

PENGEMBANGAN ALGORITME NICHING PARTICLE SWARM OPTIMIZATION UNTUK PENCARIAN TARGET PADA SISTEM MULTI-ROBOT

Siti Raehan*¹, Agus Buono², Medria Kusuma Dewi Hardhienata³

^{1,2,3}Departemen Ilmu Komputer, Fakultas Ilmu Pengetahuan dan Matematika Institut Pertanian Bogor
Email: ¹sitiraehan56@gmail.com, ²pudsha@gmail.com, ³medria.hardhienata@gmail.com
*Penulis Korespondensi

(Naskah masuk: 27 Januari 2020, diterima untuk diterbitkan: 19 Juli 2021)

Abstrak

Robot seringkali digunakan untuk mencari target, dalam hal ini target bisa korban, barang berbahaya dan tidak bisa dijangkau oleh manusia sehingga diganti menggunakan robot. Robot melakukan pencarian untuk menemukan target yang kemudian mengalokasikan diri ke target dengan asumsi bahwa targetnya dapat memancarkan sinyal. Permasalahan tersebut dipandang sebagai suatu masalah optimasi. Salah satu teknik yang dapat menyelesaikan masalah optimasi merupakan algoritme Particle Swarm Optimization (PSO). Masalah yang sering ditangani PSO sampai saat ini hanya sebatas masalah single-target. Beberapa masalah pada dunia nyata merupakan masalah multi-target, sehingga tidak dapat diselesaikan dengan algoritme PSO. Multi-target merupakan pencarian multi-robot untuk mengoptimasi pencarian target pada satu atau lebih titik optimum di dalam ruang pencarian. Masalah optimasi pada multi-target dapat diselesaikan menggunakan algoritme Niching Particle Swarm Optimization (NichePSO). Penelitian ini bertujuan untuk mengembangkan algoritme NichePSO untuk pencarian target pada sistem multi-robot. Pengembangan algoritme dilakukan dengan menggabungkan algoritme NichePSO dengan parameter robot e-puck yang merupakan kontribusi pertama pada penelitian ini. Kontribusi kedua adalah menerapkan algoritme penghindaran dan menggunakan teknik *reflecting* untuk robot yang keluar dari batas area pencarian. Pada studi ini membandingkan hasil performa antara algoritme NichePSO tanpa algoritme penghindaran dan dengan algoritme penghindaran, diuji dengan beberapa rintangan dalam lingkungan statis. Hasil penelitian menunjukkan bahwa pengembangan algoritme NichePSO pada tanpa algoritme penghindaran dan dengan algoritme penghindaran jauh berbeda dalam jumlah tabrakan tetapi tidak berbeda secara signifikan dalam waktu pencarian dan nilai fitness.

Kata kunci: *algoritme penghindaran rintangan, multi-robot, niching particle swarm optimization, pencarian target*

DEVELOPMENT OF NICHING PARTICLE SWARM OPTIMIZATION FOR TARGET SEARCH MULTI-ROBOT SYSTEM

Abstract

Robots are often used to find targets, in this case targets can be victims, dangerous goods and cannot be reached by humans so they are replaced using robots. The robot does a search to find a target which then allocates itself to the target assuming that the target can emit a signal. This problem is seen as an optimization problem. One technique that can solve optimization problems is the Particle Swarm Optimization (PSO) algorithm. The problem that is often handled by PSO to date is only limited to single-target problems. Some real-world problems are multi-target problems, so they cannot be solved by the PSO algorithm. Multi-target is a multi-robot search to optimize target search at one or more optimum points in the search space. The problem of optimization on multi-targets can be solved using the Niching Particle Swarm Optimization (NichePSO) algorithm. This study aims to develop a NichePSO algorithm for target search on multi-robot systems. The development of the algorithm is done by combining the NichePSO algorithm with the e-puck robot parameters which is the first contribution to this research. The second contribution is to apply avoidance algorithms and use reflecting techniques for robots that come out of the boundary of the search area. In this study comparing the performance results between the NichePSO algorithm without the avoidance algorithm and with the avoidance algorithm, tested with several obstacles in a static environment. The results showed that the development of the NichePSO algorithm without the avoidance algorithm and with the avoidance algorithm differed significantly in the number of collisions but did not differ significantly in search time and fitness values.

Keywords: *multi-robot, niching particle swarm optimization algorithm, obstacle avoidance algorithm, target search*

1. PENDAHULUAN

Robot secara umum dapat mempermudah pekerjaan manusia untuk melakukan tugas yang berbahaya dan beresiko. Tugas tersebut lebih mudah dilakukan dengan menggunakan multi-robot terutama untuk menemukan satu atau lebih target. Masalah pencarian target menggunakan multi-robot muncul di berbagai situasi pada dunia nyata seperti pencarian dan pembersihan ranjau (Gage 1995), pencarian dan penyelamatan korban yang terjebak dalam bangunan yang berbahaya (Luo *dkk* 2011), pencarian korban bencana (Kuswadi *dkk* 2017), pengalokasian agen untuk mencari di area yang berbahaya (Hardin *dkk* 2004; Meng & Gan 2008) dan penyelamatan korban dalam gedung yang hancur (Lopez, Suarez, & De La Rosa, 2003). Permasalahan pencarian target pada sistem multi-robot merupakan masalah pada robot untuk mencari target, dalam hal ini target bisa berupa korban, barang berbahaya dan tidak bisa dijangkau oleh manusia sehingga menggunakan robot. Robot melakukan pencarian untuk menemukan target yang kemudian mengalokasikan diri ke target dengan asumsi bahwa targetnya bisa memancarkan sinyal. Permasalahan tersebut dapat dipandang sebagai masalah optimasi. Salah satu teknik yang mampu menyelesaikan masalah optimasi adalah *Particle Swarm Optimization* (PSO).

