

Penerapan Algoritma Maze Mapping Untuk Menyelesaikan Maze Pada Line Tracer

M. Iqbal Nugraha^{#1}, Akhmad Hendriawan -1^{#2}, Reesa Akbar-2^{#3}

[#]Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya
Kampus PENS-ITS Sukolilo, Surabaya

¹iqbal_nugrah@student.eepis-its.edu

²hendri@eepis-its.edu

reesa@eepis-its.edu

Abstrak—Maze adalah suatu jaringan jalan yang rumit. Pada bidang robotika ada dua jenis maze yang umum digunakan, yaitu wall maze dan line maze. Wall maze pada umumnya dikenal dengan istilah labirin, yakni suatu jaringan jalan yang terbentuk atas lorong-lorong dengan dinding tanpa atap. Pada line maze, jaringan jalan yang terbentuk dibuat dengan menggunakan garis. Jika garis berwarna putih maka background berwarna hitam atau sebaliknya. Permasalahan yang timbul pada line maze adalah cara untuk mendapatkan jalur terpendek dari line maze. Pada proyek akhir ini, permasalahan pada line maze diselesaikan dengan 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. Kesalahan ini disebabkan kondisi baterai yang semakin melemah.

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

I. PENDAHULUAN

Perkembangan teknologi sekarang ini telah menciptakan berbagai kemajuan dibidang teknologi, khususnya teknologi dibidang robotika.

Dalam proyek akhir ini dirancang sebuah robot *line tracer*, dimana robot ini akan melewati suatu jaringan jalan yang rumit yang disebut *maze*. Robot *line tracer* adalah sebuah robot yang dirancang untuk berjalan mengikuti garis. Namun dalam proyek akhir ini, robot tidak hanya bertugas untuk berjalan mengikuti garis saja melainkan juga harus bisa mencari jalan keluar dari suatu *maze*. Hal ini 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.

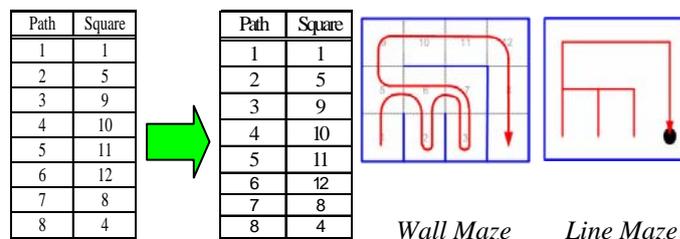
II. TEORI PENUNJANG

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

Maze mapping pada umumnya di berbagai sumber menjelaskan dengan istilah *path mapping* yang konsep dasar dalam pencariannya mengikuti aturan *wall follower* (pada *robot wall follower*) atau *left/right hand rule* (pada *robot line tracer*).

Path Mapping

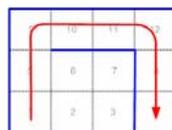
Ini adalah mode *maze mapping* yang digunakan pada robot *wall follower*, biasanya oleh robot tikus. Algoritma ini merupakan *the basic algorithm*[3]. Di dalamnya ber-opsi-kan untuk berjalan mengikuti dinding kiri atau dinding kanan pada proses memetakan *maze*. Selanjutnya, bila peta yang sudah dibuat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



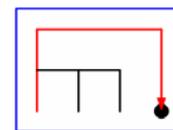
Untuk contoh kasus seperti ini dengan maze berupa *wall* dan menggunakan *right wall follower* sebagai algoritma pemetaannya, maka diperoleh:

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.



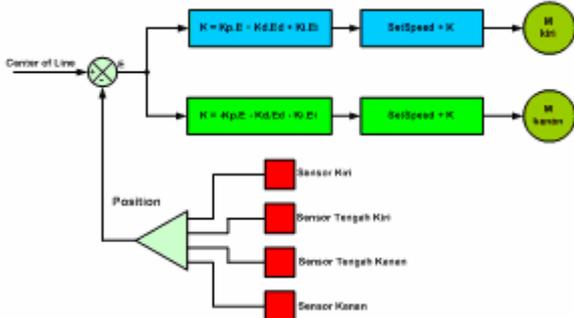
Gambar 2.1 Jalur Terpendek Wall Maze



Gambar 2.2 Jalur Terpendek Line Maze

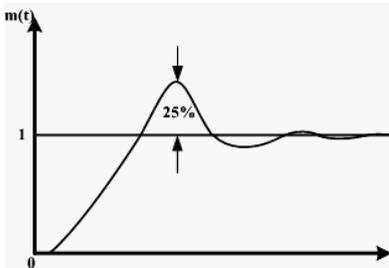
Kontrol PID

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 luaran yang cukup ideal dari yang diharapkan. Gambar di bawah menunjukkan skema kombinasi PID dalam sebuah kontroler untuk motor DC.



