

PERBANDINGAN WAKTU TEMPUH *MOBILE ROBOT* DALAM ARENA LABIRIN DENGAN ALGORITMA TANGAN KIRI DAN ALGORITMA TANGAN KANAN

Aulia Sabril¹, Nurul Mukhlisah Abdal²

¹Universitas Negeri Makassar
aulia.sabril@unm.ac.id

²Universitas Negeri Makassar
nm.abdal@unm.ac.id

ABSTRAK

Mobile robot mampu menyusuri labirin tertutup dengan menggunakan algoritma. Algoritma yang digunakan ini memberikan kemampuan robot untuk mengambil pilihan apabila dalam menyusuri jalur labirin terdapat dinding penghalang. Penelitian ini membahas dua perbandingan waktu tempuh algoritma robot dalam bernavigasi menyusuri labirin untuk mencari dan mengambil kelereng. Algoritma tersebut yakni algoritma tangan kanan dan tangan kiri. Kedua algoritma tersebut diterapkan kemudian kedua hasil uji waktu tempuh dibandingkan untuk mengetahui algoritma yang dapat memberikan waktu tempuh paling cepat. Capaian hasil penelitian ini adalah algoritma tangan kanan lebih unggul dari waktu tempuh secara keseluruhan dibandingkan algoritma tangan kiri.

Kata Kunci: Waktu tempuh, Labirin, Mobile Robot, Algoritma tangan kiri, Algoritma tangan kanan.

PENDAHULUAN

Implementasi Sistem Kecerdasan Buatan dewasa ini, khususnya pada dunia otomotif sangat pesat perkembangannya. Setidaknya terdapat 263 perusahaan baik yang telah kokoh berdiri maupun perusahaan rintisan, bahu membahu meneliti dan membuat mobil tanpa pengemudi[1]. Ada perusahaan yang fokus pada pengembangan parkir tanpa pengemudi[2], rute perjalanan[3], navigasi[4] dan berbagai sistem lainnya.

Di negara kita, Indonesia, hal ini masih dalam tahap penelitian dan pengembangan di tingkat pendidikan tinggi. Terutama penelitian dalam mengembangkan teknologi kecerdasan buatan agar kendaraan dapat bergerak tanpa pengemudi. Untuk turut serta dalam penelitian ini maka kami mencoba melakukan penelitian sederhana dengan menggunakan software dalam menjabarkan secara terperinci implementasi dari sistem kecerdasan buatan sederhana transportasi tanpa pengemudi. Kendaraan tanpa kemudi yang kami gunakan adalah sebuah mobile robot dalam software Robomind.

Robomind adalah salah satu Programming Assistant Tool (PAT)[5],

berbentuk simulasi program yang khusus digunakan untuk meningkatkan kemampuan siswa dan memahami konsep pemrograman serta meningkatkan kemampuan berpikir logis dan pemecahan masalah. Kecerdasan buatan yang kami teliti adalah algoritma mobile robot untuk bernavigasi sambil mendeteksi adanya halangan dan menghindari halangan. Algoritma yang kami gunakan dalam labirin tertutup ini yaitu dengan algoritma tangan kiri dan tangan kanan[6]. Kami mencoba menguji kecerdasan buatan yang kami gunakan tersebut untuk mencari jalur tercepat ke suatu tujuan tertentu dari titik awal yang telah kami tentukan terlebih dahulu. Kemudian membandingkan kedua metode tersebut manakah yang memiliki waktu tempuh tercepat sehingga kita dapat mengambil pilihan yang tepat dari kedua algoritma tersebut untuk kasus seperti ini kedepannya.

Algoritma tangan kiri adalah algoritma yang membuat robot akan lebih memilih untuk belok kiri jika menghadapi rintangan di depannya. Berbeda dengan algoritma tangan kiri, algoritma tangan kanan ketika berhadapan dengan rintangan.

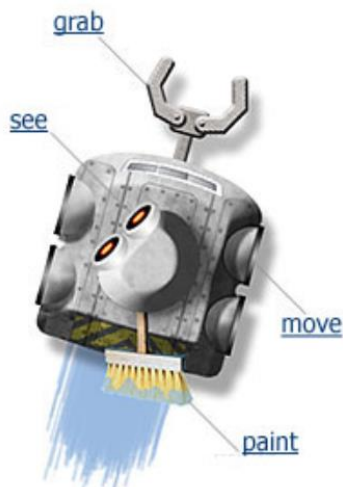
METODE PENELITIAN

Penelitian ini menggunakan perangkat lunak Robomind sebagai simulasi mobile robot dan arenanya. Program Robomind ini dapat berjalan di sistem operasi Windows dan IOS[7]. Pada program ini peneliti menggunakan sebuah *mobile robot*, sebuah kelereng dan membuat 3 variasi simulasi



