

## Implementasi Algoritma PSO Pada Multi Mobile Robot Dalam Penentuan Posisi Target Terdekat

Ikhwannuary Raditya Priyadana<sup>\*1</sup>, Bakhtiar Alldino Ardi Sumbodo<sup>2</sup>, Triyogatama Wahyu Widodo<sup>3</sup>

<sup>1</sup>Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta

<sup>2,3</sup>Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: <sup>1</sup>[\\*radityaikhwan@gmail.com](mailto:*radityaikhwan@gmail.com), <sup>2</sup>b.alldino.as@ugm.ac.id, <sup>3</sup>triyogatama@ugm.ac.id

### Abstrak

*Swarm Intelligence merupakan suatu kecerdasan buatan yang dikembangkan dengan mengadaptasi perilaku sosial sekelompok hewan di dalam suatu komunitas. Pada komunitas burung migrasi dikenal adanya perilaku gerombolan burung tersebut pada saat terbang membentuk formasi 'V' yang berperan untuk mengoptimasi kemampuan terbang burung yaitu dalam hal penghematan energi. Prinsip dasar dari sebuah swarm intelligence adalah adanya perilaku kolektif, terdesentralisasi dan self-organizing. Hal inilah yang menjadi dasar pengembangan algoritma perilaku flocking birds yang disebut dengan Particle Swarm Optimization (PSO).*

*Pada penelitian ini digunakan tiga buah autonomous mobile robot sebagai objek untuk mengimplementasikan algoritma PSO. Tiga buah robot ini bersifat homogen, yaitu serupa hardware dan software. Sekelompok robot ini akan menyelesaikan misi bersama yaitu menentukan robot dengan jarak yang terdekat dengan titik target TPr (robot handler). Ada tiga buah target TPr yang harus dieksekusi oleh robot handler sesuai dengan posisi mereka dengan titik target yang harus diselesaikan. Pengujian dilakukan dengan mengambil sampling data odometri setiap 250 milisekon dan data frame komunikasi robot.*

*Pada akhir penelitian ini didapatkan hasil berupa pemodelan sistem hasil implementasi algoritma PSO pada sekumpulan mobile robot untuk menentukan robot yang terdekat dengan target. Sistem robot yang terbentuk memenuhi prinsip PSO, yaitu adanya proses berbagi data dan proses learning.*

**Kata kunci** — swarm, mobile robot, algoritma PSO, flocking bird, misi target terdekat

### Abstract

*Swarm Intelligence is an artificial intelligence developed by adapting the social behavior of a group of animal. In the migratory birds community, it is known that the behavior of the birds during the flight forms a 'V' formation that plays a role in optimizing the bird's energy saving. The basic principle of a swarm intelligence is the existence of collective, decentralized and self-organizing behavior. This is the basis for the development of behavioral algorithms flocking birds called Particle Swarm Optimization (PSO).*

*In this research used three mobile robot as object to implement PSO algorithm. Three pieces of this robot is homogeneous, which is similar hardware and software. A group of these robots will complete the joint mission of defining the robot with the closest distance to the target TPr (robot handler). There are three TPr targets that have to be executed by the robot handler according to their position with the target point to be completed. The test is done by taking odometry data every 250 milisekon and data frame robot communication.*

*At the end of this research, the result of modeling system ~~result~~ of PSO algorithm implementation on mobile robot group to determine the robot closest to the target. The robot system that meets the principles of PSO, namely the process of data sharing and learning process.*

**Keywords** — swarm, mobile robot, PSO algorithm, flocking bird, closest target mission

## 1. PENDAHULUAN

Teknologi robotika sekarang ini sudah banyak menerapkan kecerdasan buatan sebagai algoritma utama pemroses sistem aktuator robot. *Swarm intelligence* merupakan salah satu kecerdasan buatan yang mengadaptasi perilaku sekelompok hewan dalam berinteraksi di dalam komunitasnya[1]. *Particle Swarm Optimization* atau PSO menjadi salah satu kecerdasan buatan (*artificial intelligence*) hasil pengembangan dari *swarm intelligence* umum yang berkembang cukup pesat sejak kemunculannya.

Prinsip dasar dari *swarm intelligence* adalah adanya perilaku kolektif, terdesentralisasi dan berkemampuan *self-organizing*[1], [2]. Awal mula dari pengembangan algoritma PSO adalah adanya fenomena perilaku komunitas burung-burung yang melakukan migrasi musiman[3]. Fenomena *flocking birds* ini membentuk formasi ‘V’ ketika melakukan penerbangan tersebut. Hal ini ternyata mampu menghemat pemakaian energi yang diperlukan[4].

Melalui penelitian ini penulis bermaksud untuk mengimplementasikan algoritma PSO ini dalam suatu pemodelan misi sekumpulan autonomous mobile robot. Hal ini dilakukan dengan maksud untuk mengetahui pengaruh dari implementasi pemodelan PSO dalam hal efektifitas misi yang dikerjakan oleh robot swarm.

