PERANCANGAN ALGORITMA PENGENALAN POLA LAPANGAN PADA KASUS *MAZE SOLVING ROBOT*

Ken Kinanti Purnamasari

Universitas Komputer Indonesia
Jalan Dipati Ukur 112-114 Bandung 40132
Email : guinev_gs@yahoo.co.nz

ABSTRAK

Teknologi di bidang robotika telah menciptakan berbagai inovasi yang berguna bagi kehidupan manusia. Selain aplikasi robotika dengan sistem dead reckoning yang berjalan atas instruksi manusia, telah dibangun juga sistem robot cerdas yang dapat memutuskan sendiri instruksi yang akan dijalankan.

Sistem robot cerdas dapat dimanfaatkan dalam berbagai kasus, termasuk Maze Solving. Dalam kasus ini, kecerdasan robot diharapkan dapat mengenali pola acak dari lapangan yang diberikan dan menentukan jalur mana yang paling efektif untuk dilalui dari start ke goal.

Dengan beberapa metode algoritma seperti Right Wall dan Trémaux yang dapat mengenali pola acak labirin dengan penyimpanan dan pengolahan nilai pada setiap pergerakan robot, dapat diperoleh kemampuan lebih dalam proses pengenalan pola lapangan. Konsep dari metode-metode tersebut dijadikan dasar dalam perancangan algoritma kecerdasan buatan pada robot maze solving sehingga dapat mengenali pola acak lapangan yang akan dilalui.

Algoritma robot yang dirancang, kemudian disimulasikan dan diukur untuk dapat melihat performanya dan menentukan besar kebutuhan memori yang dibutuhkan. Dari hasil pengujian diperoleh bahwa metode Trémaux dapat lebih menjamin keberhasilan pengenalan pola lapangan acak maze. Metode ini memiliki tingkat keberhasilan 100% dengan kebutuhan memori maksimum sebesar 7945 bit (994 byte), sementara metode Right Wall memiliki tingkat keberhasilan kurang dari 30% dengan kebutuhan memori maksimum sebesar 7433 bit (930 byte).

***Kata Kunci** : maze solving, algoritma, robot, right wall, trémaux*

.

1. PENDAHULUAN

Dalam pemanfaatan sistem robot cerdas pada kasus *Maze Solving*, robot diharapkan dapat mengenali pola acak dari lapangan yang diberikan dan menentukan jalur mana yang dapat untuk dilalui dari *start* ke *goal*.

Metode algoritma seperti *Right Wall* dan *Trémaux*, dapat mengenali pola acak *maze* dengan konsep penyimpanan dan pengolahan nilai dari setiap pergerakan robot. Konsep dari metode-metode inilah yang akan digunakan sebagai dasar dalam perancangan algoritma kecerdasan buatan pada robot *maze solving* sehingga dapat mengenali pola acak lapangan dengan tepat.

Algoritma yang dirancang, kemudian diukur performa dan kebutuhan memorinya sehingga dapat diintegrasikan dengan sistem *hardware* yang tepat.

Tujuan dari penelitian ini adalah :

a. Menghasilkan algoritma robot *Maze Solving* dengan keakuratan tinggi dalam pengenalan pola acak lapangan *maze*.
b. Mengukur performa dan menghitung kebutuhan memori yang dibutuhkan, sehingga dapat memperkirakan spesifikasi hardware yang tepat bagi *Maze Solving Robot.*

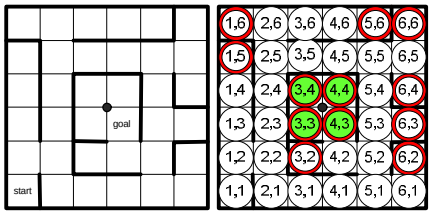
2. MODEL, ANALISA, DESAIN, DAN IMPLEMENTASI

Arena *Maze* dalam kasus perlombaan *Maze solving* di perlombaan RoboGames adalah area dengan ordo n x n dengan setiap sel berbentuk persegi. Pada arena tersebut, akan

terdapat area *start* dan *goal*. *Start* yang merupakan titik awal dari pergerakan robot, berada pada salah satu kotak dari empat bagian ujung lapangan. Sedangkan, *goal* merupakan sebuah ruangan dalam *maze* yang harus dicapai robot. Ruangan ini berukuran empat kotak persegi dan dikelilingi oleh empat buah dinding yang memiliki sebuah pintu (seukuran satu sisi persegi) di salah satu sisinya. Letak dari *goal* tepat berada di tengah *maze*.

