TESIS – EE1854401

EXPLORASI MULTI AGEN DALAM PENCARIAN TARGET MENGGUNAKAN MODIFIKASI ALGORITMA ANT COLONY PADA LINGKUNGAN YANG TIDAK DIKETAHUI

YOAN PURBOLINGGA
07111650020005

DOSEN PEMBIMBING
Prof. Dr. Ir. Achmad Jazidie, M.Eng.
Ir. Rusdhianto Effendi A.K, M.T

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK SISTEM PENGATURAN
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2019

TESIS – EE185401

EXPLORASI MULTI AGEN DALAM PENCARIAN TARGET MENGGUNAKAN MODIFIKASI ALGORITMA ANT COLONY PADA LINGKUNGAN YANG TIDAK DIKETAHUI

YOAN PURBOLINGGA
07111650020005

DOSEN PEMBIMBING
Prof. Dr. Ir. Achmad Jazidie, M.Eng.
Ir. Rusdhianto Effendi A.K, M.T

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK SISTEM PENGATURAN
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2019

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

oleh:

Yoan Purbolingga
NRP. 07111650020005

Tanggal Ujian : 28 Desember 2018
Periode Wisuda : Maret 2019

Disetujui oleh:

1. Prof. Dr. Ir. Achmad Jazidie, M.Eng. (Pembimbing I)
   NIP: 1959 0219 1986 1010 01

2. Ir. Rusdhianto Effendi A.K, M.T (Pembimbing II)
   NIP: 1957 0424 1985 0210 01

3. Prof. Ir. H. Abdullah Alkaff, M.Sc., Ph.D (Penguji)
   NIP: 1955 0123 1980 0310 02

4. Prof. Dr. Ir. Mohammad Nuh, DEA. (Penguji)
   NIP: 1959 0617 1984 0310 02

5. Dr. Trihastuti Agustinah, S.T., M.T. (Penguji)
   NIP: 1968 0812 1994 0320 01

Dekan Fakultas Teknologi Elektro

Dr. Tri Arief Sardjono, S.T., M.T.
NIP. 19700212199512100l

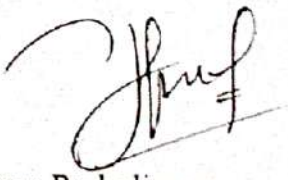
Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul "**EXPLORASI MULTI AGEN DALAM PENCARIAN TARGET MENGGUNAKAN MODIFIKASI ALGORITMA ANT COLONY PADA LINGKUNGAN YANG TIDAK DIKETAHUI**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2019



Yoan Purbolingga

NRP. 07111650020005

Halaman ini sengaja dikosongkan

EXPLORASI MULTI AGEN DALAM PENCARIAN TARGET MENGGUNAKAN MODIFIKASI ALGORITMA ANT COLONY PADA LINGKUNGAN YANG TIDAK DIKETAHUI

Nama mahasiswa : Yoan Purbolingga
NRP : 07111650020005
Pembimbing : 1. Prof. Dr. Ir. Achmad Jazidie, M.Eng.
2. Ir. Rusdhianto Effendi A.K, M.T

ABSTRAK

Beberapa dekade terakhir ini, banyak penelitian yang berhubungan dengan komputasi menggunakan perilaku alam sebagai dasar penelitiannya. Salah satu diantaranya *ant colony* (koloni semut). *Ant colony system algorithm* (ACS) termasuk dalam kelompok *swarm intelligence* yang bertujuan untuk menyelesaikan strategi masalah pencarian sasaran. Pada penelitian sebelumnya, metode koloni semut telah digunakan untuk menyelesaikan permasalahan dalam mencari jalur optimasi ke tempat target yang telah diketahui. Tetapi metode ini tidak dapat digunakan jika posisi target tidak diketahui.

Pada penelitian ini, metode yang digunakan adalah algoritma koloni semut yang telah dimodifikasi dengan menambahkan fungsi anti feromon, yaitu untuk membuat agen *swarm* selalu membuat keputusan untuk memilih jalur dengan jumlah feromon terkecil. Dalam pemilihan node selanjutnya algoritma yang diusulkan menghitung banyaknya feromon (parameter α) dan panjangnya jarak (parameter β). Pengujian dilakukan dengan beberapa keadaan, yaitu menguji kemampuan semua agen untuk mengexplorasi seluruh node tanpa target dan menguji kemampuan semua agen untuk menemukan target pada *simple* dan *complex environment* dengan kriteria minimum iterasi dan jarak tempuh.

Hasil simulasi menunjukkan bahwa algoritma koloni semut yang dimodifikasi dapat membuat agen *swarm* mengeksplorasi dalam mencari posisi target. Simulasi pertama tanpa target menggunakan 1-5 agen lebih efektif menggunakan parameter $\alpha = 1, \beta = 1$. Simulasi kedua dengan target menggunakan 1-5 agen lebih efektif menggunakan parameter $\alpha = 2, \beta = 1$ dengan minimum jarak = 380 cm, iterasi = 33 dari hasil pengujian. Simulasi ketiga mencari target pada lingkungan komplek lebih efektif menggunakan parameter $\alpha = 2, \beta = 1$ dengan minimum jarak = 470, iterasi = 57. Metode yang diusulkan mampu menemukan posisi target yang tidak diketahui dengan minimum iterasi dan jarak tempuh.

Kata kunci: *Ant Colony System, Search Problem, Swarm Algorithm, Exploration Method*

Halaman ini sengaja dikosongkan

EXPLORATION OF MULTI AGENT IN TARGET SEARCH USING MODIFICATION OF ANT COLONY ALGORITHM IN UNKNOWN ENVIRONMENT

By : Yoan Purbolingga
Student Identity Number : 07111650020005
Supervisor(s) : 1. Prof. Dr. Ir. Achmad Jazidie, M.Eng.
2. Ir. Rusdhianto Effendi A.K, M.T

ABSTRACT

In the last few decades, many studies related to computation have used natural behavior as the basis of their research. One of them is ant colony. Ant colony system algorithm (ACS) is included in the swarm intelligence group which aims to resolve the target search problem strategy. In previous studies, the method of ant colonies has been used to solve problems in finding an optimization path to a known target location. But this method cannot be used if the target position is unknown.

In this study, the method used is an ant colony algorithm that has been modified by adding anti-pheromone functions, namely to make swarm agents always make the decision to choose the path with the smallest number of pheromones. In the next node selection the proposed algorithm calculates the number of pheromones (parameters α) and the length of the distance (parameter β). Tests are carried out with several conditions, namely testing the ability of all agents to explore all nodes without targets and test the ability of all agents to find targets in simple and complex environments with minimum iteration criteria and distance traveled.

The simulation results show that the modified ant colony algorithm can make the swarm agent explore the target position. The first simulation without targets using 1-5 agents is more effective using parameters $\alpha = 1, \beta = 1$. The second simulation with targets using 1-5 agents is more effective using parameters $\alpha = 2, \beta = 1$ with a minimum distance = 380 cm, iteration = 33 from the test results. The third simulation looking for targets in complex environments is more effective using parameters $\alpha = 2, \beta = 1$ with minimum distance = 470, iterations = 57. The proposed method is able to find unknown target positions with minimum iterations and distance traveled.

Key words: *Ant Colony System, Search Problem, Swarm Algorithm, Exploration Method*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdulillahirabbil'alamin,

Segala puji dan syukur senantiasa kita panjatkan ke hadirat Allah SWT atas segala nikmat, kekuatan serta hidayah-Nya. Shalawat serta salam semoga tercurah kepada Rasulullah SAW, keluarga, sahabat dan para pengikut setianya, sehingga tesis ini dapat terselesaikan dengan baik. Tesis ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Magister pada Bidang Studi Teknik Sistem Pengaturan, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember (ITS) Surabaya yang berjudul:

"EXPLORASI MULTI AGEN DALAM PENCARIAN TARGET MENGGUNAKAN MODIFIKASI ALGORITMA ANT COLONY PADA LINGKUNGAN YANG TIDAK DIKETAHUI"

Selain itu penulis mengucapkan terima kasih penulis sampaikan kepada kedua orangtua atas kepercayaan yang telah diberikan, serta kepada semua pihak yang banyak membantu baik secara langsung maupun tidak langsung. Terima kasih kepada Prof. Dr. Ir. Achmad Jazidie, M.Eng. dan Ir. Rusdhianto Effendi A.K., M.T. selaku pembimbing Tesis atas segala bimbingan dan motivasi dari beliau hingga tesis ini terselesaikan.

Terima kasih kepada Bapak, Ibu, Saudara dan Istri penulis yang telah memberikan doa dan dukungan. Semoga buku Tesis ini dapat memberikan manfaat bagi pembaca.

Surabaya, 23 Januari 2019

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

Halaman ini sengaja dikosongkan

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Beberapa dekade terakhir ini, banyak penelitian yang berhubungan dengan komputasi menggunakan perilaku alam sebagai dasar penelitiannya, dimana salah satunya adalah *ant colony*. Yaitu sebuah algoritma dengan karakteristik perilakunya seperti semut. Penelitian mengenai *ant colony* telah menyumbangkan pengetahuan yang sangat besar, yang merupakan observasi bagaimana suatu kejadian yang ada di alam dapat digunakan untuk mengatasi permasalahan yang ada pada dunia teknologi masa kini. *Ant colony system algorithm* (ACS) termasuk dalam kelompok swarm intelligence, yang merupakan salah satu jenis pengembangan paradigma yang digunakan utnuk menyelesaikan masalah optimasi dimana inspirasi yang digunakan untuk memecahkan masalah tersebut berasal dari perilaku kumpulan (*swarm*) [8].

ACS biasanya digunakan untuk menyelesaikan kasus *search problem* dan persoalan yang kompleks dimana terdapat banyak variabel. Dalam beberapa permasalahan yang terkait dengan *searching problem* menentukan berapa banyak target yang akan dicari, kemampuan dari target apakah bersifat statis ataupun dinamis, faktor lingkungan, koordinasi antar robot. *Ant-based techniques* pertama kali digunakan oleh Dorigo et al dengan menggunakan *ant colony* optimization untuk menyelesaikan kasus *travelling salesmen problem* (TSP) [8]. Dalam ACS, setiap semut dalam kawanan yang berjalan akan meninggalkan semacam zat kimia berupa feromon pada jalur yang telah dilaluinya. Semut berikutnya, pada saat memilih jalur yang harus dipilih, biasanya cenderung memilih untuk mengikuti jalur dengan sinyal yang paling kuat, sehingga jalur terpendek akan ditemui karena lebih banyak semut yang akan melewati jalur tersebut. Beberapa penelitian yang telah dilakukan terkait *searching problem* dilakukan dengan berbagai macam pendekatan yaitu menggunakan *swarm algorithm for target searching* dimana algoritma tersebut digunakan untuk mencari target berdasarkan sumber bau.

Dengan menggunakan algoritma tersebut mendapatkan hasil dengan waktu yang seminimum mungkin untuk menemukan sumber bau [2].
Algoritma *ant colony system* masih belum masuk kualifikasi algoritma explorasi sehingga perlu adanya modifikasi.

Penelitian berikutnya membahas tentang membuat sebuah alternatif baru yaitu merubah fungsi kerja feromon normal menjadi anti feromon dimana feromon normal selalu mengutamakan jalur dengan tingkat feromon terbesar sedangkan penggunaan ati feromon cenderung memilih jalur dengan tingkat feromon terkecil yang bertujuan supaya setiap semut dapat mengexplorasi lintasan [4].

Jika dikondisikan pada kasus tertentu apabila posisi target maupun lokasi dari target tidak diketahui maka robot harus dapat menjelajah seluruh lintasan agar dapat menemukan target dengan cara memanfaatkan jejak yang dilewati pada lintasan dengan memanfaatkan anti feromon sebagai alternatif baru. Atas dasar inilah, maka ide dari penelitian ini ialah merancang sebuah algoritma *ant colony* dimana mempunyai kriteria meninggalkan jejak yang telah dilewati yang akan dimanfaatkan untuk *swarm agent* dapat menjelajah seluruh lintasan pada lingkungan yang tidak diketahui.

1.2 Rumusan Masalah

Bagaimana membuat sebuah algoritma explorasi yang dapat menjelajah lintasan pada lingkungan yang tidak diketahui dalam mencari target melalui modifikasi algoritma *ant colony*.

1.3 Tujuan

Tujuan dari penelitian ini adalah menghasilkan sebuah algoritma *ant colony* yang dapat menjelajah seluruh lintasan pada lingkungan yang tidak diketahui melalui informasi jejak dari tiap agent.

1.4 Batasan Masalah

Pada penelitian ini, batasan masalah nya meliputi :

a. Semua agen menggunakan *swarm* partikel
b. Tidak memperhitungkan batas waktu untuk menemukan target

1.5 Kontribusi

Menerapkan sebuah algoritma *ant colony* yang dimodifikasi. Dengan menambahkan fungsi berupa informasi heuristik dan fungsi global travel search serta anti feromon, dimana setiap agen dapat mencari target dengan memilih jejak feromon yang paling terkecil, sehingga tiap agen dapat mengexplorasi lintasan.

Halaman ini sengaja dikosongkan

BAB 2
KAJIAN PUSTAKA

2.1 Kajian Penelitian Terkait

2.1.1 Swarm Robotic Odor Source Localization Using Ant Colony Algorithm [2]

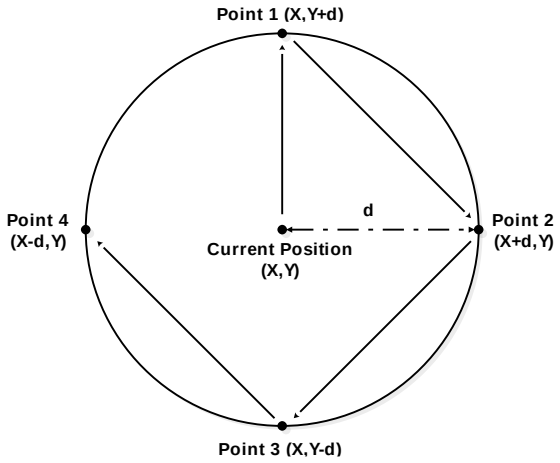
Pada penelitan yang dilakukan oleh Yuhua Zou, dkk ini bertujuan untuk menyelesaikan permasalahan *target searching*. Beberapa strategi pencarian sumber bau berdasarkan algoritma *ant colony* yang diusulkan, memungkinkan sistem multi agen untuk mencari sumber bau yang ada di lingkungan dalam ruangan. Yang berperan mengontrol *swarm particle*. Metode ini merupakan algoritma *ant colony* yang sudah dimodifikasi dengan adanya penambahan unsur *local travel search* dan *global travel search.*

Strategi pencarian mencakup dua bagian, yaitu strategi pelacakan bau dan strategi lokalisasi sumber bau. Strategi pelacakan terdiri dari algoritma koloni semut yang dimodifikasi. Strategi lokalisasi adalah prosedur verifikasi sumber bau yang dimasukkan ke dalam proses pelacakan untuk melokalisasi berbagai sumber bau.

Algoritma *ant colony* yang dimodifikasi mencakup tiga tahap yaitu :

1. *Local travel search*

   Area lokal adalah lingkaran dengan jari-jari *d*. Terlihat pada gambar 2.1.



Gambar 2.1 Daerah lokal dan urutan pencarian lokal secara traversal [2]

2. *Global travel search*

Probability migration: *searcher i* akan pindah ke posisi baru berdasarkan probabilitas berikut:

$$P_{i,j}^{k} = \frac{\tau_j(t)^{\alpha}\eta_{ij}(t)^{\beta}}{\sum_{j=1}^{m}\tau_j(t)^{\alpha}\eta_{ij}(t)^{\beta}}, (i = 1,2,3.....,n) \quad (2.1)$$

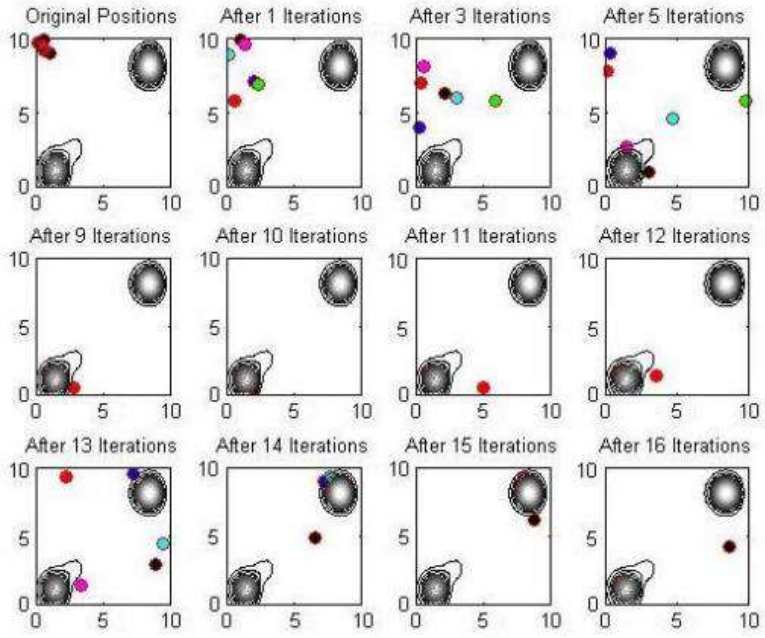
Dimana $\tau_j(t)$ mewakili feromon, $\eta_{ij}(t)$ adalah informasi heuristik *node i* ke *node j*. Pada penelitian ini juga menambahkan tiga jenis informasi heuristik yaitu :

$$\eta_{ij}(t) = \begin{cases} e^{C_{ij}(t)}/e^{D_{ij}(t)}, & j \neq i, j \in Searchers \\ e^{C_i(t)}, & j = i, j \in Searchers \\ 0, \end{cases} \quad (2.2)$$

$$\eta_{ij}(t) = \begin{cases} e^{C_{ij}(t)}.e^{D_{ij}(t)}, & j \neq i, j \in Searchers \\ e^{C_i(t)}, & j = i, j \in Searchers \\ 0, \end{cases} \quad (2.3)$$

$$\eta_{ij}(t) = \begin{cases} e^{C_{ij}(t)}, & j \neq i, j \in Searchers \\ e^{C_i(t)}, & j = i, j \in Searchers \\ 0, \end{cases} \quad (2.4)$$

Hasil simulasi menggunakan metode ant colony yang telah dimodifikasi:



Gambar 2.2 Distribusi peta robot saat mencari dua sumber [2]

Respon dari simulasi diatas menambahkan fungsi *local travel search* dan *global search* menambah performa dan akurat dalam menemukan target tetapi terdapat beberapa kelemahan dipenelitian ini penerapan metode *ant colony* yang telah dimodifikasi hanya diuji pada linkungan tanpa hambatan, sehingga algoritma tersebut belum teruji pada lingkungan yang lebih komplek dan juga setiap agen saling mengikuti agen yang lain dalam menemukan target. Oleh karena itu penulis mengajukan metode algoritma *ant colony* untuk membuat setiap agent lebih terkoordinasi dalam mencari target pada lingkungan yang lebih komplek.

