

Implementasi *Particle Swarm Optimization* untuk Optimasi *Fuzzy-Social Force Model* pada Sistem Navigasi Robot *Omnidirectional***Anugerah Wibisana¹, Bima Sena Bayu Dewantara², Dadet Pramadihanto³**¹mail.wibisana@gmail.com, ²bima@pens.ac.id, ³dadet@pens.ac.id^{1,2,3} Politeknik Elektronika Negeri Surabaya**Informasi Artikel**

Diterima : 7 Ags 2022
Direview : 10 Ags 2022
Disetujui : 30 Ags 2022

Kata Kunci

Particle Swarm Optimization, Fuzzy-Social Force Model, Navigasi, Mobile Robot.

Abstrak

Particle Swarm Optimization (PSO) adalah salah satu metode optimasi berbasis *swarm* yang implementasinya mudah dan hanya membutuhkan sedikit parameter yang diatur. Penelitian ini bertujuan untuk mengimplementasikan PSO untuk mengoptimasi *Fuzzy-Social Force Model* (FSFM). FSFM merupakan kombinasi *Social Force Model* (SFM) sebagai algoritma navigasi dan *Fuzzy Inference Rule* (FIS) untuk menghasilkan *gain* yang adaptif pada SFM sehingga menghasilkan navigasi *mobile robot* yang lebih responsif terhadap halangan. Implementasi PSO untuk mengoptimasi aturan *fuzzy* menjadi lebih optimal ketika *mobile robot* bernavigasi di ruang sosial. Dari hasil uji eksperimental pada *software* simulasi VREP, parameter *cognitive* $c1 = 1$ dan parameter *social* $c2 = 2$ menghasilkan performa navigasi paling baik dibandingkan nilai parameter uji lainnya.

Keywords

Particle Swarm Optimization, Fuzzy-Social Force Model, Navigation, Mobile Robot.

Abstrak

Particle Swarm Optimization (PSO) is a swarm-based optimization method that is easy to implement and requires only a few parameters to set. This study aims to implement PSO to optimize the *Fuzzy-Social Force Model* (FSFM). FSFM combines the *Social Force Model* (SFM) as a navigation algorithm and the *Fuzzy Inference Rule* (FIS) to produce adaptive gain on SFM to create a mobile robot navigation system that is more responsive to obstacles. The PSO implementation optimizes fuzzy rules to be more optimal when the mobile robot navigates into social spaces. From the experimental test results on the VREP simulation software, cognitive parameter $c1 = 1$ and social parameter $c2 = 2$ produced the best navigation performance compared to other test parameter values.

A. Pendahuluan

Sistem navigasi merupakan hal yang krusial pada sebuah *mobile* robot. Selain diharuskan mencapai titik tuju dengan cepat, *mobile* robot juga harus dilengkapi algoritma menghindari jika akan menabrak. Ketika *mobile* robot menabrak sebuah objek, bukan hanya kerusakan dari sisi *mobile* robot saja yang terjadi, namun juga dari sisi objek yang ditabrak, entah itu benda mati ataupun manusia. Oleh karena itu, *mobile* robot harus dilengkapi algoritma menghindari dari benda mati ataupun manusia dalam bernavigasi.

Algoritma *Social Force Model* (SFM) adalah salah satu algoritma navigasi yang tidak hanya memprioritaskan *mobile* robot untuk mencapai titik tuju, namun juga mempertimbangkan segala halangan yang ada di ruang sosial pada lintasan navigasi robot. SFM memisahkan halangan menjadi halangan statis dan halangan dinamis. Halangan statis bersifat benda mati seperti lemari, tembok, pot bunga, meja bangku dan masih banyak lagi. Sedangkan halangan dinamis bersifat benda hidup yaitu manusia.

Beberapa penelitian lain yang mengembangkan pengaplikasian SFM seperti pada navigasi robot pemandu [1], navigasi robot *tour-guide* [2], robot pengikut arah jalan manusia [3], robot *drone* [4], navigasi *mobile* robot pada area yang sangat ramai [5][6], navigasi robot pada lingkungan yang tidak diketahui [7], navigasi robot yang mampu mendekat dan berinteraksi dengan manusia [8], navigasi *service* robot [9] dan kombinasi SFM dan MPC (*Model Predictive Control*) pada *autonomous vehicle* untuk melintasi lintasan ramai pejalan kaki [10].

Algoritma *Social Force Model* (SFM) sebagai algoritma navigasi yang dikembangkan oleh Dewantara, dkk [1] mampu mencapai titik tujuan dengan tetap mempertimbangkan untuk menghindari objek disekitarnya seperti benda mati ataupun manusia. Tetapi parameter SFM yang bervariasi dan pengaturan aksi-kondisi yang cukup banyak dan beragam pada *Reinforcement Learning* membuat penerapan SFM menjadi sulit.

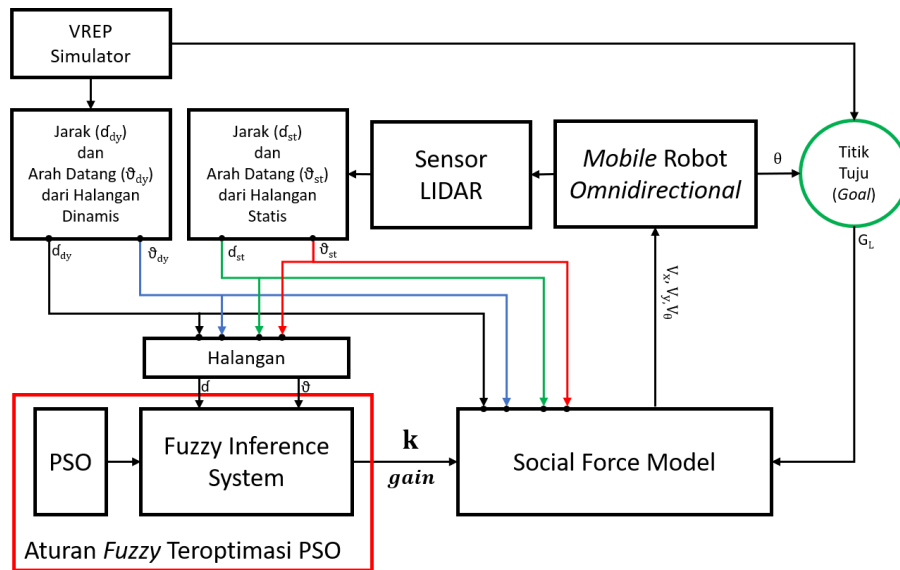
Untuk mengatasi masalah tersebut, beberapa penelitian mengembangkan sistem navigasi SFM menjadi *Fuzzy-Social Force Model* (FSFM) dengan mengkombinasikan SFM dengan *Fuzzy Inference System* (FIS) seperti pada sistem navigasi robot *healthcare* [11] agar mampu berjalan di koridor dan robot beroda pemain bola [12] guna mencapai gawang musuh tanpa bertabrakan dengan robot lainnya diatas lapangan. Namun penelitian-penelitian tersebut masih perlu mendefinisikan nilai yang tepat pada aturan *fuzzy* agar *output* dari *fuzzy* yaitu *gain* sebagai *input* untuk *Social Force Model* sesuai dengan respon robot untuk menghindari halangan.

Penelitian ini bertujuan untuk mengimplementasikan *Particle Swarm Optimization* (PSO) untuk optimasi *Fuzzy-Social Force Model* pada sistem navigasi *mobile* robot yang akan diterapkan pada robot *Omnidirectional*. Fungsi PSO disini sebagai algoritma optimasi adalah untuk mengoptimasi nilai pada aturan *fuzzy* pada *Fuzzy-Social Force Model* sehingga didapat nilai dari aturan *fuzzy* yang optimal. Proses optimasi sendiri akan dilakukan di *software* simulasi realistik 3D VREP.

Penelitian ini disusun secara berurutan dengan Bagian B berisi Metode Penelitian yang diajukan, Bagian C berisi Hasil dan Pembahasan metode yang telah diuji dan Bagian D berisi Simpulan dari penelitian ini.

