

Perencanaan Jalur Pada *Mobile Robot* Dari Obyek Nyata Dan Dinamis Berbasis Algoritma Genetika

M. Widiyanto, Bima Sena Bayu D, A.R Anom Besari
Program Studi Teknik Komputer, Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember (ITS) Surabaya
Kampus ITS, Sukolilo, Surabaya 60111
newwidiyanto@gmail.com, bima@eepis-its.edu, anom@eepis-its.edu

Abstrak

Pada setiap kendaraan dibutuhkan sistem navigasi untuk dapat mengendalikan kendaraan sesuai dengan jalur yang diinginkan. Secara umum jalur yang diinginkan yaitu jalur yang aman, dan jarak tempuh yang singkat. Saat ini sistem navigasi, untuk memutuskan jalur yang ditempuh merupakan keputusan dari awak kendaraan. Pada penelitian ini, permasalahan diatas diaplikasikan pada mobile robot dengan penentu keputusan jalur yang ditempuh tidak lagi menggunakan awak (manusia) namun ditentukan secara autonom / otomatis dengan menggunakan metode algoritma genetika. Jalur yang dihasilkan oleh sistem merupakan jalur optimal yaitu jalur yang memiliki waktu tempuh yang singkat, aman, dan kecepatan algoritma dalam merencanakan jalur yang dibuat. Proses awal sistem yaitu mengetahui arena atau lingkungan yang digunakan, jumlah halangan, posisi halangan dan posisi finish / tujuan. Setelah mengetahui jumlah, posisi halangan dan posisi finish algoritma genetik akan membangkitkan jalur optimal yaitu jalur yang mempunyai jarak terpendek terhadap garis finish serta aman, tidak menabrak halangan yang berada pada lingkungan. Apabila terjadi perubahan posisi halangan yang terdeteksi melalui kamera, maka secara cepat algoritma genetik akan melakukan penyesuaian terhadap posisi halangan dan membangkitkan atau menciptakan jalur optimal yang baru secara cepat. Tingkat keberhasilan sistem perencanaan jalur terhadap lingkungan statis tanpa menghiraukan pergerakan halangan mencapai 98% dengan panjang kromosom 10 – 20 bit. Sedangkan untuk perencanaan jalur dengan halangan yang dinamis tingkat keberhasilan mencapai 70% dan 65% untuk masing panjang kromosom 10 dan 15 bit.

Kata kunci : Algoritma genetika, halangan, jalur terpendek, kamera.

1. Pendahuluan

Sistem navigasi terus dikembangkan untuk mendukung terciptanya sistem transportasi cerdas

(*Intellegent Transport System*) dan *Intellegent Autonomous System*. Hal ini dikarenakan sistem navigasi merupakan kontrol pemberi informasi pada sistem, tentang kondisi disekitar sistem dijalankan. Salah satu sistem navigasi saat ini yaitu GPS (*Global Positioning System*). GPS mampu memberikan informasi posisi serta mampu memberikan saran dalam menempuh jalur dalam sebuah perjalanan.

Salah satu penerapan sistem navigasi untuk mendukung pengembangan *Intellegent Autonomous System*, adalah mengimplementasikan sistem navigasi pada sebuah robot. Sistem navigasi memberikan informasi arah yang akan ditempuh oleh robot dari titik awal sampai ke titik akhir, dengan membentuk jalur yang terpendek antara titik awal dan akhir serta jalur yang dibentuk aman untuk dilintasi oleh robot.

