Navigasi *Mobile Robot Nonholonomic* Menggunakan *Fuzzy-Ant Colony System*

Dimas Bintang P , Trihastuti Agustinah, Rusdhianto Effendi AK
Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember
e-mail: trihastuti.agustinah@gmail.com

***Abstrak*—Navigasi robot merupakan penentuan kedudukan (*position*) dan arah perjalanan baik di medan sebenarnya atau di peta. Navigasi atau perencanaan jalur mempunyai peranan penting dalam teknologi *mobile robot*. Secara sederhana, melakukan perencanaan jalur yaitu menghasilkan titik-titik yang dapat dilewati (*feasible)* melalui informasi yang telah ada yaitu peta lingkungan robot. Pemodelan lingkungan adalah langkah pertama untuk *path planning*. Untuk lingkungan yang tidak diketahui, pemodelan dilakukan dengan informasi yang diperoleh dari sensor. Pada lingkungan yang nyata, noise dan keterbatasan akurasi dari sensor mengarah ke masalah dasar dalam modeling, yaitu kecepatan dan akurasi, yang mempengaruhi navigasi langsung secara *real time*. Untuk mengatasi masalah tersebut diperlukan *Fuzzy Description Environment* yang terdiri atas model *fuzzy* dari informasi yang didapat di lingkungan sekitar robot. Dari model tersebut akan menjadi acuan *path planner* yang menjadikan referensi pada *path planning*. Dari pemodelan map *environtment* akan diproses oleh *ant colony* sehingga didapatkan jalur menuju titik akhir. Dari hasil simulasi program didapatkan titik-titik *waypoint* dengan jarak terpendek yaitu 363,2724 satuan jarak.**

***Kata Kunci*—Perencanaan Jalur, *Fuzzy*, *Ant Colony*, *Mobile robot* Nonholonomik.**

I. PENDAHULUAN

DALAM beberapa dekade terakhir robot merupakan salah satu fokus dalam sains. Terlebih penggunaan robot sebagai pengganti manusia dalam hal eksplorasi dan mobilisasi daerah yang tidak mudah dilalui manusia. Sebagai dasar dari penggunaan robot itu, mobilisasi robot merupakan hal yang sangat penting dimana dalam hal menghindari halangan dan pengambilan keputusannya adalah hal penting. Untuk menghindari rintangan terlebih untuk lingkungan yang tidak diketahui dapat digunakan beberapa metode salah satunya dengan menggunakan *Fuzzy* untuk memodelkan lingkungan sekitar dan *Ant Colony System(ACS)* untuk mencari jalur optimal untuk menghindari rintangan.

Ant colony system (ACS) yang dikembangkan oleh M.Dorigo et al. Titik awal dari algoritma ini adalah untuk mengoptimalkan jalan dengan mensimulasikan fenomena semut mencari makan. ACS memiliki keuntungan dari co-processing dan paralel processing maka navigasi yang agak rumit dapat dicapai dengan algoritma sederhana. [1]

Algoritma koloni semut biasa digunakan dalam penyelesaian optimal seperti *scheduling* dan *routing* [2]. Namun pada penelitian ini, koloni semut digunakan sebagai penentuan jalur atau *path planning* pada navigasi robot.

II. DASAR TEORI

A. *Mobile robot*

Mobile robot atau robot bergerak merupakan teknologi yang berkembang pesat di dunia ini. *Mobile robot* merupakan robot yang memiliki kegunaan yaitu berpindah posisi dari satu titik ke titik yang lain, bergerak di sekitar lingkungannya. [3].

Untuk dapat membuat sebuah robot mobile minimal diperlukan pengetahuan tentang mikrokontroler dan sensor-sensor elektronik. Sensor digunakan untuk mengenali lingkungan pada robot sehingga robot dapat menghindari halangan yang ada di sekitar. Untuk robot bersensor biasanya berperan sebagai AMR atau *autonomous mobile robot*, selain itu juga ada robot menggunakan sensor tapi tanpa perlu kontrol cerdas untuk menentukan jalur jalannya yaitu AGV atau *autonomous guided vehicle* dimana robot ini menggunakan sensor untuk mengikuti jalur yang telah ditentukan berupa garis terang gelap (*line follower robot)*, dinding (*wall follower robot*), ataupun cahaya. Ada juga robot yang tanpa menggunakan sensor, namun digerakkan menggunakan kontroller yang dijalankan oleh manusia, robot ini sekarang sudah banyak dijual dipasarkan berupa mainan menggunakan remot kontrol.

