

OPTIMASI PENCAPAIAN TARGET PADA SIMULASI PERENCANAAN JALUR ROBOT BERGERAK DI LINGKUNGAN DINAMIS

Yisti Vita Via

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Surabaya
Email: yistivita@gmail.com

ABSTRAK

Robot bergerak harus memiliki perencanaan jalur yang baik ketika menjalankan navigasi di lingkungan. Penelitian yang terakhir menangani permasalahan simulasi perencanaan jalur di lingkungan dengan kehadiran rintangan dan target yang bergerak. Teknik penghindaran rintangan dalam penelitian tersebut cukup baik namun jalur pencapaian target masih belum optimal. Penelitian ini memperbaiki algoritma *Q-learning* pada penelitian sebelumnya dengan menggunakan konsep *Ant Colony*. Pendekatan metode yang dilakukan bertujuan mengoptimalkan pencapaian target. Prediksi pergerakan rintangan dan target juga digunakan untuk meningkatkan efektifitas pencapaian target. Hasil evaluasi uji coba berdasarkan jumlah skenario pelatihan, menunjukkan angka kegagalan metode yang diusulkan lebih kecil 22% dibandingkan metode sebelumnya. Sedangkan berdasarkan jumlah rintangan yang digunakan, angka kegagalan metode yang diusulkan lebih kecil 11% daripada metode sebelumnya. Hasil evaluasi waktu pencapaian target menunjukkan metode yang diusulkan rata-rata mampu mencapai target lebih lama 0,9 detik dari metode sebelumnya. Metode yang diusulkan mampu mencapai target lebih cepat pada pola pergerakan target yang linear.

Kata Kunci: robot bergerak, *Ant Colony*, *Q-learning*, prediksi jalur.

1 PENDAHULUAN

Pada umumnya navigasi robot bergerak di lingkungan dinamis mempunyai dua permasalahan utama yaitu lokalisasi [1] dan perencanaan jalur [2]. Lokalisasi merupakan penentuan posisi dan orientasi robot dengan memperhatikan lingkungan. Banyak teknik yang digunakan dalam permasalahan ini, misalnya menggunakan sensor ultrasonik, infra merah, kamera, dan *Global Positioning System* (GPS). Sedangkan permasalahan lainnya yaitu perencanaan jalur, dimana robot bergerak memerlukan algoritma perencanaan jalur agar dapat menemukan jalur yang aman dalam menghindari tabrakan dan optimal dalam mencapai target di lingkungan. Beberapa penelitian yang telah dilakukan diklasifikasikan secara luas kedalam pendekatan klasik dan heuristik. Metode klasik tidak lagi mendominasi karena adanya kelemahan dalam menyelesaikan perencanaan pergerakan robot (*Robot Motion Planning* (RMP)) yang bersifat *NP-completeness* [3]. Sehingga perhatian beralih pada metode heuristik.

Metode heuristik telah digunakan pada banyak perencanaan jalur robot bergerak, mulai dari keadaan lingkungan dengan kehadiran rintangan dan target yang statis hingga dinamis. Beatriz menyelesaikan permasalahan perencanaan jalur dengan menggunakan *Ant Colony Optimization* (ACO) yang dikembangkan dengan *Genetic Algorithm* (GA) [4]. Akurasi yang dihasilkan lebih meningkat

dibandingkan dengan menggunakan ACO saja. Namun dalam penelitian ini, rintangan dan target yang digunakan masih statis. Garcia mengembangkan pencarian jalur yang jauh lebih cepat dari ACO lainnya serta mampu menghindari keadaan stagnasi [5]. Rintangan yang digunakan dalam penelitian ini sudah dinamis tetapi sifatnya masih diam atau tidak bergerak. Demikian halnya dengan target yang digunakan.

Penelitian selanjutnya dilakukan oleh Zeng yang menggunakan *Fuzzy Logistic Description* untuk pemodelan lingkungan robot dan ACO untuk pencarian jalurnya [6]. Metode ini bertujuan mengatasi ketidakpastian informasi lingkungan oleh sensor. Rintangan yang digunakan pada penelitian ini sudah bergerak tetapi target masih statis. Penelitian dengan menggunakan rintangan dan target yang mana keduanya bergerak kemudian dilakukan untuk kebutuhan aplikasi robot sepak bola [7]. Dalam kasus ini Hwang mengajukan *Adaptive Q-learning* sebagai pengembangan agen pembelajaran. Karena implementasi navigasi yang dilakukan masih menggunakan aturan dari permainan sepak bola, sehingga keadaan lingkungan yang dihadapi tidak sepenuhnya tidak dikenal oleh robot bergerak. Jaradat selanjutnya melakukan penelitian dengan fokus pada perencanaan jalur robot bergerak pada lingkungan yang sepenuhnya belum dikenal. Dalam penelitian tersebut digunakan algoritma *Q-learning* dengan menerapkan definisi baru untuk membatasi jumlah keadaan di lingkungan dinamis dan mengurangi

ukuran *Q-table* [8]. Teknik penghindaran rintangan dalam metode ini memang cukup baik namun jalur pencapaian target yang dihasilkan masih belum optimal.