## 2. METODE PENELITIAN

### 2.1 Analisis sistem

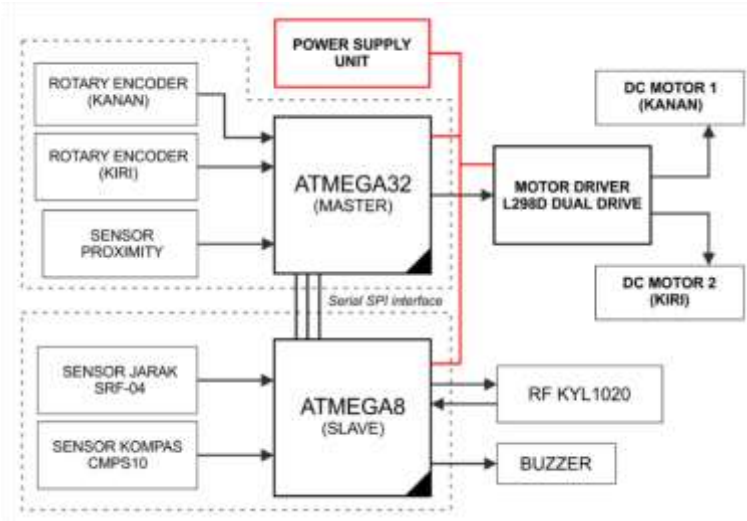
Penelitian yang dikerjakan mempunyai metodologi yang dimaksudkan untuk merancang implementasi algoritma PSO pada sekumpulan robot dalam bentuk pemodelan swarm. Sehingga dengan menguji rancangan pemodelan misi yang harus diselesaikan oleh sekumpulan robot swarm akan diketahui pengaruh serta tingkat keberhasilan hasil dari implementasi algoritma PSO yang dibuat.

### 2.2 Rancangan sistem

Rancangan sistem ini berupa bentuk robot swarm secara keseluruhan dari sistem yang dibuat, yaitu rancangan *hardware* ( mekanis dan elektronis), *software* dan rancangan pemodelan PSO yang akan dibuat. Sistem robot swarm yang dirancang merupakan robot yang berjenis *homogeneous mobile robot* yaitu robot berjumlah banyak dengan spesifikasi hardware dan software yang serupa identik. Hal ini karena di dalam konsep swarm tidak ditekankan untuk membuat satu individu yang super cerdas. Namun kecerdasannya justru muncul dari hasil interaksinya dengan individu lainnya[1], [5].

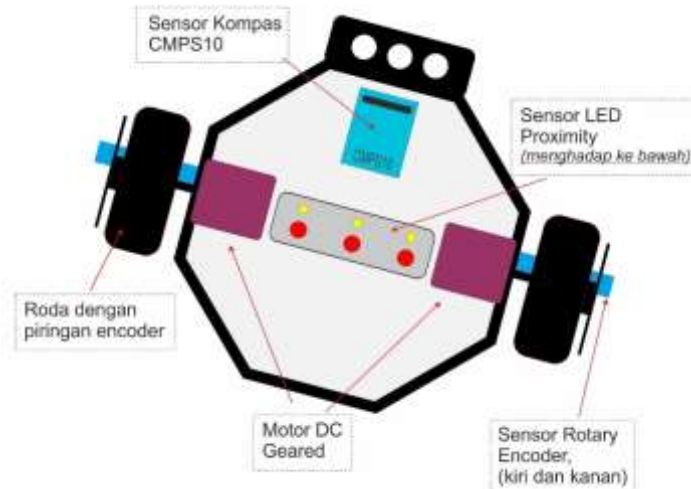
#### 2.2.1 Rancangan perangkat keras (mekanis dan elektronis)

Rancangan *hardware* ini terdiri atas bagian mekanis dan elektronis sistem robot swarm yang dibuat. Bagian elektronis sistem ditunjukkan seperti dalam Gambar 1. Setiap robot akan memiliki spesifikasi hardware yang sama. Sebagai pusat pemroses data digunakan dua buah mikrokontroler AVR 8 bit yaitu tipe ATmega32 yang berperan sebagai master dan ATmega8 berperan sebagai slave. Keduanya akan bekerja sama secara paralel.



Gambar 1 Blok diagram sistem *hardware* utama robot

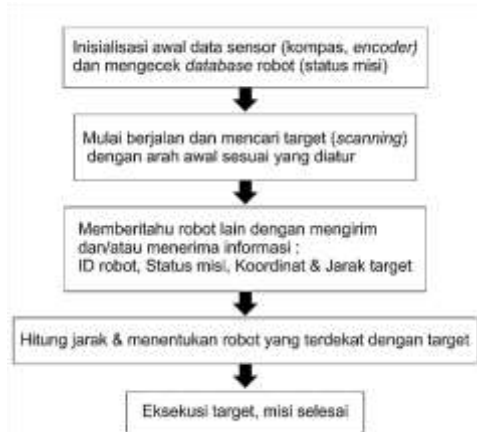
Peletakan sensor dan aktuator robot dapat dilihat seperti pada Gambar 2. Komponen sensor utama yang dipasangkan pada *autonomous mobile robot* ini yaitu sensor *proximity*, sensor kompas dan sensor *rotary encoder*. Sensor *rotary encoder* dapat dibuat dengan memakai piringan berlubang yang menempel pada bagian roda robot. Lubang ini akan digunakan untuk menghitung pulsa melalui sensor *optocoupler* yang memancarkan cahaya berbasis *proximity*. Kemudian pada bagian aktuator dipakai dua buah *geared motor DC* dengan komponen *motor driver* memakai IC L298D.



