

IMPLEMENTASI ALGORITMA *PARTICLE SWARM OPTIMIZATION* UNTUK PENENTUAN POSISI STRATEGIS AGENT PADA SIMULASI ROBOT SEPAK BOLA DUA DIMENSI

Galih Hermawan

Jurusan Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, UNIKOM
Jl. Dipati Ukur No. 112-116 Bandung
E-mail : galih.hermawan@yahoo.co.id

ABSTRAK

Robot sepak bola merupakan perpaduan antara olah raga, teknologi robotika, dan *multi agent system*. Untuk mencapai tujuan, selain membutuhkan kecerdasan individu, juga menuntut kemampuan kerja sama antar individu. Posisi robot ketika bermain mempengaruhi kemampuan robot dalam bekerja sama dan memilih aksi yang sesuai. Dalam tulisan ini akan disajikan hasil penelitian kami dalam penerapan algoritma *particle swarm optimization* (PSO) untuk menentukan posisi robot ketika bermain sepak bola. Hasil pengujian pada simulasi RoboCup Soccer dua dimensi menunjukkan bahwa tim robot sepak bola yang menggunakan algoritma PSO memiliki performa bermain lebih baik ketimbang tim sebelum menggunakan algoritma PSO.

Kata kunci : simulasi robot sepak bola, robocup soccer, *particle swarm optimization*, pemosisian *agent*, *multi agent system*.

1. PENDAHULUAN

Penelitian di bidang *artificial intelligence* (AI) atau inteligensia buatan dan teknologi robotika terus dilakukan untuk menghasilkan produk yang dapat digunakan oleh manusia untuk membantu aktivitas kesehariannya. Pengembangan dan penggunaan robot dapat ditemukan di berbagai bidang, seperti: dunia industri, pertahanan dan kemandirian nasional, rumah tinggal, perawatan medis, penyelamatan, penjelajahan laut dalam, dan lain-lain. Penggunaan banyak robot atau sering dikenal dengan istilah *swarm robotics* dalam sebuah sistem menghadapi persoalan yang lebih rumit ketimbang robot tunggal, seperti: kerja sama, koordinasi, dan negosiasi di antara robot-robot tersebut. Salah satu contoh *swarm robotics* adalah robot sepak bola, yang dibangun selain untuk mempromosikan sains dan teknologi, juga untuk membuat robot-robot yang dapat saling bekerja sama dalam mencapai tujuan yang telah ditetapkan.

Pada tahun 1993, para peneliti dari Jepang bernama Kitano, Asada, dan Kuniyoshi mengembangkan program penelitian bernama Robot J-League, yang kemudian berkembang secara internasional dan mengubah namanya menjadi Robot World Cup atau disingkat RoboCup [1]. Salah satu kompetisi yang diadengakan oleh RoboCup adalah simulasi robot sepak bola dua dimensi dengan menyediakan *tools* berupa seperangkat perangkat lunak simulator bernama *RoboCup Soccer Server*.

Salah satu tantangan yang dihadapi dalam pengembangan robot sepak bola virtual yang terdiri dari 11 *agents* adalah penentuan posisi *agent* dalam lapangan ketika bermain, dimana untuk selanjutnya berpengaruh pada mekanisme seleksi aksi dari *agents* tersebut.

Swarm Intelligence (SI) adalah salah satu teknik kecerdasan buatan yang berlandaskan kepada perilaku kolektif (*collective behaviour*) pada sistem yang terdesentralisasi dan dapat mengatur dirinya sendiri (*self-organizing*). Salah satu metoda yang digunakan untuk penentuan posisi *agent* dalam sistem SI adalah *particle swarm optimization* (PSO). Dalam hal ini, penulis juga ingin mengembangkan sebuah tim sepak bola virtual dengan menggunakan algoritma PSO untuk menentukan posisi (*agent positioning*) dan pergerakan *agent* (pemain sepak bola virtual) ketika menyerang ke daerah lawan dan bertahan di daerah sendiri.