Gambar 1. *Mobile robot* Qbot [4]

B. *Pergerakan Robot*

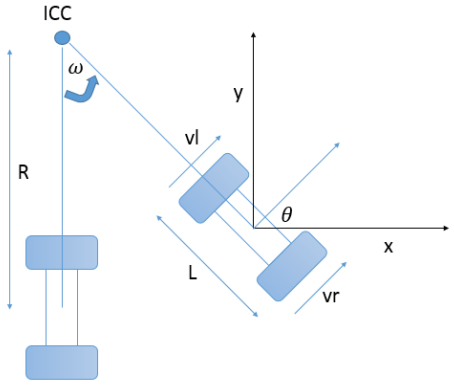
Arah gerak nonholonomik dibatasi oleh efek sentuhan arah roda yang menempel di atas permukaan. Robot tidak bisa bergerak bebas ke arah samping robot. Sehingga arah robot harus diubah terlebih dahulu sebelum berpindah karena arah gerak mengikuti arah hadap roda. Robot dengan sifat gerak nonholonomik mempunyai keterbatasan dalam arah gerak. Fungsi geometri tertentu yang berhubungan dengan orientasi harus dipenuhi untuk mendapatkan gerak yang sesuai.

Biasanya robot nonholonomik menggunakan dua roda di kanan dan kiri robot secara paralel dengan satu roda kastor

sebagai penyeimbang. Setiap roda digerakkan dengan motor yang independen atau disebut juga *Differential Drive Motor*.

C. *Differential Drive Motor*

Sebuah *differential drive robot* memiliki dua roda yang bisa bergerak secara independen. Perbedaan atau kesamaan kecepatan kedua roda menentukan apakah robot tersebut akan bergerak rotasi atau lurus.[5]



Gambar 2. Parameter *Differential Drive Robot*

Dimana L merupakan jarak antara dua roda kanan dan kiri. vr dan vl merupakan kecepatan translasi roda kanan dan kiri berturut-turut. Dan R merupakan jarak yang diukur dari ICC(*Instaneous Center of Curvature*) yang merupakan titik robot berotasi dengan titik tengah robot (titik tengah antara dua roda).[5]

$$ICC = [x - Rsin(\theta)\,, y + Rcos(\theta)] \quad (1)$$

Dengan ω merupakan perubahan rotasi robot terhadap ICC tiap waktu. Maka ketika $vl \neq vr$, ω akan sebanding dengan vl dan vr.

$$\omega\left(R + \frac{L}{2}\right) = vr \quad (2)$$

$$\omega\left(R - \frac{L}{2}\right) = vl \quad (3)$$

Jika persamaan (2) dan (3) digabungkan maka didapat persamaan (4) dan (5)

$$R = \frac{L}{2}\frac{vr + vl}{vr - vl} \quad (4)$$

$$\omega = \frac{vr - vl}{l} \quad (5)$$

Dari persamaan-persamaan diatas dapat ditarik beberapa hal, antara lain:[5]

1. Jika $vl = vr$, maka akan diperoleh gerak lurus linear ke depan. R bernilai tak terhingga dan tidak ada gerak rotasi yang mengakibatkan ω bernilai 0
2. Jika $vl = -vr$, maka R bernilai 0, dan akan diperoleh rotasi pada titik poros tengah robot $(L/2)$ atau berputar di tempat.
3. Jika $vl = 0$, maka roda kiri akan menjadi poros putar sehingga akan diperoleh $R = L/2$ dan gerak rotasi ke kiri begitu juga ketika $vl = 0$ maka akan menimbulkan gerak rotasi ke kanan

D. *Path Planning* [6]

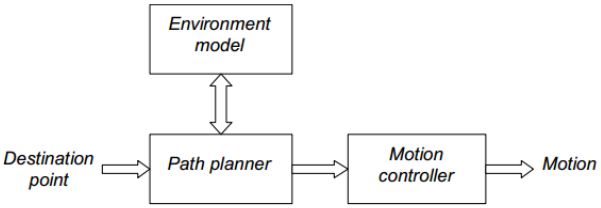
Path planning merupakan proses dimana robot menentukan jalur yang akan dilewati sehingga didapatkan titik yang merupakan titik tujuan. Dengan ini robot dapat berpindah ke titik tujuan dengan jarak terpendek dan tanpa terjadi tabrakan ke halangan yang ada di lingkungan sekitar.