Gbr.1 Program Robomind

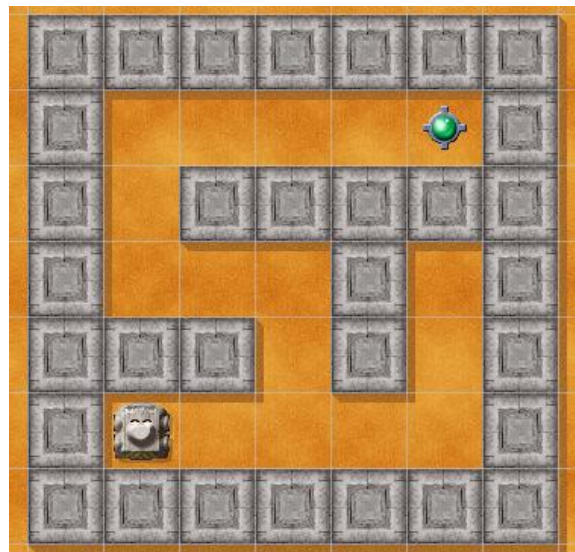
Mobile Robot yang digunakan empat roda yang digerakkan oleh dua motor. Satu motor mengendalikan masing masing satu roda depan dan roda belakang secara mandiri. Untuk navigasi robot menggunakan dua buah sensor sebagai



Gbr 2. Desain Mobile Robot mata untuk melihat, penjepit pada bagian depan untuk mengambil dan meletakkan

benda, dan sebuah kuas yang dapat diganti cat nya dengan warna hitam atau putih pada bagian belakangnya.

Misi mobile robot mencari sebuah kelereng yang diletakkan pada sisi lain labirin. Sambil bergerak, *mobile robot* mengecat jalur mana yang belum dilewati dengan warna putih dan yang sudah dilewati dengan warna hitam. Setelah menemukan lokasi kelereng, *mobile robot* mengambil kelereng tersebut dan bergerak kembali ke titik awal robot dan misipun selesai. Lama waktu tempuh mulai dari robot keadaan diam, bergerak, dan kembali diam pada tempat semula akan dicatat.



Gbr. 3 Posisi awal mobile robot dan posisi kelereng dalam arena

Prinsip dasar algoritma tangan kanan yaitu apabila dalam menyusuri jalur labirin terdapat dinding penghalang didepan maka jalur kanan yang selanjutnya diperhatikan apakah ada atau tidaknya dinding penghalang. Jika jalur kanan tidak ada maka robot akan berputar 90 derajat menghadap ke kanan dan bergerak maju lagi, dan jika ada halangan, maka robot akan berputar 90 derajat menghadap kanan lagi sampai jalur kanan tidak ada halangan. Prinsip ini berkebalikan dengan algoritma tangan kiri.

Prosedur Pengambilan Data:

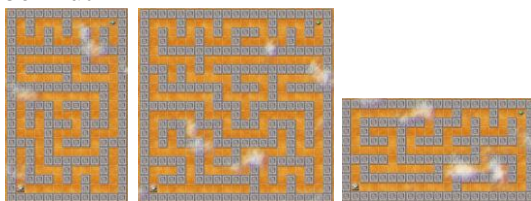
- 1) Menyiapkan ketiga arena maze dengan bantuan menu generate map, pilih ukuran tinggi (height), lebar(width) masing masing 5x10 grid, 10x10 grid, dan 10x5 grid, dengan kerumitan maze (id) yang sama yaitu 42.

- 2) Pada gambar, untuk ketiga arena secara sengaja robot diletakkan pada posisi kiri bawah maze dan kelereng berada pada kanan atas maze.
- 3) Membuat program robot secara keseluruhan dengan misi mencari kelereng sambil mengecat jalan yang dilalui agar didapatkan jalur tercepat ke kelereng setelah itu robot kembali pulang ke titik awal sambil membawa kelereng. Untuk jalur yang sudah dilalui robot dicat dengan warna hitam sedangkan untuk jalur yang belum dilalui robot dicat dengan warna putih.
- 4) Menguji program yang dibuat untuk ketiga arena
- 5) Mencatat waktu pergerakan robot mulai dari titik awal robot bergerak, mendapatkan kelereng, dan kembali ke titik awal dengan satuan detik untuk masing-masing arena.