PSO melakukan pencarian paralel secara langsung untuk meminimumkan fungsi yang diberikan dalam ruang multi-dimensi (Akat & Gazi 2008). PSO digunakan sebagai teknik untuk memecahkan masalah optimasi yang kompleks dan sulit (Kennedy & Eberhart 1995; Kennedy & Eberhart 2001; Clerc & Kennedy 2002). PSO pernah diterapkan di berbagai masalah termasuk pencarian target pada sistem multi-robot (Pugh & Martinoli 2007), (Pugh & Martinoli 2006) dan (Pugh, Martinoli, & Zhang, 2005). Hasil akhir PSO memusat di satu titik minimum atau maksimum karena menerapkan skema *single* target akan tetapi banyak masalah optimasi di dunia nyata membutuhkan solusi untuk masalah multi-target. Multi-target adalah pencarian sekumpulan robot untuk mengoptimasi pencarian target pada satu atau lebih titik optimum di dalam ruang pencarian. Multi-target merupakan pencarian robot untuk menemukan satu atau lebih target. Masalah multi-target dapat diselesaikan menggunakan metode *niching* (Li X 2010). Metode ini mampu membentuk suatu *subswarm* stabil dalam satu populasi untuk mencari beberapa solusi optimal (Li X 2010).

Penelitian tentang penggunaan metode *niching* dilakukan oleh Brits, Engelbrecht, & Van Den Bergh (2002), penelitian tersebut menjelaskan teknik yang mampu memperluas optimasi pada sejumlah partikel *single* target dengan menemukan beberapa solusi optimal yang dikenal dengan algoritme *Niching Particle Swarm Optimization* (NichePSO).

NichePSO menggunakan teknik pencarian *main swarm* dan *subswarm* yang terdiri dari kumpulan partikel di ruang pencarian untuk menemukan solusi multi target. Posisi partikel pada setiap iterasi merupakan kandidat solusi nilai optimum. Posisi partikel tersebut diperbarui terus menerus di setiap iterasi di ruang pencarian dengan memperbarui kecepatan partikel sampai menemukan beberapa posisi terbaik. Pada akhir iterasi semua partikel berkumpul di beberapa lokasi yang merupakan titik optimal. Teknik pencarian posisi titik optimal pada NichePSO dapat diterapkan di skenario pencarian target pada sistem multi-robot. Robot direpresentasikan oleh partikel, sedangkan posisi target digambarkan sebagai titik optimal yang dicari oleh partikel dalam algoritme NichePSO. Adapun penelitian yang menerapkan teknik *niching* dengan parameter robot seperti sensor dan batas kecepatan robot (Zhang, Gong, & Zhang, 2014) dan fungsi *inverse kinematic* pada robot untuk multimodal (Karla, Mahapatra, & Aggarwal, 2003). Pada penelitian ini, kami menggabungkan algoritme NichePSO dengan parameter robot e-puck yang merupakan kontribusi pertama pada penelitian ini.

Menurut Pugh & Martinoli (2007), terdapat beberapa perbedaan antara proses pencarian PSO dengan pencarian multi-robot seperti partikel memperbarui posisi disetiap iterasi pada waktu diskrit, tabrakan partikel, tidak ada batas area pencarian dan partikel tidak memiliki *noise*. Berdasarkan perbedaan tersebut maka pada penelitian ini kami memberikan kontribusi kedua yaitu merubah pergerakan partikel menjadi kontinu, mengintegrasikan algoritme penghindaran, menerapkan teknik *reflecting* (Li H *dkk* 2010), untuk robot yang keluar dari batas daerah pencarian dan robot memiliki *noise* pada sensor jarak yang digunakan.

Penelitian terkait mengenai masalah pencarian target pada sistem multi-robot dilakukan oleh Hardhienata, Merrick, & Ugrinovskii (2014), menggabungkan model komputasi motivasi pada algoritme *Guaranteed Convergence Particle Swarm Optimization* (GCPSO), yang memperkenalkan fungsi insentif dengan tiga profil motif yang diinisialisasi dari titik tunggal dan titik acak. Hasil menunjukkan bahwa jumlah *task* yang ditemukan meningkat namun masih belum mempertimbangkan *collision* (tabrakan). *Collision* dapat menyebabkan robot tidak bisa mencapai lokasi target. Pugh & Martinoli (2007), memanfaatkan teknik pencarian multi-robot untuk menemukan solusi yang optimal berdasarkan inspirasi. PSO dimodifikasi dengan meniru proses pencarian robot, parameter robot seperti jumlah robot e-puck dan jarak komunikasi digunakan namun masih *single*-target. Alfrianta, Hardhienata, & Priandana (2019), mengembangkan algoritme penghindaran rintangan pada algoritme PSO untuk pencarian target pada sistem multi-robot. Hasil penelitian Alfrianta, Hardhienata, & Priandana

(2019), menunjukkan bahwa tidak terjadi tabrakan robot, nilai fitness tidak jauh berbeda akan tetapi jauh berbeda pada waktu pencarian namun masih *single-target*.

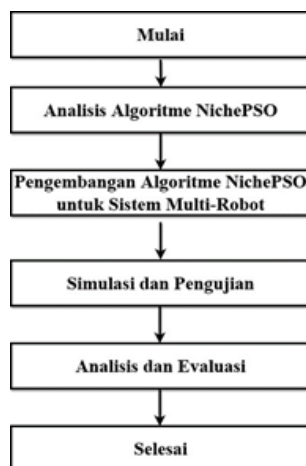
Penelitian lainnya dilakukan oleh Hereford & Siebold (2008), menggunakan algoritme *Physically-embedded Particle Swarm Optimization* (pePSO). Hasil simulasi sangat bagus dalam menemukan lokasi target di ruang pencarian yang rumit. Algoritme pada penelitian Hereford dan Siebold diuji dengan membiarkan robot menemukan titik paling terang di dalam suatu ruangan. Kekurangan pada penelitian Hereford dan Siebold masih belum ada parameter robot dan masih *single target*. Robot yang digunakan berupa robot mitEBot, masih dikendalikan oleh radio. Doctor, Venayagamoorthy, & Gudise (2004), menggunakan metode PSO pada penelitiannya dan memberikan hasil kinerja yang bagus dalam pencarian baik pada *unimodal* atau *multimodal*. Kekurangan Doctor, Venayagamoorthy, & Gudise (2004) belum memikirkan skalabilitas PSO dengan jumlah robot yang besar dan parameter robot tidak ada.

