

RANCANG BANGUN ROBOT *SOLVING MAZE* DENGAN ALGORITMA *DEPTH FIRST SEARCH*

Yultrisna ST.,MT*), Andi Syofian ST.,MT**)

*)Politeknik Universitas Andalas Padang

***)Institut Teknologi Padang

ABSTRAK

Robot *Solving Maze* dibuat sebagai salah satu penelitian *prototype* bentuk robot dalam ukuran kecil. Robot ini diberi tugas untuk mencari target dengan waktu tersingkat dalam sebuah jalur yang dirancang berbentuk *maze*. Robot ini dilengkapi dengan sensor *Photodiode* agar dapat mendeteksi jalur *maze* berupa garis menggunakan kecerdasan buatan Algoritma *Depth First Search* yaitu sebuah metoda pencarian data pada permasalahan *maze*. Robot dibuat *autonomous* melalui suatu sistem kontrol berbasis mikrokontroler *ATmega32*, memiliki dimensi panjang 18 cm, lebar 17 cm, dan tinggi 8,5 cm. Robot bekerja dengan dengan melakukan proses *mapping*, *simplifikasi* dan *solving* untuk pemecahan permasalahan *maze*.

Kata Kunci (keywords) : *solving maze, mikrokontroler, photodiode, depth first search*

I. PENDAHULUAN

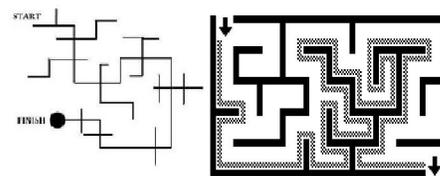
Seiring dengan kemajuan jaman, tidak dapat dipungkiri manusia semakin menginginkan otomatisasi. Berbicara tentang otomatisasi kita teringat dengan kata robot. Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu kecerdasan buaatannya.

Dengan latar belakang pembuatan jalur bentuk *maze*, manusia tidak perlu terlalu susah untuk mencari jalan keluar untuk menemukan objek dengan jalur tercepat yang akan di tempuh. Cukup dengan menggunakan Robot *Solving Maze* masalah itu bukan lagi menjadi masalah yang berarti. Karena robot memiliki kemampuan untuk menjelajah dan akan mencatat dan menyimpan semua rute yang ada. Setelah itu robot akan mampu memilih jalan dengan rute yang benar dan tercepat untuk mencapai garis *finish*.

II. ROBOT SOLVING MAZE

Robot *Solving Maze* memiliki kemampuan bergerak menelusuri garis (*line following*) dan harus mampu menyelesaikan kombinasi garis berupa *maze*, yang bentuknya beragam. Mulai dari persimpangan dengan sudut bervariasi, tikungan, kurva, dan yang lainnya. Pada bidang

robotika ada dua jenis *maze* yang umum digunakan, yaitu *wall maze* dan *line maze*. *Line maze* bentuk jalur yang dibuat berupa garis, yakni suatu jaringan jalan yang terbentuk atas lorong-lorong dengan dinding tanpa atap. Permasalahan yang timbul pada *maze* adalah cara untuk menemukan titik target dengan jalur tercepat, sehingga dibutuhkan metode untuk menyelesaikannya. *Depth First Search* merupakan algoritma yang digunakan untuk memecahkan *maze*, yakni mencari dan menggambarkan peta dari *maze* yaitu proses *mapping maze* yang konsep dasar dalam pencariannya mengikuti aturan *left tracking* (menelusuri jalur kiri). Selanjutnya, bila peta yang sudah dibuat tersebut dijalankan, maka robot bisa menemukan jalur tercepat yang benar dalam menemukan target dari posisi *start* menuju *finish* yang telah ditentukan.

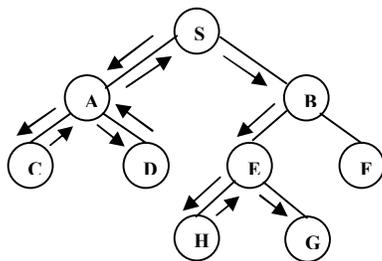


Gambar 2.1 Rancangan Jalur *Line Maze* dan *Wall Maze*

III. METODA

Algoritma DFS (*Depth First Search*)

Pada *Depth First Search* proses pencarian akan dilakukan pada semua anaknya sebelum dilakukan pencarian ke node-node yang selevel. Pencarian dimulai dari node akar (node S) ke level yang lebih tinggi. Proses ini diulang terus hingga ditemukan solusi / goal (node G) seperti gambar berikut.