Selain itu path planning juga dapat digunakan untuk meminimumkan belokan(*turning*), pengereman(*braking),* dan energi.

Path planning dapat dilakukan dalam dua keadaan, *statis* dan *dinamis*. Keadaan statis yaitu ketika lingkungan sekitar sudah diketahui oleh robot, dan tidak berubah sampai robot mengikuti path yang sudah dibentuk. Sedangkan keadaan dinamis, ketika robot mengetahui keadaan lingkungan sekitar namun terdapat halangan yang berubah-ubah/ berpindah secara tiba-tiba. Ataupun ketika robot tidak mengetahui keadaan lingkungan sama sekali namun sensor akan menangkap halangan jika ada pada jangkauannya. Metode dalam path planning bergantung pada keadaan tersebut sehingga tiap keadaan memerlukan perlakuan yang berbeda.

1) Global Path Planning

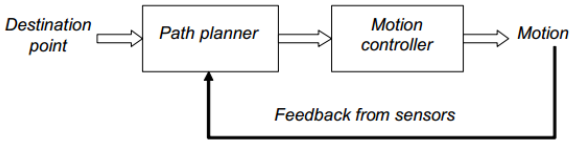
Jika telah diketahui pemodelan lingkungan sekitar robot maka path planning dapat dilakukan dengan meninjau pemodelan lingkungan secara langsung tanpa meninjau kemungkinan halangan secara dinamis. Karena Global Path Planning dapat dilakukan dengan keadaan statis



Gambar 3. Blok Diagram *Global Path Planning*

2) Local Path Planning

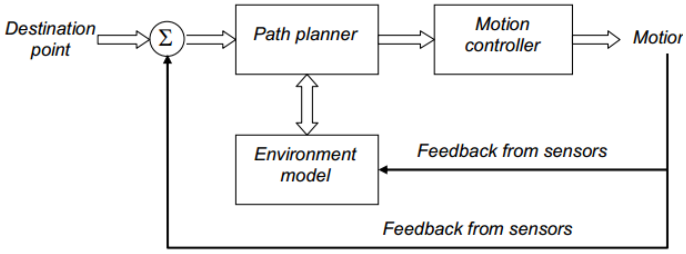
Local Path Planning dapat dilakukan tanpa mengetahui lingkungan sekitar robot. Sensor akan merasakan halangan dan tugas dari *path planner* yaitu memberikan titik ke kontroller yang sudah didapatkan sensor.



Gambar 4. Blok Diagram Local Path Planning

3) Dynamic Path Planning

Dynamic Path Planning merupakan gabungan dari Local Path Planning dan *Global Path Planning*. Metode ini yang paling sesuai dengan penelitian ini. Dimana keadaan halangan yang diraba oleh sensor akan dijadikan pemodelan lingkungan agar diproses secara global.



Gambar 5. Blok diagram *Dynamic Path Planning*

E. *Fuzzy Environment Model* [7]

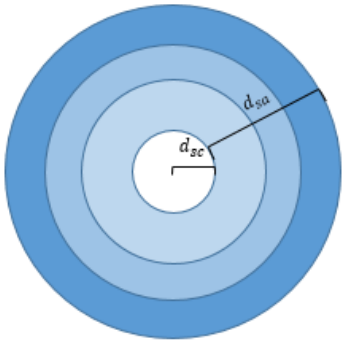
Logika *fuzzy* merupakan logika dimana penyelesaiannya bukan bernilai boolean namun bernilai kontinu dan samar bergantung pada derajat keanggotaan yang diperoleh. Semisal logika jarak yang bukan hanya jauh dan dekat, namun bernilai dekat, cukup dekat, normal, cukup jauh, jauh. Penentuan keanggotaan dan derajat kebenaran bergantung pada kita sebagai pemberi logika *fuzzy*.

Dalam hal ini *fuzzy* digunakan untuk memodelkan lingkungan robot yang berupa halangan (*obstacle*) yang digunakan sebagai parameter path yang akan dicari oleh program koloni semut.