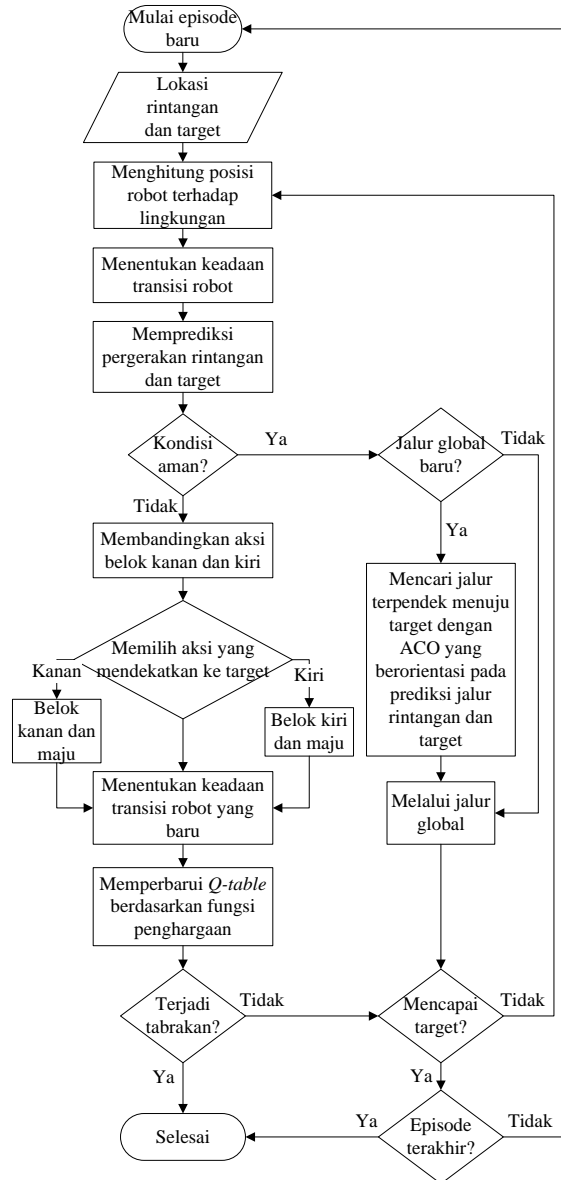
Penelitian ini bertujuan untuk memperbaiki kelemahan metode penelitian Jaradat. Konsep *Ant Colony* diajukan karena mengingat algoritma ini telah banyak digunakan untuk optimasi pencarian jalur terpendek. Di sisi lain *Ant Colony* masih digunakan pada perencanaan jalur yang mana keadaan targetnya masih statis. Keadaan lingkungan yang dinamis dalam permasalahan ini membutuhkan *Ant Colony* yang mampu bekerja dengan sangat cepat. Karakter ini dimiliki ACO yang dikembangkan oleh Garcia. Perhitungan prediksi jalur rintangan dan target juga merupakan hal yang penting dipertimbangkan untuk efektifitas pencapaian target. Dengan demikian metode yang diusulkan ini diharapkan mampu meningkatkan efektifitas perencanaan jalur agar selama pencapaian target, robot lebih sedikit berhadapan dengan rintangan dan bergerak ke arah target lebih cepat.

2 MODEL, ANALISIS, DESAIN, DAN IMPLEMENTASI

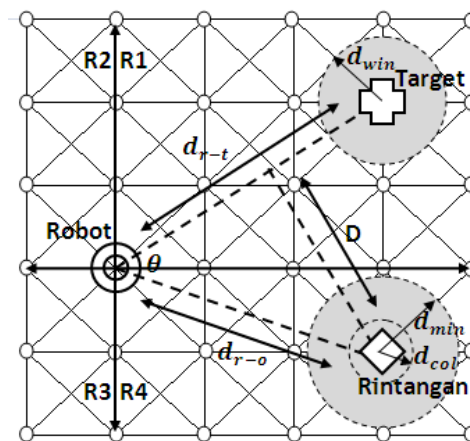
Tahap awal untuk perencanaan jalur robot bergerak adalah memodelkan lingkungan simulasi robot. Dalam penelitian ini model lingkungan digambarkan ke dalam sebuah matriks yang merepresentasikan graf berukuran 150×50 *node*, dimana nilai '0' pada *node* menandakan aman dan '1' adalah tidak aman. Setiap *node* terhubung dengan delapan *node* lain di sekelilingnya. Hal ini disesuaikan dengan sudut orientasi robot bergerak pada simulasi ini yaitu sebesar 45 derajat. Tahap berikutnya adalah mendesain algoritma perencanaan jalur. Pada Gambar 1 dijelaskan langkah-langkah yang harus dilewati robot bergerak pada tahap pelatihan sebelum robot dilepas untuk bernavigasi sendiri di lingkungan.