Gambar 3.1 Pohon Penelusuran DFS dari S ke G

Dalam algoritma DFS, node yang telah dikunjungi disimpan dalam suatu *stack* (tupukan). *Stack* ini digunakan untuk mengacu node-node yang akan dikunjungi sesuai urutan tumpukan (masuk terakhir, keluar pertama) dan mempermudah proses runut balik, jika node sudah tidak mempunyai anak (node pada kedalaman maksimal). Untuk memperjelas cara kerja algoritma DFS beserta *Stack* yang digunakannya. Berikut langkah-langkah algoritma DFS.⁹

1. Masukkan node akar ke dalam *stack*.
2. Ambil node dari *stack* teratas, lalu cek apakah node merupakan solusi.
3. Jika node merupakan solusi, pencarian selesai dan hasil dikembalikan.
4. Jika node bukan solusi, masukkan seluruh node anak ke dalam *stack*.
5. Jika *stack* kosong dan setiap node sudah dicek, pencarian selesai.
6. Ulangi pencarian mulai dari poin 2.

Keuntungan:

1. Membutuhkan memori yang relatif kecil karena hanya node-node pada lintasan yang aktif saja yang disimpan.
2. Secara kebetulan, metode *depth first search* akan menemukan solusi tanpa

harus menguji lebih banyak lagi dalam ruang keadaan.

Kelemahan:

1. Memungkinkan tidak ditemukannya tujuan yang diharapkan.
2. Hanya akan mendapatkan 1 solusi pada setiap pencarian.

IV. PERANCANGAN DAN PEMBUATAN SISTEM

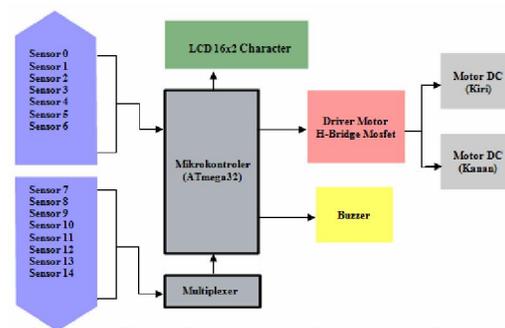
A. Spesifikasi Sistem

Definisi sistem pada tugas akhir ini adalah robot *solving maze* beserta algoritma dan kontroler pergerakannya. Yang dikerjakan dalam tugas akhir ini adalah perancangan serta implementasi algoritma *depth first search* pada penyelesaian robot *solving maze*. Spesifikasi yang diharapkan dari sistem yang akan dirancang adalah sebagai berikut.

1. Bagaimana robot mempunyai kemampuan menjelajah untuk mendeteksi *maze* berupa garis dengan *Sensor Photodiode*?
2. Bagaimana robot melakukan *proses* penyimpanan dan pengolahan data dengan mikrokontroler?
3. Bagaimanakah hasil pengujian robot setelah dilakukan sebanyak tiga kali kondisi pengujian dengan metode kecerdasan buatan algoritma *Depth First Search*?

B. Perancangan dan Implementasi Perangkat Keras

Blok berikut ini menjelaskan rancangan komponen yang membangun sistem robot *solving maze* keseluruhan secara *hardware*

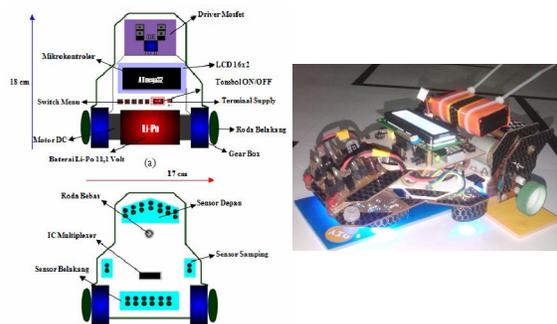


Gambar 4.1 Blok Rancangan *Hardware* Robot *Solving Maze*

Robot dibangun oleh beberapa komponen utama seperti:

1. Komponen *Input* :
 - a. Sensor *Photodiode* dengan jumlah 15 buah sensor yang berfungsi mendeteksi garis yang merupakan jalur *maze*.
 - b. *Multiplexer 8 channel I/O* dengan 1 *output* dan 3 *selector* sebagai jalur *input* dari sensor *Photodiode* menuju *port ADC* pada mikrokontroler.
 - c. Mikrokontroler *ATmega32* dengan 8 *pin I/O ADC (Analog to Digital Converter)* berfungsi untuk pembacaan sensor. Data disimpan, diolah dan dieksekusi.
2. Komponen *Output* :
 - a. *LCD 16x2* karakter untuk tampilan data dari mikrokontroler.
 - b. Motor *DC* di bagian belakang kiri dan kanan.
 - c. *Driver Mosfet H-Bridge* berfungsi mengendalikan gerak motor *DC* pada robot.
 - d. *Buzzer* sebagai indikator *ouput* suara.