Gambar 2.3 Blok diagram kontrol

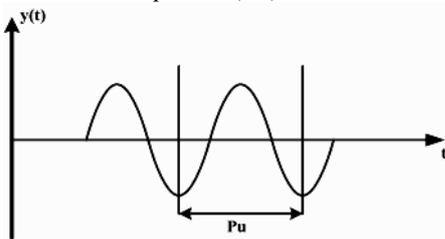
Metode Ziegler-Nichols



Gambar 2.4 kurva respons unit step tangga satuan yang memperlihatkan 25% lonjakan maksimum

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%.

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 2.5 Kurva respon sustain oscillation

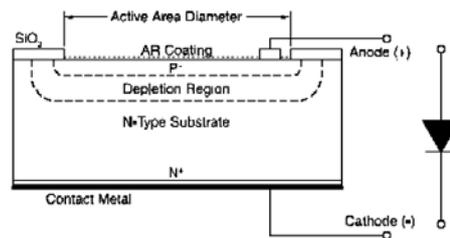
Penalaan parameter PID didasarkan terhadap kedua konstanta hasil eksperimen, K_u dan P_u . Ziegler dan Nichols menyarankan penyetulan nilai parameter K_p , T_i dan T_d berdasarkan rumus yang diperlihatkan pada tabel dibawah ini.

Tabel 2.1 Penalaan parameter PID dengan metode osilasi

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

Sensor Photodiode

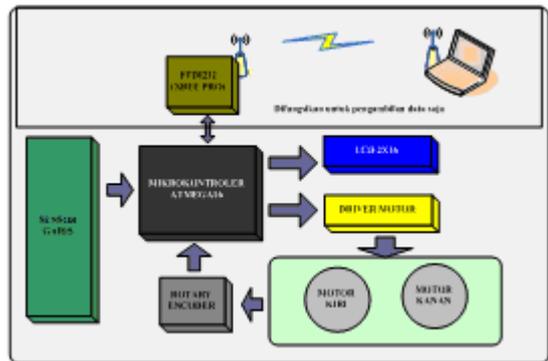
Photodiode mempunyai daerah deplesi dengan medan listrik yang tinggi yang menyebabkan pemisahan pasangan elektron hole secara photogeneration. Untuk operasi kerja yang tinggi, daerah deplesi harus dijaga ketipisannya untuk mereduksi waktu transit. Dengan kata lain peningkatan efisiensi kuantum(jumlah pasangan elektron dihasilkan perphoton yang timbul). Lapisan deplesi harus cukup tebal untuk menyerap sebagian besar cahaya tampak pada gambar



Gambar 2.6 Struktur dari photodiode

III. PERANCANGAN SISTEM

Secara umum, rancangan system yang dibuat adalah tampak pada gambar dibawah ini.



Gambar 3.1 Konfigurasi system

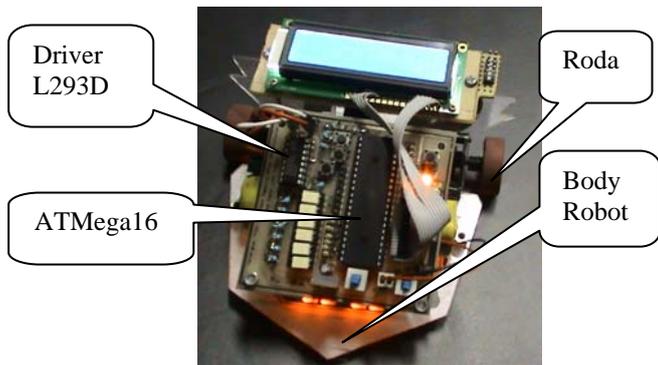
Perancangan Alat

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.