B. Metode Penelitian

Pada penelitian ini, blok diagram sistem dapat dilihat di Gambar 1.

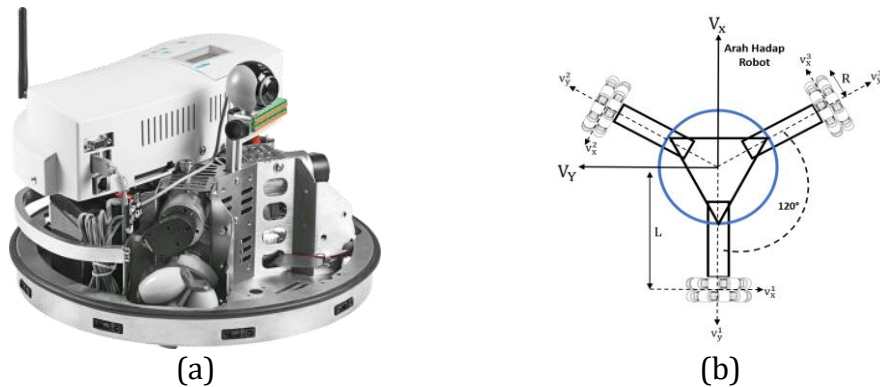


Gambar 1. Blok Diagram Sistem

Pada blok diagram sistem di Gambar 1, penelitian ini mengerjakan blok *mobile robot Omnidirectional*, *Fuzzy-Social Force Model* dan algoritma optimasi *Particle Swarm Optimization*. Penelitian ini dilakukan dan akan diuji coba pada software simulasi realistik 3D VREP. Sensor LIDAR akan digunakan untuk mendapatkan data jarak dan arah datang halangan statis. Sedangkan informasi mengenai data jarak dan arah datang halangan dinamis dalam hal ini manusia didapatkan dari data yang diberikan dari VREP. Posisi dari titik tuju atau *goal* juga akan diberikan dari VREP. Data jarak dan arah datang halangan baik statis maupun dinamis akan menjadi data masukan yang dibutuhkan dalam proses *Fuzzy Inference System* (FIS) dan *Social Force Model*. Sedangkan data posisi titik tuju akan menjadi data masukan untuk proses *Social Force Model*.

1. *Mobile Robot Omnidirectional*

Pada penelitian ini menggunakan *mobile robot* yang digunakan adalah sistem robot *Omnidirectional* dengan konfigurasi tiga roda *omni*. Model robot yang digunakan adalah robot Robotino dari Festo[13]. Model robot dan konfigurasi dari robot *Omnidirectional* dapat dilihat pada Gambar 2. Untuk mengatur kecepatan robot dalam sumbu x dan y serta pergerakan kecepatan *angular* arah hadap robot menggunakan Persamaan 1-3.



Gambar 2. Robot *Omnidirectional* (a) Robot Robotino (b) Konfigurasi dari Robot *Omnidirectional*

$$V_x = \frac{R(v_x^3 - v_x^2)}{\sqrt{3}} \quad (1)$$

$$V_y = \frac{R(-v_x^1 - 2v_x^2 + 2v_x^3)}{3} \quad (2)$$

$$V_\theta = \frac{R(v_x^1 + v_x^2 + v_x^3)}{3L} \quad (3)$$

Dimana V_x , V_y dan V_θ adalah kecepatan gerak robot dalam urutan sumbu x, sumbu y dan kecepatan *angular* arah hadap robot. R adalah radius roda *omni* dan L adalah jarak tiap roda *omni* ke titik tengah pusat *mobile* robot. Sedangkan v_x^1 , v_x^2 dan v_x^3 adalah kecepatan masing-masing roda dalam sumbu x.

2. Fuzzy-Social Force Model (FSFM)

Social Force Model (FSFM) adalah algoritma navigasi pada mobile robot yang menggabungkan mempertimbangkan halangan statis maupun dinamis dalam proses navigasi robot menuju titik tujuan. *Social Force Model* (SFM) yang diusulkan oleh Helbing, dkk [14] dengan mendefinisikan gerak jalan manusia di dalam ruang sosial menuju suatu tempat yang ingin dituju menjadi :

1. Manusia sebagai *agent* akan cenderung terus menuju ke titik yang ingin dituju. Hal ini di definisikan SFM sebagai gaya tarik terhadap titik tuju. Persamaan untuk menghitung gaya tarik ini di Persamaan 4.
2. Manusia sebagai *agent* cenderung menghindari ketika dirinya terlalu dekat dengan halangan yang bersifat solid seperti tembok, lemari ataupun benda mati. Hal ini di definisikan SFM sebagai gaya tolak terhadap halangan statik. Persamaan untuk menghitung gaya tolak ini di Persamaan 5 – 7.
3. Manusia sebagai *agent* akan menghindari bertabrakan dengan manusia lain yang tidak dikenalnya, dan akan mencoba untuk menghindari di jarak ideal dalam artian tidak terlalu dekat ataupun tidak terlalu jauh. Hal ini di definisikan SFM sebagai gaya tolak terhadap halangan dinamik. Manusia lain dianggap halangan dinamik karena tidak seperti halangan statik yang diam, manusia mempunyai kapabilitas untuk berjalan dan bergerak dalam

sebuah lintasan gerak terencana ataupun tidak terencana. Persamaan untuk menghitung gaya tolak ini di Persamaan 8 – 10.

$$F_{goal} = m \frac{v^0 - v^t}{\tau} \quad (4)$$

Dimana F_{goal} adalah gaya tarik terhadap titik tuju, m adalah massa *mobile* robot (kg), v^0 adalah kecepatan maksimal *mobile* robot (m/s) dan v^t adalah kecepatan aktual *mobile* robot (m/s). Simbol τ merepresentasikan waktu sampling (s).

$$F_{st} = F_{sts} + F_{stp} \quad (5)$$

$$F_{sts} = k_{st} \cdot \exp\left(\frac{r_{Rst} - d_{st}}{\zeta_{st}}\right) \cdot e_{st} \quad (6)$$

$$F_{stp} = k_{st} \cdot \exp(r_{Rst} - d_{st}) \cdot e_{st} \quad (7)$$

Dimana F_{st} adalah gaya tolak terhadap halangan statis, F_{sts} adalah gaya tolak bersifat sosial terhadap halangan statis dan F_{stp} gaya tolak bersifat fisik terhadap halangan statis. k_{st} adalah besaran nilai tambah atau *gain* untuk menghindari halangan statis, r_{Rst} adalah nilai total hasil penambahan dari radius *mobile* robot dengan radius halangan statis ketika keduanya saling bertabrakan di ruang sosial, sedangkan d_{st} adalah jarak halangan statis terdekat terhadap *mobile* robot. ζ_{st} merepresentasikan jarak efektif gaya tolak untuk menghindari halangan statis terhadap *mobile* robot.

$$F_{dy} = F_{dys} + F_{dyp} \quad (8)$$

$$F_{dys} = k_{dy} \cdot \exp\left(\frac{r_{Rdy} - d_{dy}}{\zeta_{dy}}\right) \cdot e_{dy} \quad (9)$$

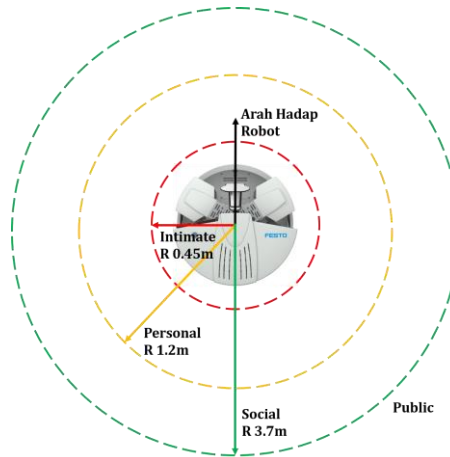
$$F_{dyp} = k_{dy} \cdot \exp(r_{Rdy} - d_{dy}) \cdot e_{dy} \quad (10)$$

Dimana F_{dy} adalah gaya tolak terhadap halangan dinamis, F_{dys} adalah gaya tolak bersifat sosial terhadap halangan dinamis dan F_{dyp} gaya tolak bersifat fisik terhadap halangan dinamis. k_{dy} adalah besaran nilai tambah atau *gain* untuk menghindari halangan dinamis, r_{Rdy} adalah nilai total hasil penambahan dari radius *mobile* robot dengan radius halangan dinamis ketika keduanya saling bertabrakan di ruang sosial, sedangkan d_{dy} adalah jarak halangan dinamis terdekat terhadap *mobile* robot. ζ_{dy} merepresentasikan nilai jarak efektif gaya tolak untuk menghindari halangan dinamis dan e_{dy} adalah arah vektor dari halangan dinamis terhadap *mobile* robot.