2.1.2 Ant Colony System Based Mobile Robot Path Planning [3]

Penelitian yang dilakukan oleh Shong Hiang Chia, dkk menyelesaikan *problem searching* target dengan program *motion path searching food*. Yaitu mencari target dengan jalur terpendek menggunakan metode *ant colony optimization* dengan jalur bebas tabrakan. Persamaan untuk menghitung probabilitas transisi dalam algoritma *ant colony* adalah sebagai berikut:

$$P_{i,j}^{k} = \frac{(\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta}}{\sum_{l \in N_i^k}(\tau_{i,j})^{\alpha}(\eta_{i,j})^{\beta}}, if \quad j \in N_i^k \tag{2.5}$$

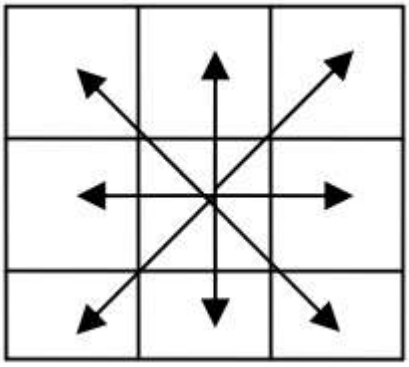
Penguapan feromon diimplementasikan oleh:

$$\tau_{i,j} = (1-\rho)\tau_{i,j} \tag{2.6}$$

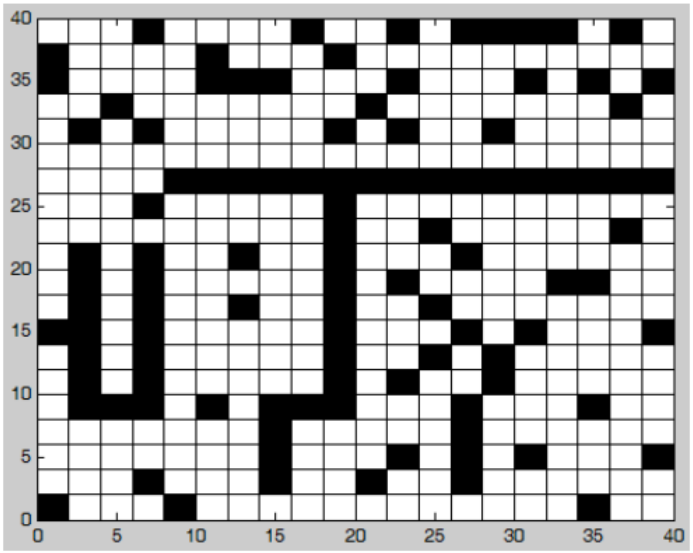
Setelah penguapan semua agen meninggalkan feromon pada jalur yang telah mereka lewati. Perubahan feromon adalah sebagai berikut:

$$\tau_{i,j} = \tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k}, \forall (i,j) \in L \tag{2.7}$$

Pada experimen memprogramkan arah gerak (*ant*) di platform grid. Arah robot *mobile* memiliki delapan pilihan yang akan ditunjukkan pada Gambar 2.3 dan platform yang digunakan pada sistem *ant colony* ditunjukkan pada Gambar 2.4.

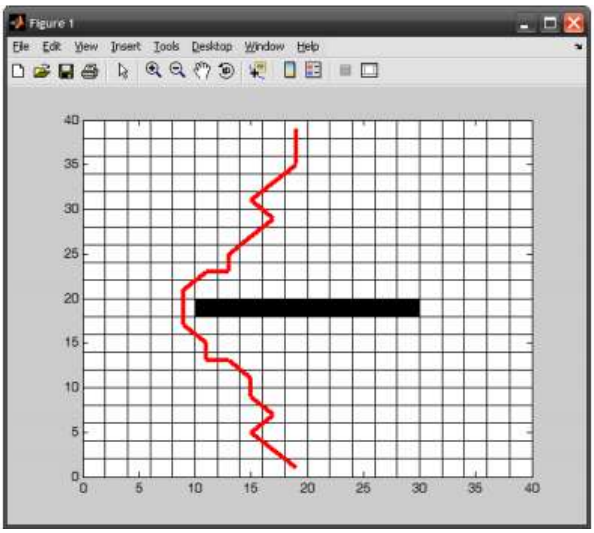


Gambar 2.3 Arah gerak mobile robot (*ant*) [3]

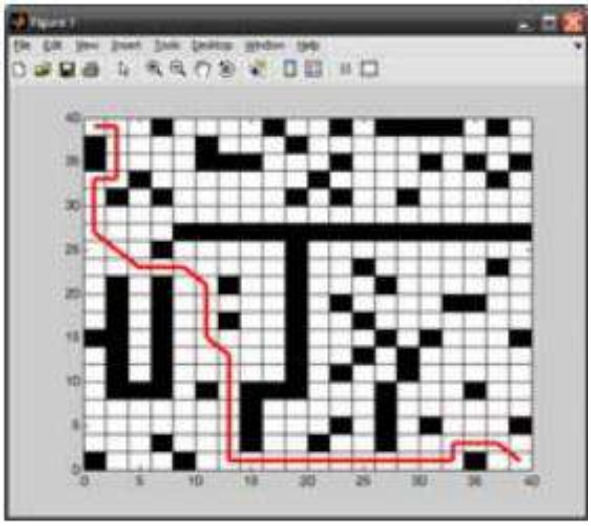


Gambar 2.4 Platform grid dari sistem koloni semut [3]

Hasil simulasi dapat dilihat pada Gambar 2.5 dan Gambar 2.6 terlihat bahwa untuk mencari suatu target menggunakan metode ant colony optimization yang disajikan pada penelitian ini mendapat hasil yang maksimal, jalur yang direncanakan telah dilalui oleh ant tanpa adanya tabrakan terhadap *obstacle* sehingga meminimumkan waktu percarian target. Tetapi penelitian ini hanya menguji untuk 1 *mobile robot* (*ant*).



Gambar 2.5 Jalur mobile robot dengan "—" tipe *obstacle* [3]



Gambar 2.6 Jalur yang dilalui *mobile robot* dilingkungan yang lebih komplek [3]

2.1.3 Anti-pheromone as a Tool for Better Exploration of Search Space [4]

Penelitian yang dilakukan oleh James Montgomery, dkk. Dengan teori banyak hewan menggunakan zat kimia yang dikenal sebagai feromon untuk menginduksi perubahan perilaku pada anggota lain dari spesies yang sama. Penggunaan feromon oleh semut secara khusus telah mengarah pada pengembangan sejumlah analog komputasional perilaku koloni semut termasuk ant colony optimisation. Tetapi interaksi antara feromon yang berbeda ini sebagian besar merupakan perluasan sederhana dari kriteria tunggal, algoritma ant koloni

tunggal. Makalah ini menginvestigasi suatu bentuk interaksi alternatif antara feromon normal dan anti feromon.

Explorer ants merupakan suatu karakteristik algoritma *ant colony* dimana sejumlah kecil semut dipilih untuk berperilaku berbeda dari semut lain tertarik ke daerah dengan feromon kecil. Semut penjelajah ini mempengaruhi lingkungan mereka dengan menyimpan feromon dengan cara yang sama seperti semut normal, hanya preferensi mereka untuk feromon yang ada dibalik. Persamaan 2.8 dan 2.9 mengungkapkan bagaimana semut penjelajah yang berada di kota *r* memilih kota berikutnya untuk pergi ke *s*.

$$s = \begin{cases} \arg\max_{u \in J_k(r)} \left\{ \left[\tau_{\max} - \tau(r,u)\right] \cdot \left[d(r,u)\right]^{\beta} \right\}, & if \ \ q \le q_0 \\ Equation\ 2.9 \end{cases} \tag{2.8}$$

$$P_k(r,s) = \begin{cases} \dfrac{\left[\tau_{\max} - \tau(r,s)\right] \cdot \left[d(r,s)\right]^{\beta}}{\sum_{u \in J_k(r)} \left[\tau_{\max} - \tau(r,u)\right] \cdot \left[d(r,u)\right]^{\beta}}, & if \ \ s \in NJ_k(r) \\ 0 \end{cases} \tag{2.9}$$

2.2 Teori Dasar

Untuk mendukung penelitian ini, dibutuhkan beberapa dasar teori yang akan dipergunakan. Pembahasan pada dasar teori ini meliputi dasar *searching algorithm*, metode *ant colony optimization*.

2.2.1 Dasar Searching Algorithm [1]

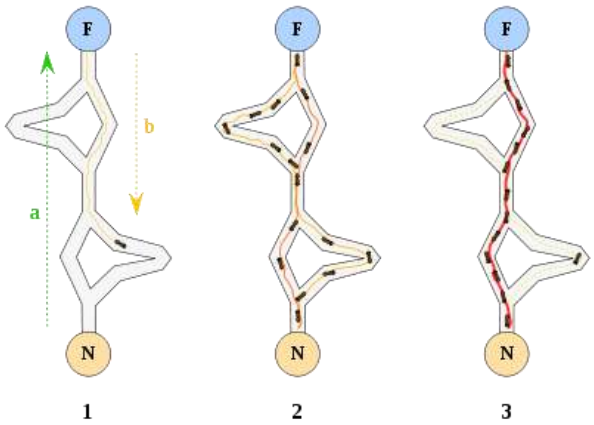
Setiap algoritma pencarian target harus memiliki batasan berikut:

a. Algoritma pencarian harus didistribusikan di antara robot. Algoritma sebaiknya tidak diimplementasikan hanya pada satu robot karena sistem akan gagal jika robot tersebut gagal.
b. Algoritma pencarian harus sederhana secara komputasi. Karena robot memiliki keterbatasan dalam kekuatan prosesor, memori dan baterai sehingga algoritma pencarian perlu disesuaikan dengan keterbatasan tersebut.
c. Algoritma harus terukur dan batas atas untuk jumlah robot akan diatur sesuai dengan hubungan komunikasi antar robot. Untuk alasan ini, algoritma pencarian membutuhkan mekanisme yang sesuai untuk berbagi informasi antar robot agar tidak menunggu lama untuk mendapat informasi dari robot lain. Jika tidak, peningkatan jumlah robot akan menyebabkan sistem *break down*.

2.2.2 Metode Ant Colony Optimization (ACO) [5]

a. Perilaku semut

Ant colony optimization didasarkan pada perilaku bekerjasama dari koloni semut dalam mencari sumber makanan yang ternyata secara alami mencari jalur terpendek.



Gambar 2.7 Perilaku semut dalam mencari makanan [5]

Seekor semut *k* pada simpul *i* akan memilih simpul *j* yang dituju pada layer berikutnya dengan probabilitas:

$$s = \begin{cases} \dfrac{\tau_{i,j}^{\alpha}}{\sum_{j \in N_i^{(k)}} \tau_{i,j}^{\alpha}}, & jika\ j \in N_i^{(k)} \\ 0 & jika\ j \notin N_i^{(k)} \end{cases} \qquad (2.10)$$

Dimana α menunjukkan derajat kepentingan feromon dan $N_i^{(k)}$ adalah pilihan yang dipunyai semut *k* pada saat ia berada pada simpul *i*. *Neighborhood* dari semut *k* pada simpul *i* akan mengandung semua simpul yang bisa dituju yang tersambung secara langsung ke simpul *i*, kecuali simpul yang sudah dikunjungi sebelumnya.

b. Penambahan dan penguapan feromon

Seekor semut *k* ketika melewati ruas akan meninggalkan feromon. Jumlah feromon yang terdapat pada ruas *i* dan *j* setelah dilewati semut *k* diberikan dengan rumus:

$$\tau_{i,j} \leftarrow \tau_{i,j} + \Delta\tau^{k} \qquad (2.11)$$

Dengan meningkatnya nilai feromon pada ruas *i* - *j* , maka kemungkinan ruas ini yang akan dipilih lagi pada iterasi berikutnya semakin besar. Setelah sejumlah

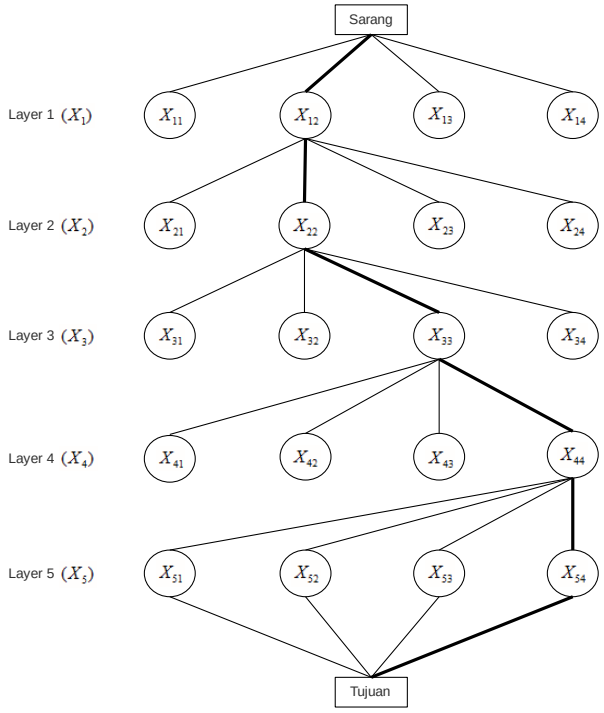
simpul dilewati maka akan terjadi penguapan feromon dengan aturan sebagai berikut:

$$\tau_{i,j} \leftarrow (1-\rho)\tau_{i,j} : \forall (i,j) \in A \tag{2.12}$$

Dimana $\rho \in (0,1)$ adalah parameter tingkat penguapan dan *A* menyatakan segmen atau ruas yang sudah dilalui oleh semut *k* sebagai bagian dari lintasan dari sarangnya menuju makanan. Penurunan jumlah feromon memungkinkan semut untuk mengekplorasi lintasan yang berbeda selama proses pencarian.ini juga akan menghilangkan kemungkinan memilih lintasan yang kurang bagus. Selain itu, ini juga membantu membatasi nilai maksimum yang dicapai oleh suatu lintasan feromon. Jumlah feromon yang ditambahkan pada ruas $i-j$ oleh semut *k* diberikan sebagai:

$$\Delta\tau_{i,j}^{(k)} = \frac{Q}{L_k} \tag{2.13}$$

Dimana *Q* adalah konstanta dan L_k adalah lintasan terpendek yang dilalui semut *k*, nilai *Q* biasanya ditentukan oleh user. Representasi grafis proses ant colony optimization dalam bentuk network multi layer terlihat seperti Gambar 2.8.



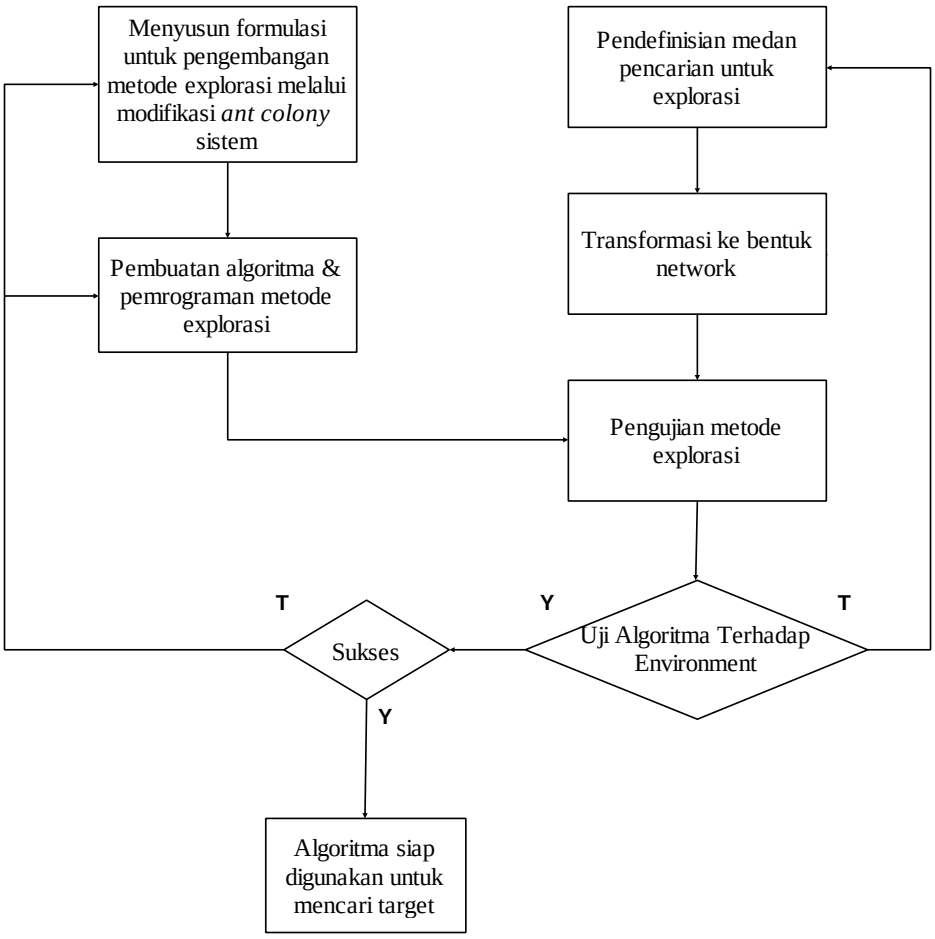
Gambar 2.8 Representasi grafis proses ACO dalam bentuk network multi-layer[5]

BAB 3
METODE PENELITIAN

Pada bab ini membahas tahapan yang akan dilakukan dalam proses perancangan sistem. Proses perancangan yang akan dilakukan meliputi perancangan algoritma *ant colony system* yang telah dimodifikasi dan pemodelan perancangan konseptual dari permasalahan pencarian target.

3.1 Rancangan konseptual sistem

Rancangan sistem pada penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Rancangan sistem

3.2 Formulasi metode explorasi

Formulasi dari metode explorasi menggunakan *ant colony* sistem didapat melalui sinyal jejak berupa feromon yang dilewati tiap agen. Untuk dapat mengexplorasi seluruh medan pencarian dilingkungan yang tidak diketahui dimana setiap agen akan memilih ruas dengan tingkat pheromones terkecil.

Tabel 3.1 Daftar simbol

No	Simbol	Keterangan
1	k	indek *Ants* (agen)
2	α	Bobot intensitas feromon
3	β	Bobot intensitas visibility
4	ρ	*Evaporation* feromon global
5	τ_0	Intensitas feromon awal
6	τ_{max}	Tingkat feromon maksimum
7	q	Random [0,1]
8	q_0	Parameter pembanding nilai *q*
9	$\tau_{i,j}$	Nilai intensitas feromon antara node *i* dan node *j*
10	$\eta_{i,j}$	Jarak antara node *i* dan node *j*
11	$N_k(i)$	list node node yang belum dikunjung oleh *ants k*
12	j	Node berikutnya yang akan dipilih
13	$P_k(i,j)$	Probabilitas *ants k* yang berada pada node *i* memilih node *j* untuk tujuan selanjutnya

Aturan transisi semut *k* yang berada pada node *i* akan memilih node *j*, menurut persamaan berikut:

$$j = \begin{cases} \arg\max_{u \in N_k(i)} \left\{ [\tau \max - \tau(i,u)]^{\alpha} \cdot [\eta(i,u)]^{\beta} \right\} \\ Equation\, 3.2 \end{cases} \quad (3.1)$$

Jika $q \le q_0 = j$ (exploitasi)

Jika tidak $P_k(i, j)$ (explorasi)

Dimana q adalah bilangan random dalam [0,1], adalah sebuah parameter pembanding bilangan random, dan persamaan 3.2 probabilitas dari semut k pada node i memilih untuk menuju node j.

$$P_k(i,j) = \begin{cases} \dfrac{[\tau_{\max} - \tau(i,j)]^{\alpha} \cdot [\eta(i,j)]^{\beta}}{\sum_{u \in N_k(i)} [\tau_{\max} - \tau(i,u)]^{\alpha} \cdot [\eta(i,u)]^{\beta}} \\ 0 \end{cases} \quad (3.2)$$

Penambahan dan penguapan *pheromone*:

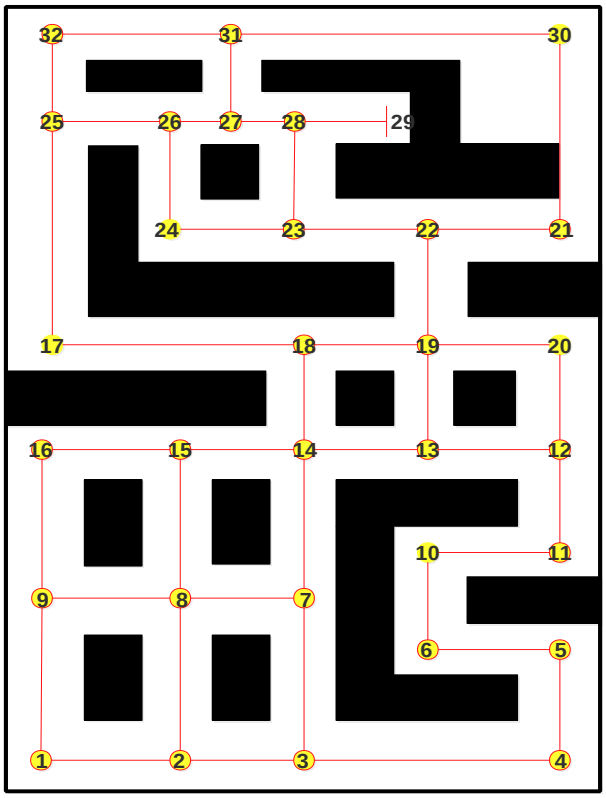
Setiap agen ketika melewati ruas akan meninggalkan feromon. Jumlah feromon yang terdapat pada ruas *i,j* setelah dilewati agen diberikan dengan persamaan 3.3

$$\tau_{i,j} \leftarrow (1-\rho)\,\tau_{i,j} \quad (3.3)$$

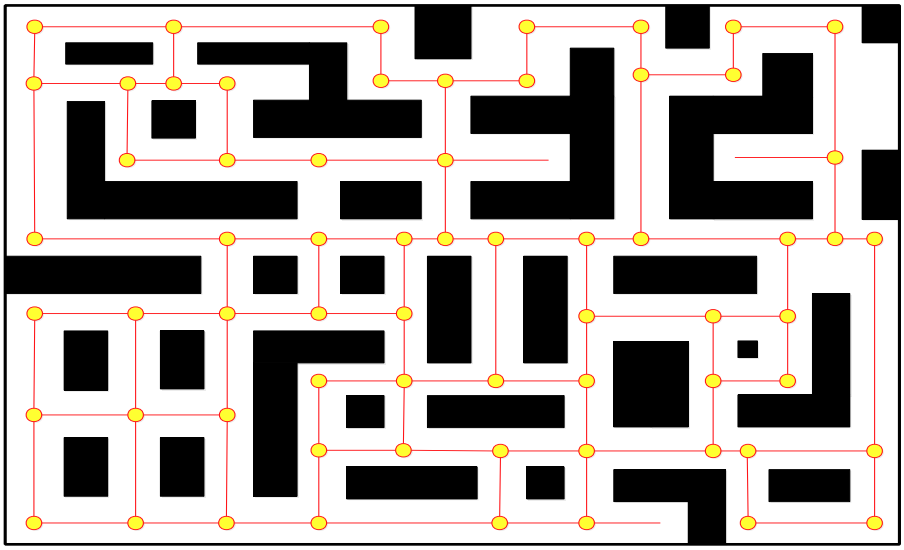
Dimana ρ adalah parameter evaporasi global, yang mempunyai nilai $0 < \rho < 1$.