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

Berikut adalah gambar alat yang dibuat:

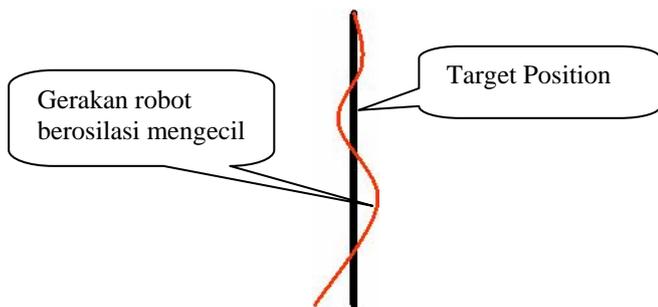


Gambar 3.2 Robot tampak atas depan

Kontrol PID

Kontrol PID dalam proyek akhir ini digunakan untuk mengontrol posisi robot saat berjalan agar bisa selalu berada di tengah-tengah garis. Luaran 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 kontroler P, I dan D. Dengan menggabungkan ketiga kontroler tersebut, maka akan diperoleh luaran yang cukup ideal dari yang diharapkan.



Gambar 3.3 Pola gerakan robot beresilasi

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:

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

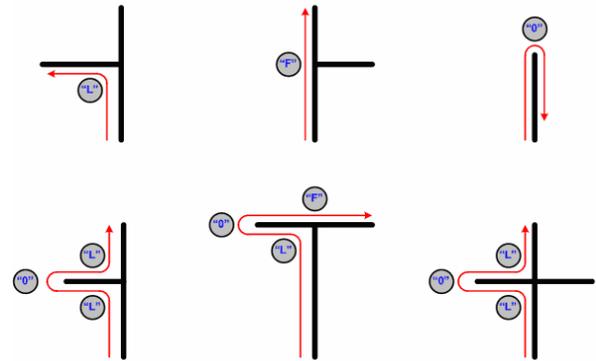
Ket:

PwmRef adalah nilai pwm yang diinginkan pada saat error = 0

Algoritma Maze Mapping

Prinsip dasar dari maze mapping ini adalah bahwa robot pada saat menjumpai persimpangan maka untuk aksinya 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.

Berikut adalah contoh gambar, bagaimana cara pemberian kode pada saat proses mapping.



Gambar 3.4 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 bahwa 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 penyederhanaan 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:

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

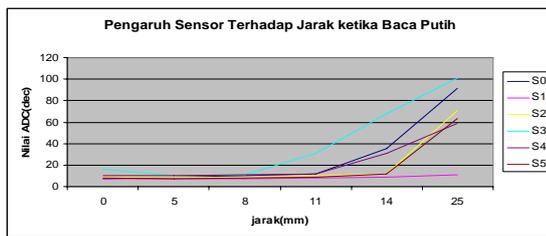
$$R-B-L = 0 \quad L-B-L = F \quad L-0-R = B \quad R-0-L = B$$

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.

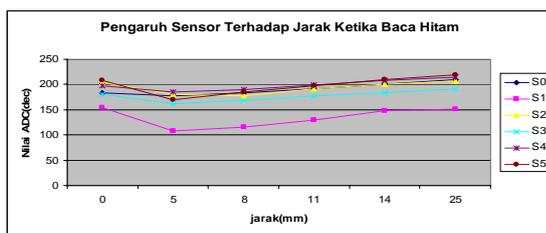
IV. PENGUJIAN

Pada bagian pengujian, telah dilakukan pengujian terhadap sensor, motor, kontroler dan algoritma untuk mengetahui karakteristik, kestabilan dan kesesuaian target yang diinginkan.

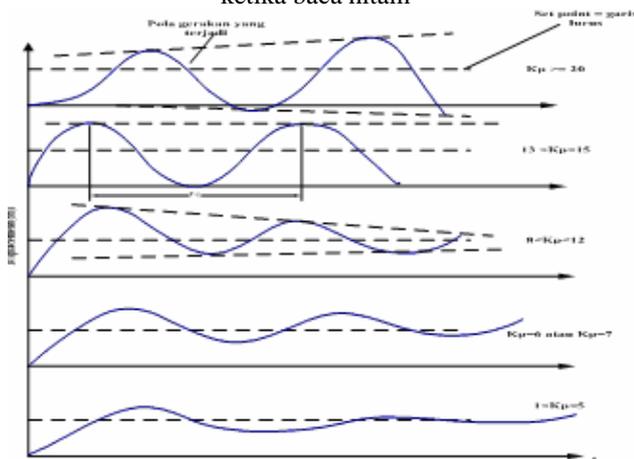
Berikut adalah gambar grafik pengujian sensor terhadap jarak permukaan/bidang yang disensor.



Gambar 4.1 Grafik hasil pengujian sensor terhadap jarak ketika baca putih



Gambar 4.2 Grafik hasil pengujian sensor terhadap jarak ketika baca hitam



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

Gambar 4.3 Hasil tuning konstanta P (Kd=0)

V. 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

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:1996dhy 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