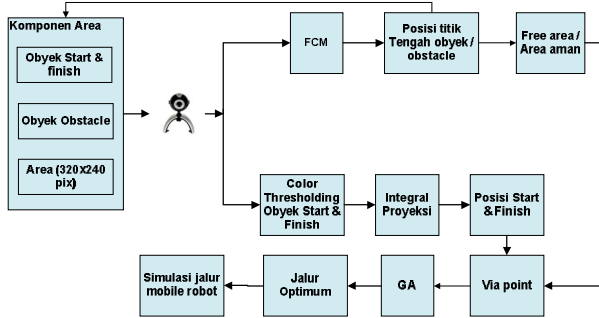
Ada banyak metode optimasi yang digunakan untuk menghasilkan jalur robot yang optimum. jalur yang optimum merupakan jalur yang akan ditempuh oleh robot adalah jalur aman, terpendek dan cepat dalam komputasi. Hacour[5] melakukan penelitian tentang perencanaan jalur pada lingkungan yang statis untuk mendapatkan jalur yang optimum dengan menggunakan *grid-map*. Giham dkk[2], Ismail dkk[4] melakukan perencanaan jalur *mobile robot* dengan menggunakan algoritma genetika untuk menentukan jalur optimal pada lingkungan yang statis, dengan metode ini akan didapatkan jalur yang optimum, terpendek dan mampu menghindari sejumlah halangan

Pada penelitian sebelumnya optimasi dilakukan hanya pada lingkungan yang statis. Oleh karena itu, pada penelitian ini akan dilakukan implementasi algoritma genetika untuk lingkungan yang dinamis. Lingkungan yang dinamis disederhanakan berupa posisi halangan yang berada diantara titik awal dan titik akhir yang posisinya dapat berubah sewaktu – waktu.

2. Perancangan Sistem

Secara keseluruhan penelitian ini terdiri dari 3 komponen utama yaitu identifikasi area, penciptaan jalur dan simulasi robot. Pada gambar 1 terlihat alur sistem secara keseluruhan.

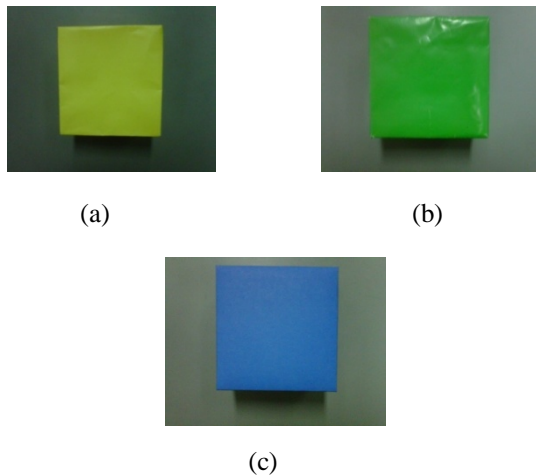
Identifikasi area digunakan untuk mengetahui kondisi area robot meliputi titik awal, titi akhir dan posisi halangan yang berada diantara titik awal dan akhir. Penciptaan jalur, memroses hasil identifikasi area untuk digunakan sebagai input sistem dari algoritma genetika (AG). Sedangkan simulasi, menyimulasikan hasil dari jalur yang telah didapat dari proses AG, dan sebagai penentu titik awal dalam proses penciptaan jalur secara dinamis.



Gambar 1. Perancangan sistem secara keseluruhan

2.1 Identifikasi Area

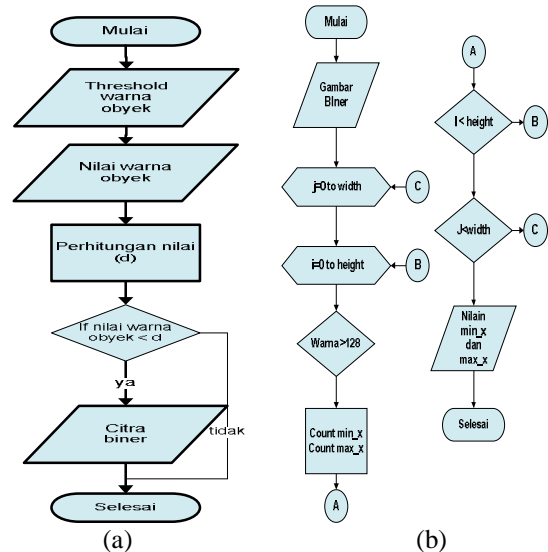
Pada tahapan ini proses identifikasi terhadap komponen – komponen area yang terdiri dari obyek *start*, *finish* dan halangan seperti terlihat pada gambar 2. Untuk komponen *start* dan *finish* proses awal dengan menggunakan *distance thresholding* untuk mengubah menjadi gambar biner. Kemudian untuk mendapatkan posisi *start* dan *finish* digunakan integral proyeksi pada masing obyek. Pada gambar 3 proses *thresholding* dan integral proyeksi.