3.3 Defenisi medan pencarian untuk explorasi

Pada penelitian ini medan pencarian menggunakan *maze environment*. Dari Gambar 3.2 dan 3.3 adalah bentuk rancangan environment dan terdapat node-node di setiap persimpangan dan juga terdapat lintasan *dead zone* (jalan buntu).



Gambar 3.2 Simpel *environment*



Gambar 3.3 komplek *environment*

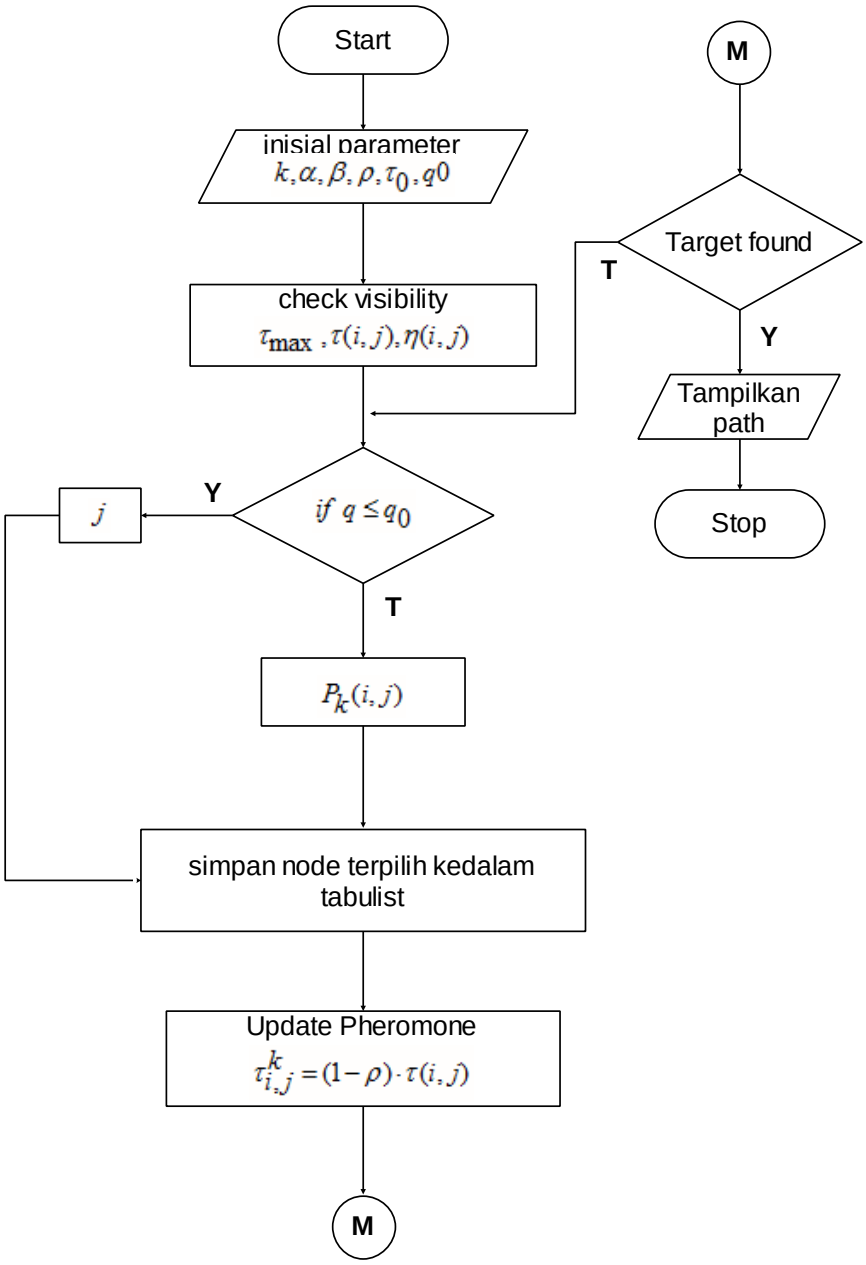
Untuk environment yang terlihat pada Gambar 3.2 menggunakan sebanyak 32 node dan terdapat 1 deadzone untuk yang pada Gambar 3.3 menggunakan sebanyak 67 node dan terdapat 3 deadzone.

3.4 Flowchart Pembuatan Algoritma Metode Explorasi

Metode explorasi pada penelitian ini dijelaskan dalam bentuk flowchart seperti gambar 3.5 merupakan persamaan yang digunakan didalam algoritma explorasi dimana kriteria algoritma yang digunakan ialah pseudo code dimana algortima nya seperti pada gambar 3.4.

```
Input    n : jumlah Agent, posisi, target,a,b,pheromone
         While posisi Agent bukan target
              For semut (s)  1 to jumlah semut
                   Pheromon tujuan  <= pheromone(posisi semut,: )
                   Hitung possibility dengan persamaan:
                   possibility = (max(phe(i,:)) − phe(i,j))^a · (η(i,j))^b
                   Hitung cumulative dari possibility
                   Bangkitkan bilangan random r
                   Posisi tujuan <− min(cumulative < r) //roulette machine
                   Pheromone(posisi semut s, posisi tujuan)=+ 1; //update pheromone
                   Posisi semut s <− posisi tujuan
              End
                   Pheromone ≤ (1 - rho) * pheromone  //evaporasi
         End
```

Gambar 3.4 Pseudo code explorasi menggunakan *ant colony algorithm*



Gambar 3.5 Flowchart Algoritma Explorasi dengan *ant colony*

Halaman ini sengaja dikosongkan

BAB 4
HASIL DAN PEMBAHASAN

Pada penelitian ini, sistem yang dibuat merupakan algoritma *ant colony system* dengan modifikasi normal feromon menjadi anti feromon. Sistem tersebut akan dibuat menggunakan fasilitas *script* dan GUI pada perangkat lunak Matlab. Eksperimen akan dilakukan 3 tahap yaitu dengan percobaan algoritma *ant colony* yang telah dimodifikasi dengan beberapa node pada lingkungan, kemudian menggunakan 2 lingkungan dengan variasi maze yang berbeda.

Untuk eksperimen pada lingkungan pertama menggunakan 1-5 agent dengan 32 node yang terdapat pada lintasan dalam lingkungan dengan model maze. Sedangkan pada lingkungan kedua dengan model maze yang lebih komplek menggunakan 1-5 agent dengan jumlah node sebanyak 67 yang terdapat pada lintasan. Pengujian dilakukan 2 variasi yaitu pengujian tanpa target dan dengan menggunakan target.

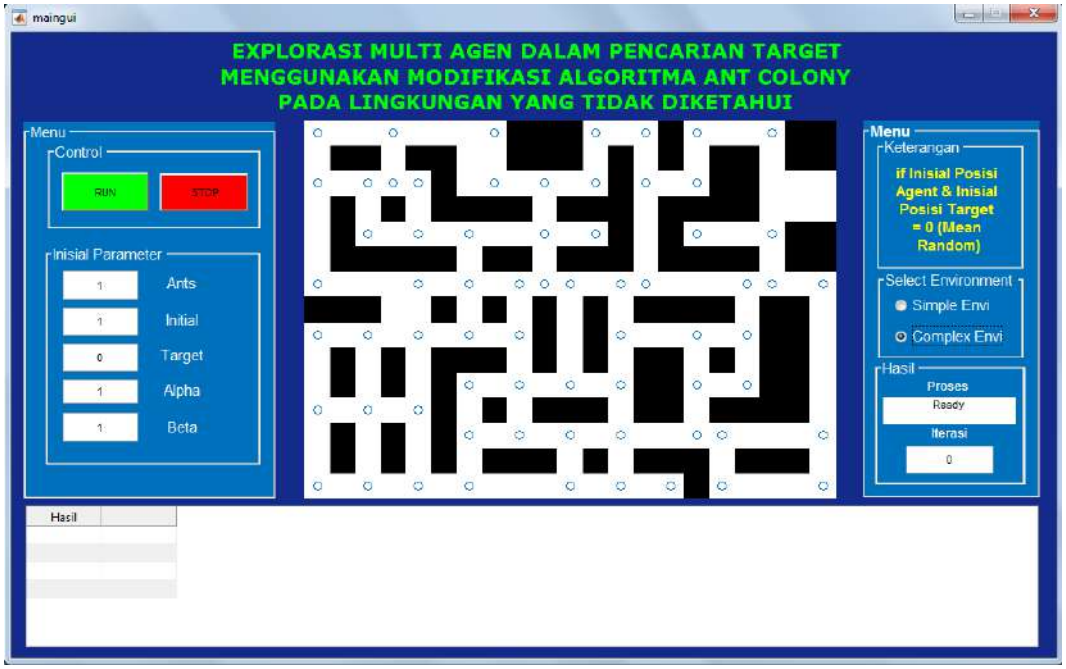
Pengujian yang akan dilakukan sebagai berikut:

a. pengujian pertama dengan kondisi tanpa target akan dilihat berapa banyak iterasi yang digunakan untuk mengunjungi semua node.
b. Pengujian kedua dengan kondisi menggunakan target akan dilihat pada iterasi keberapa robot akan mencapai target.
c. Membandingkan hasil pngujian algoritma *ant colony* pada umumnya dengan algoritma *ant colony* yang telah dimodifikasi dalam mencari target.

Pengujian yang akan dilakukan berdasarkan keterangan dari data pada Tabel 4.1 dan juga platform GUI matlab *ant colony algorithm* yang digunakan untuk mensimulasikan pengujian algoritma.

Tabel 4.1 Kriteria pengujian yang akan dilakukan

No.	Environment	Inisial Parameter						keterangan
	Tanpa Target	k	α	β	ρ	Init_pos	Init_target	Parameter
1	Simple Environment (32 Node)	1-5	1	1	0.5	1	-	(1)
2		1-5	2	1	0.5	1	-	(2)
3		1-5	1	2	0.5	1	-	(3)
4	Komplek Environment (67 Node)	1-5	1	1	0.5	1	-	(1)
5		1-5	2	1	0.5	1	-	(2)
6		1-5	1	2	0.5	1	-	(3)
	DenganTarget	k	α	β	ρ	Init_pos	Init_target	Parameter
7	Simple Environment (32 Node)	1-5	1	1	0.5	1	30	(1)
8		1-5	2	1	0.5	1	30	(2)
9		1-5	1	2	0.5	1	30	(3)
10	Komplek Environment (67 Node)	1-5	1	1	0.5	1	61	(1)
11		1-5	2	1	0.5	1	61	(2)
12		1-5	1	2	0.5	1	61	(3)



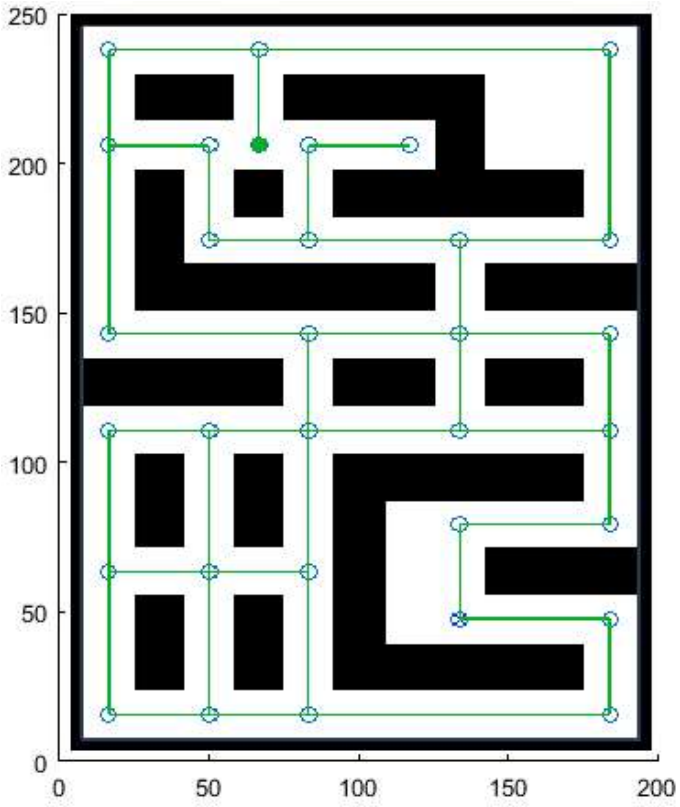
Gambar 4.1 GUI Matlab *ant colony system*

4.1 Pengujian pertama dengan kondisi tanpa target

1. Pengujian tanpa target pada simpel *environment*

Untuk gambar hasil dari simulasi pengujian 1 yaitu menggunakan medan environment dengan jumlah node sebanyak 32. Berikut ini hasil pengujian 1 dibawah ini:

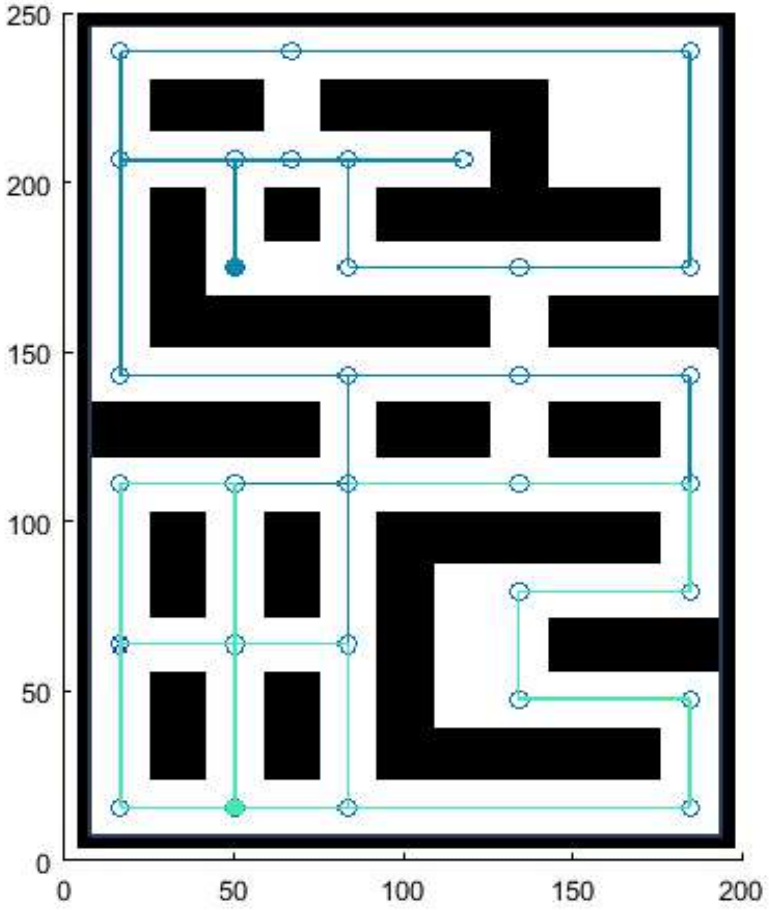
Pengujian pertama dengan input jumlah agen = 1 terlihat pada Gambar 4.2



Gambar 4.2 Simulasi 1 agent dalam mengujungi seluruh node

Agen 1 disimbolkan (●) circle berwana hijau dan untuk (—) line menandakan path atau jalur yang telah dikungi oleh agen 1. Pada Gambar 4.2 diatas dimana pengujian pada medan environment dengan jumlah node sebanyak 32. Ditetapkan inisial posisi agent 1 berada pada node 1 dibutuhkan 91 iterasi agen 1 dapat mengunjungi seluruh node. Terdapat faktor kondisi lingkungan yang dilalui agent 1 dalam mengunjung seluruh node dimana agent 1 selalu melewati jalur yang telah dilewati oleh agen 1 derajat kepentingan dari parameter (1) bobot feromon dan bobot heuristik sama sehingga kemungkinan kemungkinan jalur yang akan dilewati oleh agen 1 dikarenakan nilai nya sama antara nilai feromon dan juga informasi heuristik sehingga pilihannya menjadi seimbang.

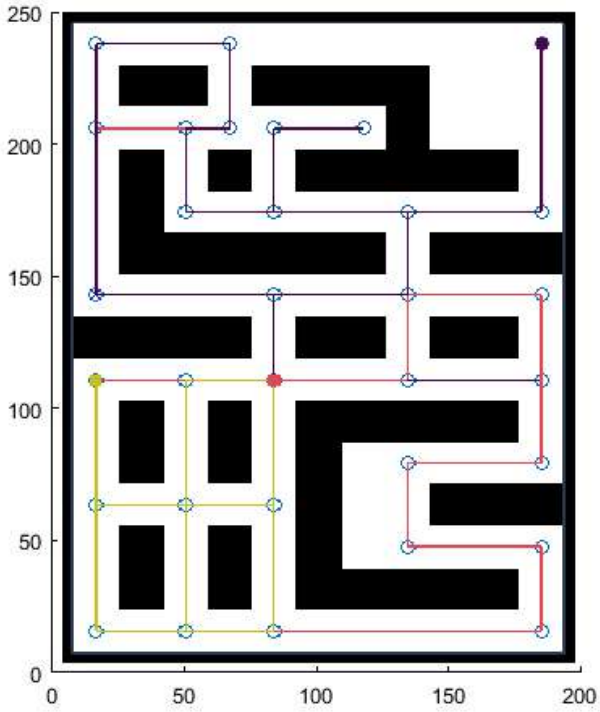
Pengujian kedua dengan input jumlah agen sebanyak 2 terlihat pada Gambar 4.3



Gambar 4.3 Simulasi 2 agent dalam mengujungi seluruh node

Simulasi 2 agen yang diposisikan pada node 1 dalam mengunjungi seluruh node sebanyak 62 iterasi. Agen 1 dengan kondisi harus melewati jalur yang sudah dilalui sebanyak 14 kali karna pada kondisi tersebut memang tidak ada jalur lain selain jalur tersebut sedangkan untuk agen 2 sebanyak 5 kali.

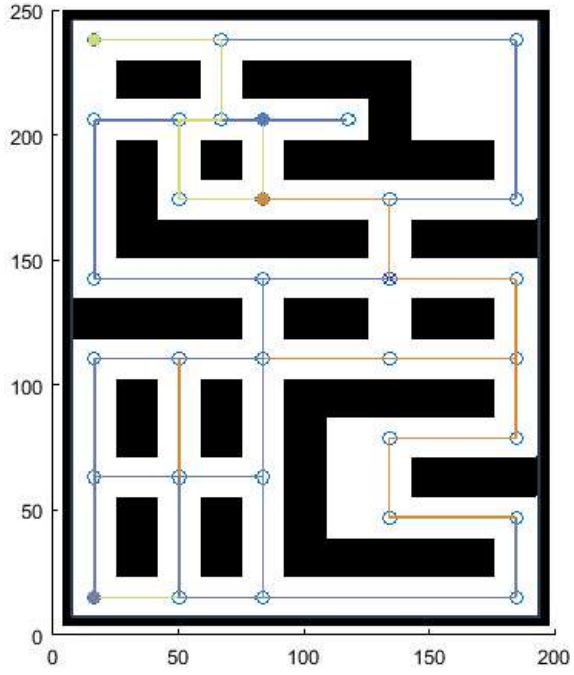
Pengujian ketiga dengan input jumlah agen sebanyak 3 terlihat pada Gambar 4.4



Gambar 4.4 Simulasi 3 agent dalam mengujungi seluruh node

Simulasi menggunakan 3 agen untuk mengunjungi seluruh node dibutuhkan 114 iterasi menggunakan parameter (1) dapat dilihat tiap tiap agent telah mengunjungi semua node.

