

Pemilihan Jalur Evakuasi Dalam Keadaan Darurat Menggunakan Algoritma *Quantum Ant-Colony*

Path Selection In Emergency Evacuation Using Quantum Ant-Colony Algorithm

Fransisca Arvevia I A¹, Jondri², Anditya Arifianto³

^{1,2,3} Prodi S1 Teknik Informatika, Fakultas Teknik, Universitas Telkom

¹fransiscaia@gmail.com, ²jondri@telkomuniversity.ac.id, ³anditya.arifianto@gmail.com³

Abstrak

Evakuasi dalam keadaan darurat pada sebuah gedung sangatlah penting untuk menyelamatkan nyawa manusia. Pemilihan jalur evakuasi ketika terjadi suatu bencana sangatlah penting, pemilihan jalur evakuasi yang tepat dapat menekan jumlah korban jiwa yang berjatuhan. Berbagai metode dan algoritma simulasi penyeleksian jalur evakuasi telah banyak dikembangkan. Di antaranya algoritma *ant-colony optimization* (ACO) dan artifisial *bee-colony* (ABC). Kedua algoritma tersebut mengadopsi perilaku individu terhadap lingkungan disekitarnya, sehingga cocok digunakan untuk seleksi jalur evakuasi. Pada penelitian ini digunakan algoritma *quantum ant-colony* (QACA) yang merupakan pengembangan dari algoritma ACO yang dikombinasikan dengan algoritma *quantum-inspired evolutionary* (QEA). Pada algoritma ini, algoritma QEA digunakan untuk memperbaharui feromon pada algoritma ACO untuk menghasilkan simulasi dengan solusi yang lebih optimal karena memiliki laju konvergensi yang cepat.

Kata kunci: Jalur evakuasi, algoritma *ant-colony optimization* (ACO), algoritma *quantum-inspired evolution* (QEA), algoritma *quantum ant-colony* (QACA), feromon, simulasi.

Abstract

Evacuation in emergency situation at the building is very important to save human lives. Selection of evacuation path when there is a disaster is very important, the right evacuation path selection can reduce the number of fatalities are falling. Various method and algorithms of selecting evacuation path simulation has been developed. Including ant-colony optimization algorithm (ACO) and artificial bee-colony (ABC). Both of these algorithms adopt individual behavior towards the surrounding environment, so it is suitable for evacuation path selection. In this study used quantum ant-colony algorithm (QACA) which is a development of the ACO algorithm combine with quantum-inspired evolutionary (QEA). In these algorithm, QEA algorithm is used to update the pheromone at ACO algorithm to generate a simulation with a more optimal solution because it has fast convergence rate.

Keywords: Selection path, ant-colony algorithm (ACO), quantum-inspired evolutionary algorithm (QEA), quantum ant-colony (QACA), pheromone, simulation.

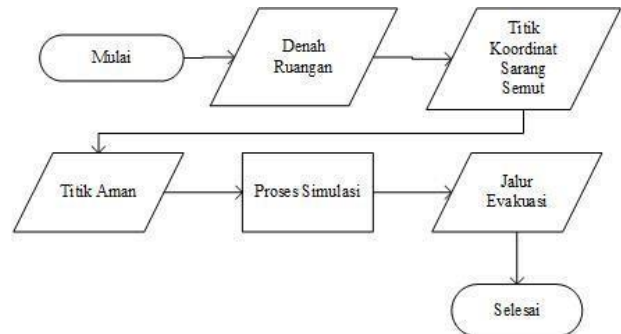
1. Pendahuluan

Evakuasi dalam keadaan darurat pada sebuah gedung sangatlah penting untuk menyelamatkan nyawa manusia. Keadaan darurat bisa disebabkan oleh bencana alam seperti banjir atau gempa bumi, kecelakaan atau kebakaran. Setiap gedung pasti memiliki banyak jalur evakuasi khusus untuk penanganan situasi darurat. Pemilihan jalur evakuasi yang salah bisa menyebabkan korban jiwa berjatuh karena waktu evakuasi yang lama sehingga memungkinkan terjadinya aksi saling dorong, bertabrakan dan mungkin juga mereka bisa jatuh lalu terinjak-injak saat berebut keluar dari gedung. Maka, diperlukan pemilihan jalur evakuasi yang tepat dan dalam waktu yang singkat untuk menyelamatkan korban ke dalam zona aman atau keluar dari gedung. Dengan adanya jalur evakuasi yang telah terpilih, diharapkan mampu mengurangi angka korban jiwa yang berjatuh.