Gambar 2. (a) Obyek Start, (b) obyek finish, (c) halangan

Input dari proses *thresholding* adalah gambar dari kamera, kemudian dengan menggunakan metode *distance threshold* dihitung jarak antara acuan dengan obyek *real*. Apabila kurang dari acuan maka warna piksel akan dirubah menjadi putih. Sebaliknya akan dirubah menjadi hitam. Pada persamaan 1 untuk menghitung jarak antara warna acuan dan warna *real*.

$$d = \sqrt{(R - r)^2 + (G - g)^2 + (B - b)^2} \tag{1}$$



Gambar 3. Proses *thresholding*(a) dan integral proyeksi(b)

Untuk komponen area lainnya yaitu halangan, digunakan metode *Fuzzy C-Means* untuk mendapatkan posisi tengah obyek. Berikut proses FCM untuk mendapatkan posisi tengah obyek dengan input berupa gambar biner dari obyek.

1. Membangkitkan data posisi x dan y

$$p(x,0) = I(x) \tag{2}$$

$$p(x,1) = I(y) \tag{3}$$
2. Mengacak harga derajat keanggotaan pusat cluster terhadap setiap data

$$m(i, j) = \text{random}() \tag{4}$$
3. Menormalisasi harga derajat keanggotaan pusat cluster terhadap setiap data.

$$m(i, j) = \frac{m(i, j)}{\sum_{j=0}^{j=1} m(j, i)} \tag{4}$$

4. Mengupdate harga center x

$$pembilang(i) = \sum_{j=0}^{jumtitik} m(i, j)^2 \cdot p(j,0) \quad (5)$$

$$penyebut(i) = \sum_{j=0}^{jumtitik} m(i, j)^2 \quad (6)$$

$$cluster_x(i,0) = \frac{pembilang(i)}{penyebut(i)} \quad (7)$$

5. Mengupdate harga center y

$$pembilang(i) = \sum_{j=0}^{jumtitik} m(i, j)^2 \cdot p(j,1) \quad (8)$$

$$penyebut(i) = \sum_{j=0}^{jumtitik} m(i, j)^2 \quad (9)$$

$$cluster_x(i,1) = \frac{pembilang(i)}{penyebut(i)} \quad (10)$$

6. Update titik tengah cluster

- a. Menghitung jarak cluster center terhadap data

$$temp_x = (cluster_x(i,0) - p(j,0))^2 \quad (11)$$

$$temp_y = (cluster_x(i,1) - p(j,1))^2 \quad (12)$$

$$d(i, j) = \sqrt{temp_x + temp_y} \quad (12)$$

- b. Menghitung total jarak semua cluster terhadap semua data

$$zd(i, j) = \sum_{k=0}^{jumlobyek} \frac{d(i, j)}{d(k, j)} \quad (13)$$

- c. Menghitung harga m untuk setiap cluster terhadap data input

$$m(i, j) = \frac{1}{zd(i, j)} \quad (14)$$

Keterangan :

- i = jumlah obyek
j = jumlah titik yang menyusun obyek
m(i,j) = derajat keanggotaan pusat cluster
d(i,j) = jarak pusat cluster terhadap data
zd(i,j) = total jarak cluster terhadap data
Cluster_x(i,0) = titik tengah x cluster ke-i.
Cluster_x(i,1) = titik tengah y cluster ke-i.

2.2 Penciptaan Jalur

Proses penciptaan jalur terdiri dua tahapan yaitu penciptaan titik – titik bantuan (*via point*) dan penciptaan jalur dengan menggunakan algoritma genetika.

2.2.1 Via Point

Via point merupakan titik yang dihasilkan secara acak untuk membantu menghasilkan jalur yang optimum.