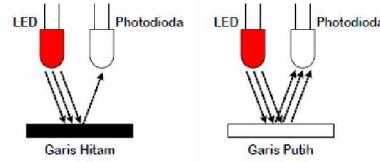
Disain mekanik robot *solving maze* dirancang dengan jenis robot memiliki 2 buah roda *caster* di bagian belakang dan 1 buah roda bebas depan. *Body* robot berdimensi 18 cm x 17 cm yang terbuat dari *board PCB*. Sensor garis dipasang di bawah bagian depan, samping dan belakang. Roda belakang terbuat dari bahan jenis karet berdiameter 2,5 cm.



Gambar 4.2 Rancangan desain mekanik robot *solving maze* dan hasil implementasi rancangan mekanik robot *solving maze*

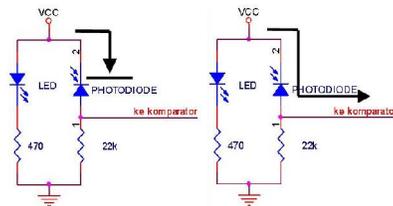
C. Sensor *Photodiode*

Konsep pemantulan sensor *photodiode* terhadap jalur garis dijelaskan sebagai berikut. Jenis LED yang digunakan yaitu LED *superbright*.



Gambar 4.3 Konsep Pemantulan LED di Lapangan

Berikut ini prinsip dan gambaran kerja dari sensor *photodiode*.



Gambar 4.4 Sistem Pembacaan ADC pada

Sensor

Saat *photodiode* tidak terkena cahaya, maka nilai resistansinya akan besar atau dapat diasumsikan tak hingga. Sehingga tidak ada arus bocor yang mengalir menuju komparator dan Saat *photodiode* terkena cahaya, maka *photodiode* akan bersifat sebagai sumber tegangan dan nilai resistansinya akan menjadi kecil, sehingga akan ada arus bocor yang mengalir ke komparator.

Untuk menghitung nilai ADC pada sensor *photodiode* dapat dicari dengan menggunakan rumus:

$$Data\ ADC = \frac{V_{in} \times 2^n - 1}{V_{ref}} \dots\dots\dots (3.1)$$

- Keterangan :
- V_{in}** : Tegangan *input*
 - V_{ref}** : Tegangan referensi
 - 2ⁿ-1** : Jumlah bit data ADC yang digunakan

D. Perancangan Algoritma DFS (*Depth First Search*) pada Robot

1. Pengenalan Bentuk Persimpangan

Pada perancangan awal untuk aplikasi algoritma DFS terlebih dahulu harus dilakukan suatu pengujian berupa pengenalan bentuk-bentuk persimpangan sebagai berikut.

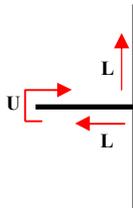
Tabel 4.1 Bentuk Persimpangan

Nama	Bentuk Track	Data yang Tersimpan	
		1	2
Left Junction		Karakter "L"	Belok Kiri
T-Junction		Karakter "L"	Belok Kiri
Cross Junction		Karakter "L"	Belok Kiri
Right Junction		Karakter "S"	Lurus
Dead End		Karakter "U"	Putar Kanan

2. Perancangan *Simplifikasi Data*

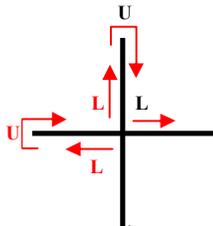
Berikut adalah proses dasar untuk simplifikasi robot *solving maze* :

a.



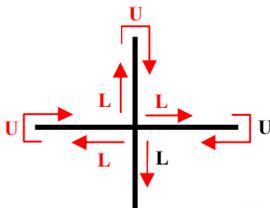
Data yang tersimpan = "LUL".
Hasil eksekusi = "S".

b.



Data yang tersimpan = "SUL".
Hasil eksekusi = "R".

c.



Data yang tersimpan = "RUL".
Hasil eksekusi = "X".

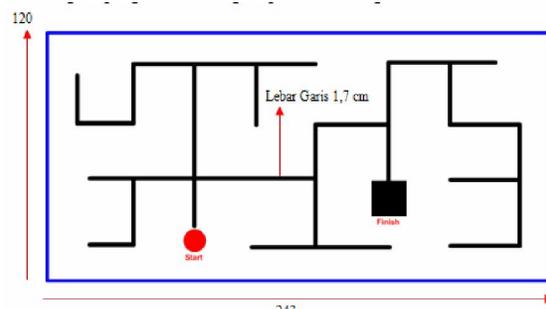
Dari hasil pengenalan bentuk persimpangan, kemudian dilakukan simplifikasi (penyederhanaan) data yang akan menjadi acuan dalam pencarian target dengan jalur tercepat saat

proses *solving maze*. Berikut ini data hasil *simplifikasi data*.

