menjumpai persimpangan. Kode-kode unik dibangkitkan setiap robot berjumpa dengan persimpangan. Pada mode return, robot sudah berjalan dari start menuju finish dengan jalur terpendeknya. Jalur terpendek diperoleh dari kode-kode unik yang telah dikonversi. Berdasarkan percobaan program konversi yang telah dilakukan, maka dibutuhkan formulasi yang handal agar diperoleh hasil yang diinginkan. Pada pengujian keseluruhan sistem, diperoleh error sebesar 10% dari 10 kali percobaan berturut-turut.

Kata kunci— robot, maze, mapping, PID, sensor garis

1. Pendahuluan

Dalam paper ini dibuat robot line tracer yang melewati suatu jaringan jalan yang rumit yang disebut maze. Robot line tracer adalah sebuah robot yang dirancang untuk berjalan mengikuti garis. Namun dalam paper ini, robot tidak hanya bertugas untuk berjalan mengikuti garis saja melainkan juga harus bisa mencari jalan keluar dari suatu maze yang tentunya membutuhkan sistem kendali yang bisa membuat robot mampu melewati maze dengan baik dan dengan tingkat error seminimal mungkin. Teknik pengenalan maze juga sangat dibutuhkan pada sistem kontrol robot. Ini bertujuan agar robot bisa melakukan pemetaan terhadap maze yang akan dilewati tersebut kemudian menemukan jalan keluar untuk rute terpendeknya. Dengan demikian, waktu yang ditempuh untuk mencapai tujuan menjadi lebih efektif.

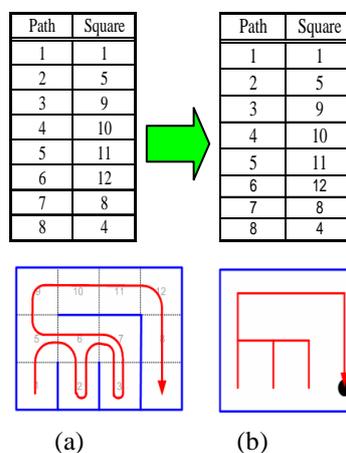
2. Teori Penunjang

Maze mapping merupakan algoritma yang digunakan untuk mapping, yakni mencari dan menggambarkan peta jalan keluar dari maze[5].

2.1 Path Mapping

Gambar path mapping yang ditunjukkan pada gambar 1 adalah mode maze mapping yang digunakan pada

robot wall follower, biasanya oleh robot tikus . Algoritma ini merupakan the basic algorithm[3]. Di dalamnya ber-opsi-kan untuk berjalan mengikuti dinding kiri atau dinding kanan pada proses memetakan maze. Selanjutnya, bila peta yang sudah dibuat tersebut dijalankan, maka robot bisa kembali ke posisi start melalui jalur terpendeknya atau mampu juga mengulangi kembali melewati jalur terpendek dari start menuju finish.



Gambar 1 Wall Maze(a) dan Line Maze(b)

Untuk contoh kasus seperti gambar1 dengan maze berupa wall dan menggunakan right wall follower sebagai algoritma pemetaannya(gambar 2), maka diperoleh:

Hasil pemetaan : 1, 5, 6, 2, 6, 7, 3, 7, 6, 5, 9, 10, 11, 12, 8, 4

Jalur terpendek : 1, 5, 9, 10, 11, 12, 8, 4.



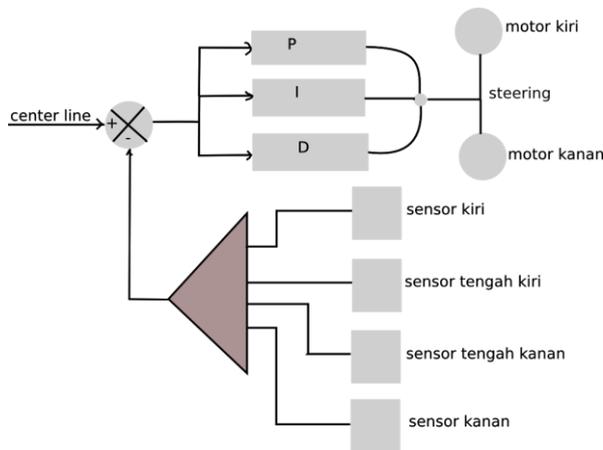
Gambar 2 Jalur Terpendek pada wallmaze(a) dan jalur terpendek pada linemaze(b)

2.2 Kontrol PID

Kontrol untuk mengendalikan kecepatan motor DC pada robot digunakan kontroler PID. Kontroler ini merupakan kombinasi antara kontrol P, I dan D. Dengan menggabungkan ketiga kontroler tersebut, maka akan diperoleh output yang cukup ideal dari yang diharapkan. Gambar 3 menunjukkan skema kombinasi PID dalam