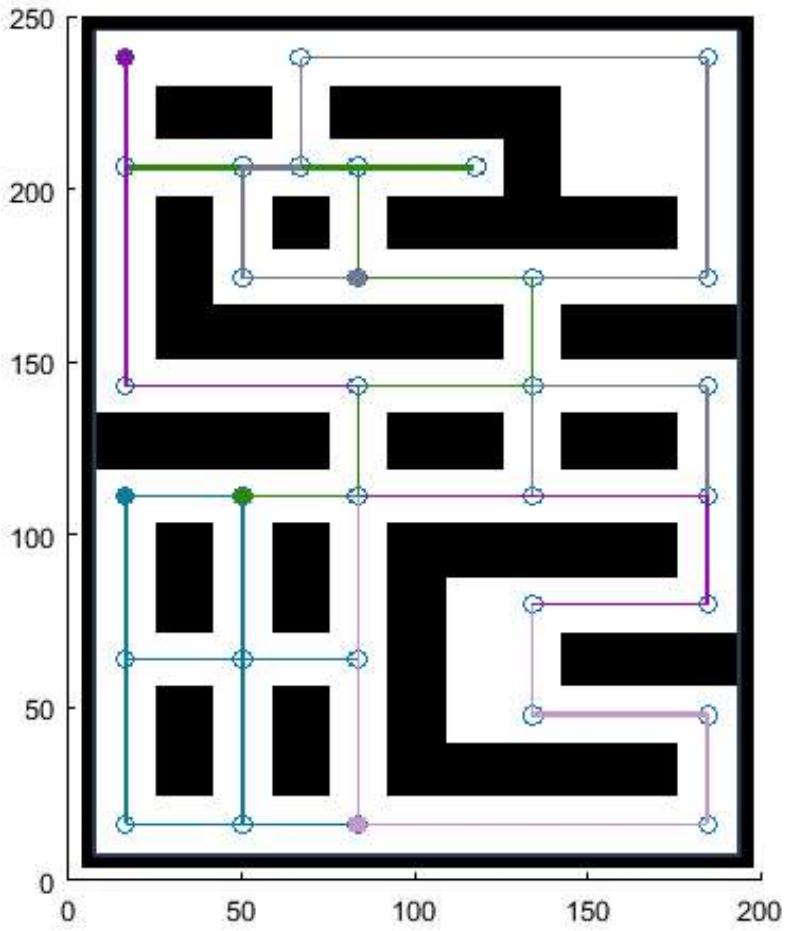
Pengujian keempat dengan input jumlah agen sebanyak 4 terlihat pada Gambar 4.5



Gambar 4.5 Simulasi 4 agen dalam mengujungi seluruh node

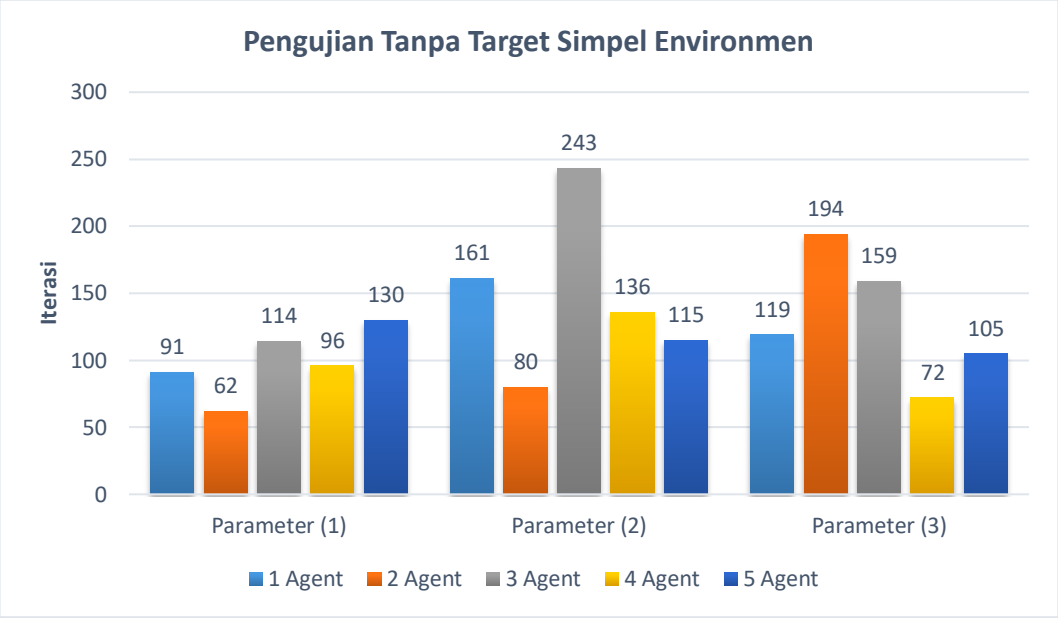
Simulasi menggunakan 4 agen untuk mengunjungi seluruh node dibutuhkan 96 iterasi menggunakan parameter (1) dapat dilihat tiap tiap agent telah mengunjungi semua node.

Pengujian kelima dengan input jumlah agen sebanyak 5 terlihat pada Gambar 4.6



Gambar 4.6 Simulasi 5 agen dalam mengujungi seluruh node

Simulasi menggunakan 5 agen untuk mengunjungi seluruh node dibutuhkan 130 iterasi menggunakan parameter (1) dapat dilihat tiap tiap agent telah mengunjungi semua node. Untuk melihat hasil dari jumlah iterasi pada pengujian 2 dan 3 yang dibutuhkan untuk dapat mengunjungi semua node berdasarkan parameter yang telah ditetapkan dapat dilihat pada Gambar 4.7.

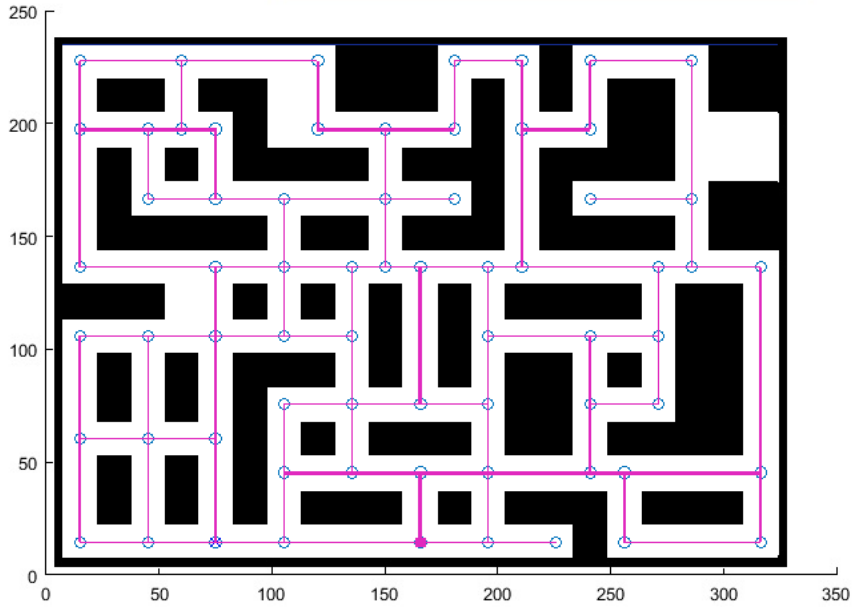


Gambar 4.7 jumlah iterasi dari 3 parameter pada pengujian simpel *environment*

Dari Gambar 4.7 menjelaskan jumlah iterasi yang diperoleh dari hasil pengujian pada simpel *environment* dengan nilai parameter $\alpha = 1, \beta = 1$ lebih efektif tiap agen dapat mengunjungi seluruh node yang ada pada simpel *environment* dikarenakan jumlah iterasi yang diperoleh lebih sedikit dibandingkan dengan menggunakan parameter (2) dan (3).

2. Pengujian tanpa target pada komplek *environment*

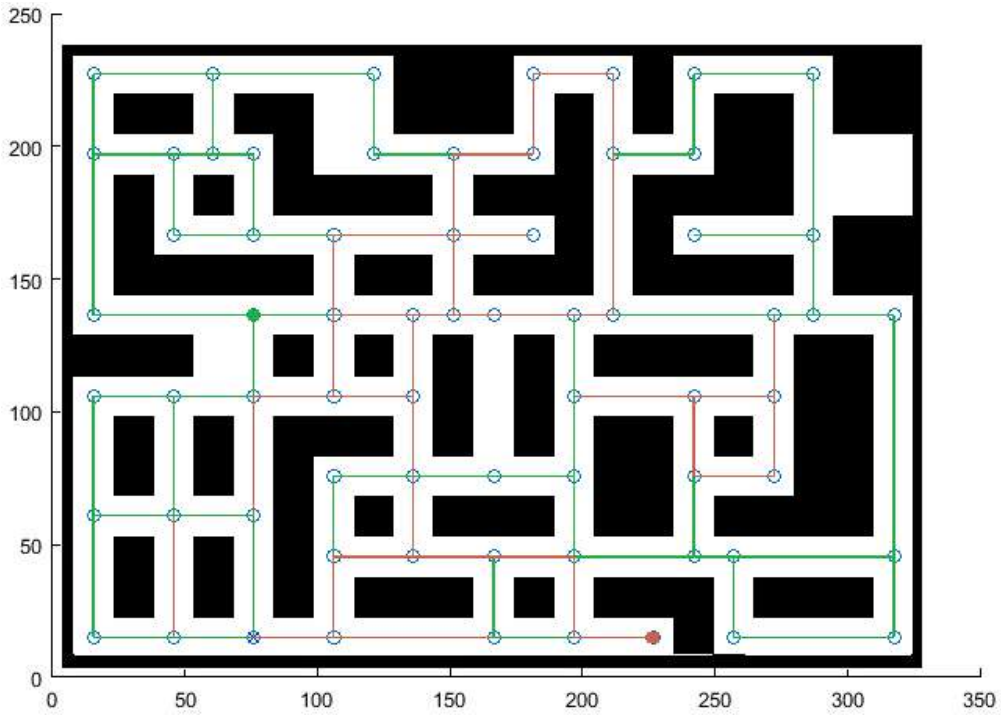
Pengujian keenam dengan input jumlah agen = 1 terlihat pada Gambar 4.8



Gambar 4.8 Simulasi 1 agen dalam mengujungi seluruh node

Simulasi menggunakan 1 agen untuk mengunjungi seluruh node dibutuhkan 160 iterasi menggunakan parameter (1) dapat dilihat agen telah mengunjungi semua node.

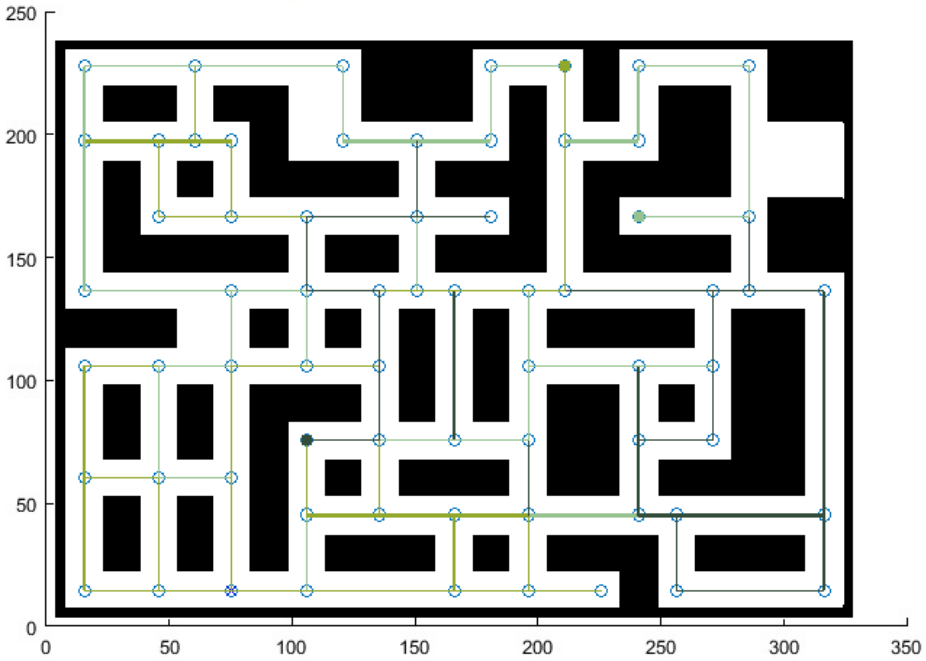
Pengujian ketujuh dengan input jumlah agen sebanyak 2 terlihat pada Gambar 4.9



Gambar 4.9 Simulasi 2 agen dalam mengujungi seluruh node

Simulasi menggunakan 2 agen untuk mengunjungi seluruh node dibutuhkan 400 iterasi menggunakan parameter (1) dapat dilihat agen telah mengunjungi semua node.

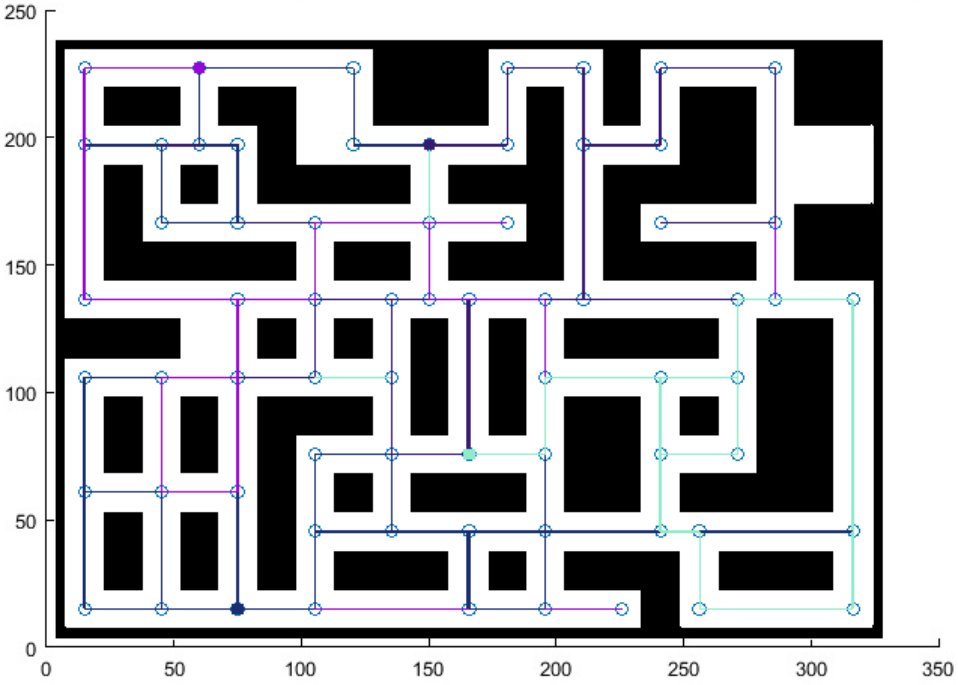
Pengujian kedelapan dengan input jumlah agen sebanyak 3 terlihat pada Gambar 4.10



Gambar 4.10 Simulasi 3 agen dalam mengujungi seluruh node

Simulasi menggunakan 3 agen untuk mengunjungi seluruh node dibutuhkan 410 iterasi menggunakan parameter (1) dapat dilihat agen telah mengunjungi semua node.

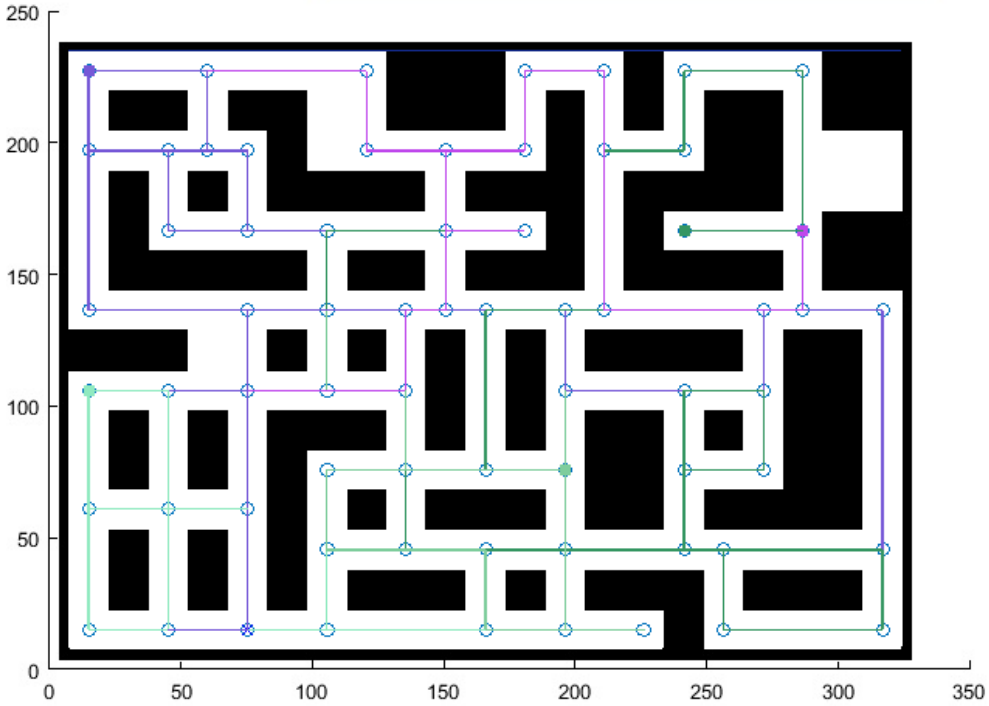
Pengujian kesembilan dengan input jumlah agen sebanyak 4 pada Gambar 4.11



Gambar 4.11 Simulasi 4 agen dalam mengujungi seluruh node

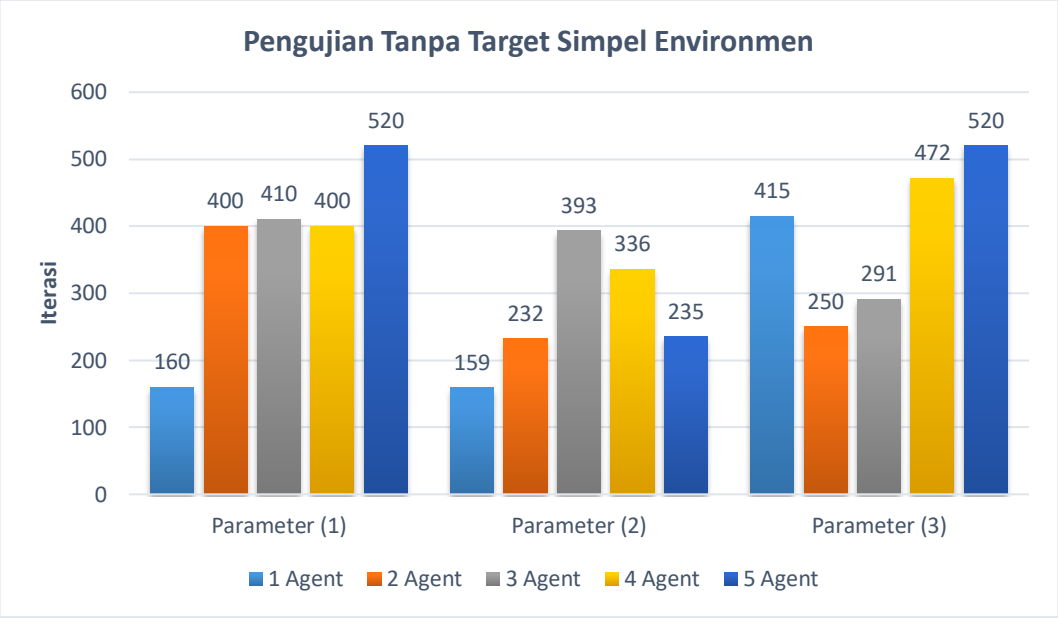
Simulasi menggunakan 4 agen untuk mengunjungi seluruh node dibutuhkan 400 iterasi menggunakan parameter (1) dapat dilihat agen telah mengunjungi semua node.