Gaya tarik terhadap titik tuju dan gaya tolak terhadap halangan baik statis maupun dinamis akan menghasilkan gaya navigasi dari SFM seperti pada Persamaan 11.

$$F_{nav} = F_{goal} + F_{st} + F_{dy} \quad (11)$$

Pada perhitungan gaya tolak baik untuk halangan statis maupun halangan dinamis di Persamaan 6, 7, 9 dan 10 terdapat nilai radius *mobile* robot dan radius halangan statis dan dinamis. Nilai radius ini menggunakan nilai *proxemics distance* yang diusulkan oleh Hall [15]. Representasi area pada *proxemics distance* dapat dilihat pada Gambar 3.



Gambar 3. Pembagian Area pada *Proxemics Distance*

Dalam bernavigasi menuju titik tuju, *mobile* robot harus mampu menerjemahkan gaya tarik-tolak yang telah di dapat ketika bernavigasi di ruang sosial menjadi bentuk perintah arah gerak dalam sumbu x, sumbu y dan arah hadap robot. Untuk menghitung perintah tersebut dapat dilihat pada Persamaan 12-15.

$$\alpha = \frac{F_{goal} \cdot \cos(G_L) + F_{st} \cdot \cos(\theta_{st}) + F_{dy} \cdot \cos(\theta_{dy})}{F_{goal} \cdot \sin(G_L) + F_{st} \cdot \sin(\theta_{st}) + F_{dy} \cdot \sin(\theta_{dy})} \quad (12)$$

$$V_x = v^t \cdot \cos(\alpha) \quad (13)$$

$$V_y = v^t \cdot \sin(\alpha) \quad (14)$$

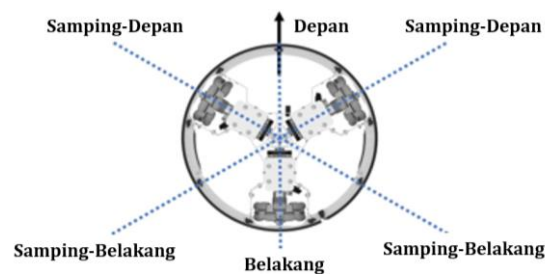
$$V_\theta = G_L - \theta \quad (15)$$

Dimana G_L adalah posisi titik tuju terhadap *mobile* robot. θ_{st} , θ_{dy} adalah posisi halangan statis terdekat terhadap *mobile* robot dan posisi halangan dinamis terdekat terhadap robot. Sedangkan θ adalah arah hadap robot saat ini.

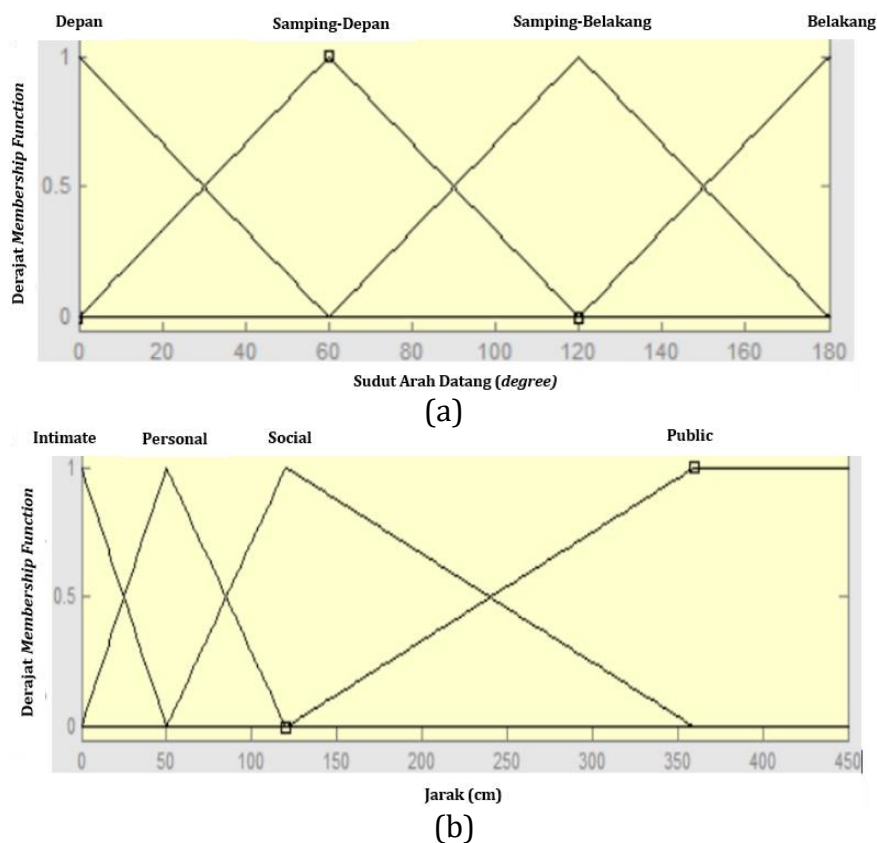
Nilai k pada Persamaan 6-7 untuk gaya tolak terhadap halangan statis dan Persamaan 9-10 untuk gaya tolak terhadap halangan dinamis adalah besaran nilai tambah atau *gain* pada SFM yang mempengaruhi gerakan navigasi *mobile* robot. Nilai ini akan mempengaruhi respon robot untuk menghindari ketika ada halangan didekatnya. Karena nilai ini bersifat tetap, robot akan menghasilkan respon yang sama baik ketika halangan statis maupun dinamis dalam jarak yang dekat atau jauh. Untuk membuat nilai k lebih adaptif, SFM akan dikembangkan menjadi *Fuzzy-Social Force Model* (FSFM) yang diusulkan oleh Dewantara, dkk[12].

Nilai k akan menjadi output dari *Fuzzy Inference System* (FIS) dengan input *membership function* berupa informasi jarak dan arah datang halangan terhadap *mobile robot*. Dalam penelitian ini, FIS untuk nilai k untuk halangan statis dan dinamis akan diatur dengan aturan *fuzzy* yang sama.