Pembangkitan titik *via point* dengan persamaan 15 – 16. Hasil dari pembangkitan titik *via point* seperti pada gambar 3.

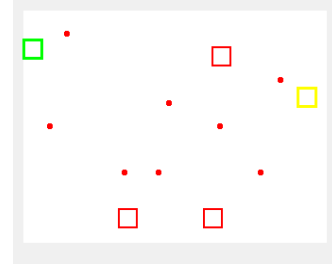
$$\Delta d = \frac{(Xb - Xa)}{N - 2} \quad (15)$$

$$posX = posX_0 + (i - 1) * rand + \Delta d * rand \quad (16)$$

$$posY = P * rand$$

Dimana :

- Δd = delta are atau range area
posX = posisi titik X
posiY = posisi titik Y
posX₀ = posisi start
i = jumlah titik (iterasi)
rand = nilai random
P = ukuran window untuk sumbu Y



Gambar 4. Hasil pembangkitan *via point*

2.2.2 Penciptaan Jalur Optimum

Pembentukan jalur yang optimum dengan menggunakan AG dengan berdasarkan input berupa titik bantu yang telah menempati pada daerah aman atau *free area*. Titik – titik bantu tersebut akan diseleksi untuk mendapatkan nilai jarak (dalam piksel) yang terdekat dan aman. Sebagai kontrol adalah fungsi fitness dari algoritma genetika seperti pada persamaan 17 - 19

$$Fs = \frac{1}{d * H * T} \quad (17)$$

$$\|d\| = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2} \quad (18)$$

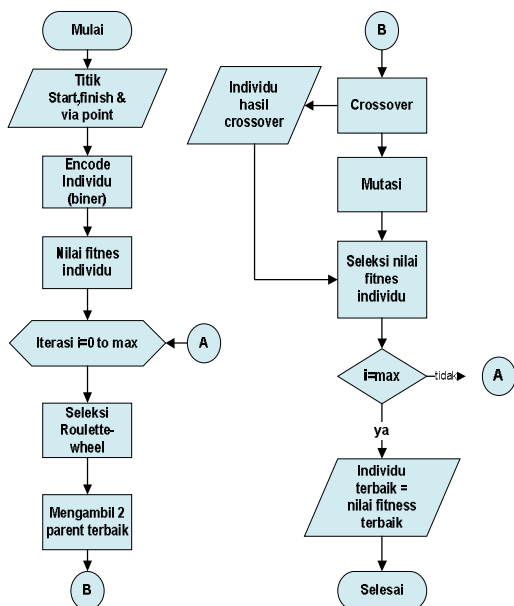
$$H = 10^{jum_tumbukan} \quad (19)$$

Keterangan:

- Fs = Fungsi fitness
d = jarak antar point
T = jumlah gen eksis
(X_A, Y_A) = Posisi kordinat titik via point A
(X_B, Y_B) = Posisi kordinat titik via point B

Dari perhitungan fungsi fitness, akan dilanjutkan proses algoritma genetika selanjutnya, yaitu seleksi *parent*, pindah silang dan mutasi. Seleksi *parent* yang

digunakan menggunakan metode seleksi *mesin roulette*. Sedangkan pindah silang menggunakan pindah silang *multi point* dengan menggunakan probabilitas pindah silang sebesar 0.6[12]. Kemudian dilakukan mutasi dengan kontrol berupa nilai probabilitas mutasi yang digunakan adalah $1/n$, dengan n adalah panjang bit[3][12][11]. Mutasi yang dilakukan dengan *single bit mutation*. Hasil keturunan dari mutasi dibandingkan dengan nilai fitness individu pada populasi dengan nilai fitness terendah. Jika nilai fitness terendah dalam populasi lebih kecil dari *offspring* maka dilakukan penggantian individu. Jika nilai fungsi fitness keturunan lebih kecil, maka keturunan dapat dihiraukan. Pada gambar 5 proses algoritma genetika secara lengkap.