Pengujian kesepuluh dengan input jumlah agen sebanyak 5 pada Gambar 4.12



Gambar 4.12 Simulasi 5 agen dalam mengujungi seluruh node

Simulasi menggunakan 5 agen untuk mengunjungi seluruh node dibutuhkan 520 iterasi menggunakan parameter (1) dapat dilihat tiap tiap agent telah mengunjungi semua node. Untuk melihat hasil dari jumlah iterasi pada pengujian 5 dan 6 yang dibutuhkan untuk dapat mengunjungi semua node berdasarkan parameter yang telah ditetapkan dapat dilihat pada Gambar 4.13.



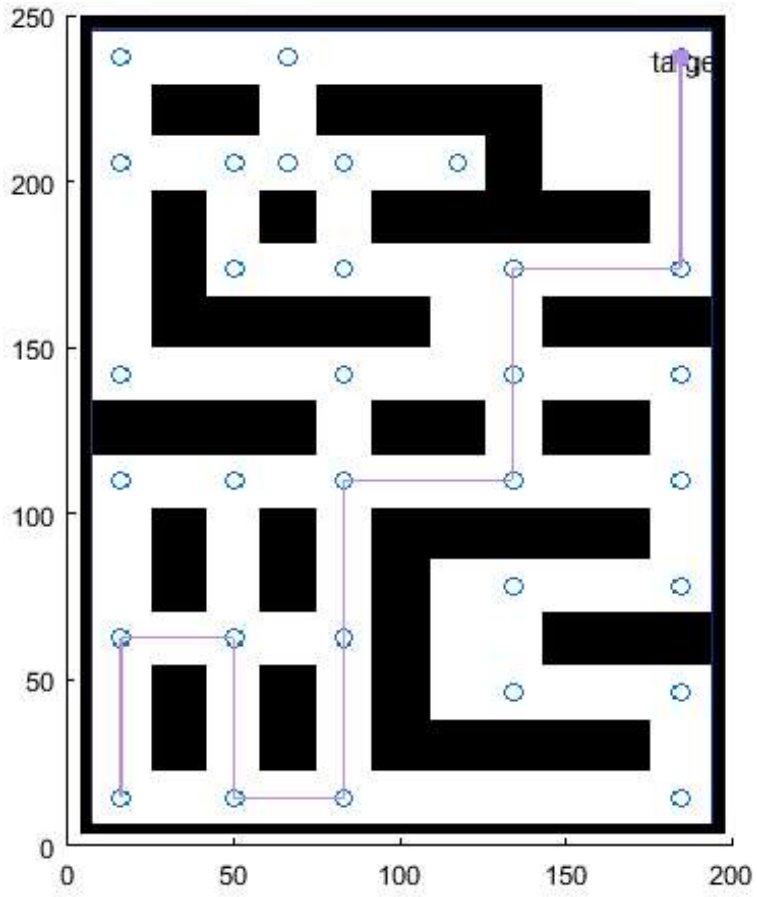
Gambar 4.13 jumlah iterasi dari 3 parameter pada pengujian komplek *environment*

Dari Gambar 4.13 menjelaskan jumlah iterasi dari masing-masing pengujian pada komplek *environment* dengan menggunakan parameter $\alpha = 2, \beta = 1$ lebih efektif algoritma ant colony dalam mengunjungi semua target dibandingkan menggunakan parameter (1) dan (3).

4.2 Pengujian Menggunakan Target

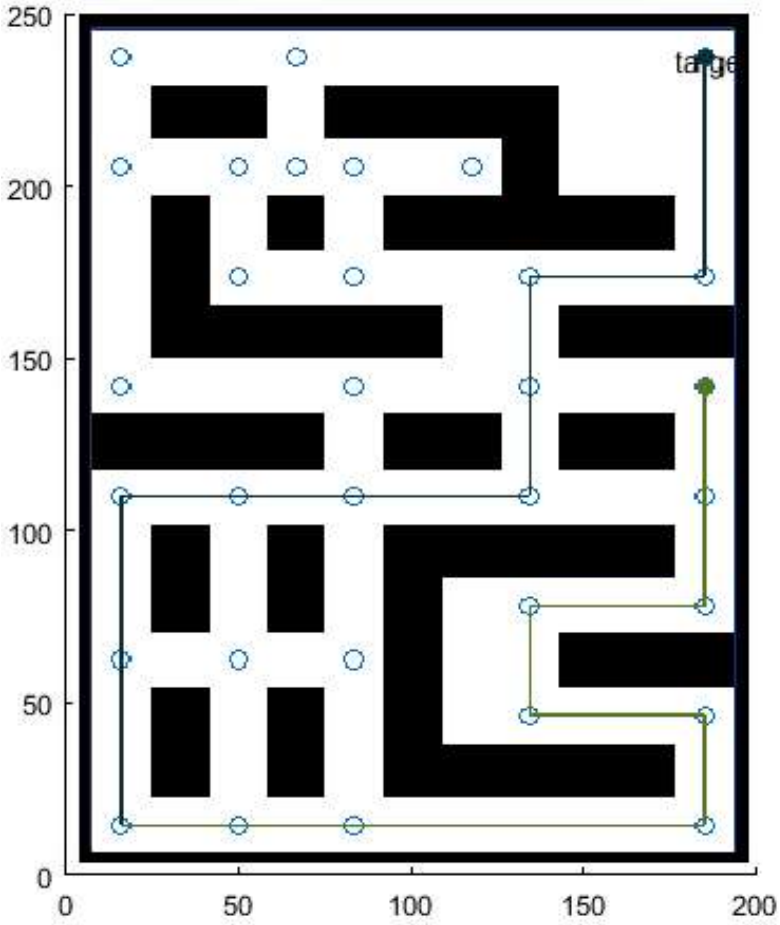
Untuk pengujian ini masing masing paramater akan diuji sebanyak 5 kali percobaan dengan tujuan mencari hasil berupa iterasi, jarak total tempuh, serta waktu yang dibutuhkan. Pengujian pada simpel environment inisial awal posisi setiap agen berada pada node 1 dengan target tujuan berada pada node 30. Time sampling untuk semua pengujian 0.5. Serta pada pengujian komplek *environment* posisi agen berada pada node 1 degan target tujuan berada pada node 61.

1. Pengujian simpel *environment*



Gambar 4.14 Simulasi 1 agen mencari posisi target

Simulasi pada Gambar 4.14 agen 1 berada pada node 1 dan target berada pada node 30. Node yang telah dilewati oleh agen 1 [1 – 9 – 8 – 2 – 3 – 7 – 14 – 13 – 19 – 22 – 21 – 30] dengan total jarak yang ditempuh 480 cm.

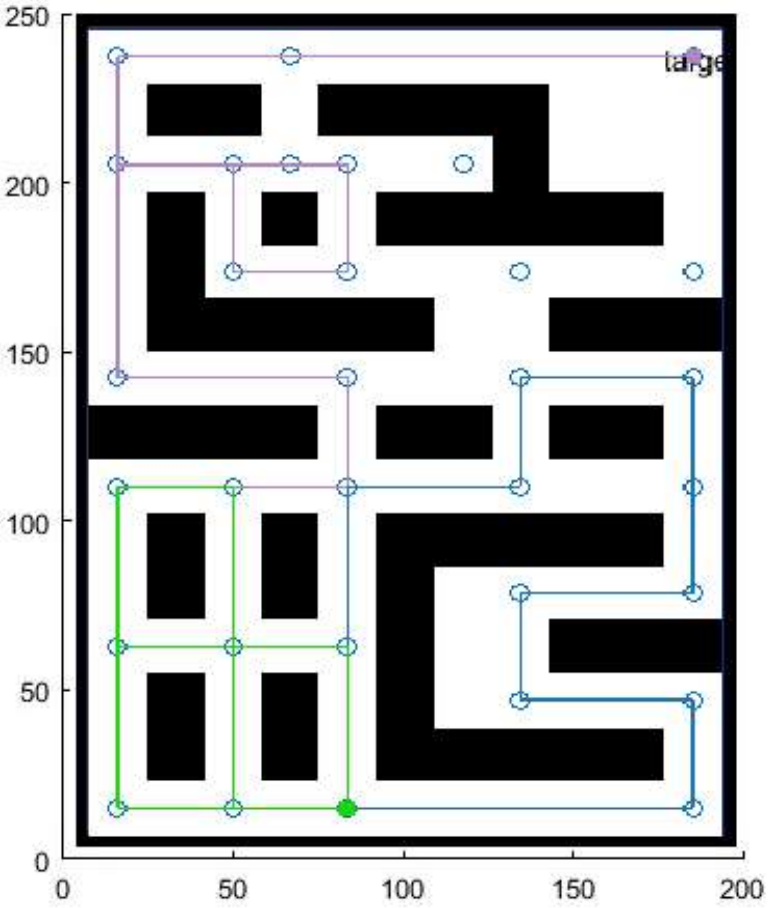


Gambar 4.15 Simulasi 2 agen mencari posisi target

Simulasi pada Gambar 4.15 menggunakan 2 agen berada diposisi node 1 dan mencari target pada posisi node 30. Agen 2 yang telah menemukan target berada pada node 30. Berikut *datalist* node yang telah dilewati tiap agen dapat dilihat pada Tabel 4.2.

Tabel 4.2 *Datalist* node yang telah dilewati oleh 2 agen.

Agen	Jarak	Node yang dikunjungi
1	370	1 – 2 – 3 – 4 – 5 – 6 – 10 – 11 – 12 – 20
2	380	1 – 9 – 16 – 15 – 14 – 13 – 19 – 22 – 21 - 30

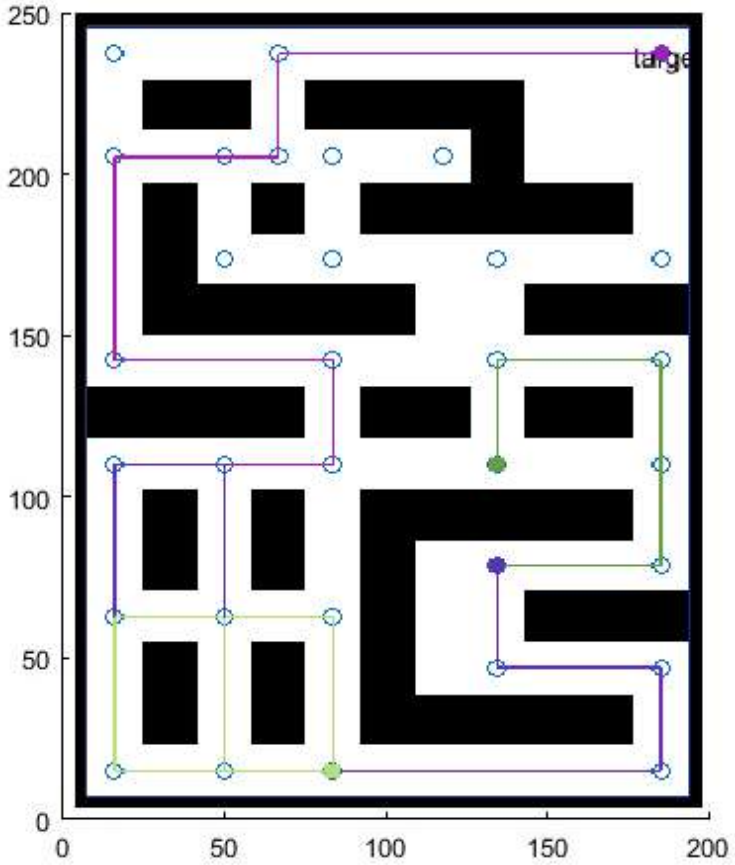


Gambar 4.16 Simulasi 3 agen mencari posisi target

Simulasi pada Gambar 4.16 menggunakan 2 agen berada diposisi node 1 dan mencari target pada posisi node 30. Agen 1 yang telah menemukan target berada pada node 30. Berikut *datalist* node yang telah dilewati tiap agen dapat dilihat pada Tabel 4.3.

Tabel 4.3 *Datalist* node yang telah dilewati oleh 3 agen.

Agen	Jarak	Node yang dikunjungi
1	760	1 – 2 – 8 – 15 – 14 – 18 – 17 – 25 – 26 – 27 – 28 – 23 – 24 – 26 – 26 – 32 – 31 - 30
2	680	1 – 9 – 16 – 15 – 8 – 7 – 14 – 13 – 19 – 20 – 12 – 11 – 10 – 6 – 5 – 4 – 3
3	720	1 – 9 – 8 – 2 – 3 – 7 – 8 – 9 – 16 – 15 – 8 – 2 – 1 – 9 – 8 – 2

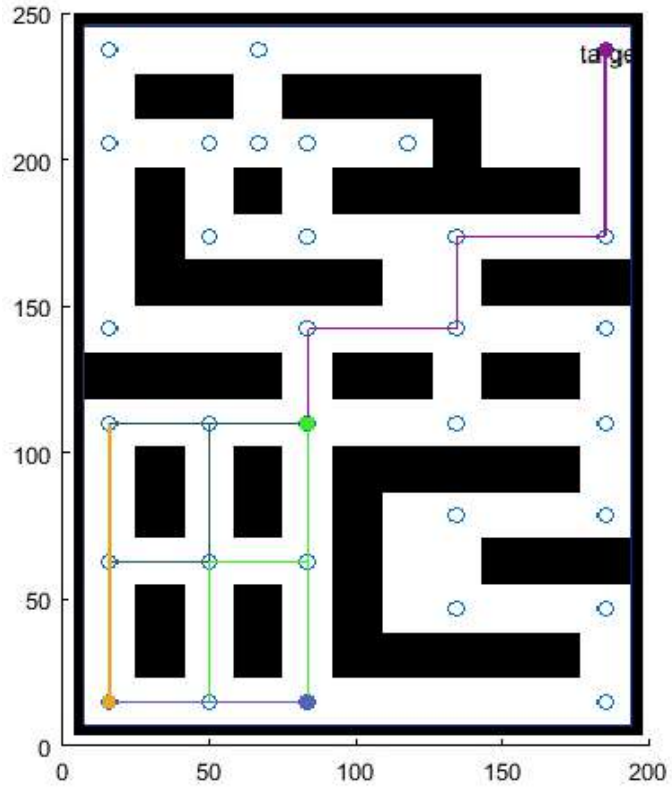


Gambar 4.17 Simulasi 4 agen mencari posisi target

Simulasi pada Gambar 4.17 menggunakan 2 agen berada diposisi node 1 dan mencari target pada posisi node 30. Agen 2 yang telah menemukan target berada pada node 30. Berikut *datalist* node yang telah dilewati tiap agen dapat dilihat pada Tabel 4.4.

Tabel 4.4 *Datalist* node yang telah dilewati oleh 4 agen.

Agen	Jarak	Node yang dikunjungi
1	440	1 – 2 – 3 – 4 – 5 – 6 – 10 – 11 – 12 – 20 – 19 – 13
2	540	1 – 9 – 8 – 15 – 14 – 18 – 17 – 25 – 26 – 27 – 31 – 30
3	440	1 – 9 – 8 – 2 – 3 – 7 – 8 – 9 – 16 – 15 – 8 – 2 – 1 – 9 – 8 – 2
4	470	1 – 9 – 16 – 15 – 8 – 2 – 3 – 4 – 5 – 6 – 10 – 10



Gambar 4.18 Simulasi 5 agen mencari posisi target

Simulasi pada Gambar 4.18 menggunakan 2 agen berada diposisi node 1 dan mencari target pada posisi node 30. Agen 1 yang telah menemukan target berada pada node 30. Berikut *datalist* node yang telah dilewati tiap agen dapat dilihat pada Tabel 4.5.

Tabel 4.5 *Datalist* node yang telah dilewati oleh 5 agen.

Agen	Jarak	Node yang dikunjungi
1	380	1 – 9 – 16 -15 – 14 – 18 – 19 – 22 – 21 – 30
2	360	1 – 2 – 8 – 7 – 14 – 15 – 16 – 9 – 1
3	360	1 – 9 – 8 – 15 – 8 – 9 – 16 – 15 – 14
4	360	1 – 2 – 3 – 7 – 3 – 2 – 8 – 7 – 14
5	360	1 – 9 – 8 – 2 – 1 – 9 – 1 – 2 – 3

Berikut ini hasil pengujian sebanyak 5 kali perulangan dalam pengujian pada simpel *environment* yang bertujuan untuk mencari posisi target. Dapat dilihat di Tabel 4.6.

Tabel 4.6 Data hasil pengujian dengan parameter (1) pada simple *environment*

No	Jumlah agen	Parameter	Simulasi	Iterasi	Jarak (cm)	Waktu	Keterangan
1	1	1	1x	12	640	6.6 s	Target ditemukan
			2x	39	1680	21 s	Target ditemukan
			3x	31	1340	17 s	Target ditemukan
			4x	17	740	9.1 s	Target ditemukan
			5x	12	640	6.7 s	Target ditemukan
2	2	1	1x	58	1260	16 s	Target ditemukan
			2x	26	560	7.3 s	Target ditemukan
			3x	65	1330	19 s	Target ditemukan
			4x	17	380	4.9 s	Target ditemukan
			5x	52	1120	15.2 s	Target ditemukan
3	3	1	1x	38	540	7.8	Target ditemukan
			2x	74	1060	15 s	Target ditemukan
			3x	25	380	5.2 s	Target ditemukan
			4x	33	460	6.5 s	Target ditemukan
			5x	50	740	10.3 s	Target ditemukan
4	4	1	1x	69	780	11 s	Target ditemukan
			2x	133	1440	21 s	Target ditemukan
			3x	130	1420	21 s	Target ditemukan
			4x	70	900	11 s	Target ditemukan
			5x	100	1100	16 s	Target ditemukan
5	5	1	1x	54	480	7.2 s	Target ditemukan
			2x	138	1080	19 s	Target ditemukan
			3x	124	1060	17 s	Target ditemukan
			4x	67	600	9.2 s	Target ditemukan
			5x	81	680	11.3 s	Target ditemukan

Pada Tabel 4.6 pengujian yang dilakukan sebanyak 5 kali dengan parameter yang digunakan $\alpha = 1, \beta = 1, \rho = 0.5$, untuk pengujian 1 agen didapat total jarak tempuh paling minimal 640 cm dengan jumlah iterasi sebanyak 12 serta total waktu 6.6 detik. Untuk pengujian 2 agen didapat total jarak tempuh paling minimal 380 cm , iterasi = 17, waktu = 4.9 detik. Untuk pengujian 3 agen didapat total jarak tempuh paling minimal 380 cm, iterasi = 25, waktu 5.2 detik. Untuk pengujian 4

agen didapat total jarak tempuh paling minimal 780 cm, iterasi = 69, waktu = 11 detik. Untuk pengujian 5 agen didapat total jarak tempuh paling minimal 480 cm, iterasi = 54, waktu 7.2 detik.

Tabel 4.7 Data hasil pengujian dengan parameter (2) pada simple *environment*

No	Jumlah agen	Parameter	Simulasi	Iterasi	Jarak (cm)	Waktu	Keterangan
1	1	2	1x	28	1150	15 s	Target ditemukan
			2x	27	1180	14 s	Target ditemukan
			3x	55	2290	30 s	Target ditemukan
			4x	14	620	7.5 s	Target ditemukan
			5x	11	480	11 s	Target ditemukan
2	2	2	1x	33	710	9.6 s	Target ditemukan
			2x	57	1360	16 s	Target ditemukan
			3x	29	660	8.5 s	Target ditemukan
			4x	28	620	8.4 s s	Target ditemukan
			5x	18	380	5 s	Target ditemukan
3	3	2	1x	49	740	10 s	Target ditemukan
			2x	65	890	13 s	Target ditemukan
			3x	79	1030	16 s	Target ditemukan
			4x	25	380	5.3 s	Target ditemukan
			5x	49	760	10.1 s	Target ditemukan
4	4	2	1x	33	380	5.5 s	Target ditemukan
			2x	33	380	5.6 s	Target ditemukan
			3x	59	640	9.4 s	Target ditemukan
			4x	49	620	8 s	Target ditemukan
			5x	42	540	6.8 s	Target ditemukan
5	5	2	1x	95	820	12 s	Target ditemukan
			2x	46	540	6.6 s	Target ditemukan
			3x	80	600	10 s	Target ditemukan
			4x	62	520	8.5 s	Target ditemukan
			5x	41	380	5.7 s	Target ditemukan

Pada Tabel 4.7 pengujian yang dilakukan sebanyak 5 kali dengan parameter yang digunakan $\alpha = 2, \beta = 1, \rho = 0.5$, untuk pengujian 1 agen didapat total jarak tempuh paling minimal 480 cm dengan jumlah iterasi sebanyak 11 serta total waktu

11 detik. Untuk pengujian 2 agen didapat total jarak tempuh paling minimal 380 cm , iterasi = 18, waktu = 5 detik. Untuk pengujian 3 agen didapat total jarak tempuh paling minimal 380 cm , iterasi = 25, waktu = 5.3 detik. Untuk pengujian 4 agen didapat total jarak tempuh paling minimal 380 cm , iterasi = 33, waktu = 5.5 detik. Untuk pengujian 5 agen didapat total jarak tempuh paling minimal 380 cm , iterasi = 41, waktu = 5.7 detik.