Agar *mobile robot* mengetahui arah datang halangan, maka robot akan dibagi menjadi beberapa area seperti Gambar 4. Sedangkan untuk *membership function* yang akan digunakan pada FSFM dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 4. Pembagian Area pada *Mobile Robot Omnidirectional*



Gambar 5. *Membership Function* (a) Arah Datang Halangan terhadap Robot
(b) Jarak Halangan terhadap Robot

3. Particle Swarm Optimization (PSO)

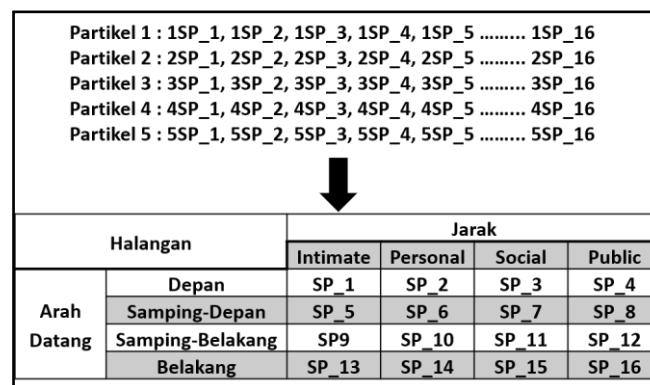
Algoritma optimasi *Particle Swarm Optimization* (PSO) digunakan untuk mengoptimasi aturan fuzzy pada *Fuzzy-Social Force Model* (FSFM). Implementasi optimasi PSO pada *Fuzzy-Social Force Model* (FSFM) merujuk ke Slowik [16], dimana optimasi PSO akan mengatur 2 dimensi partikel terdiri dari partikel itu sendiri dan sub-partikel. Untuk melakukan *update* kecepatan dan posisi dari tiap partikel dapat dilihat pada Persamaan 16-17.

$$Vel_{n,m} = Vel_{n,m} + c_1 \cdot rand_1 \cdot (PartB_{n,m} - P_{n,m}) + c_2 \cdot rand_2 \cdot (GlobB_{n,m} - Part_{n,m}) \quad (16)$$

$$Pos_{n,m} = Vel_{n,m} + P_{n,m} \quad (17)$$

Dimana $Vel_{n,m}$, $Pos_{n,m}$ adalah nilai kecepatan dan posisi partikel dengan n,m merupakan posisi partikel dan sub-partikel. Parameter c_1 dan c_2 adalah parameter PSO yang merepresentasikan parameter *cognitive* dan parameter *social*. Nilai $rand_1$ dan $rand_2$ adalah bilangan random dari 0 – 1. Nilai $P_{n,m}$ adalah nilai partikel pada koordinat n,m . Sedangkan nilai $PartB_{n,m}$ adalah nilai partikel terbaik di dalam koordinat n,m ketika iterasi optimasi berjalan dan $GlobB_{n,m}$ adalah nilai partikel terbaik dari kumpulan populasi $PartB$ dalam iterasi optimasi berjalan.

Pada penelitian ini, jumlah *swarm* partikel yang akan digunakan adalah 5 partikel. Setiap partikel akan mempunyai sub-partikel sebanyak 16 sub-partikel. Sub-partikel ini akan menampung nilai dari aturan fuzzy yang akan di optimasi. Ilustrasi distribusi partikel dan sub-partikel PSO pada Gambar 6.



Gambar 6. Distribusi Nilai Aturan Fuzzy

Pada penelitian ini, tiap-tiap partikel dan sub-partikel akan diuji di simulasi dalam setiap iterasi. Kemudian Persamaan 18 akan melakukan pengecekan perfoma dari partikel dan sub-partikel yang diuji. Nilai terbaik yang diambil dari hasil cek performansi adalah nilai yang paling kecil.

$$Objective Function = GOAL + TIME + \alpha_{Goal} \quad (18)$$

Dimana **GOAL** adalah pemberian bobot nilai jika *mobile* robot mencapai titik tuju, dimana bernilai 100 jika berhasil dan 1000 jika gagal. **TIME** adalah waktu tempuh *mobile* robot bernavigasi dari titik mulai mencapai titik tuju dan α_{Goal}

adalah selisih *error* arah hadap *mobile* robot terhadap titik tuju. Nilai ini akan dihitung setiap akhir iterasi untuk dilakukan cek performansi partikel. Untuk menghitung nilai dari α_{Goal} menggunakan Persamaan 19.

$$\alpha_{Goal} = \frac{\sqrt{(Goal\ direction - robot\ heading)^2}}{n} \quad (19)$$

Dimana *Goal direction* adalah posisi titik tuju yang berada di ruang sosial terhadap *mobile* robot, sedangkan *robot heading* adalah arah hadap robot saat ini terhadap posisi titik tuju. Nilai n merepresentasikan banyak data dalam satu iterasi berjalan.

C. Hasil dan Pembahasan

Pada penelitian ini dilakukan uji eksperimental menggunakan *software* simulasi realistik 3D VREP. Adapun spesifikasi *hardware* yang digunakan dalam uji eksperimental ini seperti yang dapat dilihat di Tabel 1.

Tabel 1. Spesifikasi *Hardware* Simulasi

<i>Hardware</i>	<i>Keterangan</i>
Laptop	Asus ROG GL552 VX
Prosesor	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
RAM	16GB
HDD	SSD Samsung EVO 860 500GB
Sistem Operasi dan <i>software</i> lainnya	WINDOWS 10 HOME
	Visual Studio C++ 2019
	VREP 3.6.2
	OpenCV 2.4.9

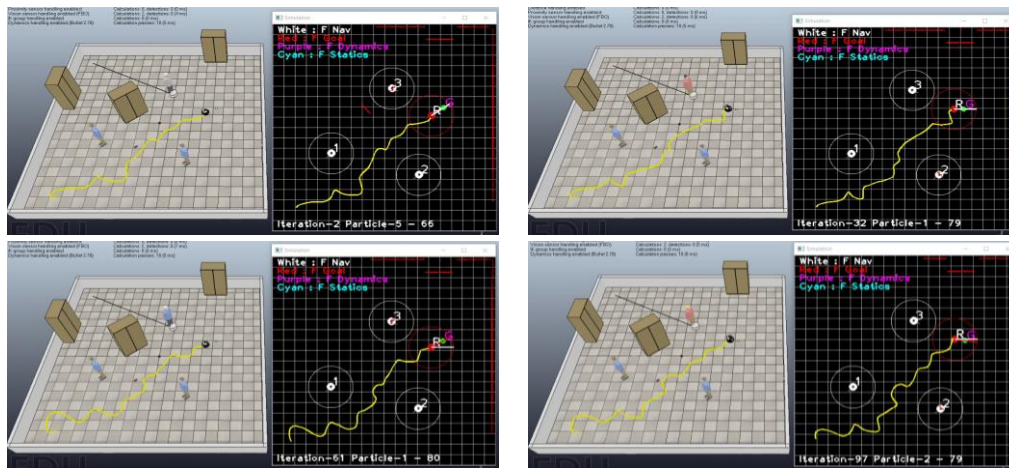
Dalam uji eksperimental ini, VREP akan memberikan data informasi guna membantu dalam proses simulasi, seperti informasi posisi robot dalam koordinat sumbu x dan y, informasi arah hadap robot, dan informasi posisi manusia sebagai halangan dinamis dalam koordinat sumbu x dan y. Posisi *goal* sebagai titik tuju juga diberikan dalam bentuk data sumbu x dan y. VREP sebagai *software* simulasi dan Visual Studio 2019 C++ sebagai *software coding* akan dihubungkan menggunakan API C++ *connection*.

Untuk data yang diambil adalah cuplikan gambar optimasi PSO yang diambil sebanyak empat kali tiap *dataset* parameter. Selain itu data yang diambil adalah data tabel yang terdiri dari keberhasilan mencapai tujuan, waktu tempuh mencapai tujuan, nilai selisih *error* α_{Goal} yang diambil dari nilai *GlobBest* tiap *dataset*. Kemudian data terakhir yang diambil adalah evolusi dari nilai *GlobBest* dari masing-masing *dataset*. Untuk parameter PSO yang digunakan dalam pengujian dapat dilihat di Tabel 2.