Telah banyak metode dan algoritma yang telah tercipta untuk mensimulasikan seleksi jalur evakuasi. Diantaranya algoritma *ant-colony* dan artifisial *bee-colony*. Algoritma ACO memiliki simulasi yang baik dalam memecahkan masalah optimasi yang terinspirasi dari perilaku koloni semut dalam mencari makan, sedangkan algoritma ABC mengadopsi perilaku lebah mencari madu. Perbedaan dari kedua algoritma ini adalah pada tahapan-tahapannya. Namun, keduanya sama-sama mencerminkan interaksi individu terhadap lingkungannya, sehingga cocok untuk menyelesaikan seleksi jalur evakuasi.

Dalam tugas akhir ini, digunakan algoritma *Quantum Ant-Colony* (QACA) untuk pemilihan jalur evakuasi darurat. QACA merupakan kombinasi algoritma ACO dengan algoritma *quantum-inspired evolutionary*(QEA). Algoritma QEA

digunakan untuk memperbaharui feromon[1]. Berdasarkan penelitian sebelumnya, simulasi dari algoritma QACA memiliki laju konvergensi yang cepat dan dapat menghindari konvergensi prematur sehingga QACA merupakan solusi baru untuk aplikasi ACO dalam pemilihan jalur evakuasi. Secara umum alur proses dari simulasi dapat dilihat pada gambar 1.

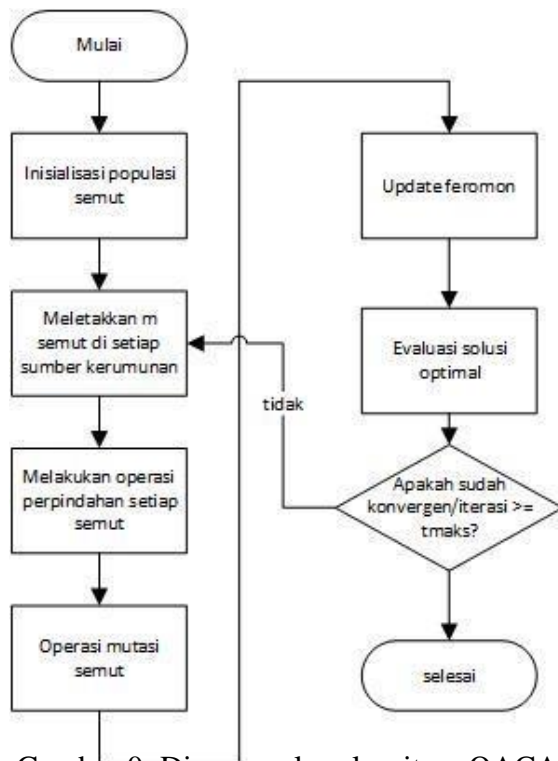


Gambar 1. Diagram Alur Sistem

2. Algoritma Quantum Ant-Colony (QACA)

QACA merupakan kombinasi dari algoritma ACO dan QEA. Dengan kombinasi tersebut, didapat Q-bit dan Q-gate yang digunakan untuk memperbaharui feromon. Penerapan QEA dalam algoritma ACO untuk mengoptimalkan jalur pilihan berdasarkan jalan yang dilalui oleh semut. Dalam QACA, Q-bit digunakan untuk merepresentasikan feromon, sedangkan untuk memperbaharui feromon menggunakan Q-gate. Probabilitas amplitudo Q-bit digunakan untuk merepresentasikan informasi dari posisi semut. Q-gate digunakan untuk memperbaharui Q-bit semut.