HASIL DAN PEMBAHASAN

A. Arena Labirin

Pembuatan arena labirin memiliki tahapan sebagai, yaitu tahapan pertama arena dibuat dengan mengklik tombol menu - Open map - Generate map [9]. Selanjutnya akan muncul kotak dialog generate map, disini variasi arena dibuat dengan ukuran lebar(width) dan tinggi(height) masing masing 10 x 5 grid, 10 x 10 grid, dan 5 x 10 grid. Untuk kolom ID nya dibiarkan dengan angka 42 untuk semua arena. Sehingga akan menghasilkan arena yang tampak seperti berikut ini



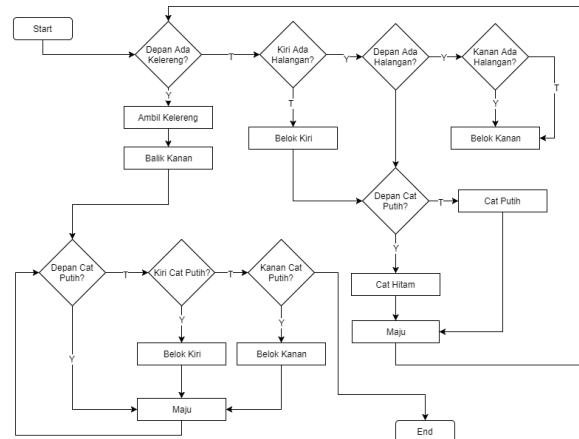
c. Kotak 10x5 b. Kotak 10x10 a. Kotak 5x10

Gbr. 4 Arena labirin

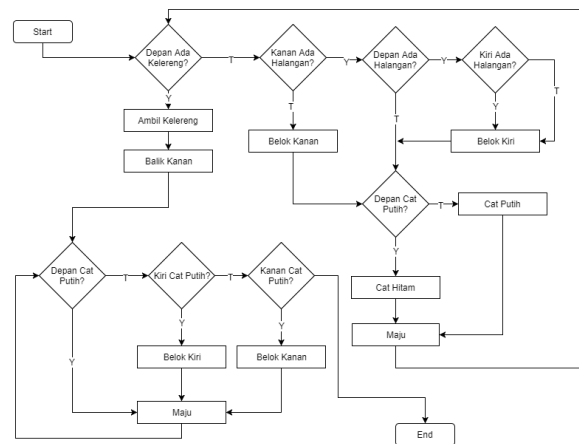
B. Diagram Alir Dan Kode Program

Dalam percobaan ini, mobile robot dapat menyelesaikan misinya menggunakan kaidah tangan kanan dan kaidah tangan kiri yang diagram alirnya dibuat secara daring

pada situs app.diagrams.net[10], dan hasilnya dapat dilihat sebagai berikut:



Gbr. 5 Diagram Alir Kaidah Tangan Kiri



Gbr. 6 Diagram Alir Kaidah Tangan Kanan

C. Perolehan data waktu tempuh

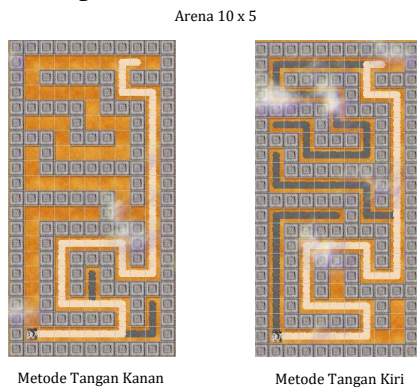
Dari percobaan simulasi yang dilakukan untuk menjalankan misi pada ketiga arena diperoleh waktu tempuh sebagai berikut:

TABEL 1. PEROLEHAN WAKTU TEMPUH

No	Arena	Algoritma	Waktu (detik)
1	10 x 5	Tangan Kanan	19,4
2	10 x 5	Tangan Kiri	38,1
3	10 x 10	Tangan Kanan	54,6
4	10 x 10	Tangan Kiri	78,3
5	5 x 10	Tangan Kanan	16,1
6	5 x 10	Tangan Kiri	11,5

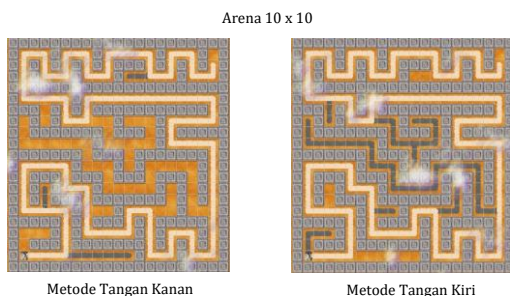
Dalam tabel, terlihat bahwa pada arena 10x5 algoritma tangan kanan menggunakan waktu tempuh yang paling singkat, yaitu 19,4 detik. Begitu juga pada arena 10x10, algoritma tangan kanan mencatat waktu tempuh yang lebih baik dibandingkan dengan algoritma

tangan kiri. Algoritma tangan kiri unggul hanya pada arena 5 x 10 dengan catatan waktu tempu 11, 5 detik.