Untuk memodelkan *fuzzy* didapat dari parameter *obstacle* yang didapat jaraknya tiap waktu $(O_t^1, O_t^2, O_t^3, \dots, O_t^q\}$ terhadap posisi robot saat waktu t, $(P_t(x_i, y_i))$. Didapatkan jarak $D(P_t, P_t^{oi}) = \sqrt{(x_t - x_t^{oi})^2 + (y_t - y_t^{oi})}$ dimana P_t^{oi} adalah state dari obstacle ke-i(i=1,2,..,q)

Membership function didapatkan dari hubungan antara jarak, radius *obstacle* jarak minimum aksi yang ditentukan dan jarak maksimum aksi yang kita tentukan. Untuk penelitian ini, aksi yang ditentukan yaitu sebagai nilai pembebanan probabilitas pada *ant colony* supaya gerakan semut tidak terlalu acak karena tidak dibatasi.

$$\mu_{p_t}(p_t, p_t^{oi})) = \begin{cases} 1, & if\ D(p_t, p_t^{oi}) \le R^{oi} + d_{sc} \\ 1 - \dfrac{D(p_t, p_t^{oi}) - d_{sc}}{d_{sa}}, & if\ d_{sc} < D(p_t, p_t^{oi}) \le d_{sc} + \\ 0, & lainnya \end{cases} \quad (6)$$

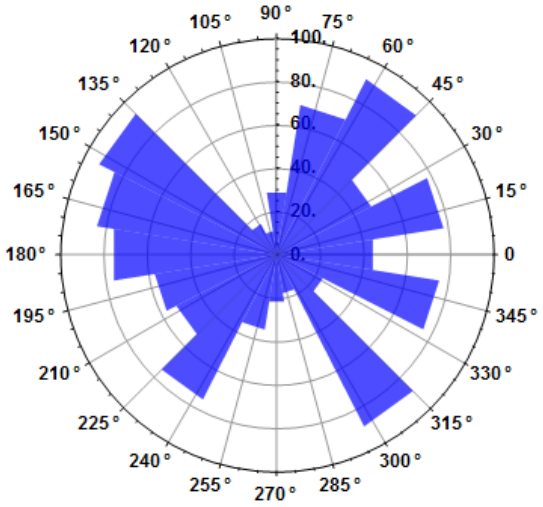


Gambar 6. *Membership function* dari Lingkungan Robot

F. *Polar Histogram* [4]

Polar histogram merupakan metode yang dapat digunakan untuk menggafmbarkan grafik dengan bentuk grafik berupa polar(lingkaran). Karena bentuknya polar, metode ini biasa digunakan sebagai pemodelan ataupun sistem penghindaran tabrakan *(collision avoidance).* Dimana besaran dari setiap derajatnya merupakan jarak *obstacle* terhadap titik robot.

Dengan mengubah jarak menjadi *density*/kepadatan halangan kita dapat mengkontrol robot untuk menghindari halangan. Jika kepadatan *obstacle* sedikit maka kita mengarahkan robot ke arah yang sedikit halangannya.



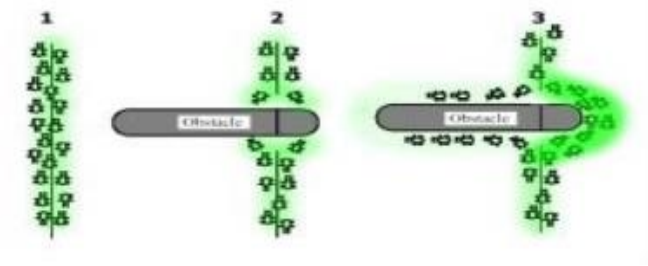
Gambar 7. *Polar Histogram*

Penggunaan *polar histogram* digunakan dalam pembebanan pemilihan *state* pada proses *ant colony.* Dimana titik yang akan dituju pada arah sudut dari robot memiliki pembebanan bergantung pada tingkat *density*. Untuk kasus ini semakin tinggi *density* maka semakin kecil nilai pembebanan pada titik tersebut. Sehingga kemungkinan besar titik tersebut tidak dipilih pada pemilihan *state.*