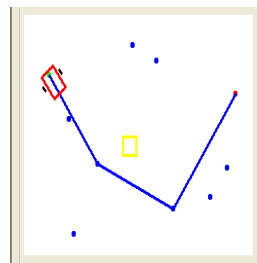


Gambar 5. Proses algoritma genetika

2.3 Simulasi

Simulasi menggunakan fasilitas grafik dari C++ 6.0. Simulasi digunakan untuk menggambarkan pergerakan dari *mobile robot* dalam mengikuti jalur yang telah dihasilkan.

Dalam simulasi sistem secara keseluruhan, pergerakan robot dalam simulasi mempengaruhi proses penciptaan jalur selanjutnya. Penciptaan jalur akan dilakukan secara kontinue jika terjadi pergerakan terhadap halangan. Pergerakan halangan akan menghentikan simulasi dan posisi terakhir robot akan dijadikan sebagai acuan titik awal penciptaan jalur yang selanjutnya. Pada gambar 6 hasil dari proses simulasi.



Gambar 6. Hasil proses simulasi

3. Uji Coba Dan Analisa

Pengujian dilakukan pada sistem identifikasi komponen area dan penciptaan jalur optimum.

3.1 Pengujian Komponen Area

3.1.1 Pengujian Identifikasi Start Dan Finish

Pengujian komponen area yang pertama yaitu pengujian nilai *distance thresholding* dengan posisi yang berbeda didalam area.

Tabel 1. Hasil pengujian nilai *distance threshold*

| No | Nilai <i>distance</i> | | deteksi Obyek lain | Waktu (ms) |
|----|-----------------------|--------|--------------------|------------|
| | Start | Finish | | |
| 1 | 41 | 51 | x | 2.59 |
| 2 | 41 | 51 | x | 2.89 |
| 3 | 41 | 51 | x | 2.7 |
| 4 | 41 | 51 | x | 2.16 |
| 5 | 37 | 51 | x | 2.73 |
| 6 | 46 | 51 | x | 2.26 |
| 7 | 35 | 51 | x | 2.72 |
| 8 | 46 | 51 | x | 2.72 |
| 9 | 35 | 51 | x | 2.84 |
| 10 | 35 | 51 | x | 2.72 |

Dari hasil pengujian untuk obyek *start* dan *finish* terlihat pada tabel 1. Penggunaan *distance thresholding* untuk melakukan *thresholding* warna sangat baik. Hal ini terlihat bahwa tidak terdapat pendeteksian obyek lain selama perubahan posisi. Hal ini dikarenakan *distance threshold* menggunakan nilai rentang antara set point dengan jarak *threshold*. Rata – rata nilai jarak yang digunakan untuk untuk obyek *start* sebesar 40 sedangkan untuk obyek *finish* sebesar 51.

3.1.2 Pengujian Identifikasi Halangan

Sedangkan untuk pengujian algoritma FCM dalam mendeteksi titik tengah obyek, pengujian dilakukan dengan menggunakan obyek mulai dari satu obyek sampai dengan 10 obyek. Tabel 2 memperlihatkan hasil pengujian FCM

Tabel 2. Hasil pengujian FCM

| Jum. Halangan | Iterasi | | | | | | |
|---------------|---------|----|----|----|----|----|----|
| | 5 | 10 | 15 | 20 | 30 | 40 | 50 |
| 1 | √ | √ | √ | √ | √ | √ | √ |
| 2 | x | √ | √ | √ | √ | √ | √ |
| 3 | x | √ | √ | √ | √ | √ | √ |
| 4 | x | √ | √ | √ | √ | √ | √ |
| 5 | x | x | √ | √ | √ | √ | √ |
| 6 | x | x | x | √ | √ | √ | √ |
| 7 | x | x | x | x | √ | √ | √ |
| 8 | x | x | x | x | x | x | x |
| 9 | x | x | x | x | x | x | x |
| 10 | x | x | x | x | x | x | x |