Gambar 2 Rancangan sistem mekanis robot tampak dari atas

### 2.2.2 Rancangan perangkat lunak (program)

Perangkat lunak yang dirancang merupakan program eksekusi yang akan ditanam ke dalam mikrokontroler setiap robot. Seperti ditunjukkan dalam Gambar 3, setiap robot akan memiliki pekerjaan yang serupa. Perancangan program algoritma PSO penentu robot jarak terdekat ini mempunyai tujuan akhir untuk menentukan robot terpilih sebagai *robot-handler*. Robot *target-handler* ini dipilih berdasarkan jarak yang terdekat di antara ketiga robot terhadap koordinat target yang diuji yaitu *Target Point robot* (TPr). Target robot ini berjumlah tiga buah koordinat yang berbeda yang kemudian disebut sebagai TPr-1, TPr-2 dan TPr-3. Bentuk dari target ini sendiri berupa lingkaran berwarna hitam yang diasumsikan sebagai titik dalam suatu koordinat kartesian dua dimensi ( $x,y$ ).

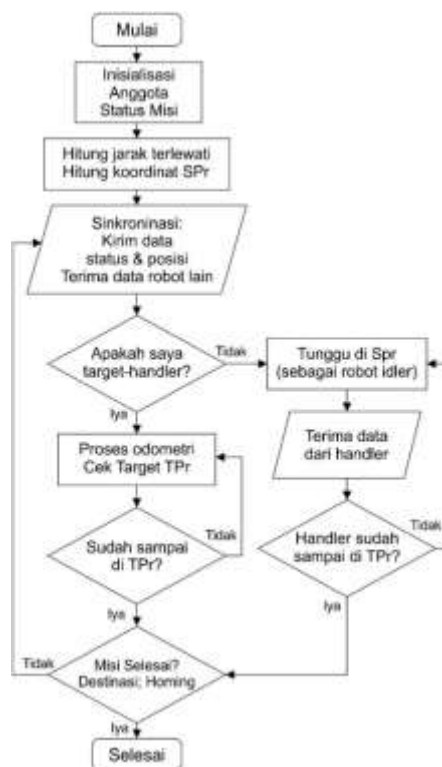


Gambar 3 Skema dasar proses pergerakan setiap robot

### 2.2.3 Rancangan pemodelan sistem dengan algoritma PSO

Misi pemodelan PSO yang dibuat pada penelitian ini adalah misi menentukan robot yang akan mengeksekusi “target bersama” yaitu robot dengan jarak yang paling dekat dengan target (TPr). TPr ini merupakan titik target yang menjadi target bersama. Dengan membandingkan posisi masing-masing robot dari SPr (*Starting Point robot*), maka akan diketahui robot mana yang harus mengeksekusi target TPr. Robot inilah yang kemudian disebut sebagai *robot target-handler*.

Untuk dapat menentukan robot mana yang merupakan target handler, maka harus diketahui jarak masing-masing individu robot terhadap titik target TPr. Proses penghitungan jarak ini akan dilakukan oleh setiap robot. Sehingga diharapkan prinsip swarm yang berkaitan dengan prinsip *decentralized system* dapat terpenuhi. Seperti ditunjukkan pada Gambar 4, misi akan berjalan secara berulang (*looping*) sampai dengan semua target telah (dieksekusi).



Gambar 4 Diagram alir skema pemodelan misi robot swarm dengan PSO

### 2.3 Implementasi sistem

Hasil dari implementasi sistem ini berupa *hardware*, *software* dan pemodelan misi PSO robot swarm. Hasil implementasi bagian hardware tampak seperti ditunjukkan Gambar 5.



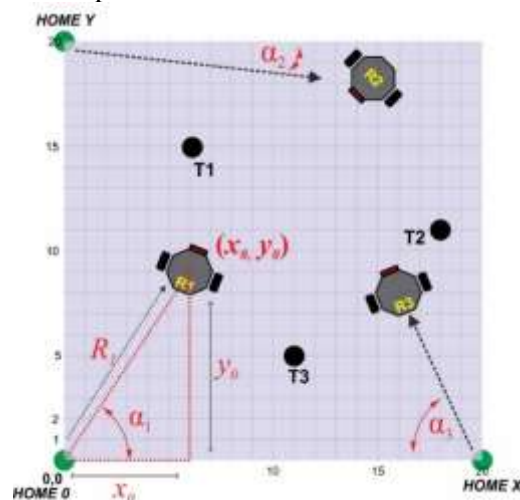
Gambar 5 Wujud dari sistem robot swarm yang dibuat

Komponen sensor yang digunakan dalam robot ini berjumlah 3 buah sensor. Ada sensor rotary encoder yang berfungsi untuk menghitung jarak perpindahan yang telah dilalui robot sekaligus untuk memindai target ketika robot melewati titik target. Sensor ini ditempatkan menempel pada chasis di atas masing-masing roda robot, sehingga total berjumlah 2 buah sensor.

### 2.4 Pemodelan misi robot dengan algoritma PSO

Pemodelan misi dengan inti menentukan robot *target-handler* ini mempunyai maksud agar proses pengujian terhadap algoritma yang dibuat menjadi lebih mudah untuk dianalisis. Untuk menentukan robot handler maka pertama kali harus diketahui dulu jarak dari setiap robot terhadap titik target yang diuji.

Pada saat pertama kali berjalan dari masing-masing titik *home*, setiap robot akan berjalan secara acak menuju suatu titik sembarang di area uji. Titik pemberhentian pertama inilah yang disebut sebagai titik *SPr* robot. Titik *SPr* ini dihitung terlebih dahulu sebagai titik *start* robot sebagai titik acuan awal untuk menghitung jarak robot ke titik target (*TPr*). Gambaran kondisinya seperti tampak dalam Gambar 6.