G. *Algoritma Koloni Semut (*Ant Colony Algorithm)

Ant Colony System (ACS) adalah suatu metode penyelesaian masalah optimasi yang berupa kumpulan beberapa algoritma yang menggunakan teknik probabilistik dan perilaku koloni semut dalam mencari makanan. Konsep ACS pertama kali diperkanalkan melalui algoritma *Ant System* (AS) pada tahun 1992 oleh Marco Dorigo dalam disertasinya.[2]

Ant Colony dari perilaku koloni semut yang dikenal sebagai sistem semut. Secara alamiah koloni semut mampu menemukan rute terpendek dalam perjalanan dari sarang menuju ke sumber makanan dan kembali lagi, pada saat semut berjalan, semut meninggalkan sebuah informasi yang disebut pheromone, di tempat yang dilaluinya dan menandai rute tersebut. Pheromone digunakan sebagai komunikasi antar semut pada saat membangun rute. Rute yang memiliki pheromone yang lebih tebal akan lebih cenderung digunakan sebagai semut lainnya. Jalur informasi dapat digunakan untuk perubahan lingkungan/rintangan.



Gambar 8. Jalur feromon semut akan lebih cenderung ke jalur terpendek

Semisal adanya obstacle yang menghalangi jalur robot secara tiba-tiba maka feromon akan di-update sehingga membentuk jalur baru dimana jalur yang kaya akan feromon cenderung pendek dan probabilitasnya lebih banyak semut yang melewati.[8]

III. PERANCANGAN SISTEM

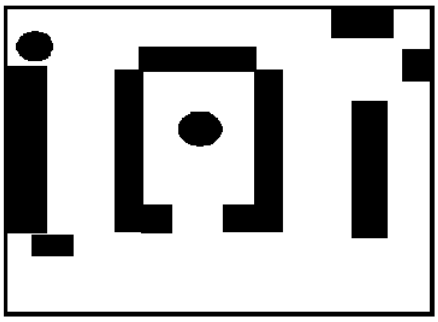
A. *Identifikasi Permasalahan*

Pada penelitian ni, permasalahan yang akan dibahas adalah permasalahan *path planning* yang bertujuan menemukan sebuah *path* terpendek antara dua titik yang ditentukan. Dengan menggunakan grafik peta yang merepresentasikan keadaan lingkungan robot secara nyata, simulasi akan bekerja mencari path yang menghubungkan antara dua titik tanpa tertabrak halangan. Peta yang dibuat berformat .png tanpa menentukan ketinggian dari halangan.

Setelah diproses dengan *software* MATLAB, gambar akan diberi koordinat, titik hitam akan bernilai 1, dan area putih merupakan yang dapat dilalui oleh robot.

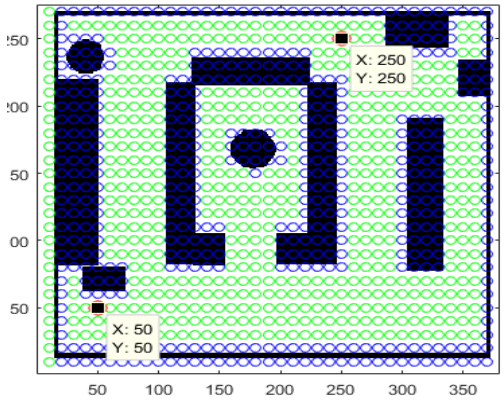
B. *Perancangan Model lingkungan Fuzzy*

Pemodelan *Fuzzy* dilakukan untuk memberikan *weight* pada algoritma koloni semut yang dilakukan pada tahap selanjutnya. Dengan menggunakan fungsi jarak, ditentukan nilai probabilitas feromon yang akan dilewati semut.



Gambar 9. Peta Lingkungan pada Simulasi Path Planning

Dari peta tersebut akan dimodelkan menjadi indeks yang merupakan halangan atau tempat yang dapat dilalui robot. Sebelumnya dilakukan pengkisian pada peta sehingga membatasi titik yang aman dilalui oleh robot. Pengkisian ini memisahkan titik yang dapat dilalui dan halangan yang ada dengan membatasi koordinat menjadi berkelipatan 10.