Dari hasil pengujian metode FCM, FCM hanya mampu mengidentifikasi posisi obyek sebanyak satu obyek sampai tujuh obyek. Dari tabel 2 didapatkan hasil bahwa semakin banyak obyek yang dideteksi, dibutuhkan iterasi yang lebih banyak. Hal ini dikarenakan semakin banyak persebaran data dibutuhkan proses iterasi yang lebih banyak juga untuk mencapai konvergen. Sedangkan ketidak berhasilan FCM untuk mendeteksi pusat titik data obyek, dapat dikarenakan variabel penampung data yang terbatas, ketidak mampuan ini yang menyebabkan pendeteksian tidak mendapatkan pusat dari seluruh obyek,

3.2 Pengujian Penciptaan Jalur

Pengujian penciptaan jalur optimum dilakukan pada dua bagian yaitu *via point* dan AG. Pengujian AG dilakukan secara dua tahap yaitu pengujian dilakukan secara *offline* dan *online*.

3.2.1 Pengujian Via Point

Titik bantu diuji berdasarkan jumlah titik yang dibangkitkan untuk menghindari posisi halangan. Posisi halangan dirubah – dirubah sebanyak sepuluh kali untuk mengetahui respon titik bantu terhadap perubahan obyek penghalang. Obyek penghalang yang diujikan mulai satu sampai dengan delapan obyek penghalang. Hasil pengujian pada tabel 3.

Hasil pengujian *via point* sangat penting untuk memastikan jalur yang dihasilkan aman, tidak menumbuk halangan. Dari tabel 4 didapatkan semakin banyak halangan dan titik bantu yang digunakan, maka semakin kecil kemungkinan titik bantu untuk menghindari halangan. Hal ini dapat dikarenakan karena area aman semakin kecil dan kontrol terhadap kondisi kepastian area aman sangat terbatas, sehingga masih dimungkinkan adanya titik bantu yang tidak berada diarea aman.

Tabel 3. Hasil pengujian titik bantu (*via point*)

| Jum. Hal. | Jumlah titik bantu | | | | | | |
|-----------|--------------------|----|----|----|----|----|----|
| | 3 | 5 | 10 | 15 | 20 | 25 | 30 |
| 0 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| 1 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| 2 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| 3 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| 4 | 10 | 10 | 10 | 10 | 8 | 9 | 9 |
| 5 | 10 | 10 | 10 | 10 | 9 | 9 | 9 |
| 6 | 10 | 10 | 10 | 10 | 8 | 7 | 8 |
| 7 | 9 | 8 | 8 | 8 | 7 | 6 | 6 |
| 8 | 8 | 6 | 5 | 4 | 4 | 3 | 2 |

3.2.2 Pengujian AG Secara *Offline*

Pengujian algoritma genetika secara *offline* dilihat pengaruh dari jumlah generasi, populasi dan panjang bit kromosomnya. Pengujian parameter tersebut diuji dengan jumlah halangan mulai dari 1 – 8 dan satu kali tanpa halangan. Hasil pengujian terlihat pada tabel 4-6 merupakan jumlah keberhasilan dari 9 kali uji coba.

Tabel 4 hasil pengujian untuk generasi 10

| populasi | Panjang bit kromosom | | | | | | |
|----------|----------------------|----|----|----|----|----|----|
| | 3 | 5 | 10 | 15 | 20 | 25 | 30 |
| 10 | 6 | 8 | 8 | 9 | 9 | 7 | 7 |
| 20 | 5 | 8 | 8 | 8 | 9 | 8 | 7 |
| populasi | Pankang bit kromosom | | | | | | |
| 3 | 5 | 10 | 15 | 20 | 25 | 30 | |
| 30 | 6 | 8 | 9 | 7 | 9 | 7 | 6 |
| 40 | 5 | 9 | 9 | 9 | 9 | 5 | 5 |

Tabel 5 hasil pengujian untuk generasi 40

| populasi | Panjang bit kromosom | | | | | | |
|----------|----------------------|---|----|----|----|----|----|
| | 3 | 5 | 10 | 15 | 20 | 25 | 30 |
| 10 | 6 | 8 | 9 | 9 | 7 | 6 | 7 |
| 20 | 6 | 7 | 9 | 8 | 7 | 7 | 5 |
| 30 | 5 | 7 | 8 | 8 | 7 | 3 | 4 |
| 40 | 4 | 7 | 8 | 9 | 8 | 5 | 5 |