2.1 Perhitungan Posisi Robot terhadap Lingkungan

Simulasi robot bergerak diasumsikan dilengkapi sensor yang menerima masukan koordinat posisi semua rintangan dan target di lingkungan. Robot, rintangan, dan target masing-masing disimulasikan menempati posisi *node-node* tertentu di lingkungan. Setiap *node* memiliki posisi spesifik terhadap x dan y yang ditentukan dengan $P = [P_x \ P_y]^T$. Posisi robot dituliskan $P_{rob} = [P_{rob-x} \ P_{rob-y}]^T$, sedangkan posisi rintangan dan target berturut-turut adalah $P_{obs} = [P_{obs-x} \ P_{obs-y}]^T$ dan $P_{tar} = [P_{tar-x} \ P_{tar-y}]^T$. Penentuan posisi ini diperlukan untuk memodelkan keadaan simulasi robot bergerak selama proses navigasi berlangsung.



Gambar 1. Algoritma Simulasi Perencanaan Jalur pada Tahap Pelatihan



Gambar 2. Perhitungan Posisi Robot Bergerak dalam Lingkungan Simulasi

Perhitungan posisi robot terhadap lingkungan dapat dilihat pada Gambar 2, dengan langkah-langkah sebagai berikut:

1. Menentukan daerah regional R masing-masing untuk rintangan dan target apakah berada di $R1$, $R2$, $R3$, atau $R4$.
2. Menentukan daerah sudut G antara $G1$ sampai $G8$ berdasarkan besar sudut θ antara garis robot ke rintangan dan garis robot ke target berdasarkan persamaan (1).

$$G = \begin{cases} G1, \theta \in [0, \pi/4] \\ G2, \theta \in [\pi/4, \pi/2] \\ G3, \theta \in [\pi/2, 3\pi/4] \\ G4, \theta \in [3\pi/4, \pi] \\ G5, \theta \in [\pi, 5\pi/4] \\ G6, \theta \in [5\pi/4, 3\pi/2] \\ G7, \theta \in [3\pi/2, 7\pi/4] \\ G8, \theta \in [7\pi/4, 2\pi] \end{cases} \quad (1).$$

3. Menentukan keadaan posisi robot di setiap waktu secara lengkap meliputi gabungan antara daerah regional target R_t , daerah regional rintangan terdekat R_o , dan jangkauan sudut θ . Keadaan ini dimodelkan menurut persamaan (2).

$$S_t = (R_t, R_o, G_n), n \in [1,8] \quad (2).$$

2.2 Penentuan Keadaan Transisi Robot

Penentuan keadaan transisi robot dilakukan untuk setiap langkah yang dijalani robot selama proses simulasi navigasi berlangsung. Keadaan transisi robot diputuskan berdasarkan pilihan kondisi pada persamaan (3) apakah WS (*Winning State*), SS (*Safe State*), NS (*Non-safe States*), atau FS (*Failure State*).

$$S = \begin{cases} WS, d_{r-t} \leq d_{win} \\ SS, d_{r-o} > d_{min} \\ NS, d_{col} < d_{r-o} \leq d_{min} \\ FS, d_{r-o} \leq d_{col} \end{cases} \quad (3).$$

dimana S merupakan keadaan transisi robot, d_{r-t} jarak robot dan target, d_{r-o} jarak robot dan rintangan, d_{min} jarak minimal aman dari rintangan, sedangkan d_{win} dan d_{col} masing-masing adalah jarak kemenangan dari target dan jarak tabrakan dari rintangan. Khusus pada keadaan lingkungan yang tidak aman, penentuan keadaan transisi dilakukan dua kali, yaitu sebelum dan sesudah melakukan aksi penghindaran rintangan. Hal ini bertujuan untuk mengetahui perubahan kondisi lingkungan akibat aksi penghindaran yang telah dilakukan, sehingga fungsi penghargaan dapat diberikan secara tepat.

2.3 Prediksi Pergerakan Rintangan dan Target

Prediksi pergerakan rintangan dan target dilakukan untuk meningkatkan efektifitas pencapaian

target. Pergerakan rintangan dan target diprediksi untuk posisi satu langkah di depan yaitu P_{n+1} . Arah pergerakan suatu obyek biasanya cenderung meneruskan arah pergerakan sebelumnya, sehingga dalam hal ini arah gerak dan posisi perpindahan dari P_{n-1} ke P_n digunakan sebagai acuan untuk pergerakan posisi dari P_n ke P_{n+1} .