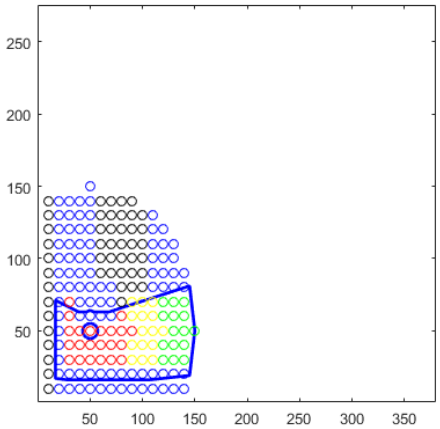
Gambar 10. Pengkisian Peta

Indeks tersebut merupakan bilangan logika dari peta yang telah dikisikan, jika indeks bernilai satu, maka koordinat pada indeks tersebut merupakan halangan dan tidak dapat dilalui robot.

C. *Pemodelan Lingkungan Kerja*

Pemodelan lingkungan kerja pada algoritma *path planning* berupa nilai *reachability* tiap titik yang ada pada radius tertentu terhadap titik yang sekarang dicapai dengan halangan, apakah titik *reach* tersebut memiliki terhalang oleh *obstacle* jika ditarik garis lurus ke titik awal, pemodelan ini merupakan representasi dari laser di sekitar robot, dimana area pada berkas laser merupakan titik titik yang dapat dilewati robot.

Hasil dari pemodelan ini yang berupa nilai *reachability* dari titik digambarkan oleh Gambar 11. Dimana warna memiliki representasi model yang akan digunakan untuk pembebanan pemilihan state pada algoritma koloni semut. Warna biru merupakan *obstacle* yang telah tereliminasi pada saat pemilihan *state*. Sedangkan warna hitam merupakan titik yang dapat dilalui robot namun tidak dapat dipilih karena robot akan menabrak jika menuju titik tersebut. Untuk titik yang dapat digunakan sebagai himpunan titik yang dapat dipilih maka direpresentasikan oleh warna merah, kuning dan hijau. Pemodelan lingkungan kerja ini dilakukan setiap *ant colony* melakukan pemilihan *state/*titik *waypoint.*



Gambar 11. Pembagian Warna Visual robot

D. *Perancangan Algoritma Path Planning*

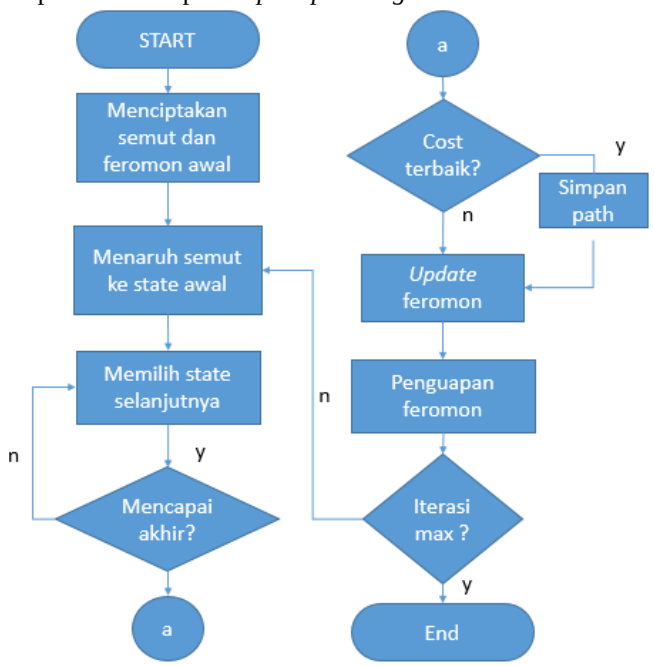
Algoritma koloni semut merupakan teknik pencarian yang meniru fenomena semut yang mencari makan. Langkah awal

yang dilakukan untuk merancang algoritma path planning berbasis koloni semut ini diawali dengan mencari titik yang memungkinkan dilalui oleh robot. Titik tersebut didapatkan dari pemodelan lingkungan kerja dan memiliki nilai pembebanan sendiri-sendiri bergantung *reachability* yang dimiliki titik tersebut.

Himpunan titik tersebut akan digunakan dalam pemilihan *state*. Yaitu melakukan pemilihan titik yang memungkinkan menggunakan *Roulette wheel selection*. Sehingga semut-semut akan berisi *titik path.*

Setelah itu dilakukan evaluasi *cost* dimana semut yang memiliki nilai path yang baik baik itu jarak maupun sudut belok yang dibuat, maka variabel dari semut tersebut akan disimpan.