Gambar 6 Mapping robot saat berjalan dari titik *home* masing-masing (*loop-0*)

Selama di perjalanan menuju titik *SPr*, setiap robot akan menghitung jarak yang telah dilewati dari titik *home* sampai dengan saat berhenti di titik *SPr*. Jarak yang telah dilalui ini akan dihitung dengan mengandalkan data dari sensor *rotary encoder* yang terpasang di bagian

roda robot. Hasil dari sensor *encoder* ini kemudian dipakai untuk menghitung jarak yang telah terlewati dengan memakai Persamaan 1 di bawah.

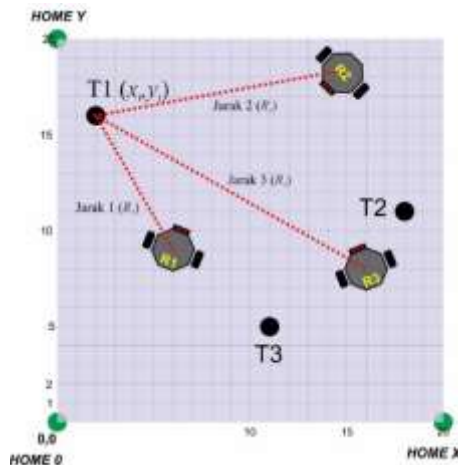
$$D \text{ tempuh} = ((Pulsa_R + Pulsa_L) * keliling) / 36 \dots\dots\dots(1)$$

Setelah nilai jarak dari titik *home* ke titik *SPr* diketahui, maka selanjutnya dicari koordinat titik *SPr* ( $X_o, Y_o$ ) dengan menerapkan konsep teorema segitiga pitagoras seperti tampak pada garis-garis imajiner Gambar 6 di atas. Kemudian untuk menghitungnya digunakan Persamaan 2 dan 3 di bawah[6]. Sudut  $\alpha$  merupakan nilai sudut yang didapatkan dari sensor kompas (*CMPS10*) pada robot.

$$X_o^2 = \frac{R^2}{(\tan \alpha)^2 + 1} \dots\dots\dots(2)$$

$$Y_o = X_o * \tan \alpha \dots\dots\dots(3)$$

Setelah koordinat titik *SPr* ( $X_o, Y_o$ ) diketahui maka semua robot akan tetap diam sejenak di titik *SPr* untuk melakukan penghitungan jarak antara dirinya dan titik target *TPr-1*. Untuk bahan uji maka dibuat empat variasi koordinat titik target *TPr*. Setiap titik *TPr* ini akan berganti secara otomatis mengikuti putaran misi robot (*loop*) yang sedang berlangsung. Tampak seperti dalam Gambar 7 berikut.

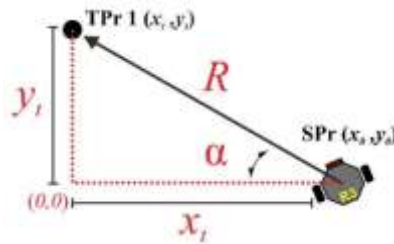


Gambar 7 Mapping robot saat berada di titik *SPr*

$$R_{robot}^2 = (X_t - X_o)^2 + (Y_t - Y_o)^2 \dots\dots\dots (4)$$

$$\alpha = \tan^{-1} \frac{(Y_t - Y_o)}{(X_t - X_o)} \dots\dots\dots (5)$$

Jarak dari titik *SPr* ke titik *TPr* ini dihitung dengan menerapkan teorema pitagoras segitiga siku. Sepeti tampak pada Gambar 8. Komponen *R* merupakan nilai jarak tempuh yang harus dihitung oleh masing-masing robot. Nilai *R* ini relatif terhadap titik *SPr* robot masing-masing. Kemudian  $\alpha$  merupakan nilai sudut yang harus digunakan robot untuk berputar memposisikan arah jalan yang sesuai. Nilai jarak (*R*) dan sudut putar ( $\alpha$ ) ini dihitung dengan memakai Persamaan 4 dan 5.



Gambar 8 Bagan segitiga imajiner pitagoras

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Pengujian sistem odometri robot

Untuk menghitung jarak yang ditempuh, robot akan mendapatkan data cacahan berupa banyaknya putaran roda dari sensor *rotary encoder*. Kecepatan pergerakan robot ini sangat dipengaruhi oleh kedali robot yang dipakai. Pulsa yang didapat dari sensor *encoder* ini akan menunjukkan banyaknya putaran yang dikerjakan oleh roda robot ketika berjalan. Dengan memakai rumus seperti Persamaan 6 banyaknya putaran robot akan linear dengan besarnya jarak yang telah ditempuh oleh robot.