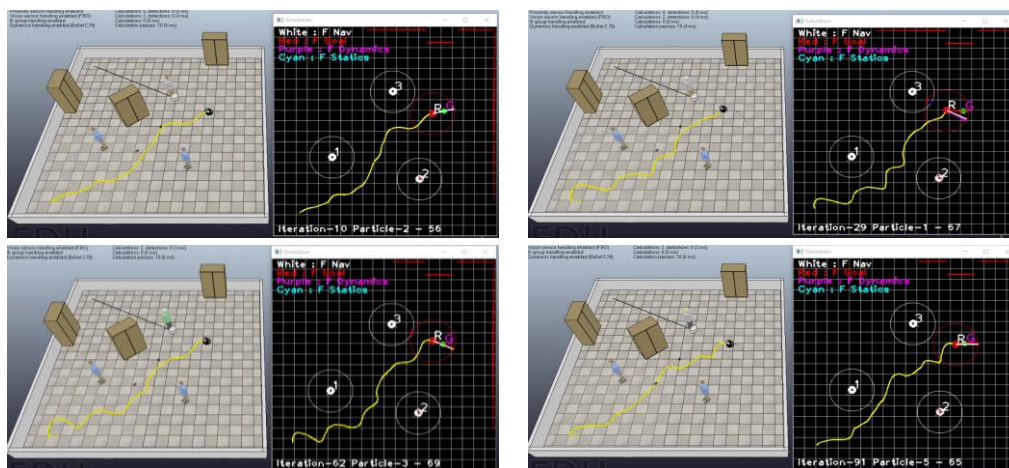
Tabel 2. Parameter Particle Swarm Optimization

Parameter	Nilai
Partikel	5
Sub-partikel	16
c1	0.5, 1, 2
c2	0.5, 1, 2
Nilai Maksimum Partikel	1000
Nilai Minimum Partikel	25
Iterasi Optimasi	100

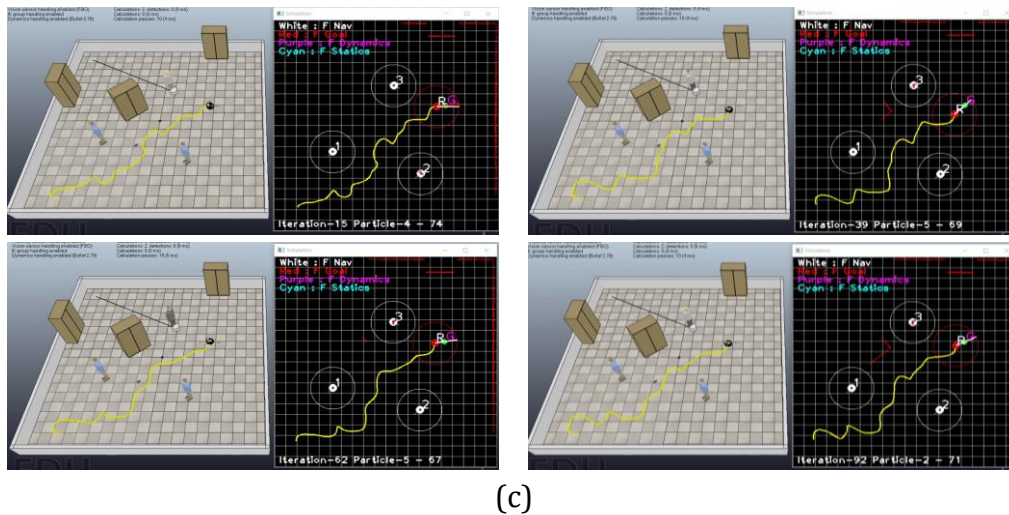
Pada pengujian pertama ini, parameter *cognitive* c1 di set 0.5 dan parameter *social* c2 di set 0.5, 1 dan 2. Cuplikan gambar navigasi dalam proses optimasi dapat dilihat pada Gambar 7, sedangkan data hasil navigasi dari *GlobBest* tiap dataset dapat dilihat pada Tabel 3. Kemudian untuk evolusi nilai *GlobBest* untuk masing-masing *dataset* dapat dilihat pada Gambar 8.



(a)

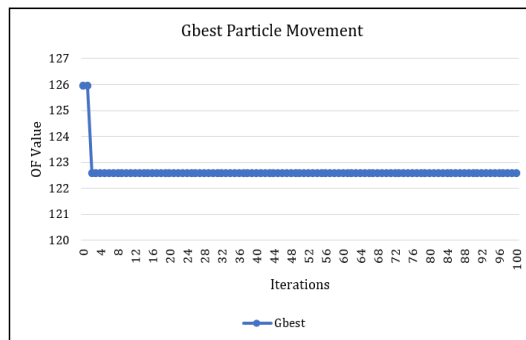


(b)

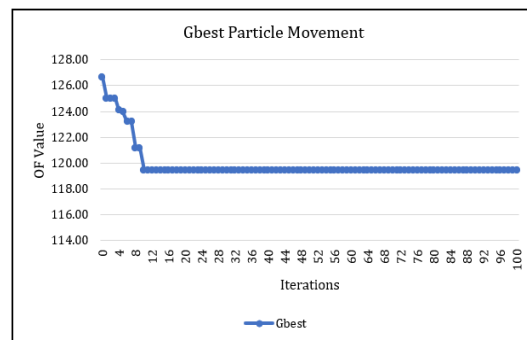


(c)

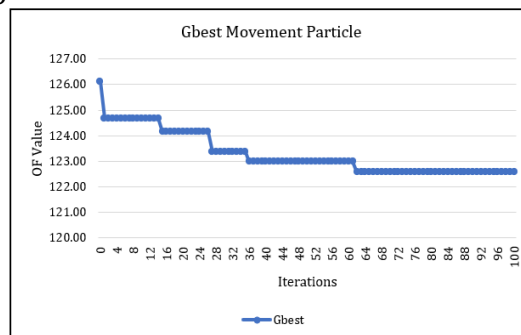
Gambar 7. Hasil Navigasi pada Proses Optimasi PSO (a) $c1 = 0.5$ dan $c2 = 0.5$ (b) $c1 = 0.5$ dan $c2 = 1$ (c) $c1 = 0.5$ dan $c2 = 2$



(a)



(b)



(c)

Gambar 8. Data Evolusi Nilai *GlobBest* pada Proses Optimasi PSO (a) $c1 = 0.5$ dan $c2 = 0.5$ (b) $c1 = 0.5$ dan $c2 = 1$ (c) $c1 = 0.5$ dan $c2 = 2$

Tabel 3. Hasil Data *GlobBest* tiap Dataset

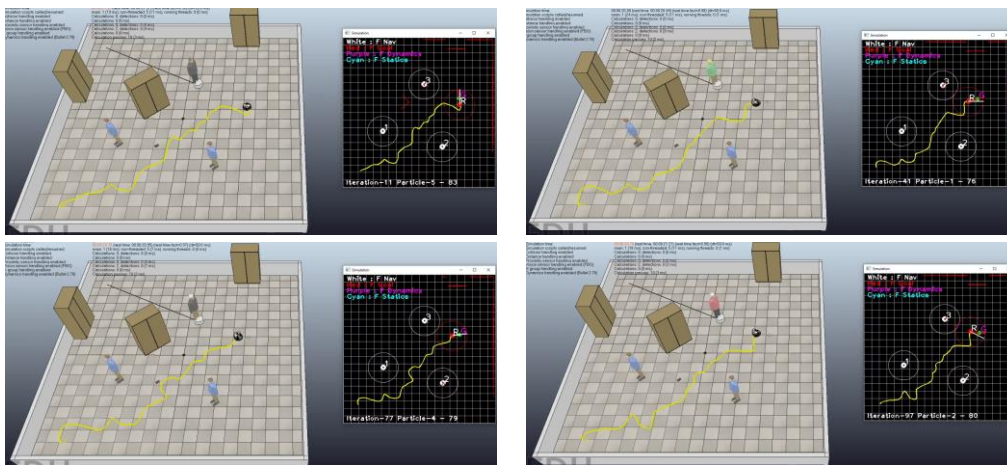
c1	c2	Mencapai Tujuan?	Waktu Tempuh (s)	α_{Goal}	Nilai <i>GlobBest</i>
0.5	0.5	Ya	22.15	0.4136	122.56
0.5	1	Ya	19.26	0.2035	119.47
0.5	2	Ya	22.21	0.3624	122.58

Pada percobaan pertama, nilai *GlobBest* paling baik didapat dengan nilai $c1 = 0.5$ dan $c2 = 1$. Untuk hasil dari aturan *fuzzy* yang dioptimasi oleh PSO dapat dilihat pada Tabel 4.