Tahap selanjutnya yaitu melakukan penguapan feromon antar titik yang dilalui semut, feromon tersebut akan menguap jika dilewati oleh semut. Penguapan terjadi agar titik tersebut memiliki kemungkinan yang lebih sedikit sehingga tidak dapat dipilih kembali saat iterasi berikutnya agar tidak terjadi *dead end*(keadaan dimana hasil iterasi tetap berkutat di titik tersebut). Langkah tersebut dilakukan sampai iterasi berhenti. Sehingga didapatkan nilai optimal *path planning.*



Gambar 12.Diagram Alir *Path Planning*

1) Roulette Wheel Selection

Dari nilai yang diberikan kepada model oleh *fuzzy*, dan pemodelan lingkungan kerja. Akan digunakan untuk berat probabilitas acak dari titik yang akan dapat dilalui robot. Semakin besar nilai berat maka akan semakin sering dipilih oleh roulette wheel yang titik tersebut akan disimpan menjadi hasil yang dibawa semut. Nilai berat dari *fuzzy* dipadukan dengan feromon yang juga sebagai beban dari fungsi acak. Berikut persamaan probabilitas kumulatif tiap titik yang dinormalisasi.

$$P_{ij}^{k}(n) = \frac{\left[\tau_{ij}(n)\right]^{\alpha}\left[\eta_{1_{ij}}(P_i)\right]^{\beta_1}\left[\eta_{2_{ij}}(P_i)\right]^{\beta_2}}{\sum\left[\tau_{ij}(n)\right]^{\alpha}\left[\eta_{1_{ij}}(P_i)\right]^{\beta_1}\left[\eta_{2_{ij}}(P_i)\right]^{\beta_2}} \qquad (7)$$

Dimana $P_{ij}^{k}(n)$ merupakan probabilitas titik J dipilih dari titik asal robot i. $\tau_{ij}(n)$ merupakan besar feromon yang ada pada titik i menuju titik J. Nilai α dan β merupakan parameter penguat antara feromon dan pembebanan. Sedangkan η merupakan *attractiveness* atau nilai beban pada titik tersebut. Terdapat 2 nilai η, yaitu $\eta_1$merupakan nilai yang didapat dari pemodelan *fuzzy* titik tersebut yang bernilai $1-\mu_{p_t}(p_t, p_t^{oi})$. Sedangkan η_2 merupakan nilai berat yang didapatkan berasal dari pemodelan lingkungan kerja yang tiap area warna (Gambar 11) ditentukan pembebanannya. Seperti pada Tabel berikut.

Tabel 1.
Tabel pembebanan $\boldsymbol{\eta_2}$

Warna	Representasi	Jarak	*Weight*
Merah	Dapat dilalui	<40	1
Kuning	Dapat dilalui	40-70	2
Hijau	Dapat dilalui	70-100	3

Seperti digambarkan pada Tabel 1 mengacu pada Gambar 11 dimana semakin jauh (area hijau) maka nilai *attractiveness* diberikan pada titik semakin besar. Ini dilakukan untuk mencegah robot berjalan dengan menuju *waypoint* dengan jarak pendek yang menyebabkan pemrosesan pencarian jalur semakin lama juga karena probabilitas semakin banyak dipilih untuk mencapai tujuan. Nilai *weight* dapat kita tentukan juga jarak pemodelannya bergantung pada jarak pandang area robot maksimum.

2) Cost Function

Dari hasil yang dibawa tiap semut akan ditinjau derajat kualitasnya, atau yang disebut *Cost*. Pada fungsi ini, dapat ditentukan baik buruknya hasil tergantung dari apa yang kita inginkan. Misal, jumlah energi yang dikeluarkan robot juka memilih jalur tersebut, kecepatan robot mencapai titik tujuan, atau jarak yang dilalui robot tanpa menghiraukan pergerakan rotasi robot.