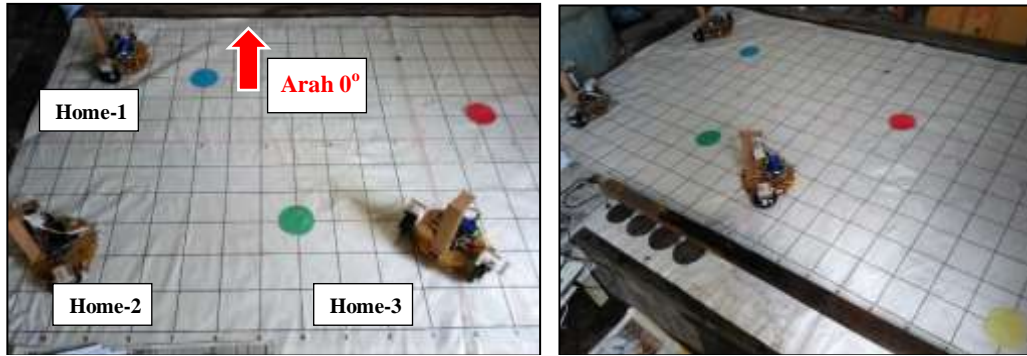
$$D \text{ tempuh} = \frac{(n * K \text{ roda})}{36} \dots (6)$$

Dari Persamaan 6 dapat dihitung nilai jarak yang telah ditempuh ( $D$ ) dalam satuan sentimeter ( $cm$ ). Komponen  $n$  merupakan banyaknya cacahan pulsa yang didapat dari sensor *rotary encoder*. Kemudian  $K$  merupakan nilai keliling roda robot. Dengan memakai rumus  $\pi * d$  maka didapatkan keliling roda sebesar  $25,12 \text{ cm}$ . Sedangkan angka 36 merupakan jumlah lubang cacahan pada roda yang dipakai oleh komponen *opto coupler* pada sensor *encoder* untuk menghitung jumlah pulsa.

Proses pengujian dilakukan dengan menjalankan robot secara lurus dan berhenti secara otomatis ketika sudah sampai pada jarak  $40 \text{ cm}$ . Pengambilan sampel data dilakukan setiap  $250 \text{ milisekon}$ . Sehingga dapat diketahui karakteristik sistem odometri masing-masing robot ketika berjalan. Perbedaan jumlah sampling data yang terekam pada ketiga robot menunjukkan bahwa kecepatan laju gerak robot ketiganya berbeda-beda. Hal ini dipengaruhi oleh sistem kendali gerak PID yang digunakan pada robot. Karakteristik dari sensor *encoder* yang dipakai di dalam sistem odometri ini menunjukkan adanya risiko *error sistemik*[7]. Seperti terlihat pada saat *timing* masih  $0$  sekon ternyata sudah ada data pulsa *encoder* yang terbaca. Hal ini dapat disebabkan oleh tingkat sensitivitas sensor yang mudah goyah terkait dengan perancangan mekanik robot.

#### 3.2 Algoritma pemilihan robot handler

Tujuan utama dari misi yang dikerjakan yaitu untuk menentukan robot dengan jarak yang paling dekat dengan target untuk kemudian mengeksekusinya. Langkah pertama yang dikerjakan untuk dapat menentukan robot terpilih tersebut adalah dengan melakukan penghitungan jarak ke target oleh masing-masing robot. Setelah masing-masing robot mengetahui jaraknya sendiri, selanjutnya dilakukan komunikasi antar robot untuk saling bertukar data tentang jarak dan posisi masing-masing terhadap titik target  $TPr$  saat itu. Seperti ditunjukkan pada Gambar 9, ada tiga buah titik  $TPr$  yang disimpan. Masing-masing robot tersebut memiliki variabel khusus *database* di dalam memori internalnya untuk menyimpan data semua robot lainnya.



Gambar 9 Posisi robot pada saat inisialisasi awal

3.3 Pengujian pemodelan misi positioning dengan algoritma PSO

Pengujian dilakukan dengan membuat tiga kali perulangan penentuan robot handler dengan tiga buah titik target yang berbeda dalam satu sesi uji. Setelah selesai sesi pertama kemudian dilanjutkan dengan pengujian sesi dua dan tiga. Sehingga total berjumlah ada sembilan kali loop pengujian dengan posisi awal home robot yang bervariasi. Tabel 2 merupakan data hasil pengujian pada saat robot berjalan dari titik home masing-masing menuju titik bebas (*SPr* = *Starting Point robot*).

Tabel 2 Hasil uji penghitungan odometri awal ke *SPr*

TES-1, <i>Ro</i> = 40 cm, <i>Loop-0</i>	Robot	Posisi Home (x, y)	Sudut awal (deg)	Jarak Riil (cm)	Koordinat Riil ( <i>Xo</i> , <i>Yo</i> )
	Robot A	H1 : 20, 100	120	48	65, 75
	Robot B	H2 : 20, 20	70	41	50, 50
	Robot C	H3 : 120, 20	70	40	170, 30

Setelah inisialisasi awal selesai, maka semua robot berjalan secara bersamaan menuju titik bebas di area uji dengan parameter : jarak tempuh sepanjang 40 cm (*Ro*), sudut awal 70 derajat dan berjalan lurus ke depan. *Timing diagram* saat ini disebut sebagai *loop-0*, yaitu pergerakan robot dari *Home* menuju titik *SPr*. Khusus untuk robot yang berada di titik *home-1* akan diarahkan sebesar 120 derajat. Ketika sudah menempuh jarak 40 cm, robot akan berhenti untuk menghitung posisi koordinat saat itu. Seperti dalam Tabel 2 setiap robot telah menghitung titik koordinatnya masing-masing. Titik inilah yang disebut sebagai titik *SPr* (*starting point robot*) dengan robot A berada di (65, 75), robot B(50,50) dan robot C(170,30). Titik *SPr* ini dihitung dengan memakai rumus trigonometri seperti pada Persamaan 7 dan 8. Penghitungan koordinat titik *SPr* setiap robot ini penting untuk menghitung jarak dari *SPr* menuju ke titik target *TPr*.