Posisi prediksi rintangan dan target pada $n + 1$ dihitung dengan menggunakan persamaan (4-6).

$$P_{(n+1)x} = 2P_{nx} - P_{(n-1)x} \quad (4).$$

$$P_{(n+1)y} = 2P_{ny} - P_{(n-1)y} \quad (5).$$

$$P_{(n+1)} = [P_{(n+1)x} \ P_{(n+1)y}]^T \quad (6).$$

dimana $P_{(n+1)x}$ dan $P_{(n+1)y}$ adalah posisi x dan y pada langkah $n + 1$, $P_{(n-1)x}$ dan $P_{(n-1)y}$ adalah posisi x dan y pada langkah $n - 1$, sedangkan P_{nx} dan P_{ny} adalah posisi x dan y pada langkah n .

2.4 Penentuan Aksi Penghindaran Rintangan

Aksi penghindaran rintangan harus dilakukan robot ketika keadaan lingkungan simulasi dinyatakan tidak aman. Pada tahap pelatihan, arah orientasi penghindaran dilakukan dengan membandingkan aksi belok ke kanan dan ke kiri. Aksi yang dipilih adalah aksi belok yang mendekatkan robot ke arah target. Sudut penghindaran sebesar 45 derajat.

Sedangkan pada aksi penghindaran tahap pengujian, robot memilih aksi berdasarkan nilai pada Q -table. Robot mengecek baris di Q -table yang sesuai dengan keadaan yang tengah terjadi dihadapi dan mengambil keputusan berdasarkan nilai Q -value yang tersimpan. Jika aksi belok kanan memiliki Q -value tertinggi, maka robot belok ke kanan. Sebaliknya, jika aksi belok kiri memiliki Q -value tertinggi, maka robot belok ke kiri. Dan jika kedua aksi memiliki Q -value yang sama, maka robot akan belok ke salah satu arah tersebut secara random.

2.5 Pencarian Jalur Terpendek pencapaian Target

Pencarian jalur terpendek dilakukan jika kondisi lingkungan simulasi robot bergerak sudah dinyatakan aman. Algoritma yang digunakan pada tahap ini adalah ACO yang sudah dikembangkan oleh Garcia dengan formula probabilistik seperti pada persamaan (7). Dalam hal ini ξ adalah jarak Euclidian antara posisi $node$ robot dengan posisi prediksi target, dan β adalah nilai yang memperkuat pengaruh ξ , $range$ yang berlaku untuk β adalah $[0, \infty]$.

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t)}{\sum_{j \in N_i^k} \xi^\beta \tau_{ij}^\alpha(t)} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases} \quad (7).$$

2.6 Penentuan Fungsi Penghargaan Aksi Robot

Fungsi penghargaan merupakan evaluasi untuk aksi yang diambil robot bergerak pada suatu keadaan tertentu. Nilai ini digunakan untuk menghitung Q -value pada Q -table. Nilai penghargaan ini diberikan setelah penentuan keadaan transisi robot atas aksi yang telah dilakukan. Fungsi penghargaan dituliskan dalam persamaan (8).

$$r = \begin{cases} 2, S \subset SS \rightarrow WS \\ 1, S \subset NS \rightarrow SS \\ -1, S \subset SS \rightarrow NS \\ -1, S \subset NS \rightarrow NS, d_{r-o}(n+1) < d_{r-o}(n) \\ 0, S \subset NS \rightarrow NS, d_{r-o}(n+1) > d_{r-o}(n) \\ -2, S \subset NS \rightarrow FS \end{cases} \quad (8).$$

2.7 Perhitungan Fungsi Nilai pada Q -table

Fungsi nilai dihitung dengan menggunakan persamaan (9). Nilai ini kemudian disimpan di Q -table. Fungsi nilai atau Q -value ini terus diperbarui selama pelatihan berdasarkan pasangan keadaan dan aksi di lingkungan robot. Nilai-nilai inilah yang akan digunakan robot bergerak sebagai peraturan navigasi pada tahap pengujian.

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \text{Max}(Q, s_{t+1}) \quad (9).$$

dimana s_t merupakan keadaan di waktu t , a_t aksi yang dilakukan robot di waktu t , dan $r(s_t, a_t)$ penghargaan dari aksi yang baru saja dilakukan. $\text{Max}(Q, s_{t+1})$ adalah Q -value maksimum yang terhitung dari pengambilan semua aksi pada keadaan baru di waktu sebelumnya. γ adalah faktor pengurang. Dan jika kedua aksi memiliki Q -value yang sama, maka robot akan belok ke salah satu arah tersebut secara random.

3 SKENARIO UJI COBA

Pengukuran kinerja metode pada penelitian ini, dilakukan dengan menerapkan metode pada beberapa skenario uji coba yang telah ditentukan. Keseluruhan pola data pada skenario ini akan dijelaskan lebih detail pada setiap pengujian yang dijelaskan pada subbab selanjutnya. Skenario uji coba yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Tahap pelatihan menggunakan beberapa skenario ujicoba pelatihan yang berbeda. Dalam skenario ini keadaan lingkungan simulasi masih belum kompleks. Lingkungan hanya terdiri dari satu robot, satu rintangan dinamis, dan satu target. Pola pergerakan baik rintangan maupun target dibuat sederhana namun mampu mewakili beberapa kondisi khusus yang harus dipelajari oleh robot.
2. Tahap pengujian menggunakan dua pola skenario uji coba pengujian.
 - a. Skenario yang pertama bertujuan untuk menguji apakah robot bergerak sudah melakukan proses pembelajaran atau tidak.