Untuk kasus ini, *cost function* yang ditentukan menggunakan persamaan jarak yang akan dilalui robot. Semakin dekat jarak, maka semakin bagus jalur tersebut.

$$Cost\ Distance = \sum_{k=1}^{n-1}\sqrt{(x_{k+1}-x_k)^2+(y_{k+1}-y_k)}$$

Dimana:

$n =$ jumlah titik yang dibawah semut

IV. HASIL PENGUJIAN DAN ANALISIS

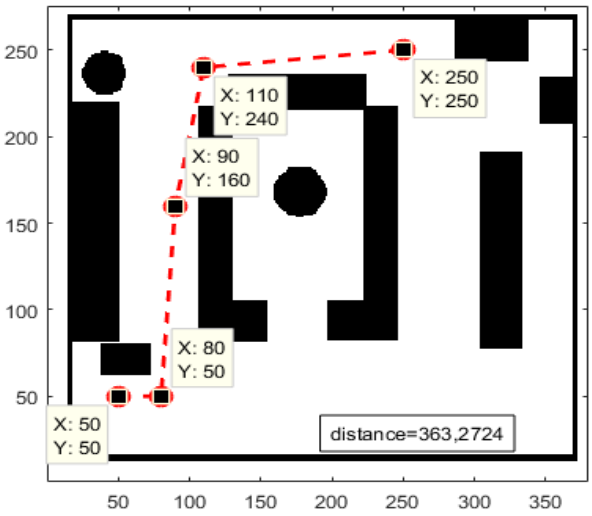
Bagian ini akan dibahas hasil yang disimulasikan oleh software matlab. Dengan aturan jumlah semut = 10 dan iterasi sebanyak 100 tiap menjalankan program, dengan titik awal=[50,50] dan titik akhir=[250,250]. Dengan melakukan beberapa eksperimen dengan parameter pada *ant colony* yang berbeda maka didapatkan hasil sebagai berikut.

Tabel 2
Tabel Nilai *Cost* untuk Tiap Eksperimen

Percobaan ke-	Alpha	Beta1	Beta2	τ_0	Best Cost
1.	1	1	1	1	363,2724

2.	1	1	1	2	390,9210
3.	1	1	2	2	387,9379
4.	1	1	2	1	385,2259
5.	1	2	1	1	385,5593
6.	1	2	1	2	385,2259
7.	2	1	1	2	392,5603
8.	2	1	1	1	384,4717

Untuk rute terpendek didapatkan dengan parameter $\alpha = 1, \beta_1 = 1,\ \beta_2 = 1, \tau_0 = 1$ sebesar 363,2724 dengan rute $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 50\ 80\ 90\ 110\ 250 \\ 50\ 50\ 160\ 240\ 250 \end{bmatrix}$.



Gambar 13. Hasil *Waypoint* dengan *Cost* Terbaik

V. KESIMPULAN

Algoritma ini mampu mendapatkan path yang memungkinkan untuk dilalui robot namun memiliki kelemahan tentang konvergensi hasil simulasi yang menyebabkan tidak mencapai titik optimal pada sekali menjalankan program

Simulasi yang dilakukan dengan 8 kali percobaan dapat memperoleh hasil *path planning* dengan jarak terpendek sebesar 363,2724 dengan rute: (x;y)=[50 80 90 110 250;50 50 160 240 250].

UCAPAN TERIMA KASIH

Penulis D.B.P mengucapkan terima kasih kepada Direktorat Pendidikan Tinggi, Departemen Pendidikan dan Kebudayaan Republik Indonesia yang telah memberikan dukungan finansial melalui Beasiswa Bidik Misi tahun 2013-2017.

DAFTAR PUSTAKA

[1] M. D. dan T. Stutzle, *Ant Colony Optimization*. The MIT Press, 2009.

[2] M. D. dan L. M. Gambardella, "Ant Colony System: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol.*, vol. 1, no. 1, pp. 53–66, 1997.

[3] T. S. dan T. R. G. B, "Design and development of an autonomous mobile smart vehicle: a mechatronics application," *Mechatronics 14*, pp. 491–514, 2014.

[4] *User Manual Quanser Qbot*. Inovative Educate, 2012.

[5] G. D. dan M. Jenkin, *Computational Principles of Mobile robotics*. New York: Cambridge University Press, 2010.

[6] T. Lehtla, "Introduction to Robotics," 2008.

[7] Y. Y. dan Y. X. B. ZENG, "Mobile robot Navigation in Unknown Dynamic Environment," in *Global Congress on Intelligent Systems*, 2009.

[8] A. R. dan D. P. Vinchurkar, "Robot Path Planning using An Ant Colony," *Int. J. Adv. Res. Artif. Intell.*, vol. 2, p. 65, 2013.