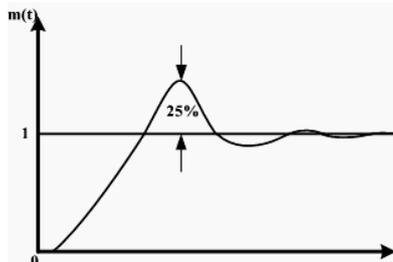
sebuah kontroler untuk motor DC. Metode yang digunakan untuk penalaan nilai dari K_p , K_i dan K_d adalah dengan metode *Ziegler-Nichols*. Metode ini memiliki dua cara, yaitu metode osilasi dan metode kurva reaksi. Kedua metode ditujukan untuk menghasilkan respon sistem dengan lonjakan maksimum sebesar 25% (gambar 3).

Nilai penguatan proporsional pada saat sistem mencapai kondisi *sustain oscillation* disebut *ultimate gain* (K_u). Periode dari K_u disebut *ultimate period* (T_u).

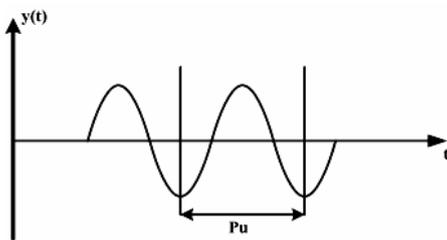


Gambar 3 Blok diagram kontrol

Metode Ziegler-Nichols



Gambar 3 kurva respons unit step tangga satuan yang memperlihatkan 25% lonjakan maksimum



Gambar 4 Kurva respon sustain oscillation

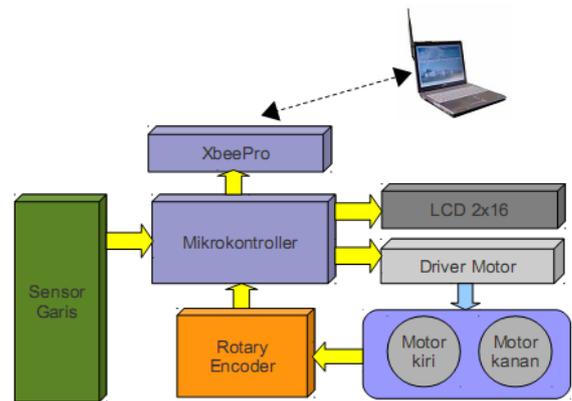
Penalaan parameter PID didasarkan terhadap kedua konstanta hasil eksperimen, K_u dan P_u (gambar 4). Ziegler dan Nichols menyarankan penyetelan nilai parameter K_p , T_i dan T_d berdasarkan rumus yang diperlihatkan pada tabel 1.

Tabel 1 Penalaan parameter PID dengan metode osilasi

Tipe Kontroler	K_p	T_i	T_d
P	$0,5.K_u$		
PI	$0,45.K_u$	$\frac{1}{2} P_u$	
PID	$0,6.K_u$	$0,5 P_u$	$0,125 P_u$

3. Perancangan Sistem

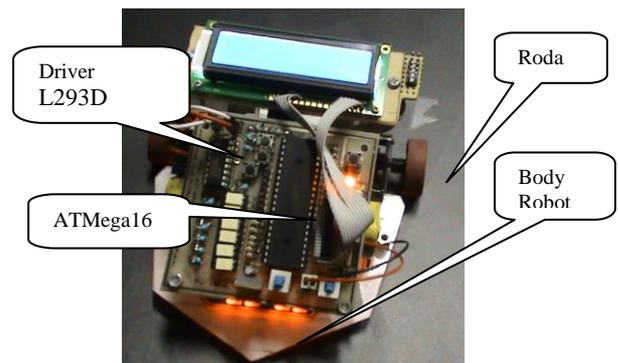
Secara umum, rancangan system yang dibuat adalah tampak pada gambar 4



Gambar 4 Konfigurasi system

Perancangan alat ini meliputi, perancangan mekanik robot, hardware, program dan algoritma. Perancangan mekanik meliputi body dan motor. Untuk perancangan hardware itu terdiri dari rangkaian system minimum, rangkaian sensor dan rangkaian penunjang lainnya yang ditunjukkan pada gambar 5

Sedangkan untuk perancangan program dan algoritma meliputi rancangan program untuk pengolahan sensor adc, program control PID dan algoritma maze mapping.