Berikut alur algoritma QACA :



Gambar 0: Diagram alur algoritma QACA

Dalam algoritma QACA, inisialisasi populasi semut yang mempunyai jumlah m semut, dengan $Q(t) = (Q_1^t, Q_2^t, \dots, Q_m^t)$ dimana t merupakan iterasi, dan $(1, 2, \dots, m)$ dilambangkan dengan j .

$$Q_j^t = \begin{pmatrix} \alpha_{j1}^t & \alpha_{j2}^t & \dots & \alpha_{jm}^t \\ \beta_{j1}^t & \beta_{j2}^t & \dots & \beta_{jm}^t \end{pmatrix} \quad (2.1)$$

Jumlah dari Q-bit adalah n , dengan inisialisasi dari α dan β sebesar $1/\sqrt{2}$. Inisialisasi $t = 0$ dan jumlah iterasi dibatasi sebesar t_{maks} . Kemudian untuk melakukan *update* feromon serta pemilihan *node* selanjutnya pada iterasi ke i menggunakan penggabungan Q-bit dengan Q-gate dengan rumus :

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \quad (2.2)$$

dimana θ_i merupakan sudut rotasi, θ_i dihitung dengan rumus :

$$\theta_i = \Delta\theta * f(\alpha_i, \beta_i) \quad (2.3)$$

$\Delta\theta$ merupakan variable yang terkait dengan jumlah iterasi, nilai sudut rotasi dan tingkat konvergensi yang didapat dari rumus :

$$\Delta\theta = 5 * \exp\left(-\frac{t}{t_{maks}}\right) \quad (2.4)$$

Sedangkan fungsi $f(\alpha_i, \beta_i)$ digunakan untuk mengontrol arah sudut rotasi dari Q-gate, yang dihitung dengan rumus :

$$f(\alpha_i, \beta_i) = (d_{ibest}/d_{inow})(\varphi_{ibest} - \varphi_{inow}) \quad (2.5)$$

d_{ibest} merupakan solusi optimal dari Q-bit yang dicari dari keseluruhan solusi yang didapat ketika mencapai iterasi ke- i . Adapun rumus untuk menghitungnya ialah :

$$d_{ibest} = \beta_{ibest}/\alpha_{ibest} \quad (2.6)$$

Sedangkan d_{inow} merupakan solusi dari Q-bit yang didapat saat iterasi ke- i . Rumusnya :

$$d_{inow} = \beta_{inow}/\alpha_{inow} \quad (2.7)$$

Kemudian, φ_{ibest} merupakan bentuk optimal dari Q-bit ke- i , dihitung dengan rumus :

$$\varphi_{ibest} = \arctan \beta_{ibest}/\alpha_{ibest} \quad (2.8)$$

Dan φ_{inow} merupakan bentuk pada Q-bit ke- i , rumusnya :

$$\varphi_{inow} = \arctan \beta_{inow}/\alpha_{inow} \quad (2.9)$$

3. Proses Simulasi

Proses penelitian ini menggunakan algoritma *quantum ant-colony* (QACA). Algoritma tersebut digunakan untuk menjalankan simulasi dengan tujuan membentuk jalur evakuasi terdekat berdasarkan feromon yang ada pada semut. Pada Gambar 1 dijelaskan bahwa sistem mulai dijalankan dengan membuat denah ruangan sebuah gedung, kemudian dilanjutkan dengan meng-inputkan titik-titik koordinat dari sarang semut. Pada titik-titik koordinat tersebut akan muncul semut-semut yang digunakan untuk proses simulasi pembentukan jalur evakuasi. Langkah selanjutnya ialah menentukan titik-titik aman yang merupakan tujuan para semut, titik aman harus berada diluar gedung. Setelah semua titik ditentukan, proses simulasi bisa dijalankan. Proses simulasi akan menghasilkan jalur evakuasi

dari setiap titik sarang semut menuju titik aman yang berada diluar gedung.

Populasi semut akan dibangkitkan pada setiap titik koordinat sarang semut. Proses pembangkitan populasi semut menggunakan Q-bit, dan prosesnya telah dijelaskan pada rumus 2.1 Setelah populasi dijalankan, sistem akan melakukan proses *update* feromon yang digunakan untuk menentukan langkah semut menuju titik aman. Proses peng-*update*-an feromon menggunakan Q-gate, sesuai yang ditulis pada rumus 2.2.