Hasil simulasi menunjukkan bahwa pengembangan algoritme NichePSO pada tanpa algoritme penghindaran dan dengan algoritme penghindaran mampu meningkatkan kinerja algoritme NichePSO. Hal ini dibuktikan dengan jumlah tabrakan yang terjadi pada algoritme NichePSO dengan penghindaran sangat jauh berbeda dengan jumlah tabrakan pada NichePSO tanpa penghindaran. Selain itu, peningkatan penemuan target dan waktu pencarian dari kedua metode tersebut secara signifikan tidak jauh berbeda.

2. METODE PENELITIAN

Tahapan Penelitian

Tahapan penelitian pengembangan algoritme NichePSO untuk pencarian target pada sistem multi-robot sesuai Gambar 1.



Gambar 1. Tahapan penelitian

2.1. Analisis Algoritme NichePSO

NichePSO adalah penyempurnaan model Parsopoulos & Vrahatis (2001), dimana beberapa *subswarm* dihasilkan dari *main swarm* untuk mencari beberapa solusi untuk masalah multi-target di ruang pencarian. NichePSO menerapkan teknik pencarian *main swarm* dan *subswarm* untuk menyeimbangkan kemampuan eksplorasi dari setiap partikel yang hanya dapat berinteraksi dengan tetangga terdekatnya.

Main swarm disebar di ruang pencarian secara merata dengan menggunakan *cognition only model* dimana partikel bergerak berdasarkan pengetahuan pribadi dari masing-masing partikel. Menurut Løvbjerg, Rasmussen, & Krink (2001), *subswarm* mampu meningkatkan keragaman *swarm* yang dapat menghindari konvergen. NichePSO menerapkan algoritme GCPSO untuk mencegah *subswarm* berhenti. GCPSO telah diuji mampu mengatasi masalah stagnasi secara otomatis sebelum menuju global minimum untuk menghindari konvergen. Ide dasar GCPSO adalah penambahan partikel yaitu pencarian posisi terbaik lokal sama dengan posisi terbaik global saat ini (Patel, Sharma, & Gupta, 2013).

2.2. Pengembangan Algoritme NichePSO untuk Sistem Multi-Robot

Algoritme NichePSO dikembangkan seperti setiap langkah partikel diubah seperti robot nyata yang melakukan pencarian untuk menemukan target. Perbedaan algoritme yang dikembangkan terletak pada inialisasi dengan mengintegrasikan parameter robot e-puck, menggerakkan robot menggunakan algoritme penghindaran rintangan dan robot keluar dari batas lingkungan pencarian menggunakan teknik *reflecting*.

2.3. Integrasi Parameter Robot E-Puck pada Algoritme NichePSO

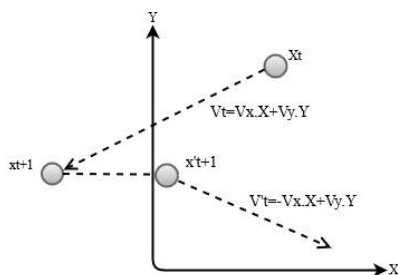
Parameter yang digunakan terdiri dari parameter algoritme NichePSO dan parameter robot e-puck. Nilai parameter NichePSO dideskripsikan pada Tabel 1 sesuai penelitian Brits, Engelbrecht, & Van Den Bergh (2002), yang memberikan pengaruh terhadap hasil kinerja simulasi.

Tabel 1. Parameter NichePSO

Parameter	Nilai
Inersia Inialisasi (ω_0)	0.7
Inersia Iterasi Terakhir (ω_1)	0.2
Komponen Kognitif (c_1)	1.2
Komponen Sosial (c_2)	1.2
Kecepatan Maksimum (v)	5
Delta (δ)	0.0001
E_Sigma (σ)	3
Miu (μ)	0.001
Rho (ρ)	1
F_c	5
S_c	15

2.4. Algoritme Penghindaran Batas Area Pencarian

NichePSO memperbarui lokasi partikel menuju x_{t+1}^n secara *teleport* namun robot melakukan pembaruan lokasi x_{t+1}^n dengan cara menggerakkan robot dan x_{t+1}^n merupakan lokasi berikutnya yang diharapkan dapat di capai. Lokasi robot selanjutnya dihitung dan diperiksa kalau ada posisi robot yang keluar dari batas area pencarian maka posisi robot ditarik ke dalam menggunakan teknik *reflecting* (Li H dkk 2010) berdasarkan Gambar 2. Nilai posisi dan kecepatan robot setelah ditarik ke dalam diubah menjadi 0 dan dinegatifkan. Penghindaran robot pada teknik ini kalau ada hambatan dipinggir area pencarian maka robot akan keluar untuk menghindari hambatan tersebut namun tidak akan melakukan pencarian diluar dan robot pasti kembali masuk ke area pencarian lagi. Batas area dilakukan untuk mencegah robot keluar dari batas area lingkungan.



Gambar 2. Teknik *reflecting*

2.5. Algoritme untuk Penghindaran Rintangan

Proses *avoidance* yaitu menggerakkan robot ke lokasi berikutnya sambil menjalankan algoritme penghindaran. Pergerakan robot menuju lokasi tujuan dihitung menggunakan sudut menuju lokasi tujuan, besarnya sudut robot harus berputar ke lokasi tujuan (θ_t^i). θ_t^i , hasil pengurangan antara *heading* robot yang sekarang dengan sudut lokasi tujuan. *Heading* robot berarti kepala robot menghadap sudut berapa (Cartesian). Kepala robot menghadap atas 90 derajat, bawah 270 derajat, kanan 0 derajat dan kiri 180 derajat. Penghitungan *heading* robot sudah di *build in* pada simulator Webots. Sudut lokasi tujuan dihitung pada Persamaan (1) (Ukil dkk 2011).

$$\tan \alpha = \frac{y}{x}, \alpha = \tan^{-1} \left(\frac{y}{x} \right) \tag{1}$$

dimana α sudut lokasi tujuan, x dan y adalah posisi robot pada sumbu x dan sumbu y koordinat cartesian.