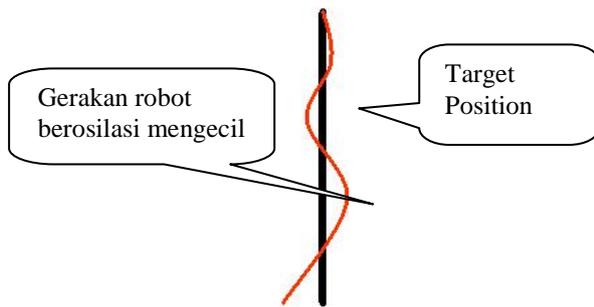
Gambar 5 Robot tampak atas depan

3.1 Kontrol PID

Kontrol PID dalam paper ini digunakan untuk mengontrol posisi robot saat berjalan agar bisa selalu berada di tengah-tengah garis seperti yang ditunjukkan pada gambar 6. Output yang diberikan adalah berupa nilai untuk pengaturan kecepatan motor.

Kontrol untuk mengendalikan kecepatan motor DC pada robot digunakan kontroler PID. Kontroler ini merupakan kombinasi antara kontrol P, I dan D. Dengan

menggabungkan ketiga kontroler tersebut, maka akan diperoleh output yang cukup ideal dari yang diharapkan.



Gambar 6 Pola gerakan robot berosilasi

Dalam penggunaannya, posisi sensor terhadap garis mengartikan error yang terjadi. Berikut adalah gambaran posisi sensor beserta nilai errornya.

- 1000 = 3
- 1100 = 2
- 0100 = 1
- 0110 = 0 (set point = target position)
- 0010 = -1
- 0011 = -2
- 0001 = -3

Dalam aplikasinya, maka peran dari kontroler ini dapat diterapkan dalam program dengan formulasi seperti berikut:

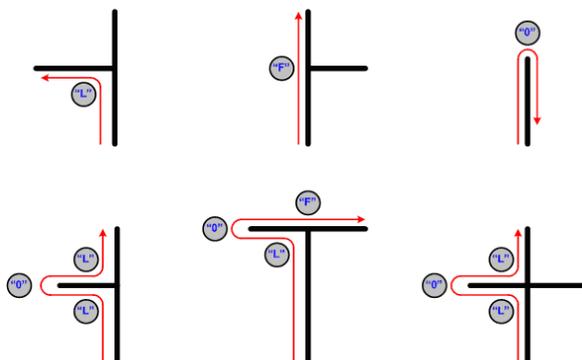
- $pwmKiri = PwmRef + (Kp.error + Kd.(error-old_error) + Ki.(error + ak_error))$
- $pwmKanan = PwmRef - (Kp.error - Kd.(error-old_error) - Ki.(error + ak_error))$

Ket:

PwmRef adalah nilai pwm yang diinginkan pada saat error = 0

3.2 Algoritma Maze Mapping

Prinsip dasar dari maze mapping ini adalah bahwa robot pada saat menjumpai persimpangan maka robot akan mengutamakan belok kiri dibanding lurus atau belok kanan, bila tidak ada pilihan belok kiri maka lurus sehingga bila tidak dijumpai pilihan belok kiri maka robot akan mengambil jalan lurus. Cara pemberian kode pada saat proses mapping ditunjukkan pada gambar



Gambar 7 Posisi saat pemberian kode

Pada saat *search*, robot akan melakukan proses *mapping* dengan cara memberikan kode pada setiap kali robot menjumpai persimpangan dan jalan yang terputus, lalu kode-kode tersebut disimpan dalam *memory* robot. Kode yang diberikan ini akan tersusun terus setiap kali robot berjumpa dengan persimpangan dan jalan yang terputus sampai dengan robot mencapai posisi target (*finish*). Adapun kode-kode yang digunakan pada saat mapping adalah:

- “L” berarti *left*. Ini menandakan bahwa robot telah melakukan belok kiri karena melewati persimpangan.
- “F” berarti *forward* atau jalan terus. Ini menandakan kalau robot melakukan jalan terus karena bertemu dengan persimpangan tiga dengan pilihan lurus atau belok kanan.
- “0” berarti robot berjumpa dengan jalan yang terputus dan berjalan kembali ke persimpangan yang terakhir.

Setelah kode-kode tersusun, maka untuk menemukan jalan keluar dari *maze* tersebut perlu dilakukan penyederhanaan. Sehingga diperoleh susunan kode-kode yang baru. Kunci dari jalan keluar adalah bila tidak ada lagi kode “0”. Itu mengartikan bahwa sudah tidak bertemu lagi dengan jalan terputus.