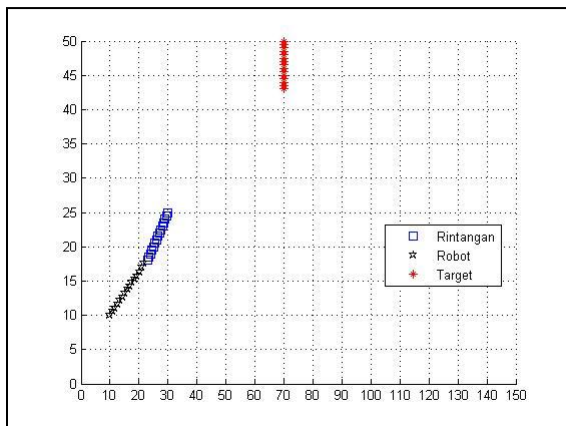
Pengujian ini dilakukan dengan meletakkan robot pada skenario uji coba pelatihan yang mana robot mengalami kegagalan dalam mencapai target atau bertabrakan. Pada skenario pengujian yang pertama ini sebuah rintangan statis juga ditambahkan agar lingkungan sedikit lebih kompleks.

- b. Skenario pengujian kedua menggunakan skenario kondisi lingkungan yang lebih kompleks. Dalam skenario ini diletakkan rintangan lebih banyak. Rintangan dapat bersifat semuanya dinamis, semuanya statis, atau sebagian dinamis dan sebagian statis. Masing-masing posisi awal rintangan dibuat acak dan pola pergerakan rintangan dinamis dibuat *random walk*. Demikian halnya dengan pola pergerakan target dibuat lebih kompleks dengan bentuk *sinusoidal*.
3. Tahap evaluasi menggunakan skenario lingkungan simulasi yang memiliki pola sama dengan skenario pengujian yang kedua. Pada tahap ini ada tiga parameter yang dievaluasi yaitu jumlah skenario pelatihan, angka kegagalan, dan waktu pencapaian target.
 - a. Evaluasi jumlah skenario pelatihan dilakukan dalam beberapa uji coba. Setiap uji coba menggunakan jumlah skenario pelatihan yang berbeda mulai dari 50 sampai 100. Sedangkan setiap pengujianya menggunakan jumlah skenario yang sama yaitu 500. Di akhir pengujian dihitung persentase kegagalan robot mencapai target. Metode yang diusulkan dinyatakan lebih baik jika jumlah skenario pelatihan yang dibutuhkan untuk mampu menghasilkan angka keberhasilan pencapaian target yang tinggi, lebih kecil daripada metode sebelumnya.
 - b. Evaluasi untuk mengukur angka kegagalan juga dilakukan dalam beberapa uji coba. Pada setiap uji coba dilakukan penambahan jumlah rintangan yang digunakan. Uji coba yang pertama menggunakan dua rintangan dinamis dan satu rintangan statis. Kemudian untuk uji coba selanjutnya dilakukan penambahan satu rintangan dinamis dan satu rintangan statis. Setiap pengujian menggunakan jumlah skenario yang sama yaitu 500 buah dan kemudian dihitung persentase kegagalan robot mencapai target di setiap akhir pengujian. Jika seiring penambahan rintangan persentase kegagalan robot lebih kecil dibanding metode sebelumnya, maka metode yang diusulkan dinyatakan lebih baik.
 - c. Evaluasi terhadap waktu pencapaian target dilakukan dengan membandingkan antara waktu robot mencapai target menggunakan metode yang diusulkan dengan metode pada

penelitian sebelumnya. Keadaan lingkungan skenario yang digunakan pada keduanya di setiap uji coba adalah sama. Sehingga dapat dihasilkan perbandingan waktu yang relevan. Metode yang mampu menghasilkan waktu pencapaian target lebih kecil akan dinyatakan lebih baik.

4 HASIL UJI COBA

Hasil uji coba pelatihan dan pengujian dengan skenario sederhana ditunjukkan pada Gambar 3 dan 4. Sedangkan hasil uji coba pengujian dengan skenario kompleks ditunjukkan pada Gambar 5.