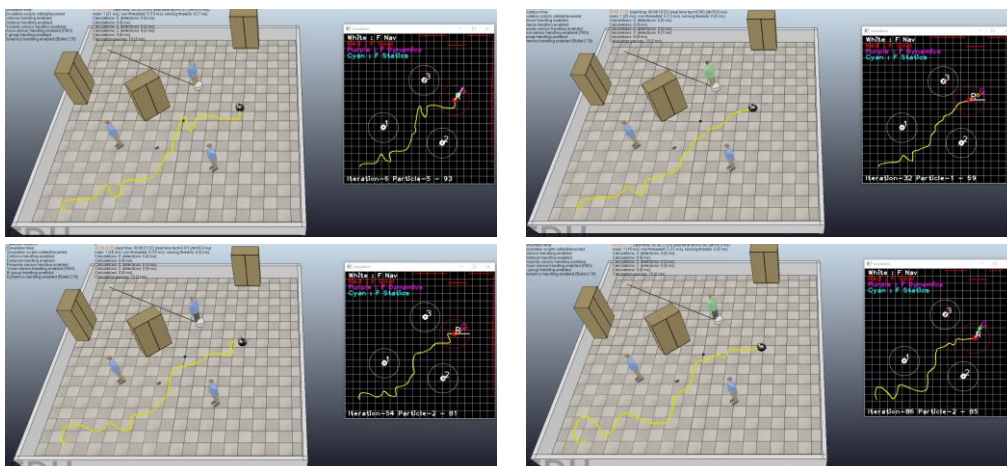
Tabel 4. Hasil Optimasi PSO untuk Nilai Aturan *Fuzzy*

Halangan		Jarak			
		Intimate	Personal	Social	Public
Arah Datang	Depan	939	25	25	430
	Samping-Depan	574	25	25	1000
	Samping-Belakang	136	25	25	25
	Belakang	1000	25	562	798

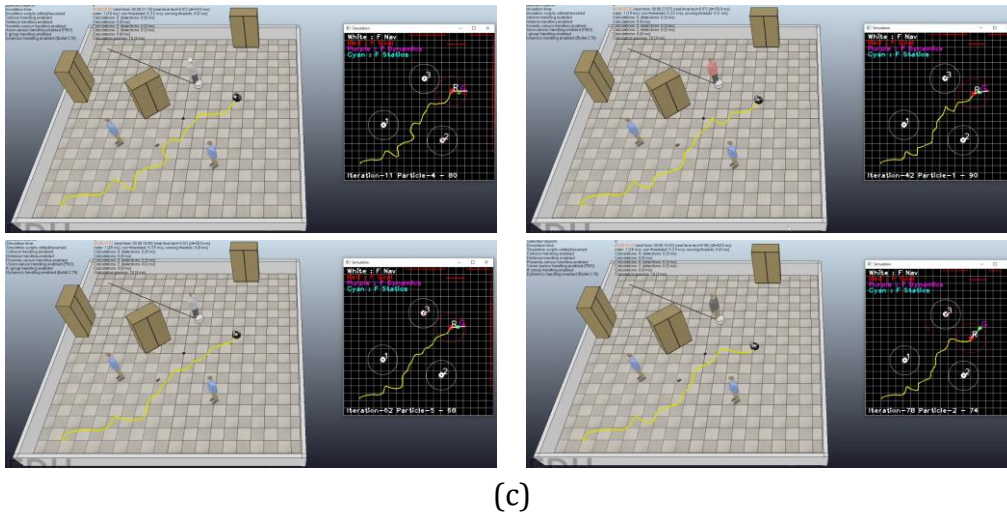
Pada pengujian kedua, parameter *cognitive* $c1$ di set 1 dan parameter *social* $c2$ di set 0.5, 1 dan 2. Cuplikan gambar navigasi dalam proses optimasi dapat dilihat pada Gambar 9, sedangkan data hasil navigasi dari *GlobBest* tiap dataset dapat dilihat pada Tabel 5. Kemudian untuk evolusi nilai *GlobBest* untuk masing-masing *dataset* dapat dilihat pada Gambar 10.



(a)

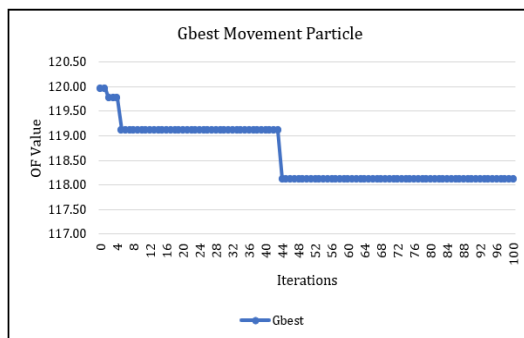


(b)

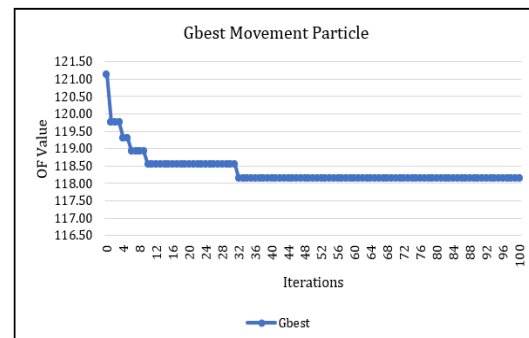


(c)

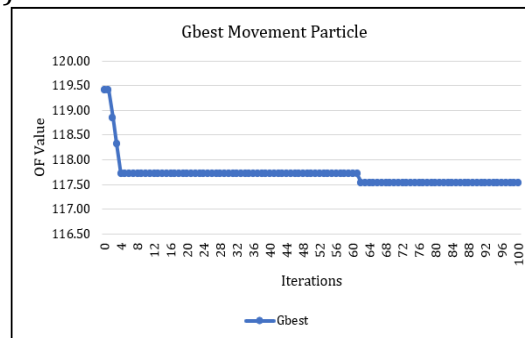
Gambar 9. Hasil Navigasi pada Proses Optimasi PSO (a) $c1 = 1$ dan $c2 = 0.5$
(b) $c1 = 1$ dan $c2 = 1$ (c) $c1 = 1$ dan $c2 = 2$



(a)



(b)



(c)

Gambar 10. Data Evolusi Nilai *GlobBest* pada Proses Optimasi PSO
(a) $c1 = 1$ dan $c2 = 0.5$ (b) $c1 = 1$ dan $c2 = 1$ (c) $c1 = 1$ dan $c2 = 2$

Tabel 5. Hasil Data *GlobBest* tiap Dataset

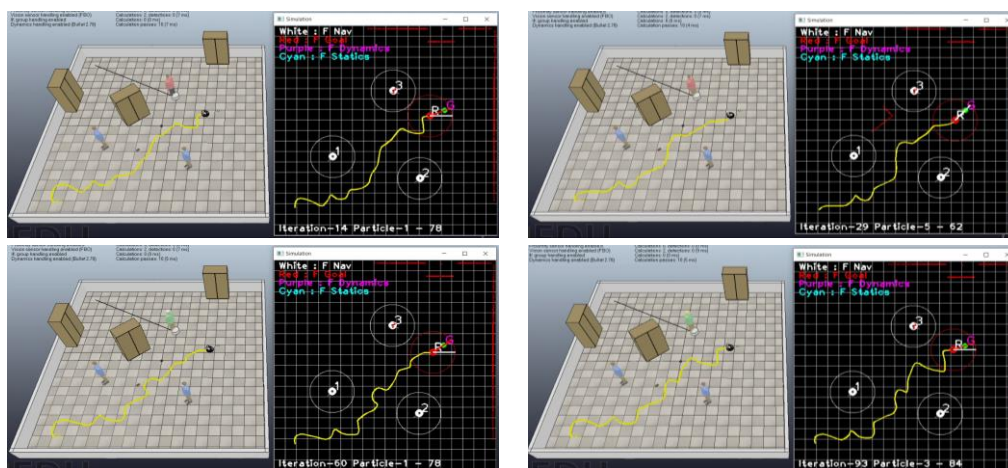
c1	c2	Mencapai Tujuan?	Waktu Tempuh (s)	α_{Goal}	Nilai <i>GlobBest</i>
1	0.5	Ya	17.75	0.3710	118.12
1	1	Ya	17.82	0.3439	118.16
1	2	Ya	17.09	0.4354	117.53

Pada percobaan kedua, nilai *GlobBest* paling baik didapat dengan nilai $c1 = 1$ dan $c2 = 2$. Untuk hasil dari aturan *fuzzy* yang dioptimasi oleh PSO dapat dilihat pada Tabel 6.