$$X_o^2 = \frac{R^2}{(\tan \alpha)^2 + 1} \dots\dots\dots(7)$$

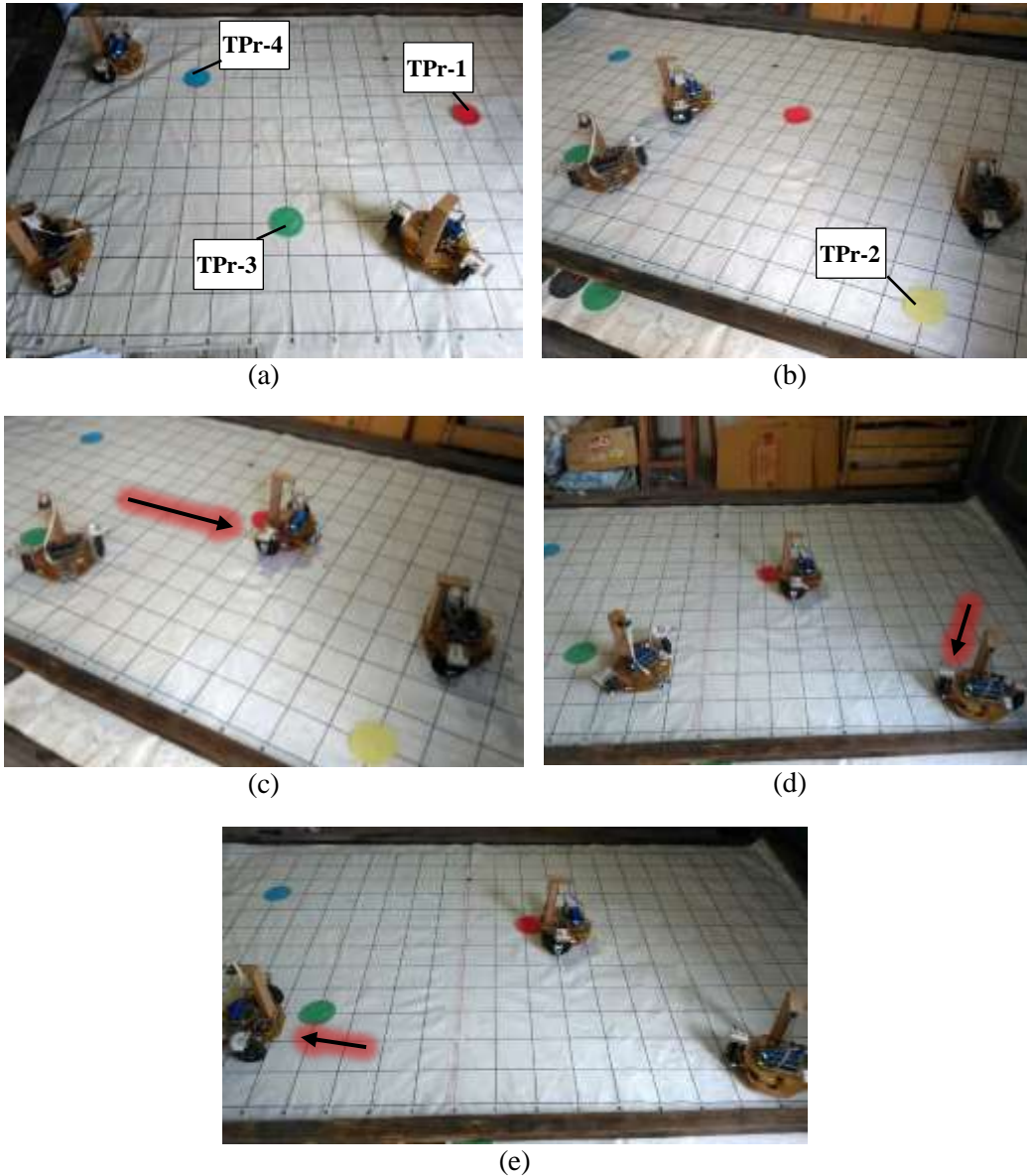
$$Y_o = X_o * \tan \alpha \dots\dots\dots(8)$$

Pada *loop-1* ini didapatkan hasil bahwa robot B merupakan robot yang terdekat dengan target *TPr-1* dengan jarak 46 cm. Maka robot B bergerak menuju *TPr-1*. Sedangkan kedua robot lainnya diam di titik semula (titik *SPr*) menunggu hasil dari robot B yang sedang dalam perjalanan menuju titik *TPr-1*. Seperti tampak dalam Gambar 10 (c) di bawah. Ketika robot B sudah berhenti di *TPr-1*, maka status misi masuk pada *loop-2*.

Pada *loop-2* ini proses yang dilakukan setiap robot akan diulangi, yaitu menghitung jarak ke titik target selanjutnya (*TPr-2*) sebagai titik target kedua. Seperti dalam Gambar 10 (d)



di bawah ini, posisi robot B berada di di titik  $TPr-1$ , sedangkan robot A dan C masih berada di titik  $SPr$  masing-masing. Perbedaannya adalah robot B akan menghitung jarak ke  $TPr-2$  dengan menganggap posisi koordinat saat ini ( $TPr-1$ ) sebagai posisi awal ( $Xo, Yo$ ). Sedangkan robot A dan C menghitung jarak dari titik  $SPr$  masing-masing.