Pergerakan robot pada saat bergerak menuju lokasi tujuan akan menghadapi rintangan yang dapat mengakibatkan terjadinya tabrakan baik dengan robot atau rintangan. Untuk menghindari terjadinya tabrakan maka robot harus melakukan penghindaran

dan mencari posisi yang tidak ada rintangan. Ben-Ari dan Mondada (2018), memperkenalkan suatu algoritme penghindaran rintangan yang mampu mencegah terjadinya tabrakan antara robot dengan hambatan atau robot lainnya dengan menggunakan teknik penelusuran dinding (*wall following*), dapat dilihat di Gambar 3. Algoritme ini menggunakan sensor jarak atau sensor sentuh untuk mendeteksi rintangan. Sensor jarak yang digunakan untuk mendeteksi rintangan dari sudut sebelah kanan merupakan sensor kanan dan rintangan di depan dideteksi oleh sensor depan. Ada tiga kondisi penghindaran yang dipertimbangkan pada penelitian Ben-Ari dan Mondada (2018), jika ada rintangan di depan maka robot berputar ke kiri sebesar 90 derajat, jika sudut rintangan di sebelah kanan maka robot berputar ke kanan 90 derajat dan yang terakhir jika rintangan di sebelah kanan maka robot maju terus. Jenis rintangan berupa rintangan statis yang berbentuk dinding.

```

1 while belum mencapai target do
2   | if target terdeteksi then
3   |   | robot bergerak ke target
4   | else if dinding di depan then
5   |   | robot putar heading ke kiri 90 derajat
6   | else if titik sudut dinding sebelah kanan then
7   |   | robot putar heading ke kanan 90 derajat
8   | else if dinding di sebelah kanan then
9   |   | maju terus
10  | else
11  |   | maju terus
12  | end
13end
    
```

Gambar 3. Algoritme *Wall-Following* (Ben-Ari dan Mondada 2018)

Penghindaran rintangan di penelitian Ben-Ari dan Mondada (2018) hanya menggunakan sensor depan dan kanan berbeda pada penelitian ini menggunakan sensor depan kanan, depan kiri, sudut kanan dan sudut kiri yang dapat mendeteksi rintangan dari segala arah seperti yang pernah dilakukan oleh Alfrianta, Hardhienata, & Priandana (2019). Rintangan tersebut dideteksi menggunakan sensor jarak yang terdiri dari delapan sensor jarak yaitu ps0, ps1, ps2, ps3, ps4, ps5, ps6 dan ps7 dengan masing-masing jangkauan 0.04 m. Ps0 dan ps1 mendeteksi rintangan sebelah kanan depan maka robot berputar ke kiri. Ps6 dan ps7 mendeteksi rintangan dari kiri depan sehingga robot berputar ke kanan. Ps2 mendeteksi rintangan dari sudut kanan dan robot maju sambil belok kiri sedikit dan ps5 mendeteksi rintangan dari sudut kiri dan robot maju sambil belok kiri sedikit. Besar derajat perputaran robot untuk menghindari rintangan yang ada sampai sensor jarak pada robot yang mendeteksi rintangan tidak aktif.

2.6. Simulasi dan Pengujian

Tujuan simulasi untuk menguji keefektifan algoritme dalam menyelesaikan masalah pencarian target pada sistem multi-robot. Fungsi uji yang

digunakan merupakan fungsi Himmelblau, Griewank dan Rastrigin. Luas area pencarian $-5 < x < 5$ dan $-5 < y < 5$ dengan jumlah robot 20 yang diinisialisasi secara acak. Fungsi uji yang digunakan diuraikan secara rinci pada Tabel 2. Jumlah target mendeskripsikan banyaknya target yang harus dicari oleh robot. Maksimum nilai fitness menunjukkan seberapa dekat robot dengan lokasi target. Semakin besar nilai fitness robot maka robot dikatakan mampu menemukan lokasi target.

Tabel 2. Fungsi uji, jumlah target dan maksimum fitness

Nama Fungsi	Fungsi Uji	Jumlah Target	Maksimum Fitness
Himmelblau	$f(x,y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2$	4	200
Griewank	$f(x,y) = \frac{x^2 + y^2}{4000} - \cos\left(x \cos\left(\frac{y}{\sqrt{2}}\right) + 1\right)$	5	2
Rastrigin	$f(x,y) = 100 - (20 + (x^2 - 10\cos(2\pi x^2)) + (y^2 - 10\cos(2\pi y^2)))$	1	100

Untuk mendapatkan hasil yang bagus, dilakukan simulasi sebanyak 10 kali pada masing-masing fungsi uji dengan maksimum iterasi 100. Maksimum iterasi ditetapkan sebagai kondisi partikel berhenti bergerak. Semakin kecil iterasi dapat menyebabkan robot harus berhenti bergerak sebelum sampai lokasi target dan semakin besar iterasi mengakibatkan penambahan kompleksitas komputasi yang tidak penting dengan waktu yang lama (Engelbrecht 2007). Tabrakan robot dengan rintangan terjadi apabila dinding robot dan rintangan saling berpotongan. Tabrakan antar robot terjadi jika jarak antara robot satu dengan yang lainnya kurang dari dua kali radius robot yaitu 0.07 m. Setiap simulasi pergerakan robot diuji rintangan statis yang diletakkan secara *random* di dalam ruang pencarian. Jenis dan parameter rintangan dapat dilihat di Tabel 3. Hasil algoritme yang sudah dikembangkan dibandingkan dengan algoritme yang sama namun perbedaannya terletak pada algoritme penghindaran rintangan dengan parameter yang sama. Setiap simulasi dihitung jumlah tabrakan, waktu pencarian dan nilai fitness terbaik.