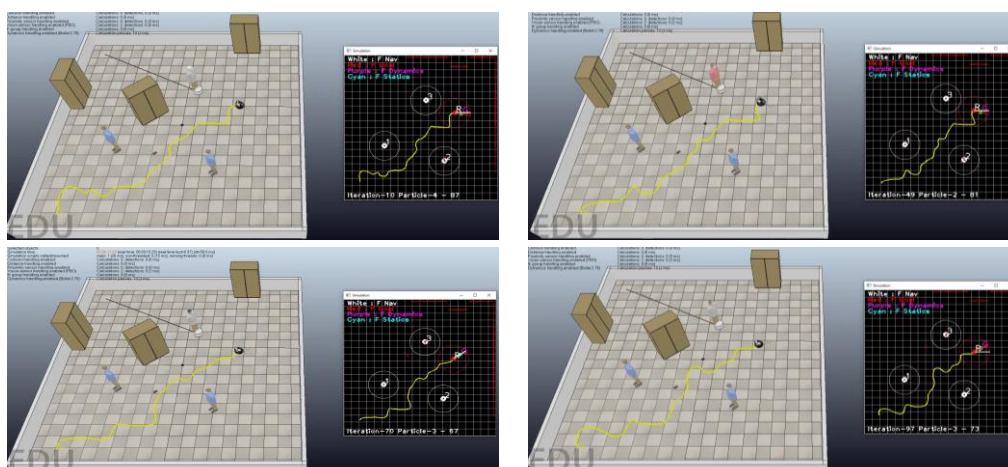
Tabel 6. Hasil Optimasi PSO untuk Nilai Aturan *Fuzzy*

Halangan		Jarak			
		Intimate	Personal	Social	Public
Arah Datang	Depan	254	346	152	146
	Samping-Depan	670	577	358	1000
	Samping-Belakang	595	866	700	128
	Belakang	607	987	280	240

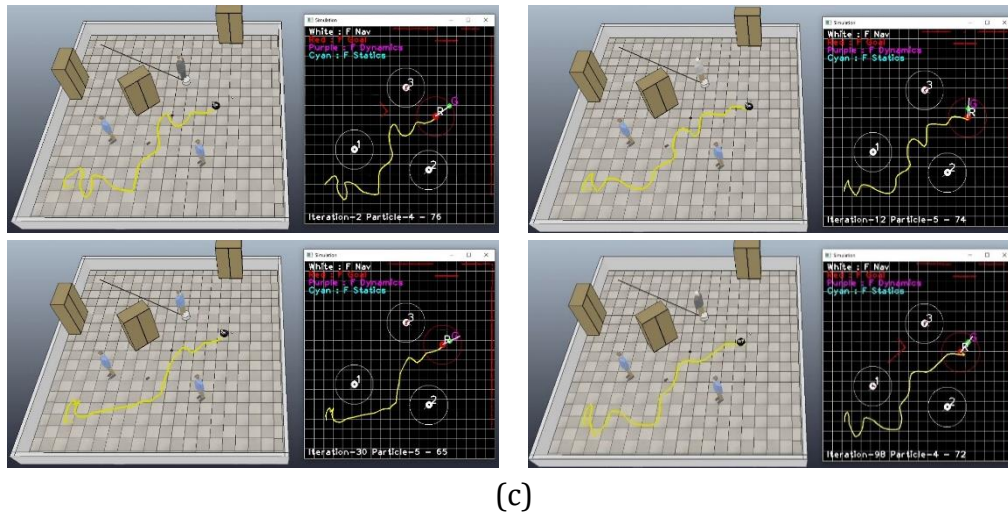
Pada pengujian ketiga, parameter *cognitive* $c1$ di set 2 dan parameter *social* $c2$ di set 0.5, 1 dan 2. Cuplikan gambar navigasi dalam proses optimasi dapat dilihat pada Gambar 11, sedangkan data hasil navigasi dari *GlobBest* tiap dataset dapat dilihat pada Tabel 7. Kemudian untuk evolusi nilai *GlobBest* untuk masing-masing *dataset* dapat dilihat pada Gambar 12.



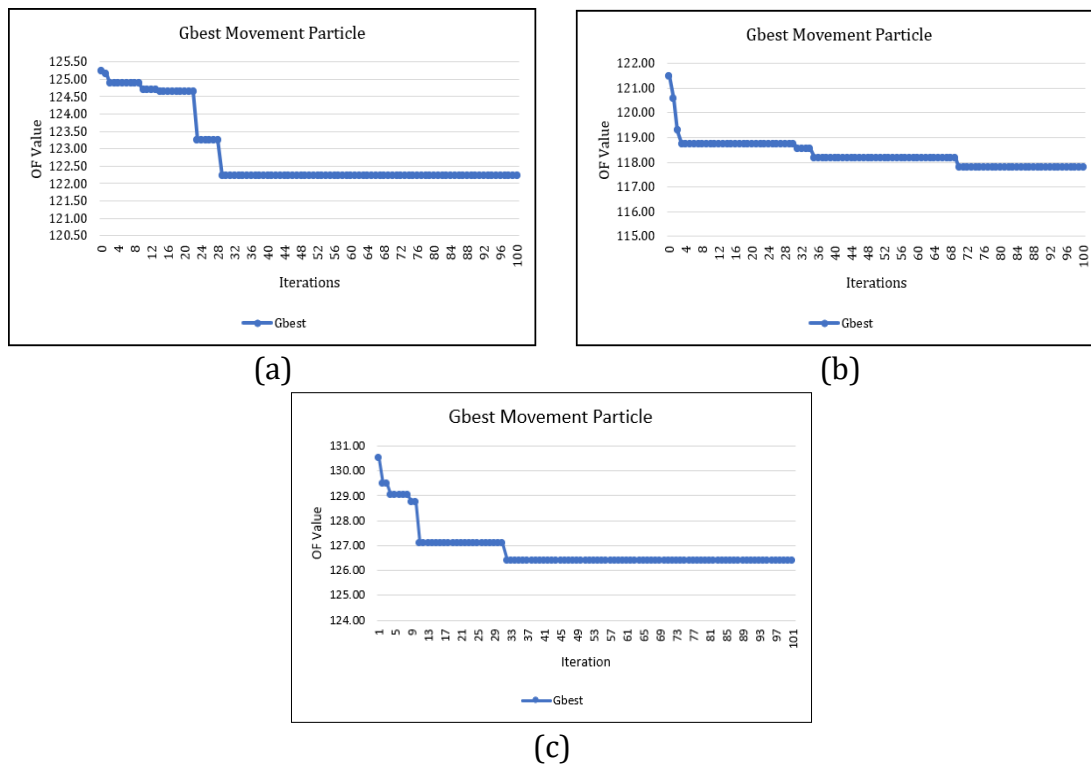
(a)



(b)



Gambar 11. Hasil Navigasi pada Proses Optimasi PSO
(a) $c1 = 2$ dan $c2 = 0.5$ (b) $c1 = 2$ dan $c2 = 1$ (c) $c1 = 2$ dan $c2 = 2$



Gambar 12. Data Evolusi Nilai *GlobBest* pada Proses Optimasi PSO
(a) $c1 = 2$ dan $c2 = 0.5$ (b) $c1 = 2$ dan $c2 = 1$ (c) $c1 = 2$ dan $c2 = 2$

Tabel 7. Hasil Data *GlobBest* tiap Dataset

c1	c2	Mencapai Tujuan?	Waktu Tempuh (s)	α_{Goal}	Nilai <i>GlobBest</i>
2	0.5	Ya	21.92	0.3172	122.24
2	1	Ya	17.38	0.4140	117.79
2	2	Ya	25.96	0.4173	126.38

Pada percobaan ketiga, nilai *GlobBest* paling baik didapat dengan nilai $c1 = 2$ dan $c2 = 1$. Untuk hasil dari optimasi aturan *fuzzy* yang dioptimasi oleh PSO dapat dilihat pada Tabel 8.