Tabel 4.2 Data *Simplifikasi*

No	Karakter Pemetaan		Karakter Simplifikasi
	1	2	
1	LUL	LXL	S
2	RUR	RXR	S
3	SUS	SXS	X
4	RUL	RXL	X
5	LUR	LXR	X
6	SUL	SXL	R
7	LUS	LXS	R

4. Perancangan Lapangan Robot *Solving Maze*.



Gambar 4.5 Disain Lapangan Robot *Solving Maze*

Bentuk *track* berdimensi 243 cm x 120 cm yang bahannya terbuat dari papan putih dan lebar garisnya 1,7 cm yang terbuat dari isolasi warna hitam.

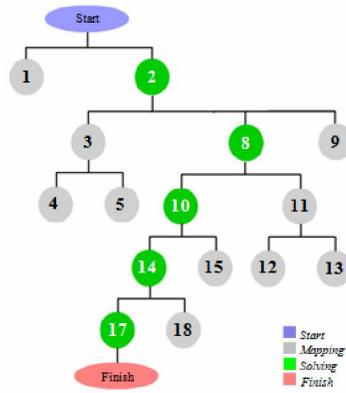
V. PENGUJIAN DAN ANALISA

A. Pengujian Sensor *Photodiode*

Pengujian sensor *Photodiode* dilakukan untuk mendapatkan nilai perbandingan antara besarnya tegangan keluaran dari sensor *Photodiode* terhadap warna dari bidang pantul di lapangan.

Tabel 5.1 Data Pengukuran Sensor *Photodiode*

Data ADC (Decimal)	Tegangan (V)	Warna
224	4.4	Hitam
191	3.7	Coklat
134	2.55	Abu - abu
13	0.25	Putih



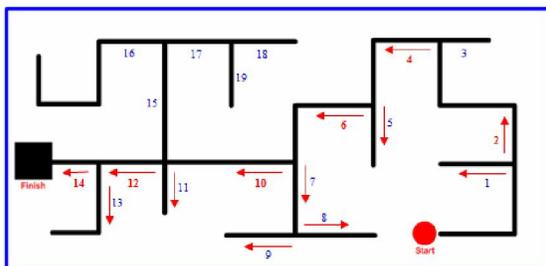
Gambar 5.4 Tree DFS 2

Tabel 5.5 Data Hasil Pengujian 2

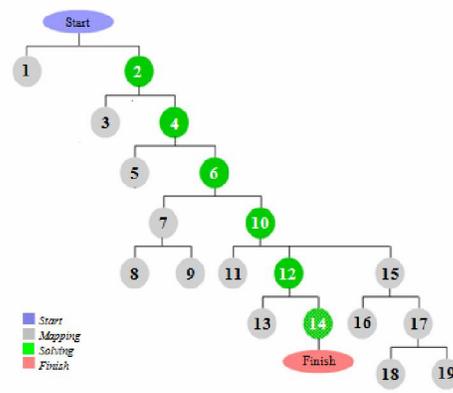
No.	Bentuk Simpang	Arah Gerak Robot	Data Tersimpan	Simplifikasi	Eksekusi
1	Right Junction	Lurus	S	S	S
2	Cross Junction	Belok Kiri	L	SL	
3	T-Junction	Belok Kiri	L		
4	Dead End	Putar Kanan	U	R	SLR
5	Right Junction	Lurus	S		
6	Right Junction	Lurus	S		
7	Dead End	Putar Kanan	U	R	SLRR
8	Left Junction	Belok Kiri	L		
9	Dead End	Putar Kanan	U		
10	T-Junction	Belok Kiri	L	SLRRUL	SLRX
11	Left Junction	Belok Kiri	L	SLRXL	SLX
12	Cross Junction	Belok Kiri	L	SLXL	SS
13	T-Junction	Belok Kiri	L		SSL
14	T-Junction	Belok Kiri	L		SSLL
15	Right Junction	Lurus	S		SSLLS
Solving					SSLLS

Data dari hasil pengujian pada kondisi 2 di atas menjelaskan bahwa jalur yang tepat dilewati robot sebagai jalur terpendek seperti terlihat pada tree DFS 2 yaitu adalah jalur dengan node (nomor simpang) 2, 8, 10, 14 dan 17 dengan instruksi SSLLS (Lurus, Lurus, Kiri, Kiri, Lurus).