Tabel 3. Jenis dan parameter rintangan

Parameter	Jumlah	Ukuran	
		Maksimum	Minimum
Lingkaran	10	0.35 m	0.1 m
Persegi	10	0.35 m	0.1 m
Persegi Panjang	10	0.35 m	0.1 m

2.7. Analisis dan Evaluasi

Hasil simulasi pada pencarian robot e-puck di jadikan sebagai bahan analisis terhadap performa algoritme dalam menyelesaikan pencarian target yang ada. Jika terjadi kesalahan atau hasil yang

kurang baik maka dianalisis penyebab kesalahan tersebut. Evaluasi dilakukan dengan menggunakan *performance metrics*. Ada tiga variabel yang dievaluasi seperti jumlah tabrakan, waktu pencarian dan nilai fitness terbaik yang dijelaskan pada uraian berikut ini:

1. Jumlah tabrakan

Jumlah tabrakan digunakan untuk mengukur hasil performa algoritme penghindaran yang digunakan pada NichePSO tanpa algoritme penghindaran dan dengan algoritme penghindaran. Jumlah tabrakan dihitung berdasarkan Persamaan (2).

$$\bar{Y} = \frac{1}{I} \sum_{i=1}^I \bar{Y}_i \quad (2)$$

I menandakan jumlah simulasi, \bar{Y}_i jumlah tabrakan antar robot dan hambatan pada simulasi ke I .

2. Rata-rata waktu pencarian

Waktu pencarian pada penelitian ini merupakan waktu yang di butuhkan robot untuk sampai ke lokasi target yang dihitung pada setiap iterasi pada semua simulasi. Waktu yang di butuhkan robot untuk bergerak tanpa rintangan 2 detik dan jika ada rintangan maka robot membutuhkan waktu untuk menghindari hambatan. Rata-rata waktu pencarian robot dihitung pada Persamaan (3).

$$\bar{U} = \frac{1}{I} \sum_{i=1}^I \bar{U}_i \quad (3)$$

dimana \bar{U}_i merupakan jumlah waktu pencarian disimulasi I .

3. Nilai fitness terbaik

Robot di katakan dapat menemukan target apabila nilai fitness terbaik robot mencapai maksimum. Nilai fitness sebagai penanda bahwa robot sudah menemukan lokasi target. Penghitungan nilai rata-rata fitness terbaik pada Persamaan (4).

$$\bar{O} = \frac{1}{I} \sum_{i=1}^I \bar{O}_i \quad (4)$$

\bar{O}_i merupakan jumlah nilai fitness terbaik robot pada simulasi I .

3. HASIL DAN PEMBAHASAN

3.1. Jumlah Tabrakan

Robot bergerak di dalam ruang pencarian untuk memperoleh lokasi terbaru pada setiap iterasi dengan mengecek apakah posisi berikutnya akan menghadapi tabrakan atau tidak yang dideteksi oleh sensor jarak pada. Faktor yang paling mempengaruhi dalam pencarian robot merupakan adanya rintangan, tabrakan antar robot dan tabrakan robot dengan

rintangan. Robot mengalami tabrakan apabila terjadi kontak poin lebih kecil atau sama dengan dua kali. Robot akan melakukan penghindaran apabila mendeteksi rintangan namun tidak berarti merubah arah tujuannya. Tabel 4 dan 5 menunjukkan bahwa algoritme NichePSO tanpa algoritme penghindaran jumlah tabrakan secara signifikan jauh berbeda bila di bandingkan NichePSO dengan penghindaran pada selang kepercayaan sebesar 95%. Pada tanpa penghindaran jumlah tabrakan yang terjadi paling banyak fungsi Griewank. Hal ini disebabkan karena fungsi Griewank memiliki jumlah target paling banyak sehingga robot mengalami tabrakan semakin banyak. Beda halnya pada fungsi Rastrigin jumlah tabrakan paling sedikit karena jumlah target yang harus dicari robot hanya satu, robot mengalami stagnasi dan terjebak di antara rintangan.

Tabel 4. Jumlah tabrakan tanpa algoritme penghindaran

Simulasi	Jumlah Tabrakan		
	NichePSO tanpa Algoritme Penghindaran		
	H	G	R
1	4386	3567	5011
2	4331	2886	935
3	3742	6101	4187
4	4108	4342	3705
5	5896	4205	1458
6	4986	7082	1452
7	4344	7822	3007
8	4208	3918	1507
9	4439	5049	4186
10	4398	4098	7377
Rata-rata	4483.8±418.2	4907±1143.53	3282.5±1450.13

Keterangan: (H) Himmelblau (G) Griewank (R) Rastrigin

Tabel 5. Jumlah tabrakan dengan algoritme penghindaran

Simulasi	Jumlah Tabrakan		
	NichePSO dengan Algoritme Penghindaran		
	H	G	R
1	945	939	842
2	1344	791	708
3	1410	920	1826
4	1115	1294	1698
5	1271	710	1393
6	1256	888	509
7	1115	1037	786
8	1285	847	655
9	1207	401	1029
10	1117	1371	529
Rata-rata	1206.5±97.22	919.8±198.62	997.5±324.47

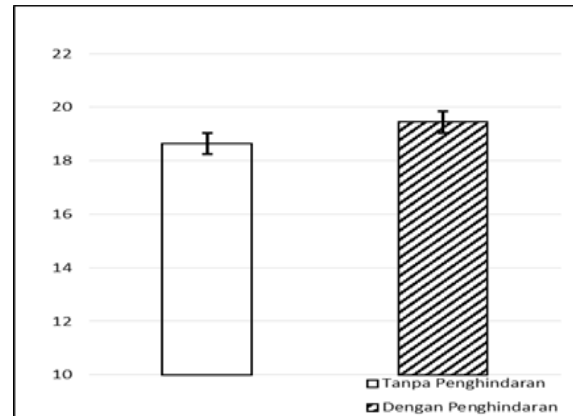
NichePSO dengan penghindaran terdapat beberapa jumlah tabrakan. Tabrakan tersebut disebabkan oleh sensor jarak pada setiap robot memiliki *noise* dan akibatnya robot dapat mendeteksi rintangan dengan jarak yang berbeda. Nilai sensor jarak pada setiap robot sudah di

tetapkan hanya dapat mendeteksi hambatan dengan jarak 0.04 m. Namun, dengan adanya *noise* mengakibatkan robot dapat mendeteksi rintangan dengan jarak kurang atau lebih dari 0.04 m.