Semut akan kembali ke sarangnya setelah menemukan titik aman, semut-semut akan berjalan mundur menggunakan jalur terakhir yang dilewati untuk menuju ke sarangnya. Hal tersebut dilakukan untuk meninggalkan feromon pada jalur tersebut sebagai petunjuk jalan bagi semut lain.

Proses tersebut yang akan menciptakan jalur evakuasi yang tepat dan efektif dari titik sarang menuju titik aman. Simulasi akan berhenti ketika mencapai iterasi maksimal yang telah ditentukan, dimana pada iterasi tersebut sudah mencapai keadaan konvergen. Keseluruhan alur algoritma QACA telah digambarkan pada Gambar 2.

4. Pengujian dan Analisis

4.1 Skenario Pengujian

Sistem yang telah dirancang akan diuji dengan skenario uji yang telah ditentukan. Pengujian sistem pada penelitian ini dilakukan untuk mengetahui jalur yang terbentuk dari simulasi semut yang menerapkan algoritma QACA. Dalam

pengujian ini akan digunakan 3 denah ruangan yang berbeda. Pengujian ini juga bertujuan untuk melihat pengaruh jumlah iterasi maksimal terhadap laju konvergensi populasi semut dalam sistem.

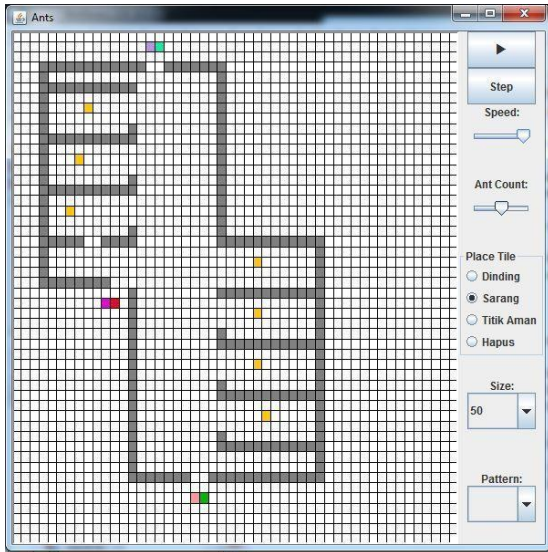
Pada pengujian sistem ini, terdapat beberapa pengaturan yang dilakukan sebagai parameter dalam sistem yang telah dibangun. Berikut pengaturan yang dilakukan :

1. Jumlah maksimal populasi semut adalah 100 semut.
2. Nilai awal α dan β sebesar $1/\sqrt{2}$.
3. Dinding ditandai dengan warna abu-abu, sarang semut ditandai dengan warna kuning, dan titik aman mempunyai warna *random*.
4. Warna hijau pada aplikasi menandakan feromon.

Dalam penelitian ini, pengujian sistem dilakukan dengan menggunakan 3 denah ruangan yang berbeda, dimana setiap denah memiliki jumlah ruangan dan pintu keluar yang berbeda. Pengujian sistem dibagi menjadi 3 skenario, yaitu setiap denah diuji dengan 3 jumlah iterasi maksimal (t_{maks}) yang berbeda.

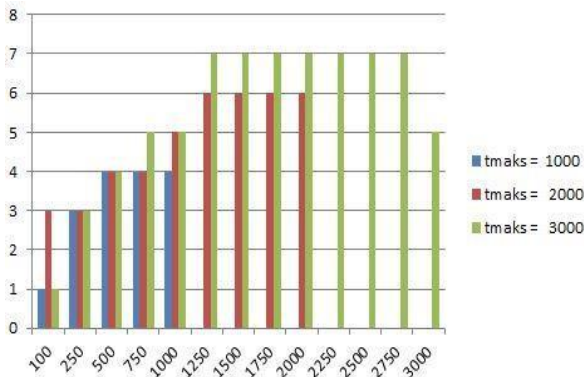
4.2 Analisis Hasil Pengujian

Pada pengujian yang pertama, sistem diuji dengan menggunakan denah A. Denah A memiliki 7 ruangan, dimana setiap ruangnya terdapat 1 sarang semut. Pada denah tersebut terdapat 3 pintu keluar serta 3 daerah aman yang berada diluar gedung namun tidak jauh dari pintu keluar. Berikut gambaran dari denah A :