Gambar 10 Mapping posisi robot (a) Titik Home, (b) Berhenti di titik  $SPr$ , (c) Ke titik  $TPr-1$ , (d) Ke titik  $TPr-2$ , (e) Ke titik  $TPr-3$

Perulangan ini akan berlangsung dengan alur yang sama untuk proses *looping* penghitungan jarak titik target seterusnya, sampai dengan misi dinyatakan selesai yaitu ketika *loop* sudah mencapai 3 kali putaran. Tampak seperti dalam Tabel 3 di bawah ini bahwa untuk *loop-2* dieksekusi oleh robot C (ke  $TPr-2$ ) dengan jarak 64 cm dan dilanjutkan *loop-3* oleh robot A (ke  $TPr-3$ ) dengan jarak 43 cm.

Tabel 3 Hasil data sampling robot handler ketika di titik target TPr

No	Waktu sampling (ms)	Loop 1 : Robot B				Loop 2 : Robot C				Loop 3 : Robot A			
		Pulsa Kanan	Pulsa Kiri	Jarak (cm)	Arah (deg)	Pulsa Kanan	Pulsa Kiri	Jarak (cm)	Arah (deg)	Pulsa Kanan	Pulsa Kiri	Jarak (cm)	Arah (deg)
0	0	0	0	0	332	1	0	0	122	0	1	0	140
1	250	2	0	0	331	4	4	2	122	1	1	0	140
2	500	6	0	2	330	7	7	4	122	1	5	2	139
3	750	9	1	3	330	12	11	7	121	6	12	6	140
4	1000	13	5	6	329	18	15	11	121	13	19	11	141
5	1250	17	9	9	329	25	16	13	124	22	28	17	141
6	1500	21	13	11	329	30	19	16	124	27	36	21	142
7	1750	24	17	13	328	36	24	20	124	27	43	24	140
8	2000	29	20	16	328	42	30	25	125	31	48	27	142
9	2250	34	20	18	329	50	38	30	126	36	54	31	140
10	2500	40	20	20	330	58	40	34	127	44	62	36	140
11	2750	45	21	23	330	66	41	36	128	51	70	41	141
12	3000	50	25	25	330	76	51	43	130	52	74	43	143
13	3250	53	28	27	330	86	61	50	127	52	75	43	143
14	3500	56	32	30	331	93	63	54	124	<b>52</b>	<b>75</b>	<b>43</b>	<b>143</b>
15	3750	61	36	33	332	97	63	55	124				
16	4000	65	40	36	333	102	68	59	125				
17	4250	68	41	37	334	105	72	61	124				
18	4500	73	41	39	334	107	74	62	123				
19	4750	77	41	41	334	<b>108</b>	<b>76</b>	<b>64</b>	<b>123</b>				
20	5000	<b>88</b>	<b>47</b>	<b>46</b>	<b>333</b>								

Dari Tabel 3 di atas, sampling data diambil setiap 250 ms direkam melalui terminal serial di komputer. Perbedaan jumlah sampling data setiap robot disebabkan oleh kecepatan laju robot dan jarak tempuh yang berbeda satu dengan lainnya[8]. Selama putaran sampai dengan *loop-3* pada sesi pengujian pertama ini didapatkan hasil pencapaian robot *handler* pada masing-masing *loop* hanya robot C yang tepat berada di titik target (*TPr-2-kuning*) pada *loop-2* yang berada di titik koordinat (200,15). Robot B pada *loop-1* ketika menuju *TPr-1* berhenti di titik (140,85) dengan selisih *error* sekitar 20 cm (2 petak). Robot A pada *loop-3* berhenti di titik (100,25) dengan *error* riil sekitar 15 cm (1,5 petak). Seperti tampak pada Tabel 4 di bawah ini merupakan hasil eksekusi target pada masing-masing *loop*.

Tabel 4 Hasil pengujian kesesuaian eksekusi titik target TPr

Loop	Target terpilih (x, y)	Robot A		Robot B		Robot C	
		Koordinat Riil (Xt,Yt)	Jarak <i>TPr</i> (cm)	Koordinat Riil (Xt,Yt)	Jarak <i>TPr</i> (cm)	Koordinat Riil (Xt,Yt)	Jarak <i>TPr</i> (cm)
<i>loop 1</i>	<i>TPr 1</i> : 140, 70	- diam -		140, 85	46	- diam -	
<i>loop 2</i>	<i>TPr 2</i> : 200, 10	- diam -		- diam -		200, 15	64
<i>loop 3</i>	<i>TPr 3</i> : 80, 30	100, 25	43	- diam -		- diam -	

### 3.4 Analisis pengujian algoritma PSO dengan pemodelan

Persamaan umum PSO dapat dibuat seperti dalam Persamaan 9 dan 10[10,11]. Pengujian robot dilakukan dengan mengambil sampling data setiap 250 *milisekon*. Data yang diperoleh dari hasil uji tersebut berupa jarak yang telah ditempuh oleh robot, data posisi sekaligus data jejak odometri robot (*dead reckoning*)[9,10] seperti tampak dalam Tabel 3 di atas.