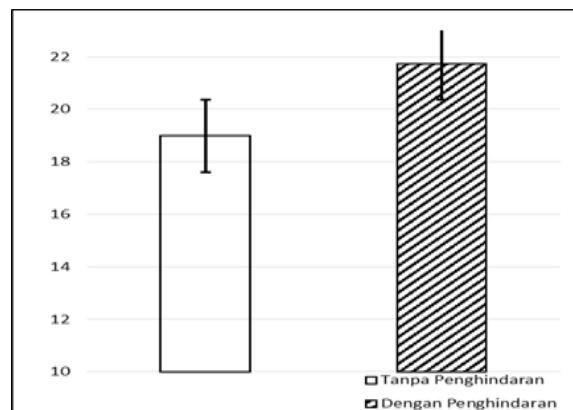
Jumlah tabrakan dengan penghindaran paling banyak terjadi pada fungsi Himmelblau dan paling sedikit pada fungsi Griewank. Hal ini terjadi karena pada fungsi Himmelblau pencarian robot berhasil menemukan semua target bahkan tidak ada robot yang berhenti sebelum sampai di lokasi target. Sedangkan, fungsi Griewank pencarian robot berhasil menemukan empat target dari total lima target dan ada satu target yang tidak ditemukan. Fungsi Rastrigin jumlah tabrakan lebih banyak daripada Griewank padahal jumlah targetnya lebih sedikit. Ini merupakan penyebab dari robot yang mengalami stagnasi dan terjebak oleh rintangan. Robot tetap bisa bergerak untuk menemukan lokasi target sehingga tabrakan tetap terjadi. Jumlah tabrakan dengan penghindaran lebih sedikit karena robot sudah menyesuaikan diri dengan algoritme penghindaran yang ada sedangkan tanpa penghindaran, jumlah tabrakan lebih banyak karena robot belum menerapkan algoritme penghindaran.

3.2. Rata-rata Waktu Pencarian

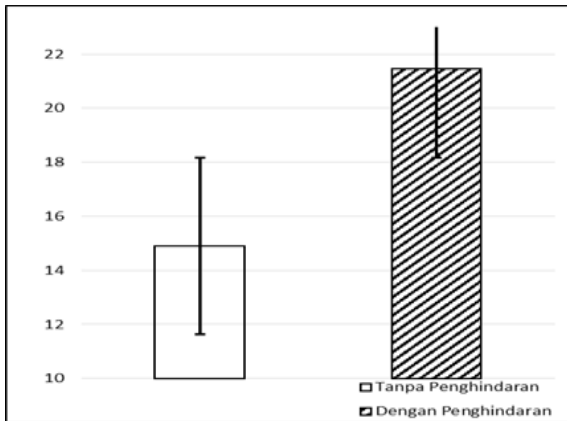
Gambar 4, 5 dan 6 memperlihatkan waktu pencarian semua robot, baik dengan penghindaran dan tanpa penghindaran sama-sama memberikan hasil yang sangat baik pada semua fungsi pengujian.



Gambar 4. Waktu pencarian target untuk fungsi Himmelblau



Gambar 5. Waktu pencarian target untuk fungsi Griewank



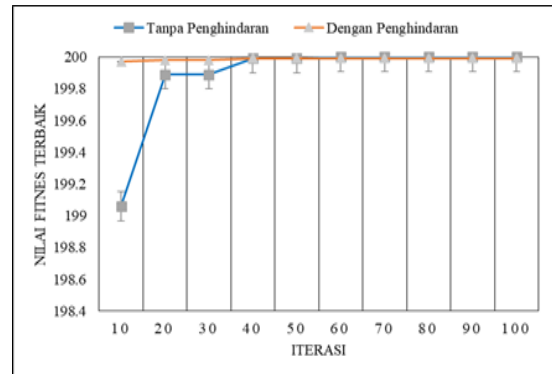
Gambar 6. Waktu pencarian target untuk fungsi Rastrigin

Waktu pencarian kedua metode secara signifikan tidak jauh berbeda di fungsi Himmelblau dan Griewank namun fungsi Rastrigin sedikit jauh perbedaannya pada selang kepercayaan 95%. Robot yang berkumpul sebelum sampai di lokasi target mengakibatkan robot bergerak terus melakukan pencarian sehingga membutuhkan waktu yang lebih banyak. Waktu yang lebih sedikit pada dengan penghindaran disebabkan robot membutuhkan waktu lebih untuk menghindari rintangan. Pengamatan seperti ini bisa terjadi karena pada tanpa penghindaran belum menerapkan algoritme penghindaran yang ada sehingga tidak membutuhkan sumber daya perhitungan tambahan. Terbukti dengan waktu tersebut algoritme NichePSO baik tanpa penghindaran atau dengan penghindaran dalam pencarian target cukup bagus dan efisien bahkan pada masalah multi-target.

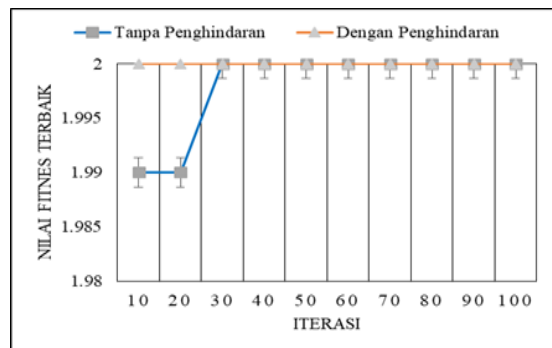
3.3. Nilai Fitnes Terbaik

Pada Gambar 7, 8 dan 9 nilai fitnes dengan penghindaran lebih baik dari iterasi awal sampai iterasi terakhir dibandingkan tanpa penghindaran dengan selang kepercayaan 95%. NichePSO dengan penghindaran lebih konsisten dengan nilai standar deviasi yang lebih kecil dan akurasi yang lebih baik bila dibandingkan tanpa penghindaran. Fungsi Himmelblau dan Rastrigin menunjukkan pada iterasi 20 dan 30 di tanpa penghindaran nilai fitnes tidak ada perubahan yang signifikan namun kembali meningkat sedangkan fungsi Griewank di iterasi 10 dan 20. Ini disebabkan robot terjebak di lokal minima, terjebak di antara rintangan dan tidak konvergen. Nilai fitnes pada NichePSO dengan penghindaran tidak ada robot satupun yang terjebak di lokal minima karena robot tidak dapat tumpang tindih sehingga menyebabkan robot tidak dapat berkumpul semua di satu titik. Hanya ada satu robot yang berada di lokal minima sementara robot lainnya berada di sekitarnya mengeksplor area lainnya untuk menemukan fitnes yang lebih tinggi. Rata-rata nilai fitnes terbaik pada fungsi Himmelblau, Griewank dan Rastrigin tanpa penghindaran adalah sebesar 197.33 ± 17.78 ,