Gambar 3 : Denah A

Pada denah A dilakukan pengujian sebanyak 3 kali dengan masing-masing iterasi maksimal sebanyak 1000, 2000, dan 3000. Pada setiap iterasi tertentu akan terbentuk sejumlah jalur yang berasal dari sebuah sarang semut menuju satu titik aman yang akan dijelaskan pada gambar 4.

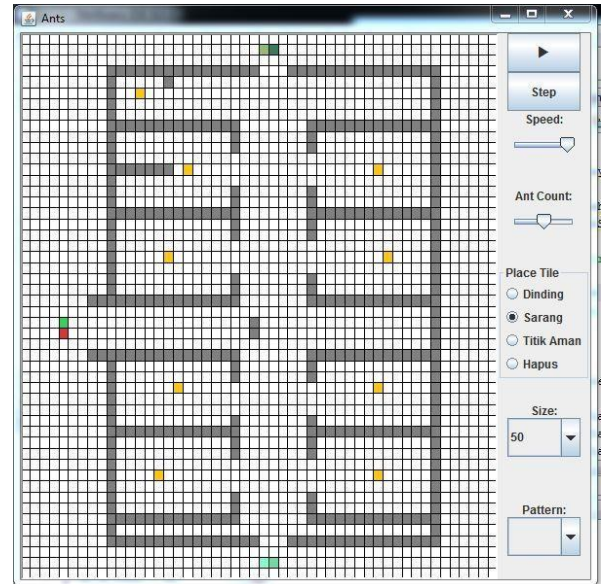


Gambar 0 : Grafik Pengujian Denah A

Hasil pengujian menggunakan denah A dapat dilihat pada grafik diatas. Berdasarkan grafik diatas, jumlah jalur yang terbentuk dengan jumlah iterasi maksimal 1000 kali hanya bisa membentuk 4 jalur saja. Jika jumlah iterasi maksimal ditambah menjadi 2000 kali, simulasi dapat menghasilkan 6 jalur. Ketika iterasi maksimal menjadi 3000 kali, jalur yang terbentuk menjadi 7 jalur meskipun pada akhir iterasi jumlah jalur berkurang menjadi 5 jalur. Berdasarkan grafik pada Gambar 4 4, denah A memiliki kondisi konvergensi

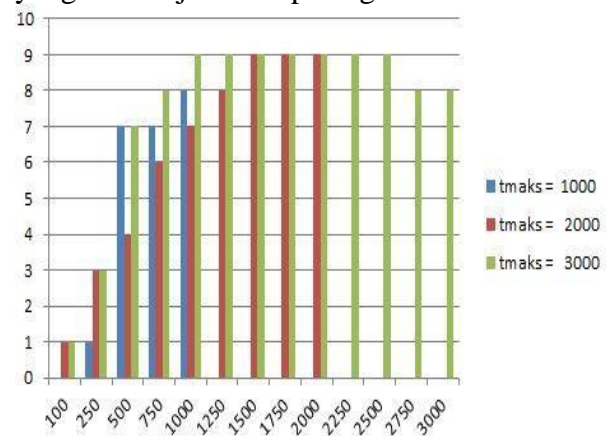
yang hampir sempurna ketika nilai t_{maks} sebesar 3000. Laju konvergensi yang paling bagus yaitu ketika simulasi berjalan pada iterasi ke-1500 hingga 2750.