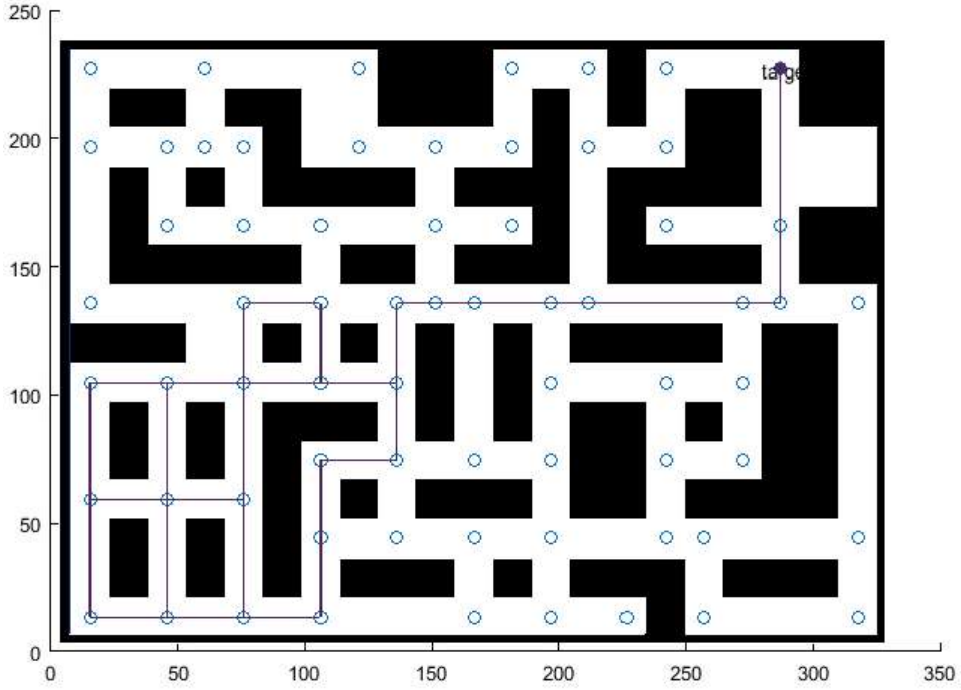
Tabel 4.8 Data hasil pengujian dengan parameter (3) pada simple *environment*

No	Jumlah agen	Parameter	Simulasi	Iterasi	Jarak (cm)	Waktu	Keterangan
1	1	3	1x	31	1320	16.9 s	Target ditemukan
			2x	40	1640	21 s	Target ditemukan
			3x	35	1560	19 s	Target ditemukan
			4x	49	1980	26 s	Target ditemukan
			5x	159	1350	88 s	Target ditemukan
2	2	3	1x	17	380	4.9 s	Target ditemukan
			2x	70	1300	20.1 s	Target ditemukan
			3x	18	380	5 s	Target ditemukan
			4x	26	560	7.2 s	Target ditemukan
			5x	38	1300	16.7 s	Target ditemukan
3	3	3	1x	110	1630	22.4 s	Target ditemukan
			2x	31	480	6.5 s	Target ditemukan
			3x	44	660	8.9 s	Target ditemukan
			4x	217	3000	45 s	Target ditemukan
			5x	80	980	16 s	Target ditemukan
4	4	3	1x	49	520	8 s	Target ditemukan
			2x	85	920	14 s	Target ditemukan
			3x	90	870	14.7 s	Target ditemukan
			4x	172	1900	27 s	Target ditemukan
			5x	197	1860	32 s	Target ditemukan
5	5	3	1x	151	1320	21 s	Target ditemukan
			2x	137	1190	18.8 s	Target ditemukan
			3x	51	460	7.8 s	Target ditemukan
			4x	97	960	13.4 s	Target ditemukan
			5x	125	1080	16.8 s	Target ditemukan

Pada Tabel 4.8 pengujian yang dilakukan sebanyak 5 kali dengan parameter yang digunakan $\alpha = 1, \beta = 2, \rho = 0.5$, untuk pengujian 1 agen didapat total jarak tempuh paling minimal 1320 cm dengan jumlah iterasi sebanyak 31 serta total waktu 16.9 detik. Untuk pengujian 2 agen didapat total jarak tempuh paling minimal 380 cm , iterasi = 17, waktu = 4.9 detik. Untuk pengujian 3 agen didapat total jarak tempuh paling minimal 480 cm , iterasi = 31, waktu = 6.5 detik. Untuk pengujian 4 agen didapat total jarak tempuh paling minimal 520 cm , iterasi = 49, waktu = 8 detik. Untuk pengujian 5 agen didapat total jarak tempuh paling minimal 460 cm , iterasi = 51, waktu = 7.8 detik.

Dari kesemua pengujian agen dapat menemukan target. Hanya saja untuk untuk menentukan parameter mana yang lebih baik digunakan dalam pencarian target supaya lebih efektif dapat dilihat dilihat pada Tabel 4.7 menggunakan parameter (2) dengan jarak yang lebih minimum dibandingkan dengan parameter yang digunakan pada Tabel 4.6 dan Tabel 4.8.

2. Komplek *environment*

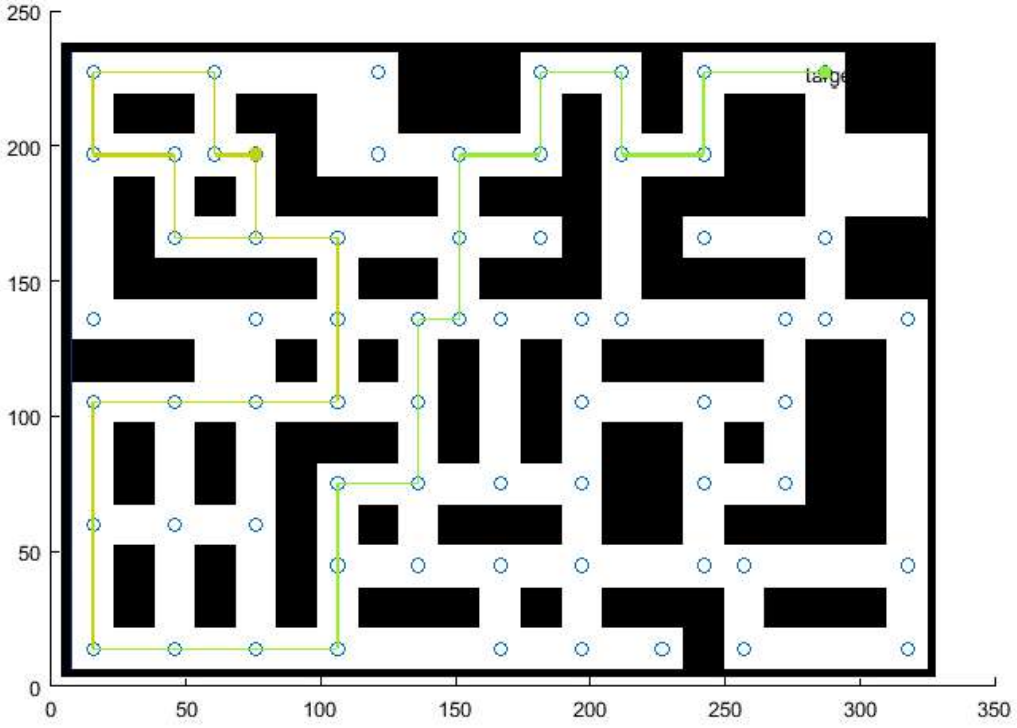


Gambar 4.19 Simulasi 1 agen mencari posisi target dilingkungan komplek

Simulasi pada Gambar 4.19 menggunakan 1 agen berada diposisi node 1 dan mencari target pada posisi node 61. Agen 1 yang telah menemukan target berada pada node 61. Berikut *datalist* node yang telah dilewati tiap agen dapat dilihat pada Tabel 4.9.

Tabel 4.9 *Datalist* node yang telah dilewati oleh 1 agen.

Agen	Jarak	Node yang dikunjungi
1	1740	1 – 19 – 18 – 17 – 3 – 2 – 1 – 19 – 33 – 32 – 18 – 2 – 3 – 17 – 31 – 30 – 29 – 21 – 20 – 16 – 4 – 3 – 17 – 18 -19 – 33 – 32 – 31 – 30 – 36 – 35 – 31 – 32 – 33 19 – 18 – 2 – 3 – 4 – 16 – 20 – 21 – 29 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 45 – 61

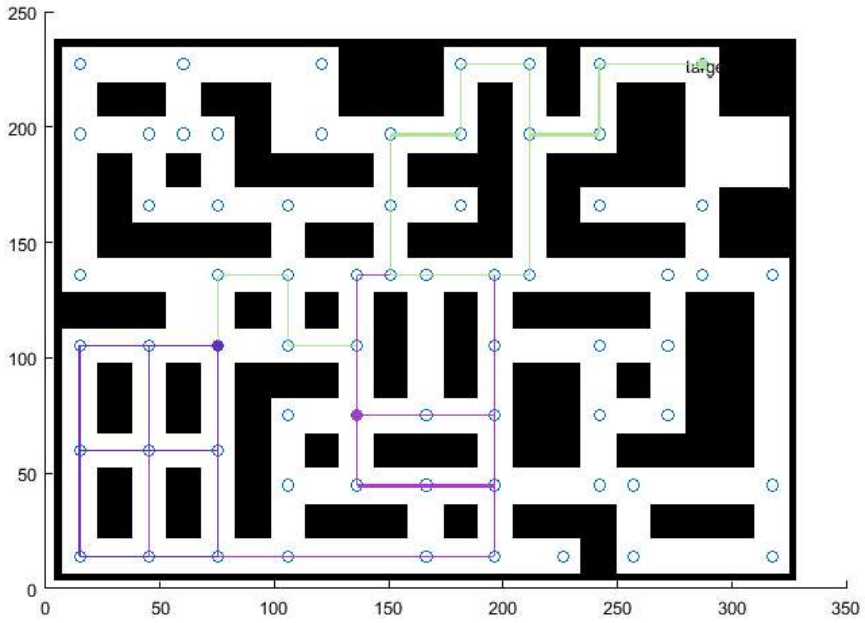


Gambar 4.20 Simulasi 2 agen mencari posisi target dilingkungan komplek

Simulasi pada Gambar 4.20 menggunakan 2 agen berada diposisi node 1 dan mencari target pada posisi node 61. Agen 1 yang telah menemukan target berada pada node 61. Berikut *datalist* node yang telah dilewati tiap agen dapat dilihat pada Tabel 4.10.

Tabel 4.10 *Datalist* node yang telah dilewati oleh 2 agen.

Agen	Jarak	Node yang dikunjungi
1	530	1 – 2 – 3 – 4 – 16 – 20 – 21 – 29 – 37 – 38 – 48 – 57 – 58 – 64 – 63 – 59 – 60 – 62 – 61
2	580	1 – 19 – 33 – 32 – 31 – 30 – 36 – 49 – 50 – 55 – 54 – 66 – 67 – 52 – 53 – 51 – 50 – 55

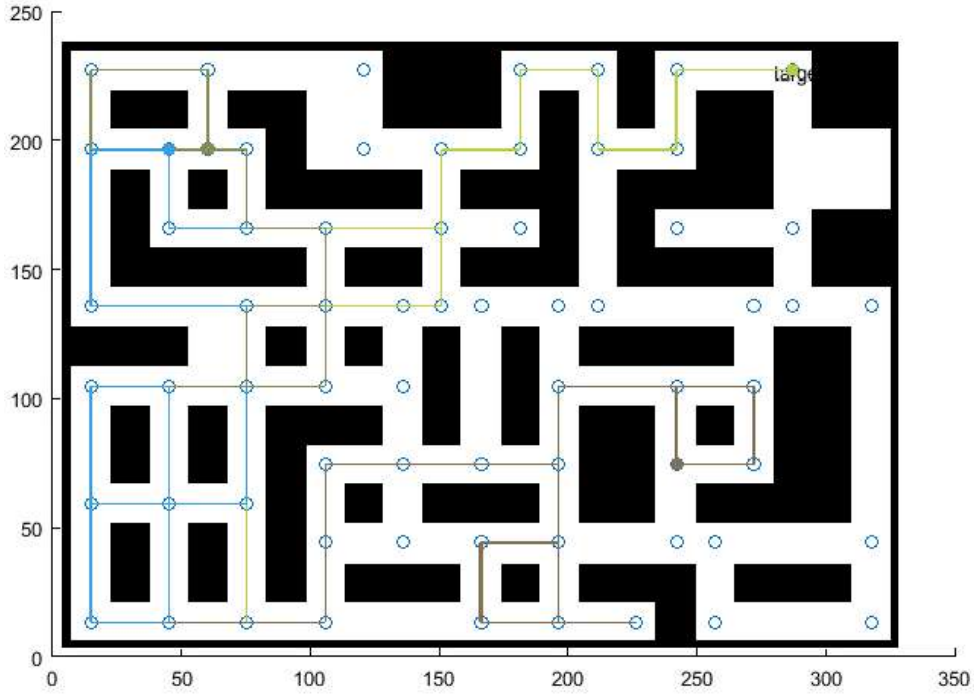


Gambar 4.21 Simulasi 3 agen mencari posisi target dilingkungan komplek

Simulasi pada Gambar 4.21 menggunakan 3 agen berada diposisi node 1 dan mencari target pada posisi node 61. Agen 1 yang telah menemukan target berada pada node 61. Berikut *datalist* node yang telah dilewati tiap agen dapat dilihat pada Tabel 4.11.

Tabel 4.11 *Datalist* node yang telah dilewati oleh 3 agen.

Agen	Jarak	Node yang dikunjungi
1	830	1 – 2 – 3 – 17 – 31 – 35 – 36 -30 – 29 – 37 – 38 – 39 – 40 – 59 – 63 – 64 – 58 – 57 – 48 – 38 – 39 – 40 – 41 – 59 – 60 – 62 – 61
2	960	1 – 19 – 33 – 32 – 31 – 17 – 18 – 19 – 33 – 32 – 31 – 17 – 3 – 2 – 1 – 19 – 18 – 32 – 33 – 19 – 18 – 32 – 33 – 19 – 18 – 32 – 31
3	740	1 – 2 – 18 – 17 – 3 – 4 – 5 – 6 – 13 – 14 – 15 – 21 – 22 – 23 – 28 – 40 – 39 – 38 – 37 – 29 – 21 – 22 – 23 – 13 – 14 – 15 – 21 - 21

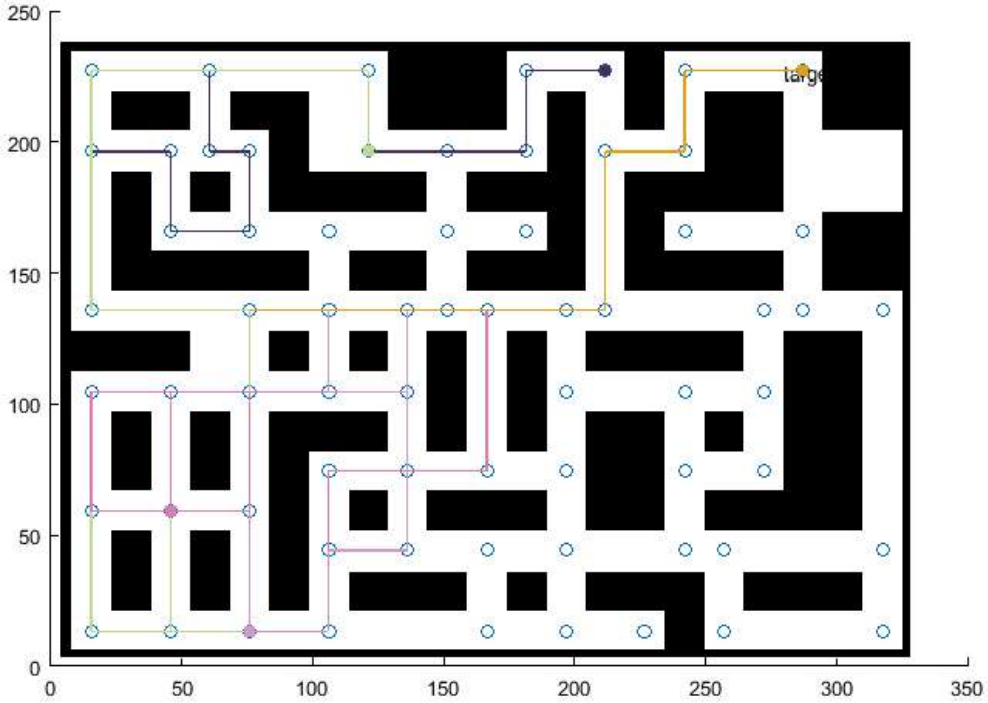


Gambar 4.22 Simulasi 4 agen mencari posisi target dilingkungan komplek

Simulasi pada Gambar 4.22 menggunakan 4 agen berada diposisi node 1 dan mencari target pada posisi node 61. Agen 3 yang telah menemukan target berada pada node 61. Berikut *datalist* node yang telah dilewati tiap agen dapat dilihat pada Tabel 4.12.

Tabel 4.12 *Datalist* node yang telah dilewati oleh 4 agen.

Agen	Jarak	Node yang dikunjungi
1	960	1 – 2 – 18 – 32 – 31 – 35 – 35 – 34 – 52 – 67 – 66 – 54 – 53 – 52 – 67 – 66 – 54 – 55 – 50 – 49 – 36 – 35 – 31 – 30 – 36 – 49 – 50
2	820	1 – 19 – 33 – 32 – 18 – 2 – 3 – 4 – 16 – 20 – 21 – 22 – 23 – 13 – 6 – 7 – 6 – 5 – 14 – 13 – 23 – 28 – 27 – 24 – 25 – 26 – 27 - 24
3	850	1 – 2 – 3 – 17 – 31 – 30 – 36 – 49 – 50 – 55 – 54 – 55 – 50 – 49 – 36 – 37 – 38 – 48 – 49 – 48 – 57 – 58 – 64 – 63 – 59 – 60 – 61
4	1010	1 – 19 – 18 – 17 – 19 – 33 – 32 – 18 – 19 – 33 – 32 – 18 – 19 – 1 – 2 – 18 – 17 – 31 – 35 – 52 – 53 – 51 – 50 – 51 - 53



Gambar 4.23 Simulasi 5 agen mencari posisi target dilingkungan komplek

Simulasi pada Gambar 4.23 menggunakan 5 agen berada diposisi node 1 dan mencari target pada posisi node 61. Agen 5 yang telah menemukan target berada pada node 61. Berikut *datalist* node yang telah dilewati tiap agen dapat dilihat pada Tabel 4.13.

Tabel 4.13 *Datalist* node yang telah dilewati oleh 5 agen.

Agen	Jarak	Node yang dikunjungi
1	870	1 – 19 – 33 – 32 – 31 – 35 – 34 – 52 – 67 – 66 – 54 – 55 – 50 – 51 – 53 – 52 – 67 – 66 – 65 – 56 – 57 – 58 – 64 - 63
2	730	1 – 2 – 3 – 4 – 16 – 20 – 21 – 29 – 37 – 38 – 39 – 22 – 29 – 30 – 31 – 32 – 33 – 19 – 18 – 32 – 31 – 17 - 18
3	790	1 – 19 – 18 – 32 – 33 – 19 – 18 – 2 – 18 – 17 – 31 – 30 – 29 – 37 – 36 – 30 – 29 – 21 – 15 – 16 – 4 – 3 – 17 - 3
4	930	1 – 2 – 18 – 2 – 1 – 19 – 1 – 2 – 1 – 19 – 33 – 32 – 18 – 2 – 3 – 17 – 31 – 35 – 34 – 52 – 67 – 66 – 65 – 56
5	760	1 – 2 – 3 – 17 – 18 – 19 – 33 – 32 – 31 – 35 – 36 – 30 – 31 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 59 – 60 – 62 – 61

Berikut ini hasil pengujian sebanyak 5 kali perulangan dalam pengujian pada komplek *environment* yang bertujuan untuk mencari posisi target. Dapat dilihat di Tabel 4.14.