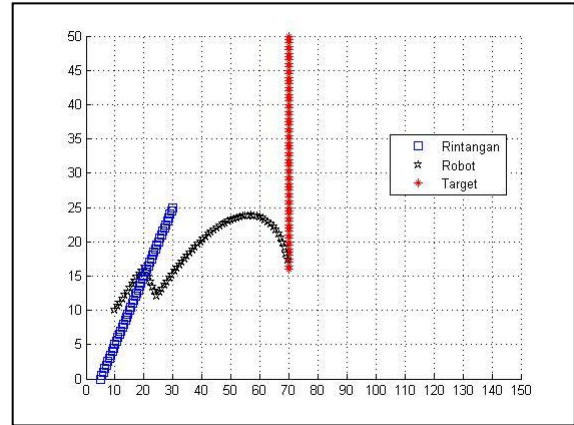


Gambar 3. Hasil Uji Coba Tahap Pelatihan di Lingkungan Sederhana

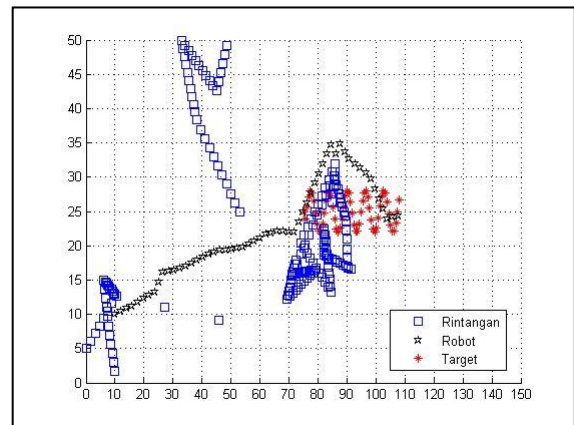
Pada skenario pelatihan Gambar 3, target bergerak vertikal dari titik $[70\ 50]^T$ dengan kecepatan $[0\ -0,5]^T$. Sebuah rintangan bergerak dari titik $[30\ 25]^T$ dengan kecepatan $[-0,5\ -0,5]^T$. Dan robot bergerak dengan kecepatan $[1\ 1]^T$ dari titik $[10\ 10]^T$. Pada langkah-langkah awal robot bergerak lurus menuju orientasi target sampai pada titik $[22\ 18]^T$. Pada posisi ini robot memasuki keadaan tidak aman yang mana rintangan berada di daerah R1 dan target di R1. Karena ini merupakan keadaan tidak aman pertama yang dilalui robot pada posisi ini, maka robot bergerak ke kiri yang mendekatkan ke posisi target. Namun aksi ini menghasilkan tabrakan dengan rintangan yang mengakibatkan nilai penghargaan negatif. Sehingga Q -value untuk aksi pada keadaan ini menurun dan proses navigasi berhenti pada detik ke-15 tepat di titik $[21,7\ 17,6]^T$. Pada waktu lain robot akan melewati kondisi ini dengan bergerak ke kanan.

Pengujian pada Gambar 4 menggunakan skenario yang sama dengan skenario pelatihan pada Gambar 3. Pada skenario tersebut di tahap pelatihan, robot gagal mencapai target karena bertabrakan dengan rintangan. Pada skenario pengujian ini, dilakukan pengecekan apakah robot berhasil mencapai target atau tidak setelah melalui pembelajaran tahap pelatihan. Pada posisi $[23\ 18]^T$

robot berhasil menghindari rintangan dengan belok ke kanan. Tidak seperti pada tahap pelatihan sebelumnya dimana robot pada posisi ini bertabrakan setelah belok ke kiri. Proses navigasi pada pengujian ini berakhir sampai robot berhasil mencapai target di posisi $[69,5\ 17]^T$ detik ke-69.



Gambar 4. Hasil Uji Coba Tahap Pengujian di Lingkungan Sederhana



Gambar 5. Hasil Uji Coba Tahap Pengujian di Lingkungan Kompleks

Pada lingkungan simulasi skenario pengujian Gambar 5 menggunakan enam rintangan kombinasi, yaitu dua rintangan statis dan empat rintangan dinamis. Keseluruhan posisi awal rintangan ditentukan secara acak. Demikian juga dengan jalur pergerakan rintangan yang dinamis berpola *random walk*. Rintangan statis 1 dan 2 walaupun ditentukan acak, tetapi untuk pengujian kali ini hasilnya disamakan dengan skenario pengujian pada penelitian sebelumnya yaitu masing-masing $[13\ 17]^T$ dan $[14\ 47]^T$. Posisi awal rintangan dinamis 3, 4, 5, dan 6 berurutan adalah $[72,1\ 33,3]^T$, $[65,9\ 34,7]^T$, $[80,2\ 28,5]^T$, dan $[96,2\ 43,4]^T$. Titik awal target ditentukan di $[75\ 25]^T$ dengan kecepatan $[0,6\ 3 * \sin(x)]^T$. Sedangkan robot tetap dimulai di titik awal $[10\ 10]^T$ dengan kecepatan $[2\ 2]^T$. Robot berhasil menghindari rintangan statis di awal navigasi. Robot memutuskan belok ke kiri karena ini merupakan aksi

penghindaran rintangan yang mendekatkan robot ke target. Setelah berada dalam kondisi aman, robot membangun jalur kembali dan terus bergerak ke arah target. Namun ketika akan mencapai target, robot menghadapi keadaan tidak aman lagi. Robot memilih aksi belok ke kiri dan bergerak mendekati target dengan terus menghindari dari posisi pergerakan rintangan. Robot berhasil mencapai target yang bergerak *sinusoidal* dalam waktu 65 di posisi [111 22]^T.