Untuk melakukan penyederhaan susunan kode-kode ini, maka diperlukan sebuah formulasi. Adapun formulasi tersebut adalah sebagai berikut:

Konversi1:

$$L-0-L = F \quad F-0-L = R \quad L-0-F = R \quad F-0-F = B$$

Konversi2:

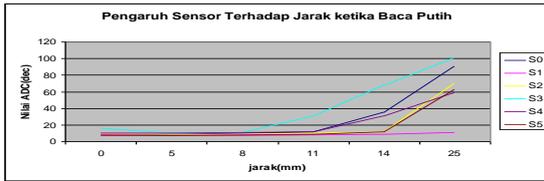
$$\begin{matrix} F-B = 0 & F-B-F = 0 & L-B-F = R & F-B-L = R \\ R-B-L = 0 & L-B-L = F & L-0-R = B & R-0-L = B \\ L-B-R = B & & & \end{matrix}$$

Penyederhanaan prosesnya dilakukan bisa dengan satu kali konversi atau dua kali konversi. Pertama kali dilakukan dengan aturan konversi1, bila setelah dikonversi menemukan kode ”B” maka harus disederhanakan lagi dengan aturan konversi2.

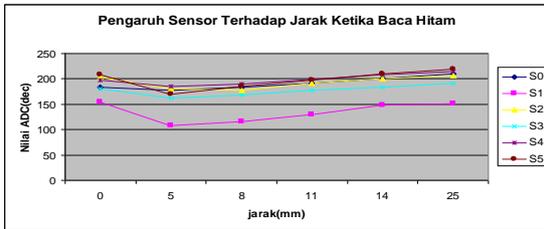
4. Pengujian

Pengujian dilakukan terhadap sensor, motor, kontroler dan algoritma untuk mengetahui karakteristik, kestabilan dan kesesuaian target yang diinginkan.

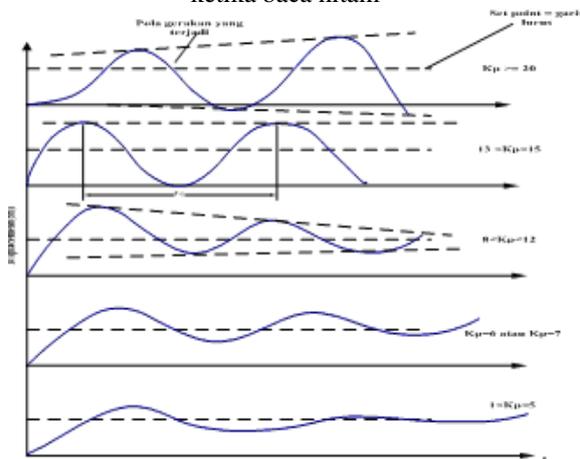
gambar 8 dan gambar 9 adalah grafik pengujian sensor terhadap jarak permukaan/bidang yang disensor. Pada gambar 8 dan gambar 9 menunjukkan walaupun tiap sensor mempunyai jenis yang sama tetapi dengan jarak yang sama memberikan output yang berbeda sehingga jumlah variabel nilai tengah antara pembacaan warna hitam dan warna putih mempunyai variabel sebanyak jumlah sensor.



Gambar 8 Grafik hasil pengujian sensor terhadap jarak ketika baca putih



Gambar 9 Grafik hasil pengujian sensor terhadap jarak ketika baca hitam



Gambar 10 Hasil tuning konstanta P ($K_d=0$)

Tc yang dihasilkan berkisar 2,6 s.d 3 detik.

5. Kesimpulan

- Hasil penerapan algoritma *maze mapping* yang telah dibuat untuk sistem pencarian jalan keluar optimal dengan tingkat keberhasilan 90 %.
- Perlu perlakuan khusus terhadap sensor photodiode, karena setiap photodiode tidak lah sama sensitivitasnya.
- Rangkaian sensor yang dihubungkan dengan pin ADC lebih efektif dan praktis dibanding menggunakan rangkaian comparator analog, karena proses komparasi dilakukan secara program

6. Daftar Pustaka

- [1] Maeda, Y. Kuswadi, Son. M, Nuh. Sulistyio MB. *Kontrol Automatik*. Politeknik Elektronika Surabaya; 1993.
- [2] Pitowarno, Endra. (2006). *Robotika: Desain, Kontrol, Dan Kecerdasan Buatan*. ANDI. Yogyakarta.
- [3] Wahyu Dwi Hartanto, Thomas. Wahyu Agung P. *Analisis dan Desain Sistem Kontrol dengan Matlab*. Yogyakarta. ANDI: 2004
- [4] Ogata, Katsuhiko. *Teknik Kontrol Automatik*. Jilid I edisi kedua – Jakarta. Erlangga: 1996 dhy Sutanta. *Komunikasi Data & Jaringan Komputer*. Edisi Pertama – Yogyakarta; Graha Ilmu, 2005.
- [5] Mishra, Swati. *Maze Solving Algorithm for Micro Mouse*. IEEE International Conference on Signal Image Technology and Internet Based Systems : 2008
- [6] CodeVisionAVR, *help*.
- [7] Wesley H. Huang and Kristopher R. Beevers. (2005). *Topological Mapping with Sensing-Limited Robots*. International Journal of Robotics Research