Tabel 6 hasil pengujian untuk generasi 70

| populasi | Panjang bit kromosom | | | | | | |
|----------|----------------------|---|----|----|----|----|----|
| | 3 | 5 | 10 | 15 | 20 | 25 | 30 |
| 10 | 5 | 7 | 8 | 9 | 9 | 5 | 5 |
| 20 | 5 | 6 | 8 | 9 | 8 | 6 | 5 |
| 30 | 5 | 6 | 9 | 9 | 9 | 5 | 4 |
| 40 | 5 | 7 | 8 | 6 | 7 | 6 | 5 |

Dari hasil pengujian yang dilakukan metode algoritma genetika dalam menghasilkan jalur optimum berdasarkan tabel 4 – 6 pengujian diatas didapatkan panjang bit kromosom yang cukup baik dalam menghasilkan jalur optimum. panjang bit kromosom antara 10 – 20 bit. Hasil ini didapatkan tingkat keberhasilan antara 90% hingga 98% untuk penerapan dalam keadaan *offline*.

3.2.3 Pengujian AG Secara Online

Pengujian penciptaan jalur juga dilakukan secara *online/* dinamis, dimana perubahan posisi halangan akan memberikan pengaruh pada posisi terakhir robot dan *generate* titik bantu serta proses AG yang baru untuk menghasilkan individu baru yang sesuai dengan kondisi area setelah perubahan posisi halangan. Jumlah titik bantu yang dibangkitkan akan berkurang sesuai dengan posisi terakhir dari robot. Posisi titik bantu yang dihilangkan yaitu kurang dari posisi terakhir robot. Prosedur pengujian, jalur yang sudah terbentuk akan disimulasikan terhadap robot, kemudian posisi halangan akan dirubah sebanyak dua kali untuk setiap data pengujian. Data yang diambil berupa kriteria jalur optimum. Hasil pengujian pada tabel 7.- 9. Angka pada tabel merupakan jumlah keberhasilan dari setiap pengujian yang dilakukan.

Tabel 7 hasil pengujian untuk generasi 10

| populasi | Panjang bit kromosom | | |
|----------|----------------------|----|----|
| | 10 | 15 | 20 |
| 10 | 7 | 7 | 5 |
| 20 | 7 | 8 | 7 |
| 30 | 7 | 7 | 7 |
| 40 | 5 | 7 | 6 |

Tabel 8 hasil pengujian untuk generasi 40

| populasi | Panjang bit kromosom | | |
|----------|----------------------|----|----|
| | 10 | 15 | 20 |
| 10 | 7 | 6 | 6 |
| 20 | 5 | 7 | 7 |
| 30 | 4 | 6 | 5 |
| 40 | 5 | 6 | 6 |

Tabel 9 hasil pengujian untuk generasi 70

| populasi | Panjang bit kromosom | | |
|----------|----------------------|----|----|
| | 10 | 15 | 20 |
| 10 | 5 | 6 | 6 |
| 20 | 6 | 6 | 7 |
| 30 | 5 | 5 | 4 |
| 40 | 5 | 6 | 4 |

Dari hasil pengujian AG secara *online/dinamis*, individu dengan panjang bit 10 dan 15 memiliki tingkat keberhasilan yang besar untuk penerapan perencanaan jalur optimum dengan halangan yang dinamis. Seperti halnya pada kasus penerapan secara *offline/statis* jumlah titik bantu 10 dan 15 cukup efektif karena tidak terlalu memenuhi area dan mampu menghindari diluar halangan yang ada karena kisaran nilai posisi yang cukup lebar antara titik bantu. Tingkat keberhasilan dari seluruh pengujian ada pada kisaran 70.8 % - 74 %.