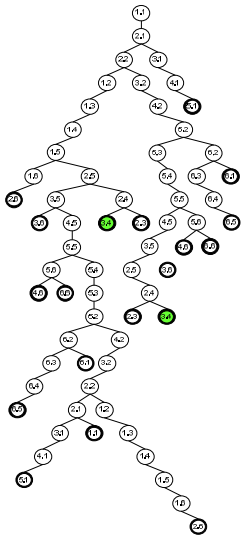
Maze memiliki beberapa jenis kondisi, diantaranya belokan kiri, belokan kanan, simpang dua, dan simpang tiga. Pola keseluruhan dari kombinasi kondisi-kondisi ini dalam *maze* bersifat acak. Hal ini menjadi semakin rumit ketika pola lapangan dibentuk sedemikian rupa sehingga algoritma susur biasa tidak dapat menyelesaikannya.

Lapangan *Maze solving* yang akan dilalui, pada dasarnya ber-ordo n x n. Sebagai contoh untuk analisis *maze* secara keseluruhan, akan digunakan *maze* ber-ordo 6 x 6.



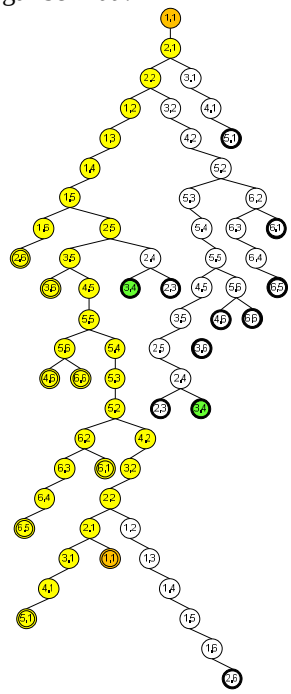
Gambar 1 *Maze* 6 x 6

Dari lapangan di atas, akan didapatkan tree sebagai berikut :



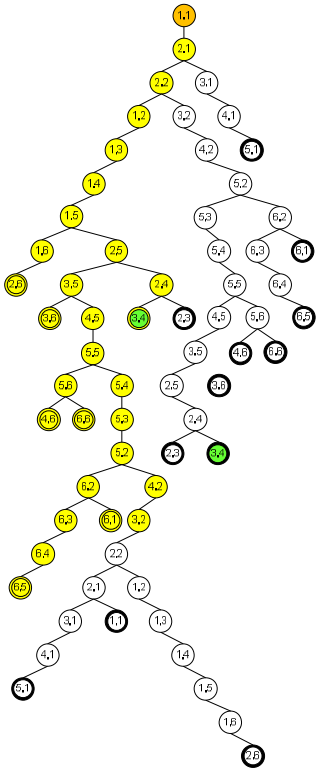
Gambar 2 *Tree* 6 x 6

Dengan algoritma *Right Wall*, didapatkan tree sebagai berikut :



Gambar 3 *Tree* 6 x 6 dengan *Right Wall*

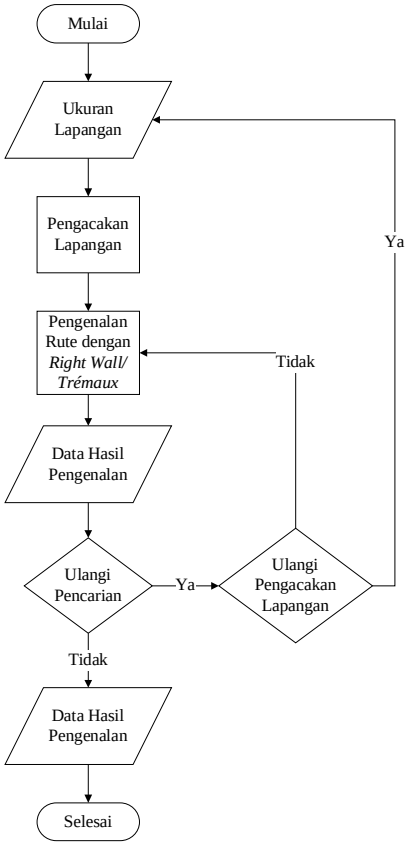
Sedangkan dengan *Trémaux*, dihasilkan *tree* sebagai berikut :



Gambar 4 *Tree* 6 x 6 dengan *Trémaux*

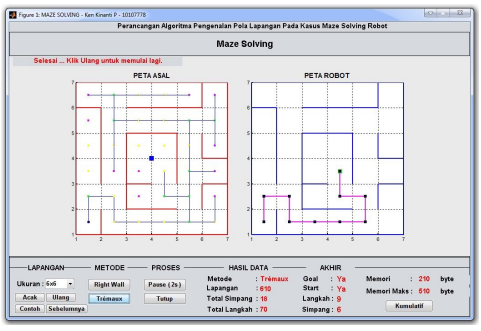
Dapat terlihat bahwa algoritma *Right Wall* tidak bisa mendeteksi adanya jalur ke *goal*, sedangkan *Trémaux* dapat menemukannya.