$$V_i \text{ akhir} = \omega V_i \text{ awal} + C_1(pbest[ ] - X_i \text{ awal}) + C_2(gbest - X_i \text{ awal}) \dots \dots (9)$$

$$X_i \text{ akhir} = X_i \text{ awal} + V_i \text{ akhir} \dots \dots \dots (10)$$

Pada awal misi setelah proses inialisasi di titik *Home*, setiap robot akan bergerak menuju titik *SPr* masing-masing. Titik inilah yang berperan sebagai komponen *pbest*. Setiap robot akan merekam posisi *SPr* ini untuk selanjutnya “didiskusikan” dengan semua robot anggota. Setelah mendapatkan koordinat *SPr*, setiap robot akan menghitung jarak dirinya sendiri ke arah titik target *TPr* (*desentralized*). Kemudian setiap robot akan berdiskusi lagi tentang siapa robot yang memiliki jarak terdekat dengan target. Di sinilah proses *learning* oleh masing-masing robot akan berjalan. Pembelajaran untuk menentukan jarak yang terdekat dengan cara membandingkannya.

## 4. KESIMPULAN

Kesimpulan yang didapat dari hasil penelitian laporan tugas akhir ini adalah sebagai berikut.

1. Konsep dasar dari algoritma swarm telah berhasil diimplementasikan yaitu meliputi perilaku kolektif robot, sistem terdesentralisasi dan kemampuan *self-organizing*.
2. Pemodelan algoritma PSO berhasil diterapkan dengan indikasi berupa adanya proses *data sharing* dan proses *learning*.
3. Berhasil merancang dan mengimplementasikan protokol dan *frame* komunikasi untuk kebutuhan robot swarm, yaitu fungsi inialisasi (COMM\_INIT), fungsi *data sharing* (COMM\_DATA) dan fungsi *synchronize* (COMM\_STAT).
4. Penerapan pemodelan algoritma PSO memberikan pengaruh terhadap hasil penentuan robot *handler* yang lebih efisien waktu dan jarak, yaitu dalam durasi waktu total  $\pm 13,25$  detik dengan pengambilan jarak terdekat oleh robot *handler*.

## 5. SARAN

Saran yang diharapkan ke depan untuk dapat memperbaiki dan mengembangkan sistem yang ada adalah sebagai berikut.

1. Pemakaian komponen pendukung pengolah posisi robot perlu ketelitian lebih tinggi, seperti sensor *rotary encoder* dan pemakaian motor DC bertorsi tinggi (aktuator).
2. Komponen hardware utama (CPU) untuk memproses bagian komunikasi antar robot membutuhkan spesifikasi terpisah untuk keperluan *realtime data sharing* secara *streaming*.
3. Pemodelan algoritma PSO yang dibuat perlu diuji lagi dengan membuat simulasi animasi di komputer khususnya tentang *mapping* robot, misalnya melalui program simulasi “MATLAB” dan sejenisnya.
4. Algoritma untuk menjalankan fungsi *dead reckoning* yang akurat sangat diperlukan untuk mendukung ketersediaan data *pbest* dan *gbest* dalam PSO.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan segala bentuk dukungan terhadap penulis dalam menyelesaikan penelitian ini.

## DAFTAR PUSTAKA

- [1] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*, no. 1. Oxford university press, 1999.
- [2] Y. Liu and K. M. Passino, "Swarm Intelligence : Literature Overview," no. 614, 2015.
- [3] J. Kennedy and R. Eberhart, "Particle swarm optimization," *1995 IEEE Int. Conf. Neural Networks (ICNN 95)*, vol. 4, pp. 1942–1948, 1995.
- [4] F. T. Muijres and M. H. Dickinson, "Fly with a little flap from your friends," *Nature*, vol. 505, pp. 295–6, 2014.
- [5] M. G. C. A. Cimino, A. Lazzeri, and G. Vaglini, "Combining stigmergic and flocking behaviors to coordinate swarms of drones performing target search," *IISA 2015 - 6th Int. Conf. Information, Intell. Syst. Appl.*, 2016.
- [6] E. Papadopoulos and M. Misailidis, "On Differential Drive Robot Odometry with Application to Path Planning," *Control*, pp. 5492–5499, 2007.
- [7] R. K. Rasul, "Implementasi Sistem Gerak pada Autonomous Swarm Mobile Robot dengan Metode Odometri dan Algoritma Perilaku Semut untuk Menemukan Target," *Skripsi. Fak. MIPA, Universitas Gadjah Mada, Yogyakarta*, 2013.
- [8] A. Djoewahir, K. Tanaka, and S. Nakashima, "Adaptive PSO-based self-tuning PID controller for ultrasonic motor," *Int. J. Innov. Comput. Inf. Control*, vol. 9, no. 10, pp. 3903–3914, 2013.
- [9] H. Sa'adi, "Prototipe Robot Pengikut pada Implementasi Robot Swarm untuk Membentuk Formasi Pengikut Pemimpin." Institut Teknologi Sepuluh Nopember, Surabaya, 2010.
- [10] A. E. Turgut, C. Huepe, H. Çelikkanat, F. Gökçe, and E. Şahin, "Modeling phase transition in self-organized mobile robot flocks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5217 LNCS, pp. 108–119, 2008.
- [11] K. Mukhtar, "Konfigurasi Autonomous Routing untuk Jaringan Sensor Nirkabel Berbasis XBee," *IJEIS*, vol 2, no. 1, 2013.