Pada pengujian yang kedua, sistem diuji dengan menggunakan denah B. Denah B memiliki 9 ruangan, dimana setiap ruangnya terdapat 1 sarang semut. Pada denah tersebut terdapat 3 pintu keluar serta 3 daerah aman yang berada diluar gedung namun tidak jauh dari pintu keluar. Berikut gambaran dari denah B :



Gambar 5 : Denah B

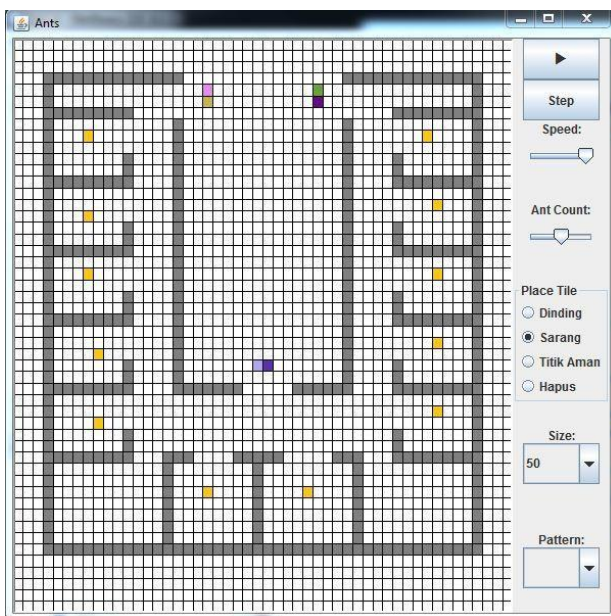
Pada denah B dilakukan pengujian sebanyak 3 kali dengan masing-masing iterasi maksimal sebanyak 1000, 2000, dan 3000. Pada setiap iterasi tertentu akan terbentuk sejumlah jalur yang berasal dari sebuah sarang semut menuju satu titik aman yang akan dijelaskan pada gambar 6.



Gambar 6 : Grafik Pengujian Denah B

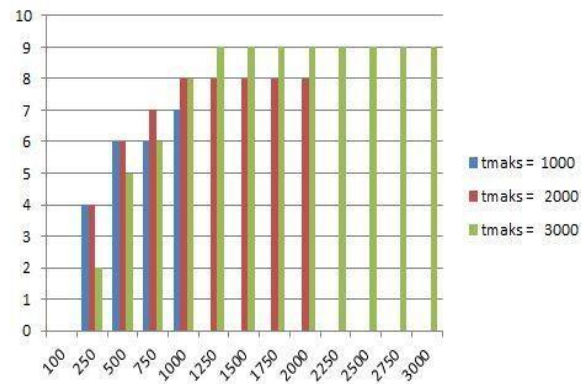
Hasil pengujian menggunakan denah B dapat dilihat pada grafik diatas. Berdasarkan grafik diatas, jumlah jalur yang terbentuk dengan jumlah iterasi maksimal 1000 kali hanya bisa membentuk 8 jalur saja. Jika jumlah iterasi maksimal ditambah menjadi 2000 kali, simulasi dapat menghasilkan 9 jalur. Ketika iterasi maksimal menjadi 3000 kali, jalur yang terbentuk sejumlah 9 jalur meskipun pada akhir iterasi jumlah jalur berkurang menjadi 8 jalur. Berdasarkan grafik diatas, dengan menggunakan nilai t_{maks} sebesar 2000, simulasi pembentukan jalur telah mencapai keadaan konvergen. Laju konvergensi untuk denah B ini tergolong cepat.

Pada pengujian yang kedua, sistem diuji dengan menggunakan denah C. Denah C memiliki 12 ruangan, dimana setiap ruangnya terdapat 1 sarang semut. Pada denah tersebut terdapat 3 pintu keluar serta 3 daerah aman yang berada diluar gedung namun tidak jauh dari pintu keluar. Berikut gambaran dari denah C :



Gambar 7 : Denah C

Pada denah C dilakukan pengujian sebanyak 3 kali dengan masing-masing iterasi maksimal sebanyak 1000, 2000, dan 3000. Pada setiap iterasi tertentu akan terbentuk sejumlah jalur yang berasal dari sebuah sarang semut menuju satu titik aman yang akan dijelaskan pada gambar 8.