4. Kesimpulan

Dari hasil pengujian sistem secara keseluruhan dapat diambil kesimpulan sebagai berikut :

1. *Thresholding* dengan menggunakan metode *distance threshold* cukup baik digunakan dalam sistem ini. Hal ini dibuktikan dengan merubah posisi obyek, *distance thresholding* masih mampu mendeteksi obyek dengan hanya merubah satu parameter, yaitu nilai *distance*. Jika dibandingkan dengan *thresholding* RGB yang harus merubah tiga parameter jika terjadi perubahan.
2. FCM hanya mampu untuk mendeteksi obyek pada jumlah 1 sampai dengan 8.
3. Semakin banyak obyek untuk dideteksi maka FCM membutuhkan banyak iterasi pula. Dikarenakan semakin banyanya jumlah sebaran data.
4. Pembangkitan titik bantu pada area bebas dibutuhkan nilai toleransi untuk visualisasi dari titik bantu dan halangan.
5. Titik bantu yang dibangkitkan baik pada kisaran 10 – 20 titik bantu
6. Pada sistem *offline* tingkat keberhasilan algoritma genetika dalam menghasilkan jalur optimum pada kromosom dengan 10 – 20 bit dengan tingkat keberhasilan mencapai 98%.
7. Implementasi algoritma genetika secara online hasil yang maksimal pada individu dengan panjang bit 10 dan 15 dengan tingkat keberhasilan sebesar 70.8% dan 74%.

Referensi

- [1] Bima Sena Bayu Dewantara, “Perencanaan Jalur Mobile Secara Nyata Pada Lingkungan Dinamis Berbasis Compact Genetic Algorithm”. Tesis Program Magister Teknik Elektro-Institut Teknologi Sepuluh Nopember. Surabaya. 2010.
- [2] Gihan Nagib and W.Gharieb,c, “Path Planning For A Mobile Robot Using Genetic Algorithm”, Electrical Engineering Dept., Faculty of Engineering Cairo University-Fayoum Branch.
- [3] Suyanto, ST, Msc, “Artificial Intelligence Searching Reasoning Planning and Learning”. Informatika. Bandung. 2007
- [4] Ismail AL-Taharwa, Alaa Sheta and Mohammed Al-Weshah, “A Mobile Robot Path Planning Using

- Genetic Algorithm in Static Environment”, Journal of Computer Science, Department of Information Technology, Faculty of Science and Information Technology, Al-Balqa Applied University. Al-Salt. Jordan.2008.
- [5] O.Hachour,” Path planning of Autonomous Mobile robot”, INTERNATIONAL JOURNAL OF SYSTEMS APPLICATIONS, ENGINEERING & DEVELOPMENT.2008.
- [6] Torvald Ersson, Xiaoming Hu,“Path Planning and Navigation of Mobile Robots in Unknown Environments”. Optimization and Systems Theory / Centre for Autonomous Systems Royal Institute of Technology.
- [7] Supriadi Ahmad, “ Perencanaan Jalur untuk Menghindari Tabrakan Pada Robot POEMAV”,Departemen Teknik Elektro - Institute Teknologi Bandung.
- [8] <http://www.csharp-indonesia.com/2011/01/program-fuzzy-c-mean-clustering-di-c-c.html>
- [9] <http://www.megenep.com/article/785/mengenal-fuzzy-c-means--part-1--sebuah-pengenalan-.htm>
- [10] Joni I Made dan Raharjo Budi,” Pemrograman C dan Implementasinya Edisi Kedua”.Informatika.Bandung.2006
- [11] www.eepis-its.edu/lecture/~basuki/lecture.html.
“AlgoritmaGenetika”
- [12] Modul Ajar: Basuki Achmad,” Strategi Menggunakan Algoritma Genetika ”.PENS-ITS.Surabaya.2003
- [13] MunakataToshinori.”Fundamentals of The New Artificial Intelligence neural,Evolution, Fuzzy ang More”.Springer.London.2008
- [14] Randy L Haupt and Sue Ellen Haupt.”Practical Genetic Algorithms”. Wiley – Interscience.New Jersey.2004
- [15] Basuki Ahmad,Ramadijanti Nana. Modul Ajar “Aplikasi Pengolahan Citra Deteksi warna(pertemuan_11.ppt)”.PENS-ITS.2009