3. Pengujian Kondisi 3 (Ketiga)



Gambar 5.5 Lapangan Maze 3



Gambar 5.6 Tree DFS 3

Tabel 5.6 Data Hasil Pengujian 3

No.	Bentuk Simpang	Arah Gerak Robot	Data Tersimpan	Simplifikasi	Eksekusi
1	Left Junction	Belok Kiri	L		
2	Dead End	Putar Kanan	U	S	S
3	T-Junction	Belok Kiri	L		
4	T-Junction	Belok Kiri	L		SL
5	Right Junction	Lurus	S		
6	Dead End	Putar Kanan	U	R	SLR
7	Left Junction	Belok Kiri	L		
8	Right Junction	Lurus	S		SLRS
9	T-Junction	Belok Kiri	L		
10	Dead End	Putar Kanan	U	R	SLRSR
11	Right Junction	Lurus	S		
12	Dead End	Putar Kanan	U	SLRSRUL	SLRSX
13	Left Junction	Belok Kiri	L	X	
14	Left Junction	Belok Kiri	L		SLRR
15	Cross Junction	Belok Kiri	L		SLRRL
16	Dead End	Putar Kanan	U	SLRRLUL	SLRRS
17	Cross Junction	Belok Kiri	L	S	
18	Left Junction	Belok Kiri	L		SLRRSS
19	Dead End	Putar Kanan	U	S	
20	T-Junction	Belok Kiri	L		SLRRSS
Solving					SLRRSS

Data dari hasil pengujian pada kondisi 3 di atas menjelaskan bahwa jalur yang tepat dilewati robot sebagai jalur terpendek seperti terlihat pada tree DFS 3 yaitu adalah jalur dengan node (nomor simpang) 2, 4, 6, 10, 12 dan 14 dengan instruksi SLRRSS (Lurus, Kiri, Kanan, Kanan, Lurus, Lurus).

VI. PENUTUP

Dari hasil penelitian yang telah dilakukan pada pengerjaan tugas akhir ini, maka dapat diperoleh beberapa simpulan diantaranya:

1. Robot melakukan memiliki kemampuan menjelajah jalur maze berupa garis dengan sensor photodiode yaitu dengan melakukan pembacaan ADC (Analog to Digital Converter) yang mana data berupa tegangan

- diubah menjadi data digital dan kemudian diproses oleh mikrokontroler.
2. Proses Penyimpanan dan pengolahan data dilakukan oleh mikrokontroler 8 bit yaitu jenis *ATmega32* menggunakan 8 *channel I/O ADC* untuk pembacaan data dari sensor *photodiode*.
 3. Pengujian yang dilakukan sebanyak tiga kali secara *random*, menghasilkan data yang diperoleh sesuai dengan penggunaan metoda *Algoritma Depth First Search* tanpa ada kesalahan perhitungan data.

Untuk kelanjutan riset yang akan datang, diharapkan adanya pengembangan metode kontroler dengan menggunakan kontroler cerdas seperti logika fuzzy, neuro-fuzzy, algoritma genetic atau kontroler cerdas lainnya. Selain itu juga perlu dikembangkan metode atau cara dalam *solving maze* dengan menggunakan robot jenis *wall maze*.

DAFTAR PUSTAKA

- Malvino. 1985. *Prinsip-Prinsip Elektronika*. Jilid 1. Jakarta: Erlangga.
- Pitowarno, Endra. 2006. *ROBOTIKA: Desain, Kontrol dan Kecerdasan Buatan*. Yogyakarta: ANDI
- Millman, Jacob. 1986. *MIKROELEKTRONIKA, Sistem Digital dan Rangkaian Analog*. Jakarta: Erlangga.
- Daryanto. 2005. *Pengetahuan Teknik Elektronika*. Jakarta: Bumi Aksara.
- Paul Fay, Roy Pickup, Clive Braithwaite, and Jaffrey Hall. 1988. *Pengantar Ilmu Teknik ELEKTRONIKA*. Jakarta: Gramedia.
- Horowitz, Paul. 1987. *Seni dan Disain Elektronika Volume 2*. Jakarta: Gramedia.
- Horowitz, Paul. 1987. *Seni dan Disain Elektronika Volume 3*. Jakarta: Gramedia.
- Heryanto, Ary M. 2008. *Pemrograman Bahasa C untuk Mikrokontroler ATMEGA 8535*. Yogyakarta: ANDI.
- Sutojo T, Mulyanto Edy, Suhartono Vincent.2008. *Kecerdasan Buatan*. Yogyakarta: ANDI.