Gambar 8 : Grafik Pengujian Denah C

Hasil pengujian menggunakan denah C dapat dilihat pada grafik diatas. Berdasarkan grafik diatas, jumlah jalur yang terbentuk dengan jumlah iterasi maksimal 1000 kali hanya bisa membentuk 7 jalur saja. Jika jumlah iterasi maksimal ditambah menjadi 2000 kali, simulasi dapat menghasilkan 8 jalur. Ketika iterasi maksimal menjadi 3000 kali, jalur yang terbentuk menjadi 9 jalur saja.

Pada denah ini, meskipun nilai t_{maks} sebesar 3000 tidak terjadi pengurangan jalur. Laju konvergensi pada denah ini tergolong cepat meskipun kondisi konvergensi belum terpenuhi. Dan masih terdapat 3 titik sarang semut yang belum menemukan titik aman. Untuk mencapai kondisi konvergen mungkin nilai t_{maks} perlu ditambah lagi.

Jumlah jalur dapat berkurang karena terjadinya penguapan feromon yang menyebabkan semut kehilangan arah menuju titik aman. Feromon pada jalur-jalur tertentu dapat menguap karena feromonnya tidak diperbaharui. Hal itu disebabkan oleh semut yang tidak kembali ke sarang dimana dia muncul, sehingga jalur yang telah terbentuk hanya diperbaharui oleh semut yang membawa nilai feromon yang kecil yang melewati jalur tersebut menuju titik aman.

Setiap semut yang berhasil menemukan titik aman akan kembali ke sarang terdekat dari titik aman melalui jalur yang mempunyai nilai feromon paling besar. Nilai feromon yang dibawa semut setelah menemukan titik aman digunakan untuk memperbaharui feromonon pada jalur

yang dilaluinya. Kondisi konvergen akan terpenuhi ketika setiap jalur selalu diperbaharui nilai feromonnya sehingga setiap semut yang keluar dari sarangnya akan melewati jalur terbentuk dari sarang menuju ke titik aman.

Besarnya nilai t_{maks} mempengaruhi jalur yang terbentuk karena nilai t_{maks} digunakan untuk menentukan langkah semut dari satu titik ke titik selanjutnya. Hal tersebut ditunjukkan pada rumus 2.2, dimana perhitungan sudut rotasi dipengaruhi oleh besarnya nilai t_{maks} . Adanya sudut rotasi menyebabkan membuat feromon tidak menyebar ke sembarang arah, serta membuat laju konvergensi semakin cepat.

5. Kesimpulan dan Saran

Berdasarkan pengujian dan analisis pengujian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Sistem simulasi pemilihan jalur evakuasi menggunakan algoritma *quantum ant-colony* (QACA) menghasilkan laju konvergensi yang lebih cepat dibandingkan dengan algoritma *ant-colony* biasa.
2. Besarnya nilai t_{maks} mempengaruhi sudut rotasi yang digunakan untuk operasi mutasi semut.
3. Laju konvergensi yang cepat disebabkan penyebaran feromon yang terarah.

Adapun saran untuk perbaikan dari sistem prediksi ini selanjutnya antara lain :

1. Perbanyak variasi besaran nilai t_{maks} agar setiap denah ruangan mempunyai kondisi konvergen.
2. Algoritma *quantum ant-colony* (QACA) dapat dimodifikasi atau dikombinasikan dengan algoritma lain untuk disesuaikan dengan kebutuhan.
3. Sebaiknya simulasi dihentikan ketika keadaan telah mencapai kondisi konvergen tanpa menunggu iterasi selesai.

Daftar Pustaka:

- [1] Feng Zhang, Min Liu*, Zhuo Zhou, Wei-ming Shen, 2013, *Quantum Ant Colony Algorithm-Based Emergency Evacuation Path Choice Algorithm*, Tongji University Shanghai 201804, China.
- [2] Habilitation, Lehprobe Zur, dan Hammer, Barbara , 2003, *Ant Colony Optimization* , Osnabruck University
- [3] Han K-H., Kim J-H, 2002, *Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization*, *IEEE Trans on Evolutionary Computation*