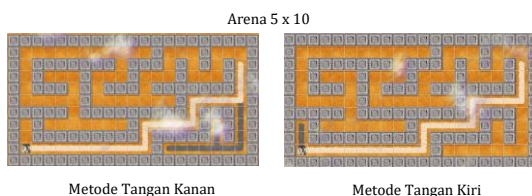


Gbr. 7 Simulasi robot pada arena 10 x 5

Gambar di atas merupakan gambar ketika robot bergerak dengan metode algoritma tangan kanan dan tangan kiri pada labirin 10 x 5. Area yang dilewati oleh robot ditandai dengan cat putih. Simulasi mobile robot dari posisi robot mulai bergerak menuju posisi kelereng menurut algoritma tangan kanan pada gambar sebelah kiri, sedangkan pada gambar sebelah kanan dengan algoritma tangan kiri.



Gbr. 8 Simulasi robot pada arena 10 x 10



Gbr. 9 Simulasi robot pada arena

Gambar 8 dan gambar 9 berturut-turut merupakan gambar ketika robot bergerak dengan metode algoritma tangan kanan dan tangan kiri pada labirin 10 x 10 dan labirin 5 x 10. Area yang dilewati oleh robot ditandai dengan cat putih. Area tersebut merupakan jarak terpendek yang terpilih

dalam algoritma tangan kanan dan tangan kiri. Simulasi mobile robot dari posisi robot mulai bergerak menuju posisi kelereng menurut algoritma tangan kanan pada gambar sebelah kiri, sedangkan pada gambar sebelah kanan dengan algoritma tangan kiri.

KESIMPULAN

Waktu tempuh tercepat mobile robot dalam labirin dari enam bentuk labirin yang digunakan mutlak diperoleh dengan algoritma tangan kanan. Penelitian ini dipandang perlu untuk dilanjutkan kepada percobaan dengan mobile robot sesungguhnya supaya didapat data aktual. Tentunya metode untuk pergerakannya lebih kompleks lagi karena dalam bernavigasi mobile robot mendapat kendala yang lebih kompleks pula dibanding dengan simulasi.

DAFTAR PUSTAKA

- [1] I. Flinsenberg. *Route Planning Algorithms for Car Navigation*. Technische Universiteit Eindhoven: Netherlands. 2004.
- [2] Y. Anshori, A. Y. E. Dodu, dan F. Kurniawan. Perancangan Robot Penelusur Menggunakan Algoritma Dijkstra dan Metode Maze Solver. Dalam *Jurnal Techno.COM*. Volume 18, Nomor 2, Mei 2019. Hlm 166-177.
- [3] M. R. Ariyati, dan A. R. Musthafa. Autonomous Robot Path Planning Menggunakan Perbandingan Metode Particle Swarm Optimization dan Genetic Algorithm. Dalam *Jurnal Buana Informatika*. Volume 9, Nomor 2, Oktober 2018. Hlm 61-70.
- [4] Budianto, Lasguido, dan F. Fathurrahman, dan A. Wibowo. Implementasi Algoritma Breadth First Search Dan Obstacle Detection Dalam Penelusuran Labirin Dinamis Menggunakan Robot Lego. Dalam *Jurnal Ilmu Komputer dan Informasi*. Volume 4, Nomor 1, Februari 2011. Hlm 15.22.
- [5] R. Siegwart dan I. R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. The MIT Press, Massachusetts, 2004.

- [6] T. Braunl, *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*, Springer, Berlin, 2008.
- [7] D. Lee. *The Map-Building and Exploration Strategies of a Simple Sonar-Equipped Mobile Robot*. Cambridge University Press: Cambridge. 2003.
- [8] L. House & J. Hill. *Designing Autonomous Mobile Robots: Inside the Mind of an Intelligent Machine*. Elsevier: New York, 2004.
- [9] S. Zhao, W. Tan, C. Wu, & C. Li. "Research on Robotic Popular Science System Based on LEGO BrickV". *International Conference on Computer Science and Software Engineering*. Vol. 5. Hlm. 741-744. 2008.
- [10] T.H. Cormen, C.E. Leiserson, R.L. Rivest, & C. Stein. *Introduction to Algorithm*. 2nd ed. The MIT Press: Massachusetts. 2001.
- [11] X. Ming. *Fundamentals of Robotics: linking perception to action*. World Scientific Publishing: Singapore. 2003.