Hasil uji coba metode yang diusulkan dibandingkan dengan metode sebelumnya. Tabel evaluasi kedua metode ini ditunjukkan pada Tabel 1, Tabel 2, dan Tabel 3.

Tabel 1. Evaluasi Jumlah Pelatihan

Jumlah Skenario Pelatihan	Kegagalan Metode 1 (%)	Kegagalan Metode 2 (%)
5	16.2	9.8
15	15.8	8.8
30	13	8.2
45	11	5
65	2.6	10.2
75	2	11
90	15.6	10.6
100	17.6	9.4

Pada uji coba evaluasi terhadap parameter jumlah skenario pelatihan, robot dilatih menggunakan jumlah skenario yang beragam. Kemudian robot diuji dengan menggunakan 500 skenario pengujian. Pola skenario lingkungan untuk pelatihan dan pengujiannya adalah skenario acak. Angka kegagalan robot mencapai target kemudian dihitung pada setiap pengujian dan dibandingkan dengan metode penelitian sebelumnya. Hasil uji coba ini ditunjukkan pada Tabel 1.

Hubungan antara jumlah skenario pelatihan dengan angka kegagalan pada kedua metode, diketahui dengan menghitung nilai standar deviasi. Berdasarkan Tabel 1, nilai standar deviasi persentase angka kegagalan untuk metode 1 adalah 6,17 dan untuk metode 2 adalah 1,90. Ini menunjukkan bahwa dengan jumlah skenario pelatihan yang beragam, metode yang diusulkan lebih stabil menjaga performansinya daripada metode pada penelitian sebelumnya. Untuk mengukur peningkatan kinerja metode yang diusulkan terhadap metode penelitian sebelumnya, maka dihitung nilai rata-rata dari persentase angka kegagalan kedua metode. Nilai rata-rata dari persentase angka kegagalan metode 1 adalah 11,7% dan metode 2 adalah 9,1%. Dari kedua nilai ini dapat disimpulkan bahwa metode yang diusulkan memiliki performansi yang lebih baik dengan penurunan nilai kegagalan sebesar 22% dari metode penelitian sebelumnya.

Tabel 2. Evaluasi Jumlah Rintangan

Jumlah Rintangan	Kegagalan Metode 1 (%)	Kegagalan Metode 2 (%)
3	4	4.2
5	4.4	4.4
7	5.2	8
9	5.4	9
11	18.6	13.6
13	24.4	16

Parameter selanjutnya yang dievaluasi adalah angka kegagalan robot dalam mencapai target berdasarkan jumlah rintangan yang digunakan pada skenario lingkungan uji coba. Dalam uji coba ini robot dilatih dengan 50 skenario pelatihan. Sama seperti pada tahap evaluasi sebelumnya, pola skenario lingkungan untuk pelatihan dan pengujiannya pada evaluasi ini juga sama dengan skenario acak. Namun jumlah kombinasi rintangan statis dan dinamis dibuat berbeda. Jumlah rintangan yang digunakan dimulai dari dua rintangan dinamis dan satu rintangan statis. Kemudian untuk skenario selanjutnya setiap jenis rintangan ditambahkan satu per satu. Persentase angka kegagalan pencapaian target setiap uji coba kemudian dibandingkan dengan metode penelitian sebelumnya. Hasil uji coba ditunjukkan pada Tabel 2.

Tabel 2 menunjukkan bahwa jumlah rintangan sangat berpengaruh terhadap jumlah angka kegagalan robot mencapai target. Jika jumlah rintangan semakin sedikit, maka kecenderungan robot gagal mencapai target kecil. Hal ini dikarenakan probabilitas robot bertemu rintangan menjadi kecil, sehingga kemungkinan bertabrakan pun juga kecil. Dengan semakin bertambahnya jumlah rintangan, keadaan lingkungan simulasi robot akan semakin kompleks. Dengan jumlah rintangan yang kecil, metode yang diusulkan memiliki persentase angka kegagalan yang relatif seimbang dibandingkan dengan metode penelitian sebelumnya. Namun seiring kenaikan jumlah rintangan, persentase angka kegagalan metode yang diusulkan lebih kecil daripada metode penelitian sebelumnya.

Peningkatan kinerja metode yang diusulkan terhadap metode penelitian sebelumnya dapat diukur dengan menghitung nilai rata-rata dari persentase angka kegagalan kedua metode. Nilai rata-rata dari persentase angka kegagalan metode 1 adalah 10,3% dan metode 2 adalah 9,2%. Sehingga dapat disimpulkan bahwa metode yang diusulkan memiliki performansi yang lebih baik dengan penurunan nilai kegagalan sebesar 11% dari metode penelitian sebelumnya.