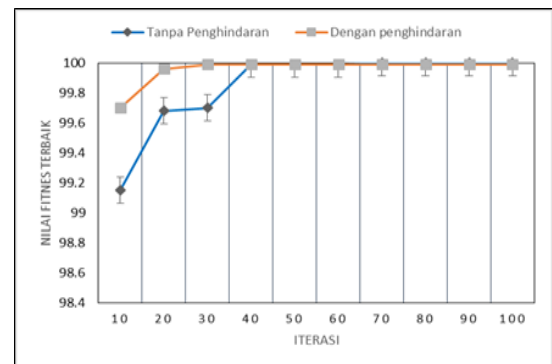
1.91 ± 0.25 , 93.34 ± 8.45 dan dengan penghindaran sebesar 199.52 ± 0.74 , 1.91 ± 0.22 , 91.57 ± 11.21



Gambar 7. Fitnes terbaik fungsi Himmelblau



Gambar 8. Fitnes terbaik fungsi Griewank



Gambar 9. Fitnes terbaik fungsi Rastrigin

4. KESIMPULAN

Berdasarkan hasil penelitian ini bahwa pengembangan algoritme NichePSO baik tanpa penghindaran atau dengan penghindaran mampu menemukan target secara efektif dan efisien untuk pencarian target pada sistem multi-robot. Jumlah tabrakan algoritme NichePSO tanpa algoritme penghindaran rintangan di fungsi Himmelblau adalah sebesar 4483.8 ± 418.2 , Griewank 4907 ± 1143.53 dan Rastrigin 3282.5 ± 1450.13 . Sedangkan, dengan algoritme penghindaran rintangan jumlah tabrakan pada fungsi Himmelblau adalah sebesar 1206.5 ± 97.22 , Griewank 919.8 ± 198.62 dan Rastrigin 997.5 ± 324.47 . Hasil penelitian ini menunjukkan bahwa secara umum

algoritme NichePSO dengan algoritme penghindaran rintangan mampu menurunkan jumlah tabrakan dibandingkan algoritme NichePSO tanpa algoritme penghindaran rintangan. Dalam penelitian ini juga dihitung nilai rata-rata waktu pencarian target oleh robot. Rata-rata waktu pencarian target oleh robot tanpa penghindaran rintangan pada fungsi Himmelblau, Griewank dan Rastrigin masing-masing adalah sebesar 18.64 ± 8.25 menit, 18.98 ± 8.08 menit dan 14.90 ± 2.55 menit sedangkan dengan penghindaran rintangan adalah sebesar 19.65 ± 1.43 menit, 21.73 ± 3.34 menit dan 21.45 ± 2.05 menit. Hal ini menunjukkan meskipun digunakan algoritme penghindaran rintangan pada NichePSO, waktu pencarian robot tidak signifikan berbeda dengan tanpa algoritme penghindaran rintangan. Hal ini membuktikan bahwa dengan menerapkan algoritme penghindaran rintangan pada algoritme NichePSO dapat menghasilkan performa pencarian yang baik dan mampu memenuhi kriteria yang diharapkan. Penelitian ini juga mengukur nilai rata-rata fitness terbaik yang diperoleh oleh robot pada fungsi yang diusulkan. Secara umum ketiga fungsi yang diusulkan dapat mencapai nilai optimum yang membuktikan bahwa robot mampu menemukan lokasi target.

UCAPAN TERIMA KASIH

Terima kasih penulis ucapkan atas kesempatan yang diberikan dari Lembaga Pengelola Dana Pendidikan (LPDP) melalui program beasiswa Afirmasi untuk daerah 3T sehingga hasil penelitian ini dapat dipublikasikan.

DAFTAR PUSTAKA

- AGASSOUNON, W., MARTINOLI, A., EASTON, K., 2004. Macroscopic modeling of aggregation experiments using embodied agents in teams of constant and time-varying sizes. *Autonomous Robots*, 17(2–3), p. 163–191.
- AKAT, S.B., GAZI, V., 2008. Particle swarm optimization with dynamic neighborhood topology three neighborhood strategies and preliminary results. *IEEE Swarm Intelligence Symposium St.*
- ALFRIANTA, A., HARDHIENATA, M.K.D., PRIANDANA, K., 2019. Pengembangan algoritme penghindaran rintangan pada algoritme particle swarm optimization untuk sistem multi-robot pencari target [skripsi]. Bogor (ID): Institut Pertanian Bogor.
- AME, J.M., MILLOR, J., HALLOY, J., SEMPO, G., 2006. Collective decision making based on individual discrimination capability in pre social insects. InProc. 9th International Conference on Simulation of Adaptive Behavior (SAB), p. 713-724.
- BEN-ARI, M., MONDADA, F., 2018. Elements of Robotics. Springer.
- BENI, G., 2004. From swarm intelligence to swarm robotics. InProc. of the SAB International Workshop on Swarm Robotics, p. 3342:1–9. doi: 10.1007/978-3-540-30552-1_1.
- BONABEAU, E., DORIGO, M., THERAULAZ, G., 1999. Swarm intelligence from natural to artificial systems. Oxford University Press. New York.
- BRITS, R., ENGELBRECHT, A.P., VAN DEN BERGH, F., 2002. A niching particle swarm optimizer. Department of computer science, University of Pretoria: Pretoria.
- CHRISTOPHER, M., CIANCI, RAEMY, X., PUGH, J., MARTINOLI, A., 2006. Communication in a swarm of miniature robots: the e-puck as an educational tool for swarm robotics. *International Workshop on Swarm Robotics*, 4433, p. 103-115.
- CLERC, M., KENNEDY, J., 2002. The particle swarm explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1), p. 58–73.
- DOCTOR, S., VENAYAGAMOORTHY, G.K., GUDISE, V.G., 2004. Optimal pso for collective robotic search applications. In *Congress on Evolutionary Computation*, 2, p. 1390–1395.
- EBERHART, R.C., KENNEDY, J., 1995. A new optimizer using particle swarm theory. In *Sixth International Symposium on Micromachine and Human Science*, p. 39–43.
- GAGE, D.W., 1995. Many robot MCM search systems. proceedings of autonomous vehicles in mine countermeasures symposium, p. 9–55. Monterey:California.
- GHOSH, P., ZAFAR, H., DAS, S., ABRAHAM, A., 2011. Hierarchical dynamic neighborhood based particle swarm optimization for global optimization. *IEEE Congress of Evolutionary Computation*.
- HARDHIENATA, M.K.D., MERRICK, K.E., UGRINOVSKII, V., 2014. Task allocation under communication constraints using motivated particle swarm optimization. *IEEE congress on evolutionary computational*, p. 3135-3134.
- HARDIN, X., CUI, R., RAGADE, J., GRAHAM, ELMAGHRABY, A., 2004. A modified particle swarm algorithm for robotic mapping of hazardous environments. in Proc. of Automation World Congress, 17(1), p. 31–36.
- HEREFORD, J., SIEBOLD, M., 2008. Multi robot search using a physically embedded particle swarm optimization. *International Journal of Computational Intelligence Research*, 4(2), p. 197–209.