Tabel 4.14 Data hasil pengujian dengan parameter (1) pada komplek *environment*

No	Jumlah agen	Parameter	Simulasi	Iterasi	Jarak (cm)	Waktu	Keterangan
1	1	1	1x	122	3570	70 s	Target ditemukan
			2x	152	4820	87 s	Target ditemukan
			3x	84	2960	48.7 s	Target ditemukan
			4x	140	4400	80 s	Target ditemukan
			5x	162	5360	94.6 s	Target ditemukan
2	2	1	1x	378	6410	122 s	Target ditemukan
			2x	66	1110	21.2 s	Target ditemukan
			3x	101	1820	32 s	Target ditemukan
			4x	80	1450	25 s	Target ditemukan
			5x	30	730	12.2 s	Target ditemukan
3	3	1	1x	85	910	20.2 s	Target ditemukan
			2x	302	3030	71 s	Target ditemukan
			3x	154	1630	36 s	Target ditemukan
			4x	73	830	17.5 s	Target ditemukan
			5x	117	1230	27.6 s	Target ditemukan
4	4	1	1x	143	1190	27.4 s	Target ditemukan
			2x	217	1710	41 s	Target ditemukan
			3x	183	1560	34.9 s	Target ditemukan
			4x	207	1640	39.7 s	Target ditemukan
			5x	117	1050	22.7 s	Target ditemukan
5	5	1	1x	130	880	21.5 s	Target ditemukan
			2x	95	610	15.7 s	Target ditemukan
			3x	173	1050	28.8 s	Target ditemukan
			4x	183	1150	30.3 s	Target ditemukan
			5x	115	760	19 s	Target ditemukan

Pada Tabel 4.14 pengujian yang dilakukan sebanyak 5 kali dengan parameter yang digunakan $\alpha = 1, \beta = 1, \rho = 0.5$, untuk pengujian 1 agen didapat total jarak tempuh paling minimal 2960 cm dengan jumlah iterasi sebanyak 84 serta total waktu 48.7 detik. Untuk pengujian 2 agen didapat total jarak tempuh paling minimal 730 cm , iterasi = 30, waktu = 12.2 detik. Untuk pengujian 3 agen didapat total jarak tempuh paling minimal 830 cm, iterasi = 73, waktu 17.5 detik. Untuk

pengujian 4 agen didapat total jarak tempuh paling minimal 1050 cm, iterasi = 117, waktu = 22.7 detik. Untuk pengujian 5 agen didapat total jarak tempuh paling minimal 610 cm, iterasi = 95, waktu 15.7 detik.

Tabel 4.15 Data hasil pengujian dengan parameter (2) pada komplek *environment*

No	Jumlah agen	Parameter	Simulasi	Iterasi	Jarak (cm)	Waktu	Keterangan
1	1	2	1x	216	7270	125 s	Target ditemukan
			2x	73	2450	42 s	Target ditemukan
			3x	20	690	11.5 s	Target ditemukan
			4x	45	1470	26.1 s	Target ditemukan
			5x	130	4120	75.8 s	Target ditemukan
2	2	2	1x	45	810	14.6	Target ditemukan
			2x	157	2590	50.4 s	Target ditemukan
			3x	93	1460	30.1 s	Target ditemukan
			4x	62	990	20 s	Target ditemukan
			5x	35	530	11.5 s	Target ditemukan
3	3	2	1x	252	2410	59 s	Target ditemukan
			2x	100	1170	24 s	Target ditemukan
			3x	142	1550	33 s	Target ditemukan
			4x	197	2170	46.2 s	Target ditemukan
			5x	118	1230	28.2 s	Target ditemukan
4	4	2	1x	223	1790	42 s	Target ditemukan
			2x	234	2100	45.1 s	Target ditemukan
			3x	130	950	22.3 s	Target ditemukan
			4x	57	470	12 s	Target ditemukan
			5x	107	850	20.6	Target ditemukan
5	5	2	1x	280	1670	46.9 s	Target ditemukan
			2x	115	790	19.2 s	Target ditemukan
			3x	264	1570	44.5 s	Target ditemukan
			4x	174	1290	29 s	Target ditemukan
			5x	164	950	27.5 s	Target ditemukan

Pada Tabel 4.15 pengujian yang dilakukan sebanyak 5 kali dengan parameter yang digunakan $\alpha = 2, \beta = 1, \rho = 0.5$, untuk pengujian 1 agen didapat total jarak tempuh paling minimal 690 cm dengan jumlah iterasi sebanyak 20 serta

total waktu 11.5 detik. Untuk pengujian 2 agen didapat total jarak tempuh paling minimal 530 cm , iterasi = 35, waktu = 11.5 detik. Untuk pengujian 3 agen didapat total jarak tempuh paling minimal 1170 cm, iterasi = 100, waktu 24 detik. Untuk pengujian 4 agen didapat total jarak tempuh paling minimal 470 cm, iterasi = 57, waktu = 12 detik. Untuk pengujian 5 agen didapat total jarak tempuh paling minimal 790 cm, iterasi = 115, waktu 19.2 detik.

Tabel 4.16 Data hasil pengujian dengan parameter (3) pada komplek *environment*

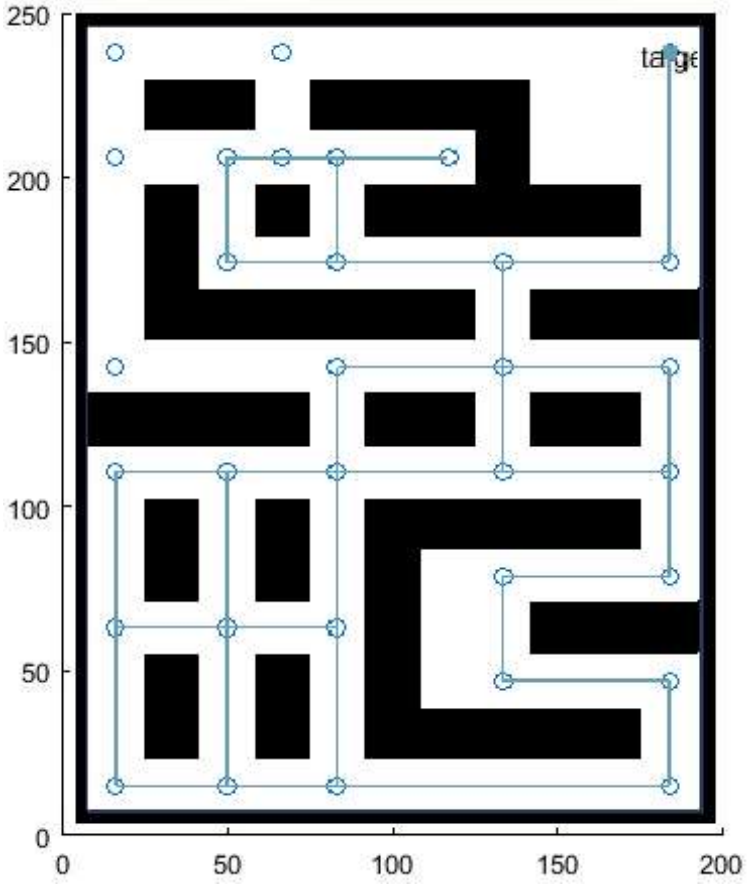
No	Jumlah agen	Parameter	Simulasi	Iterasi	Jarak (cm)	Waktu	Keterangan
1	1	3	1x	32	1010	18.7 s	Target ditemukan
			2x	198	5760	115 s	Target ditemukan
			3x	228	7250	132 s	Target ditemukan
			4x	140	4090	81 s	Target ditemukan
			5x	51	1740	29.7 s	Target ditemukan
2	2	3	1x	37	730	12.2 s	Target ditemukan
			2x	162	2370	51.9 s	Target ditemukan
			3x	159	2670	51.2	Target ditemukan
			4x	64	1040	20.8 s	Target ditemukan
			5x	49	730	15.9 s	Target ditemukan
3	3	3	1x	123	1270	30.7	Target ditemukan
			2x	64	710	16 s	Target ditemukan
			3x	46	530	11.3 s	Target ditemukan
			4x	238	2460	56 s	Target ditemukan
			5x	79	830	18.7	Target ditemukan
4	4	3	1x	239	1890	45.5 s	Target ditemukan
			2x	230	1670	44 s	Target ditemukan
			3x	77	650	15 s	Target ditemukan
			4x	129	1170	25 s	Target ditemukan
			5x	135	1170	25.7	Target ditemukan
5	5	3	1x	128	830	21.9 s	Target ditemukan
			2x	259	1710	42.5 s	Target ditemukan
			3x	91	590	15.8 s	Target ditemukan
			4x	155	1030	25.6 s	Target ditemukan
			5x	297	1870	48.8 s	Target ditemukan

Pada Tabel 4.16 pengujian yang dilakukan sebanyak 5 kali dengan parameter yang digunakan $\alpha = 1, \beta = 2, \rho = 0.5$, untuk pengujian 1 agen didapat total jarak tempuh paling minimal 1010 cm dengan jumlah iterasi sebanyak 32 serta total waktu 18.7 detik. Untuk pengujian 2 agen didapat total jarak tempuh paling minimal 730 cm , iterasi = 37, waktu = 12.2 detik. Untuk pengujian 3 agen didapat total jarak tempuh paling minimal 530 cm, iterasi = 46, waktu 11.3 detik. Untuk pengujian 4 agen didapat total jarak tempuh paling minimal 650 cm, iterasi = 77, waktu = 15 detik. Untuk pengujian 5 agen didapat total jarak tempuh paling minimal 590 cm, iterasi = 91, waktu 15.8 detik.

Dari kesemua pengujian agen dapat menemukan target. Hanya saja untuk untuk menentukan parameter mana yang lebih baik digunakan dalam pencarian target. Supaya lebih efektif dapat dilihat dilihat pada Tabel 4.14 menggunakan parameter (2) dengan jarak yang lebih minimum dari keseluruhan pengujian dibandingkan dengan parameter yang digunakan pada Tabel 4.13 dan Tabel 4.15.

4.3 Pengujian algoritma *ant colony* anti feromon tanpa modifikasi

Pada pengujian ini membandingkan algoritma yang telah dimodifikasi dengan algoritma *ant colony* anti feromon tanpa modifikasi. Cara yang digunakan yaitu diuji dengan jumlah iterasi dan waktu yang sama pada pengujian mencari target menggunakan algoritma yang telah dimodifikasi apakah bisa menemukan target atau sebaliknya menggunakan persamaan (2.8) dan (2.9).

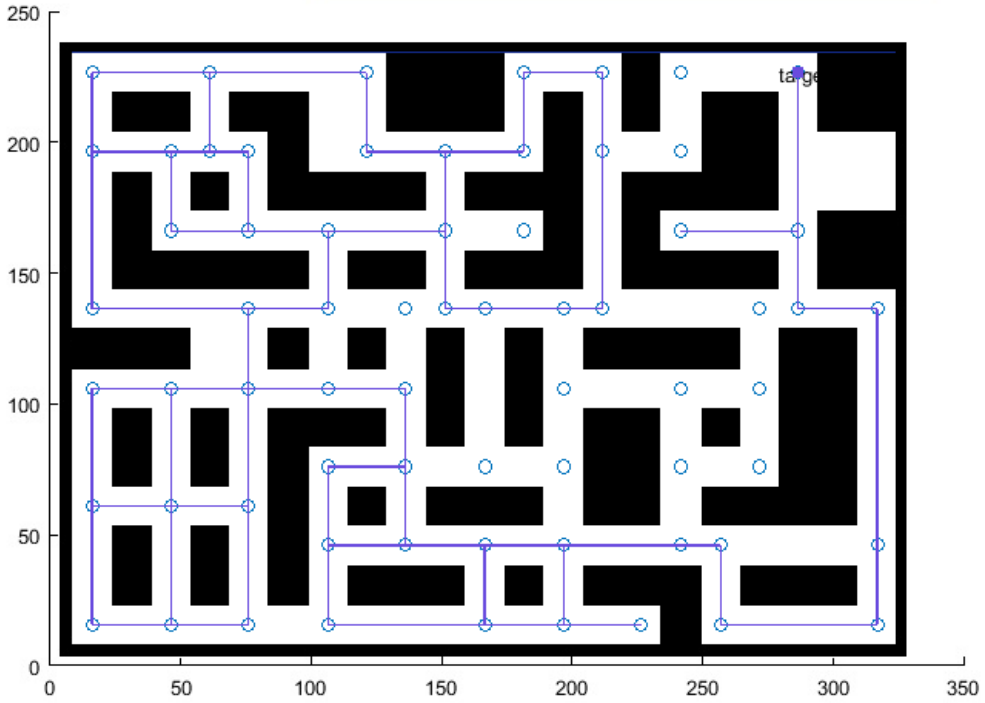


Gambar 4.24 Simulasi 1 agen mencari posisi target di simpel *environment*

Simulasi pada Gambar 4.24 menggunakan 1 agen berada diposisi node 1 dan mencari target pada posisi node 30. Agen 1 yang telah menemukan target berada pada node 30. Berikut *datalist* node yang telah dilewati tiap agen dapat dilihat pada Tabel 4.17.

Tabel 4.17 *Datalist* node yang telah dilewati oleh 1 agen.

Agen	Jarak	Node yang dikunjungi
1	2960	1 - 9 - 8 - 7 - 14 - 13 - 12 - 11 - 10 - 6 - 5 - 4 - 3 - 2 - 1 - 9 - 16 - 15 - 14 - 7 - 3 - 2 - 8 - 7 - 14 - 13 - 19 - 20 - 12 - 11 - 10 - 6 - 5 - 4 - 3 - 2 - 8 - 7 - 14 - 18 - 19 - 20 - 12 - 11 - 10 - 6 - 5 - 4 - 3 - 2 - 1 - 9 - 8 - 15 - 14 - 13 - 12 - 20 - 19 - 22 - 23 - 28 - 29 - 28 - 23 - 24 - 26 - 27 - 28 - 29 - 28 - 23 - 22 - 21 - 30



Gambar 4.25 Simulasi 1 agen mencari posisi target di komplek *environment*

Simulasi pada Gambar 4.25 menggunakan 1 agen berada diposisi node 1 dan mencari target pada posisi node 61. Agen 1 yang telah menemukan target berada pada node 61. Berikut *datalist* node yang telah dilewati tiap agen dapat dilihat pada Tabel 4.18.

Tabel 4.18 *Datalist* node yang telah dilewati oleh 1 agen.

Agen	Jarak	Node yang dikunjungi
1	4330	1 - 2 - 3 - 17 - 31 - 32 - 18 - 2 - 1 - 19 - 18 - 17 - 31 - 30 - 29 - 21 - 20 - 16 - 15 - 14 - 5 - 6 - 7 - 6 - 13 - 14 - 15 - 16 - 4 - 5 - 14 - 15 - 16 - 20 - 21 - 29 - 30 - 31 - 35 - 34 - 52 - 67 - 66 - 54 - 53 - 51 - 50 - 49 - 48 - 38 - 39 - 40 - 41 - 59 - 63 - 64 - 58 - 57 - 48 - 38 - 39 - 40 - 41 - 59 - 63 - 64 - 58 - 57 - 56 - 65 - 66 - 67 - 52 - 53 - 51 - 50 - 55 - 54 - 53 - 51 - 50 - 49 - 36 - 35 - 31 - 32 - 33 - 19 - 1 - 2 - 18 - 32 - 33 - 19 - 18 - 17 - 31 - 30 - 29 - 21 - 20 - 16 - 15 - 14 - 5 - 6 - 7 - 6 - 13 - 14 - 15 - 16 - 20 - 21 - 15 - 14 - 5 - 6 - 7 - 6 - 13 - 12 - 11 - 8 - 9 - 10 - 44 - 43 - 45 - 46 - 45 - 61

Tabel 4.19 Data hasil pengujian pada simpel *environment*

No	Jumlah agen	Parameter			Jarak (cm)	Iterasi	Waktu	Keterangan
		α	β	ρ				
1	1	-	1	0.5	2960	71	40.3 s	Target ditemukan
	2	-	1	0.5	1630	61	17.9 s	Target ditemukan
	3	-	1	0.5	1430	68	13.7 s	Target ditemukan
	4	-	1	0.5	1530	101	16.2 s	Target ditemukan
	5	-	1	0.5	1320	132	17.2 s	Target ditemukan
2	1	-	2	0.5	2380	66	35.2 s	Target ditemukan
	2	-	2	0.5	2520	50	58.7 s	Target ditemukan
	3	-	2	0.5	1980	30	55 s	Target ditemukan
	4	-	2	0.5	1030	25	50 s	Target ditemukan
	5	-	2	0.5	2170	28	45 s	Target ditemukan

Tabel 4.12 Data hasil pengujian pada komplek *environment*

No	Jumlah agen	Parameter			Jarak (cm)	Iterasi	Waktu	Keterangan
		α	β	ρ				
1	1	1	1	0.5	4330	131	75.2 s	Target tidak ditemukan
	2	1	1	0.5	3520	188	58.7 s	Target tidak ditemukan
	3	1	1	0.5	2980	238	55 s	Target tidak ditemukan
	4	1	1	0.5	3030	240	50 s	Target tidak ditemukan
	5	1	1	0.5	2270	259	45 s	Target tidak ditemukan
2	1	2	1	0.5	6020	200	120 s	Target tidak ditemukan
	2	2	1	0.5	3090	170	51.5 s	Target tidak ditemukan
	3	2	1	0.5	2870	239	56 s	Target tidak ditemukan
	4	2	1	0.5	1980	230	45 s	Target tidak ditemukan
	5	2	1	0.5	4070	260	40 s	Target tidak ditemukan

Dari Tabel 4.19 dan 4.20 dapat dilihat hasil dari pengujian pada simpel environment dan juga komplek environment dimana pengunaan algoritma original anti feromon juga telah bisa menemukan target, hanya saja dari jumlah iterasi dan total jarak yang dibutuhkan untuk menemukan target lebih besar dari algoritma yang telah dimodifikasi sehingga dapat disimpulkan algoritma yang dimodifikasi lebih efektif dengan estimasi jarak tempuh yang lebih sedikit dibandingkan dengan anti feromon yang belum dimodifikasi.

BAB 5
KESIMPULAN

5.1 Kesimpulan

Hasil simulasi menunjukkan untuk kategori tanpa target pada simpel *environment* didapat dengan parameter α sama dengan 1 β sama dengan 1, minimum iterasi ialah 62 telah dapat mengunjungi semua node. Pada pengujian dengan target pada simpel environment dengan parameter α sama dengan 1 β sama dengan 1 didapat menggunakan 3 agen lebih efektif dalam mencari target dengan total jarak minimum sama dengan 380 cm, iterasi sama dengan 25, waktu sama dengan 5.2 detik. Untuk parameter α sama dengan 2 β sama dengan 1 lebih efektif menggunakan 4 agen dengan total jarak minimum sama dengan 380 cm, iterasi sama dengan 33, waktu sama dengan 5.5 detik. Untuk parameter α sama dengan 2 β sama dengan 2 efektif menggunakan 2 agen dengan total jarak minimum sama dengan 380 cm, iterasi sama dengan 17, waktu 4.9 detik.

Sedangkan pada lingkungan komplek untuk parameter α sama dengan 1 β sama dengan 1 efektif menggunakan 5 agen dengan jarak minimum sama dengan 610 cm, 95 iterasi, waktu 15.7 detik, untuk parameter α sama dengan 2 β sama dengan 1 efektif menggunakan 4 agen dengan jarak minimum sama dengan 470cm, 57 iterasi, waktu 12 detik, untuk parameter α sama dengan 1 β sama dengan 2 efektif menggunakan 3 agen dengan jarak minimum sama dengan 530 cm, 46 iterasi, waktu 11.3 detik. Semua pengujian menggunakan algoritma yang telah dimodifikasi dapat menemukan posisi target.

5.2 Saran

Untuk mengembangkan penelitian ini perlu ada tambahan beberapa target yang tidak diketahui dimana target tersebut statis dan dinamis, sehingga perlu membuat strategi dalam menemukan target dinamis dengan memanfaatkan algoritma ini

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

[1] Madhubhashi Senariyake, Ilankaikone Senthooran, Jan Carlo Barca, Hoam Chung, Joarder Kamruzzaman, Manzur Murshed "Search and tracking algorithms for swarms of robots : A survey". Journal Robotic and Autonomous Systems 2016, vol 422-434.