Tabel 8. Hasil Optimasi PSO untuk Nilai Aturan *Fuzzy*

Halangan		Jarak			
		Intimate	Personal	Social	Public
Arah Datang	Depan	25	343	858	470
	Samping-Depan	641	1000	1000	986
	Samping-Belakang	304	1000	1000	179
	Belakang	25	264	25	1000

Untuk melihat nilai parameter $c1$ dan $c2$ terbaik, maka semua hasil data uji akan dibandingkan dan dapat dilihat pada Tabel 9.

Tabel 9. Hasil Data *GlobBest* tiap *Dataset*

$c1$	$c2$	Mencapai Tujuan?	Waktu Tempuh (s)	α_{Goal}	Nilai <i>GlobBest</i>
0.5	0.5	Ya	22.15	0.4136	122.56
0.5	1	Ya	19.26	0.2035	119.47
0.5	2	Ya	22.21	0.3624	122.58
1	0.5	Ya	17.75	0.3710	118.12
1	1	Ya	17.82	0.3439	118.16
1	2	Ya	17.09	0.4354	117.53
2	0.5	Ya	21.92	0.3172	122.24
2	1	Ya	17.38	0.4140	117.79
2	2	Ya	25.96	0.4173	126.38

Pada hasil uji eksperimen ini seperti yang tertera di Tabel 9, nilai $c1 = 1$ dan $c2 = 2$ menghasilkan menghasilkan nilai *GlobBest* terbaik dengan nilai 117.53. Pada nilai parameter ini, waktu tempuh *mobile* robot mencapai titik tuju adalah waktu terbaik dari keseluruhan data uji lainnya yaitu 17.09s. Namun nilai α_{Goal} terbaik berada di parameter $c1 = 0.5$ dan $c2 = 1$ di 0.2035. Secara keseluruhan data, parameter terbaik berada di nilai *GlobBest* terkecil dimana persamaan *objective function* yang digunakan adalah jumlah dari penambahan data keberhasilan *mobile* robot mencapai titik tuju, waktu tempuh robot mencapai titik tuju dan selisih error arah hadap robot terhadap titik tuju.

D. Simpulan

Penelitian ini melakukan implentasi *Particle Swarm Optimization* (PSO) sebagai algoritma optimasi yang diterapkan untuk mengoptimasi *Fuzzy-Social Force Model* (FSFM) pada sistem navigasi robot *Omnidirectional*. Hasil uji eksperimental menunjukkan bahwa perubahan parameter *cognitive* ($c1$) dan parameter *social* ($c2$) pada algoritma PSO menghasilkan perbedaan hasil optimasi. Dari hasil uji, nilai $c1 = 1$ dan $c2 = 2$ menghasilkan nilai *GlobBest* yang terkecil dibandingkan nilai parameter lainnya yaitu 117.53 dan waktu tempuh tercepat menuju titik *goal* di 17.09s. Sedangkan untuk nilai *error* α_{Goal} terkecil berada di nilai parameter $c1 = 0.5$ dan $c2 = 1$ di 0.2035. Secara keseluruhan, parameter $c1 = 1$ dan $c2 = 2$ adalah parameter terbaik hasil optimasi PSO. Di penelitian di masa depan, kami akan melakukan

peningkatan untuk optimasi aturan *fuzzy* yang independen untuk masing-masing **k** pada gaya tolak untuk halangan statis dan dinamis. Selain itu, kami akan melakukan implementasi hasil optimasi PSO disimulasi VREP ini ke *mobile* robot nyata dan aktual ruang sosial untuk mengukur performansi optimasi PSO.

E. Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada seluruh anggota lab AIS (*Active Intelligence System*) dan Politeknik Elektronika Negeri Surabaya (PENS) untuk dukungan finansial dan non-finansial demi terselesaikannya penelitian ini.

F. Referensi

- [1] B. S. B. Dewantara and J. Miura, "Generation of a socially aware behavior of a guide robot using reinforcement learning," in *Proceedings - 2016 International Electronics Symposium, IES 2016*, pp. 105–110, Sep. 2017.
- [2] A. Bellarbi, S. Kahlouche, N. Achour, and N. Ouadah, "A social planning and navigation for tour-guide robot in human environment," *Proc. 2016 8th Int. Conf. Model. Identif. Control. ICMIC 2016*, pp. 622–627, 2017.
- [3] M. Kuderer and W. Burgard, "An approach to socially compliant leader following for mobile robots," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8755, pp. 239–248, 2014.
- [4] Ó. Gil, A. Garrell, and A. Sanfeliu, "Social robot navigation tasks: Combining machine learning techniques and social force model," *Sensors*, vol. 21, no. 21, 2021.
- [5] M. Fuad, T. Agustinah, and D. Purwanto, "Modified Headed Social Force Model Based on Hybrid Velocity Obstacles for Mobile Robot to Avoid Disturbed Groups of Pedestrians," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 3, pp. 222–241, 2021.
- [6] C. T. Yang, T. Zhang, L. P. Chen, and L. C. Fu, "Socially-aware navigation of omnidirectional mobile robot with extended social force model in multi-human environment," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, vol. 2019-Octob, pp. 1963–1968, 2019.
- [7] H. Kivrak, F. Cakmak, H. Kose, and S. Yavuz, "Social navigation framework for assistive robots in human inhabited unknown environments," *Eng. Sci. Technol. an Int. J.*, vol. 24, no. 2, pp. 284–298, 2021.
- [8] E. Repiso, A. Garrell, and A. Sanfeliu, "Adaptive Side-by-Side Social Robot Navigation to Approach and Interact with People," *Int. J. Soc. Robot.*, 2019.
- [9] S. F. Chik *et al.*, "Gaussian pedestrian proxemics model with social force for service robot navigation in dynamic environment," *Commun. Comput. Inf. Sci.*, vol. 751, pp. 61–73, 2017.
- [10] D. Yang and Ü. Özgüner, "Combining Social Force Model with Model Predictive Control for Vehicle's Longitudinal Speed Regulation in Pedestrian-Dense Scenarios," Jul. 2019.
- [11] A. T. Rifqi, B. S. B. Dewantara, D. Pramadihanto, and B. S. Marta, "Fuzzy Social Force Model for Healthcare Robot Navigation and Obstacle Avoidance," *Int. Electron. Symp. 2021 Wirel. Technol. Intell. Syst. Better Hum. Lives, IES 2021 - Proc.*, pp. 445–450, 2021.
- [12] B. S. B. Dewantara and B. N. D. Ariyadi, "Adaptive Behavior Control for Robot Soccer Navigation Using Fuzzy-based Social Force Model," *Smart Sci.*, vol. 9,

- no. 1, pp. 14–29, 2021.
- [13] C. B. M. Bliesener, Ch. Weber, K. Kling, U. Karras, D. Zitzmann, *Workbook Robotino*. 2013.
 - [14] D. Helbing and P. Molnár, “Social force model for pedestrian dynamics,” *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, May 1995.
 - [15] R. E. Spencer, “Book Reviews : The Hidden Dimension by Edward T. Hall. New York: Doubleday and Company, Inc., 1966. Pp. xii + 193,” *Educ. Psychol. Meas.*, vol. 26, no. 4, pp. 1118–1119, Dec. 1966.
 - [16] A. Slowik, *Swarm intelligence algorithms. A tutorial*, First edit., vol. 1999, no. December. 2020.