Waktu pencapaian target juga dievaluasi pada tahap evaluasi ini. Ini bertujuan untuk mengetahui apakah metode yang diusulkan mampu membuat robot mencapai target lebih optimal daripada metode

penelitian sebelumnya. Dalam uji coba ini robot dilatih menggunakan sebanyak 8 skenario uji coba pelatihan acak yang sama. Pola skenario lingkungan yang digunakan adalah skenario lingkungan acak. Di setiap akhir pengujian dihitung waktu keberhasilan robot mencapai target kemudian dilakukan perbandingan dengan metode penelitian sebelumnya. Hasil uji coba ditunjukkan pada Tabel 3.

Tabel 3. Evaluasi Waktu Pencapaian Target

Skenario	Waktu Metode 1 (detik)	Waktu Metode 2 (detik)	Selisih (detik)
ke-1	49	51	-2
ke-2	57	54	+3
ke-3	51	52	-1
ke-4	58	56	-2
ke-5	42	42	0
ke-6	52	52	0
ke-7	48	51	-3
ke-8	60	62	-2

Pengukuran perbandingan waktu pencapaian target kedua metode dilakukan dengan menghitung nilai rata-rata dari selisih waktu yang dihasilkan kedua metode. Berdasarkan Tabel 3 dihasilkan nilai rata-rata selisih waktu pencapaian target antara metode yang diusulkan dengan metode penelitian sebelumnya adalah sebesar -0,9. Nilai ini menunjukkan bahwa metode yang diusulkan rata-rata mampu mencapai target lebih lama satu detik daripada metode penelitian sebelumnya. Penurunan waktu ini disebabkan karena faktor perhitungan prediksi posisi target yang belum akurat untuk pola pergerakan *sinusoidal*. Perhitungan prediksi dalam penelitian ini lebih akurat jika digunakan pada pola pergerakan jalur yang linear. Hal ini dibuktikan pada tahap pelatihan, bahwa waktu robot mencapai target dengan menggunakan metode yang diusulkan menjadi lebih cepat daripada metode penelitian sebelumnya jika pola pergerakan target adalah linear.

5 KESIMPULAN

Dari hasil evaluasi uji coba berdasarkan jumlah skenario pelatihan, menunjukkan angka kegagalan metode yang diusulkan lebih kecil 22% dibandingkan metode sebelumnya. Sedangkan berdasarkan jumlah rintangan yang digunakan, angka kegagalan metode

yang diusulkan lebih kecil 11% daripada metode sebelumnya. Hasil evaluasi waktu pencapaian target menunjukkan metode yang diusulkan rata-rata mampu mencapai target lebih lama 0,9 detik dari metode sebelumnya. Metode yang diusulkan mampu mencapai target lebih cepat pada pola pergerakan target yang linear. Sebagai pengembangan penelitian selanjutnya dapat dilakukan dengan membuat prediksi jalur untuk pola pergerakan lain yang mampu mendefinisikan fungsi dari pergerakan baik rintangan ataupun target. Sehingga hasil prediksi jalur menjadi lebih akurat dan pencapaian target menjadi lebih optimal.

6 DAFTAR PUSTAKA

- [1] Filliat, G. dan Mayer, J. 2003. "Map Based Navigation in Mobile Robots: I. A Review of Localization Strategies". *Cognitive System Research*, 4:243-82.
- [2] Mayer, J. dan Filliat, G. 2003. "Map Based Navigation in Mobile Robots: II. A Review of Map Learning and Path Planning Strategies", *Cognitive System Research*, 4:283-317.
- [3] Canny, J.F. 1988. *The Complexity of Robot Motion Planning*. MIT Press. Cambridge.
- [4] Beatriz, A., Sossa, H., dan Vazquez, R. 2007. "Evolving Ant Colony System for Optimizing Path Planning in Mobile Robots". *IEEE*.
- [5] Garcia, P., Montiel, O., Castillo, O., Sepulveda, R., dan Melin, P. 2009. "Path Planning for Autonomous Mobile Robot Navigation with Ant Colony Optimization and Fuzzy Cost Function Evaluation". *ScienceDirect, Applied Soft Computing* 9, 1102-1110.
- [6] Zeng, B., Yimin, Y., dan Yisan, X., 2009. "Mobile Robot Navigation in Unknown Dynamic Environment Based on Ant Colony Algorithm". *IEEE*.
- [7] Hwang, K., Tan, S., dan Chen, C. 2004. "Cooperative Strategy on Adaptive Q-learning for Robot Soccer System". *IEEE Transactions on Fuzzy Systems*, 12(4):569-76.
- [8] Jaradat, M., Al-Rousan, M., dan Quadan, L. 2011. "Reinforcement Based Mobile Robot Navigation in Dynamic Environment". *ScienceDirect, Robotics and Computer-Integrated Manufacturing*, 135-149.