[2] Yuhua Zou, Dehan Luo, Weihai Chen, "Swarm Robotic Odor Source Localization Using Ant Colony Algorithm". IEEE International Conference on Control and Automation Christchurch, New Zealand, December 9-11-2009.

[3] Song-Hiang Chia, Kuo-Lan Su, Jr-Hung Guo, Cheng-Yun Chung, "Ant Colony System Based Mobile Robot Path Planning", Fourth International Conferennce on Geneetic and Evolutionery Computing, 2010.

[4] James Montgomery, Marcus Randall, "Anti-Pheromone as a Tool for Better Exploration of Search Space", International Conference, School of Information Technology,Bond University - Australia September 2002 Proceedings.

[5] Marco Dorigo, V. Maniezzo, A. Colorni, “The ant system: An autocatalytic optimizing process”, Technical Report No. 91-016 Revised, Politecnico di Milano, Italy, 1991

[6] Marco Dorigo, V. Maniezzo, and A. Colorni, “Ant system: optimization by a colony of cooperating agents”, IEEE Trans. Syst. Man Cybern. B, vol.26, no.1, pp. 29-41, 1996

[7] Antonio Franchi, Luigi Freda, Giuseppa Oriolo, Marielena Vendittelli, "A Decentralized Strategy for Cooperative Robot Exploration", First International Conference on Robot Communication and Coordination (ROBOCOMM 2007) October 15-15, 2007, Athens, Greece.

[8] Budi Santosa, Paul Willy, " Metaheuristic Method Concept and Implementation ", 2011 GunaWidya

[9] Marco Dorigo, Stutzle, “Ant Colony Optimization”. The MIT Press. Cambridge, Massachusetts. London, England. 2004

[10] Denis Darquennes, "Implementation and Aplications of Ant Colony Algorithms". Master Thesis, Faculty University Notre-Dame de la Paix, Namur Institut d'Informatique, 2004-2005

Halaman ini sengaja dikosongkan

LAMPIRAN

Tabel 3.1 matrik nilai dari $\eta(i,j)$ (jarak) 32 node

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
1	0	40	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	40	0	40	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
3	0	40	0	80	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
4	0	0	80	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
5	0	0	0	30	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
6	0	0	0	0	40	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
7	0	0	50	0	0	0	0	40	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
8	0	50	0	0	0	0	40	0	40	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
9	50	0	0	0	0	0	0	40	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9
10	0	0	0	0	0	40	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
11	0	0	0	0	0	0	0	0	0	40	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11
12	0	0	0	0	0	0	0	0	0	0	30	0	40	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	12
13	0	0	0	0	0	0	0	0	0	0	0	40	0	40	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	13
14	0	0	0	0	0	0	50	0	0	0	0	0	40	0	40	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14
15	0	0	0	0	0	0	0	50	0	0	0	0	0	40	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15
16	0	0	0	0	0	0	0	0	50	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	60	0	0	0	0	0	0	0	17
18	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	80	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	18
19	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	40	0	40	0	30	0	0	0	0	0	0	0	0	0	0	19
20	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	20
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	60	0	0	21
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	40	0	40	0	0	0	0	0	0	0	0	0	22
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	40	0	0	0	30	0	0	0	0	23
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	30	0	0	0	0	0	0	24
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	40	0	0	0	0	0	30	25
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	40	0	20	0	0	0	0	0	26
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	20	0	0	30	0	27
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	20	0	0	0	0	0	28

29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	29
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	100	0	30
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	100	0	60	31
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	60	0	32
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	

Tabel 3.2 matrik nilai dari $\eta(i,j)$ (jarak) 67 node

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
1	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	30	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
3	0	40	0	40	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
4	0	0	40	0	50	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
5	0	0	0	50	0	30	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
6	0	0	0	0	30	0	30	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
7	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
8	0	0	0	0	0	0	0	0	40	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
9	0	0	0	0	0	0	0	40	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9
10	0	0	0	0	0	0	0	0	30	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
11	0	0	0	0	0	0	0	30	0	40	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11
12	0	0	0	0	0	0	0	0	0	0	10	0	50	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	12
13	0	0	0	0	0	30	0	0	0	0	0	50	0	30	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	13
14	0	0	0	0	30	0	0	0	0	0	0	0	30	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14
15	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	30	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	15
16	0	0	0	30	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
17	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	17
18	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	40	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	18
19	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	19
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	20
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	30	0	20	0	0	0	0	0	0	20	0	0	0	0	0	21
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	30	0	0	0	0	0	0	0	0	0	0	0	22
23	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	30	0	0	0	0	0	20	0	0	0	0	0	0	23

24	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	20	0	20	0	0	0	0	0	0	0	24
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	20	0	0	0	0	0	0	25
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	20	0	0	0	0	0	26
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	20	0	50	0	0	0	0	27
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	50	0	0	0	0	0	0	28
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	30	0	0	0	0	29
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	40	0	0	0	30
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	40	0	40	0	0	31
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	40	0	30	0	32
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	33
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	70	35
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	36
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	37
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	39
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	40
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	41
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	42
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	43
44	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	45
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50
51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	52

53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	57
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	61
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	62
63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	63
64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	64
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65
66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	66
67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	67
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	

	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28
29	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29
30	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30
31	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	34
	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	

	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	
35	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35
36	40	0	30	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36
37	0	30	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37
38	0	0	10	0	10	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38
39	0	0	0	10	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39
40	0	0	0	0	30	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40
41	0	0	0	0	0	20	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	41
42	0	0	0	0	0	0	50	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	42
43	0	0	0	0	0	0	0	10	0	20	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	43

44	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	44
45	0	0	0	0	0	0	0	0	30	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	45
46	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	46
47	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47
48	0	0	0	30	0	0	0	0	0	0	0	0	30	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48
49	0	30	0	0	0	0	0	0	0	0	0	0	0	40	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	40	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	50
51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	51
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	30	52
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	30	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	53
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	20	0	0	0	0	0	0	0	0	0	30	0	0	54
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	55
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	30	0	0	56
57	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	30	0	30	0	0	0	0	0	0	0	0	0	57
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	30	0	0	0	58
59	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	30	0	0	0	0	59
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	30	0	0	0	0	0	60
61	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	61
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	30	0	0	0	0	0	0	62
63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	30	0	0	0	63
64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	30	0	0	0	0	64
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	70	0	65	
66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	70	0	50	66	
67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	50	0	67	
	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	

List Code Matlab :

GUI MATLAB :

```
function varargout = maingui(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @maingui_OpeningFcn, ...
                   'gui_OutputFcn',  @maingui_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```
function maingui_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
global mode
mode=1;
simul;
```

```
function varargout = maingui_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;
```

```
function but_run_Callback(hObject, eventdata, handles)
global mode
handles.textprocess.String=(['Processing ...']); pause(0.5);
[D J]=DistanceMatrix();
if mode==2
    D=J;
end
aco
```

```
function edit_initial_Callback(hObject, eventdata, handles)
function edit_initial_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_target_Callback(hObject, eventdata, handles)
function edit_target_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end
global first target
```

```
function edit_n_Callback(hObject, eventdata, handles)
function edit_n_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_alpha_Callback(hObject, eventdata, handles)
function edit_alpha_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_beta_Callback(hObject, eventdata, handles)
function edit_beta_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function radio_env1_Callback(hObject, eventdata, handles)
global mode
mode=1;
simul
```

```
function radio_env2_Callback(hObject, eventdata, handles)
global mode
mode=2;
simul2
```

```
function Iterasi_Callback(hObject, eventdata, handles)
function Iterasi_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

ACS :

```
%=====================================================================%
%=                "Thesis S2 Teknik Sistem Pengaturan ITS          =%
%=                          YOAN PURBOLINGGA                       =%
%=====================================================================%
% ========================== main program ACO ==================%

clc;
delete('proses.txt');
diary('proses.txt');
D = makezeroinfinity(D);
d = 1./D;
N = size(d);          %calculate dimension

clc;
finish = 0;

first  = str2num(handles.edit_initial.String);
target = str2num(handles.edit_target.String);  % target
n      = str2num(handles.edit_n.String);       % ants number
a      = str2num(handles.edit_alpha.String);;%weight of pheromone
b      = str2num(handles.edit_beta.String);;%heuristik
rho    = 0.5;                                  % evaporation rate
phe0   = 1;                                    % initial pheromone
phe    = phe0*ones(N(1),N(2));

av = d>0;                                   %declare availibility


%% ============== define first and target position ============%%

target = round(target); first = round(first);
if target == 0 || target<1 || target>N(1)
     target = randi([1 N(1)]);
end

%% ================ ant initially in first position
====================%%

if first == 0 || first<1 || first>N(1)
        posit = randi([1 N(1)],1,n);
else
        posit = first*ones(1,n);
end

for i=1:length(posit)
        while posit(i)== target
            posit(i) = randi([1 N(1)]);
        end
display(['Posisi awal agent ' num2str(i) ' : 'num2str(posit(i))]);
end
display(['===> Target: ' num2str(target)]);
```

```
%% ==================== End define first and target ===========%%

%% ============================ make color ====================%%

colsem=randi([1 240],n,3); colsem=colsem/255;

tic  %counting

% display GUI
if mode == 1
   simul;
else
   simul2;
end

[y x] = find(idx==target); %define target point
hold on;
plot(x,y,'xb','Parent',handles.axes_graf);
text((x-0.5),y,'target','Parent',handles.axes_graf);
matpos = posit';
mat_h  = {};
for sem = 1:n
    [y x] = find(idx == posit(sem));
    g = plot(x,y,'--
gs','Parent',handles.axes_graf,'Color',colsem(sem,:),'MarkerEdgeCo
lor',colsem(sem,:),'MarkerFaceColor',colsem(sem,:));
    mat_h = cat(1,mat_h,{g});
end
dist = zeros(1,n);
display(['======= Pheromone awal ======= ']);
mat2teks2Dshow(phe);
iterasi=0;

while 1
    matpos = [matpos matpos(:,end)];
    for sem = 1:n     % untuk setiap semut

        iterasi = iterasi+1;
        handles.Iterasi.String = (iterasi);

        display(['===> Iterasi : ' num2str(iterasi)]);
        display(['-------   Agent ' num2str(sem) ' -----------']);
        display(['===> Posisi sekarang : ' num2str(posit(sem))]);

        % calculate possibility and availibility
        pos = zeros(1,N(1)); h = av.*d;

display(['===> Pheromone tujuan : 'mat2teks(phe(posit(sem),:)) ]);
display(['===> Max pheromone :'num2str(max(phe(posit(sem),:))) ]);
display(['===> Bobot pheromone:'mat2teks((max(phe(posit(sem),:))-
phe(posit(sem),:)).^a)  ]);
display(['===> Bobot Jarak : ' mat2teks( h(posit(sem),:).^b ) ]);
```

```
    for j = 1:N(1)
        pos(j)=( (max(phe(posit(sem),:))-phe(posit(sem),j)).^a )*
( h(posit(sem),j).^b );

%pos(j)=(phe(posit(sem),j).^a)*(h(posit(sem),j).^b);
    end
            if sum(pos) == 0
                for j = 1:N(1)
                    if h(posit(sem),j)>0 && j ~= posit(sem)
                        pos(j) = 1;
                    end
                end
                pos = pos/sum(pos);
            else
                pos = pos/sum(pos);
            end

            display(['===> Possibility : ' mat2teks(pos) ]);

            % calculate cumulative
            cum   = zeros(1,N(1));
            cum(1)= pos(1);
            for j = 2:N(1)
                cum(j) = cum(j - 1) + pos(j);
            end
            display(['===> Cumulative : ' mat2teks(cum) ]);

            % generate random number
            r = rand();
            display(['===> Random : ' num2str(r) ]);

            % take cum<r minimum
            take = cum < r;
            take = find(take == 0);
            take = min(take);
            display(['===> Tujuan : ' num2str(take)]);

            % ants give pheromone
            phe(posit(sem),take) = phe(posit(sem),take) + phe0;
            phe(take,posit(sem)) = phe(take,posit(sem)) + phe0;
            posit(sem) = take;

            % save
            matpos(sem,end) = take;

     %calculate distance
     dist(sem) = dist(sem) + D(matpos(sem,end-1),matpos(sem,end));

% check target
if  take == target
    handles.textprocess.String = (['Found: ' num2str(toc) ' s']);
    finish = 1;
    break;
    end

end
```

```
% plot grafik

for sem = 1:n
    [ya xa] = find(idx == matpos(sem,end-1));
    [yb xb] = find(idx == matpos(sem,end));
    plot([xa xb],[ya yb],'Color',colsem(sem,:),'LineWidth',1);
    set(mat_h{sem},'Visible','off')
    [y x] = find(idx == posit(sem));
    g = plot(x,y,'--
gs','Parent',handles.axes_graf,'Color',colsem(sem,:),'MarkerEdgeCo
lor',colsem(sem,:),'MarkerFaceColor',colsem(sem,:));
    mat_h{sem} = g;
end

% evaporating
phe = (1-rho)*phe;

% check is finish?
if finish
   textstring = {}; maxstring = 0;
   clc;
for sem = 1:n
    getstr = strcat([' Agent ' num2str(sem) ' ('
num2str(dist(sem)) ' cm) : ' mat2teks(matpos(sem,:))]);

    if length(getstr) > maxstring
       maxstring = length(getstr);
    end
       textstring = cat(1,textstring,getstr);
end


set(handles.table_hasil,'Data',textstring,'ColumnWidth',{maxstring
*8,0});
   break;
end

pause(0.5);
end
hold off; diary off
```

Make zero Infinity :

```
function out=makezeroinfinity(D)
    a=size(D);
    out=D;
    for i=1:a(1)
        for j=1:a(2)
            if out(i,j)==0
                out(i,j)=inf;
            end
        end
    end

end
```

Mat2Text :

```
function out=mat2teks(matriks)

    out=':';
    if length(matriks)>=1
        out=num2str(matriks(1));
        for i=2:length(matriks)
            out=strcat([out ' - ' num2str(matriks(i))]);
        end
    end

end
```

Environment 32 Node :

```
    dot=[...
    1 0 0 1 0 0 0 0 0 0 1;
    0 0 0 0 0 0 0 0 0 0 0;
    1 0 1 1 1 0 1 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0;
    0 0 1 0 1 0 0 1 0 0 1;
    0 0 0 0 0 0 0 0 0 0 0;
    1 0 0 0 1 0 0 1 0 0 1;
    0 0 0 0 0 0 0 0 0 0 0;
    1 0 1 0 1 0 0 1 0 0 1;
    0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 1 0 0 1;
    1 0 1 0 1 0 0 0 0 0 0;
    0 0 0 0 0 0 0 1 0 0 1;
    0 0 0 0 0 0 0 0 0 0 0;
    1 0 1 0 1 0 0 0 0 0 1];

    I=[...
    1 1 1 1 1 1 1 1 1 1 1;
    1 0 0 1 0 0 0 0 1 1 1;
    1 1 1 1 1 1 1 0 1 1 1;
    1 0 1 0 1 0 0 0 0 0 1;
    1 0 1 1 1 1 1 1 1 1 1;
    1 0 0 0 0 0 1 1 0 0 0;
    1 1 1 1 1 1 1 1 1 1 1;
    0 0 0 0 1 0 1 1 0 1 1;
    1 1 1 1 1 1 1 1 1 1 1;
    1 0 1 0 1 0 0 0 0 0 1;
    1 0 1 0 1 0 1 1 1 1 1;
    1 1 1 1 1 0 1 1 0 0 0;
    1 0 1 0 1 0 1 1 1 1 1;
    1 0 1 0 1 0 0 0 0 0 1;
    1 1 1 1 1 1 1 1 1 1 1];

    I=I*255;


    idx=[...
    32 0 0 31 0 0 0 0 0 0 30;
```

```
    0 0 0 0 0 0 0 0 0 0;
    25 0 26 27 28 0 29 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0;
    0 0 24 0 23 0 0 22 0 0 21;
    0 0 0 0 0 0 0 0 0 0 0;
    17 0 0 0 18 0 0 19 0 0 20;
    0 0 0 0 0 0 0 0 0 0 0;
    16 0 15 0 14 0 0 13 0 0 12;
    0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 10 0 0 11;
    9 0 8 0 7 0 0 0 0 0 0;
    0 0 0 0 0 0 0 6 0 0 5;
    0 0 0 0 0 0 0 0 0 0 0;
    1 0 2 0 3 0 0 0 0 0 4];

    imshow(I,'Parent',handles.axes_graf)
    %image(I,'Parent',handles.axes_graf,'CDataMapping','scaled')
    hold on
    listdot=[];
    for i=1:size(dot,1)
        for j=1:size(dot,2)

            if dot(i,j)==1
                listdot=[listdot;j i];
            end

        end
    end

plot(listdot(:,1),listdot(:,2),'o','Parent',handles.axes_graf);hol
d off;
```

Environment 67 Node :

```
    dot=[...
    1 0 0 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 1 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    1 0 1 1 1 0 0 1 0 1 0 1 0 1 0 1 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 1 0 1 0 1 0 0 1 0 1 0 0 0 1 0 0 1 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    1 0 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 1 1 0 1;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    1 0 1 0 1 0 1 0 1 0 0 0 1 0 0 1 0 1 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 1 0 1 0 0 0;
    1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 1 1 0 0 0 1;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0 1];
```

```
    I=[...
    0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 1 1;
    0 1 1 0 1 1 0 0 1 1 1 0 1 0 1 0 1 1 0 1 1;
    0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0;
    0 1 0 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 0 0 0;
    0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1;
    0 1 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 0 1 1;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    1 1 1 0 0 1 0 1 0 1 0 1 0 1 1 1 1 0 1 1 0;
    0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0;
    0 1 0 1 0 1 1 1 0 1 0 1 0 1 1 0 1 0 1 1 0;
    0 1 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1 1 0;
    0 0 0 0 0 0 1 0 1 0 1 1 1 0 1 1 0 1 1 1 1 0;
    0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 1 0 1 0 1 0 1 1 1 0 1 0 1 1 1 0 1 1 1 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0];
    I=I==0;

    I=I*255;


    idx=[...
    67 0 0 66 0 0 0 65 0 0 0 64 0 63 0 62 0 0 61 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    52 0 53 54 55 0 0 56 0 57 0 58 0 59 0 60 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 51 0 50 0 49 0 0 48 0 47 0 0 0 46 0 0 45 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    34 0 0 0 35 0 36 0 37 38 39 0 40 41 0 0 0 42 43 0 44;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    33 0 32 0 31 0 30 0 29 0 0 0 28 0 0 27 0 26 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 20 0 21 0 22 0 23 0 0 24 0 25 0 0 0;
    19 0 18 0 17 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 16 0 15 0 14 0 13 0 0 12 11 0 0 0 10;
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    1 0 2 0 3 0 4 0 0 0 5 0 6 0 7 0 8 0 0 0 9];

    imshow(I,'Parent',handles.axes_graf)
    %image(I,'Parent',handles.axes_graf,'CDataMapping','scaled')
    hold on
    listdot=[];
    for i=1:size(dot,1)
        for j=1:size(dot,2)

            if dot(i,j)==1
                listdot=[listdot;j i];
            end

        end
    end
    size(listdot)

plot(listdot(:,1),listdot(:,2),'o','Parent',handles.axes_graf);hol
d off;
```

Halaman ini sengaja dikosongkan

BIODATA

Yoan Purbolingga dilahirkan di Bengkalis pada Tanggal 27 Maret 1993. Putra Pertama dari Bapak Zahari dan Ibu Misnawati. Pendidikan formalnya dimulai dari SDN 003 – Bengkalis, MTSN 1 – Bengkalis, SMKN 1 – Bengkalis, kemudian melanjutkan studi di D3 Jurusan Teknik Elektro – Politeknik Negeri Bengkalis pada tahun 2010, selanjutnya di D4 Jurusan Teknik Elektronika – Politeknik Elektronika Negeri Surabaya pada tahun 2013, penulis telah menyelesaikan pendidikan sarjananya dan mendapat gelar Sarjana Sains Terapan. Pada tahun 2016, penulis melanjutkan pendidikan magister di Jurusan Teknik Elektro, bidang keahlian Teknik Sistem Pengaturan di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Penulis telah menyelesaikan sidang tesis pada tanggal 28 Desember 2018 sebagai salah satu syarat untuk mendapatkan gelar **Magister Teknik (M.T.)**