Algoritma ini, kemudian dirancang dalam suatu simulasi dengan alur proses sebagai berikut :



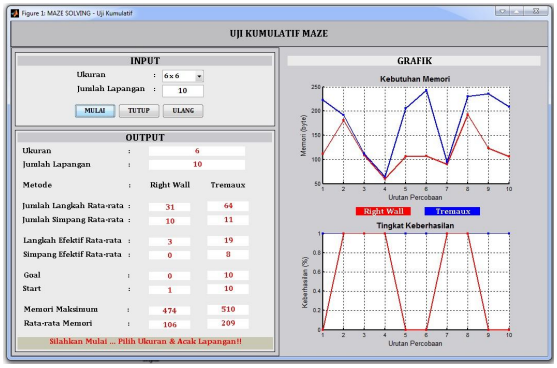
Gambar 5 *Flowchart* simulasi

Contoh tampilan simulasi adalah :



Gambar 6 Tampilan Simulasi

Dengan uji kumulatif sebagai berikut :



Gambar 7 Tampilan Simulasi Kumulatif

3. HASIL DAN DISKUSI

Dengan jumlah sampel 50 buah untuk setiap ukuran lapangan, diperoleh hasil sebagai berikut.

Tabel 1 Hasil Uji

No	Lap	Kebutuhan Memori (bit)		Keber-hasilan	
		RW	*Tré*	*RW*	*Tré*
1	6 x 6	168	172	1	50
2	8 x 8	217	391	0	50
3	10 x 10	268	696	0	50
4	12 x 12	278	1049	0	50
5	14 x 14	316	1331	0	50
6	16 x 16	534	1787	0	50

Dari hasil pengujian di atas, diperoleh tingkat keberhasilan *Trémaux* adalah 100% untuk semua ukuran dan *Right Wall* sebesar 2% untuk ukuran 6 x 6 dan 0% untuk semua ukuran lainnya.

Pada contoh kasus yang sama, besar memori maksimal yang dibutuhkan *Right Wall* adalah 534 bit, sedangkan maksimal memori yang dibutuhkan oleh *Trémaux* adalah 1787 bit.

4. KESIMPULAN DAN SARAN

Berdasarkan dari hasil penelitian, maka dapat diambil kesimpulan bahwa Metode *Trémaux* dapat lebih menjamin keberhasilan pengenalan pola dari metode *Right Wall* untuk jalur dari *start* ke *goal* dengan kebutuhan memori maksimum sebesar 7945 bit (994 byte).

Algoritma yang digunakan dalam perancangan ini, dapat dikembangkan dengan menggunakan teori algoritma *artificial intelligence* lainnya untuk menghasilkan waktu yang lebih cepat dengan tetap memperhatikan kebutuhan memori yang dibutuhkan.

5. DAFTAR PUSTAKA

[1]. Away, Gunaidi Abdia. 2006. *The Shortcut To MATLAB Programming.* Bandung : Informatika.

[2]. Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya).* Yogyakarta : Graha Ilmu.

[3]. Sridadi, Bambang. 2009. *Pemodelan dan Simulasi Sistem (Teori, Aplikasi, dan Contoh Program dalam Bahasa C).* Bandung : Informatika.

[4]. Sugiharto, Aris. 2006. *Pemrograman GUI dengan MATLAB*. Yogyakarta : ANDI.

[5]. Suyanto. 2007. *Artificial Intelligence (Searching Reasoning Planning and Learning).* Bandung : Informatika.

[6]. *Maze Solving/Micromouse Rules* Robogames 2011, tersedia pada http://robogames.net/rules/maze.php diakses pada tanggal 10 Mei 2011.

[7]. *Maze*, tersedia pada http://en.wikipedia.org/wiki/Maze, diakses pada tanggal 10 Mei 2011.

[8]. *Maze Solving Algorithm*, tersedia pada http://en.wikipedia.org/wiki/Maze_ solving_algorithm, diakses pada tanggal 10 Mei 2011.