- KARLA, P., MAHAPATRA, P.B., AGGARWAL, D.K., 2003. On the solution of multimodal robot inverse kinematic functions using real-coded genetic algorithms. Proceedings of IEEE on systems, man and cybernetics, Washington (USA): United States.
- KENNEDY, J., EBERHART, R.C., 1995. Particle swarm optimization. Proceedings of the IEEE international conference on neural networks, 4, p. 1942–1948.
- KENNEDY, J., EBERHART, R.C., 2001. Swarm Intelligence. San Mateo California: Morgan Kaufmann.
- KHAN, K., ASHOK, S., 2012. A comparison of BA, GA, PSO, BP and LM for training feedforward neural networks in e-learning context. Intelligent Systems and Applications, 7, p. 23-29.
- KUSWADI, S., SIGIT, R., SULISTIJO, I.A., TAMARA, M.N., 2017. Disaster swarm robot development: On going project. International Conference on Electrical Engineering and Informatics, p. 45–50. doi: 10.1109/ICELTICS.2017.8253258.
- LI, H., HE, X., XIE, X., LI, L., ZHOU, J., LI, X., 2010. A New Boundary Condition for Particle Swarm Optimization. Journal of Convergence Information Technology, 5(9), p. 215-221. doi: 10.4156/jcit.vol5.issue9.22.
- LI, X., 2010. Niching without niching parameters: Particle swarm optimization using a ring topology. IEEE Transactions on Evolutionary Computation, 14(1), p. 150–169.
- LOPEZ, B., SUAREZ, DE LA ROSA, J.L., 2003. Task allocation in rescue operations using combinatorial auctions. Artificial Intelligence Research and Development, p. 233–243.
- LØVBJERG, M., RASMUSSEN, T.K., KRINK, T., 2001. Hybrid Particle Swarm Optimizer with Breeding and Subpopulations. In Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco: California.
- LUO, C., ESPINOSA, A.P., PRANANTHA, D., GLORIA, A.D., 2011. Multi robot search and rescue. IEEE International Symposium on Safety, Security and Rescue Robotics, p. 296–301.
- MENG, Y., GAN, J., 2008. Self adaptive distributed multi task allocation in a multi robot system. In IEEE Congress on Evolutionary Computation, p. 398–404.
- MOHCENE, B., MOUSSA, S., KAHLA, S., 2014. Maximum power point tracking for photovoltaic systems: A comparison between GA and PSO approaches. International Journal of Hydrogen Energy, 10(6), p. 393-398.
- PASSARO, A., STARITA, A., 2008. Particle swarm optimization for *multimodal* functions a clustering approach. Journal of Artificial Evolution and Applications.
- PARSOPOULOS, K.E., VRAHATIS, M.N., 2001. Modification of the Particle Swarm Optimizer for Locating All the Global Minima. Artificial Neural Nets and Genetic Algorithms, p. 324-327. doi: 10.1007/978-3-7091-6230-9_80.
- PATEL, P.K., SHARMA, V., GUPTA, K., 2013. Guaranteed convergence particle swarm optimization using personal best. International Journal of Computer Applications, 73(7), p. 0975-8887.
- PUGH, J., MARTINOLI, A., ZHANG, Y., 2005. Particle swarm optimization for unsupervised robotic learning. Proceedings IEEE Swarm Intelligence Symposium, Pasadena: USA.
- PUGH, J., MARTINOLI, A., 2006. Multi-Robot Learning with Particle Swarm Optimization. Proceedings of the fifth international joint conference on Autonomous agents and multiagent system.
- PUGH, J., MARTINOLI, A., 2007. Inspiring and modeling multi-robot search with particle swarm optimization. In IEEE Swarm Intelligence Symposium, p. 332–339.
- UKIL, A., SHAH, V.H., DECK, B., 2011. Fast computation of arctangent functions for embedded applications: A comparative analysis. IEEE International Symposium on Industrial Electronics.
- VAN DEN BERGH, F., ENGELBRECHT, A.P., 2002. A new locally convergent particle swarm optimizer. IEEE International Conference on Systems, Man and Cybernetics, 3.
- VAN DEN BERGH F., 2002. An analysis of particle swarm optimizers [Thesis]. South Africa: University of Pretoria.
- ZHANG, J., GONG, D., ZHANG, Y., 2014. A niching PSO-based multi-robot cooperation method for locating odor sources. Neurocomputing, p. 123:308-317.

Halaman ini sengaja dikosongkan