2. ISI PENELITIAN

Pada bagian ini akan dipaparkan mengenai RoboCup, simulator RoboCup Soccer, arsitektur *agent* dan formasi tim, desain penentuan posisi *agent*, dan hasil uji.

2.1 RoboCup

RoboCup [2] adalah sebuah federasi yang dibangun sejak tahun 1993 untuk mengakomodasi para peneliti utamanya pada robot sepak bola. Salah satu kegiatannya adalah dengan cara menyelenggarakan turnamen piala dunia setiap tahun. Anggota-anggota Federasi RoboCup adalah para peneliti yang aktif melakukan penelitian dalam

bidang ini dan mereka adalah wakil-wakil dari universitas-universitas atau perusahaan-perusahaan yang berkepentingan dan berkontribusi di bidang ini.

Dengan semakin berkembangnya jumlah peneliti dan jumlah negara-negara yang ikut bergabung, maka dibentuk jugalah komite-komite lokal yang bertugas untuk mempromosikan *events* yang berkaitan dengan RoboCup di area geografis setempat.

Federasi RoboCup telah menetapkan tujuan-tujuan dan suatu jadwal penelitian. Penetapan tujuan ini dimaksudkan untuk mendesak perkembangan terkini (*the state-of-the-art*) selanjutnya, sehubungan dengan *test-beds* yang telah diformalkan. Artinya bahwa sebenarnya tujuan utama yang paling penting dari RoboCup adalah memajukan tingkat teknologi masyarakat di segala bidang, dan tujuan pragmatisnya adalah mencapai hal berikut ini:

“By mid-21st century, a *tim of fully autonomous humanoid robot soccer pemsains shall win a soccer game, complying with the official rules of the FIFA, against the winner of the most recent World Cup*” [3].

Hingga saat ini, federasi RoboCup internasional telah mempromosikan kompetisi/liga robot sepak bola sebagai berikut.

- a) *RoboCup Soccer*, terdiri atas:
 - *Simulation League*, terdiri dari:
 - *2D Simulation League*,
 - *3D Simulation League*,
 - *3D Development*.
 - *Small Size Robot League*;
 - *Middle Size Robot League*;
 - *Standard Platform League*;
 - *Humanoid League*, terdiri dari:
 - *Kid Size (30-60 cm height)*,
 - *Teen Size (100-160 cm height)*.
- b) *RoboCup Rescue*, terdiri atas:
 - *Rescue Simulation League*,
 - *Rescue Robot League*.
- c) *RoboCup Junior*, terdiri atas:
 - *Soccer Challenge*,
 - *Dance Challenge*,
 - *Rescue A Challenge*,
 - *Rescue B Challenge*.

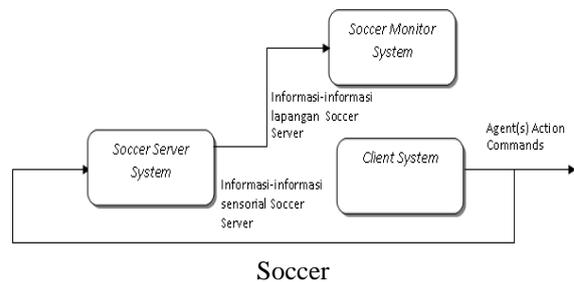
Adapun topik yang disampaikan dalam tulisan ini berfokus pada simulasi dua dimensi.

2.2 Simulator RoboCup Soccer

Simulator yang telah disediakan untuk liga RoboCup Soccer terdiri dari 3 komponen utama, yaitu: Soccer Server, Soccer Monitor dan Log Player. Sebuah simulasi pertandingan sepak bola dilakukan dalam bentuk *client-server*. Soccer Server menyediakan sebuah domain (sebuah lapangan sepakbola virtual) yang mensimulasikan semua gerakan-gerakan obyek dalam domain ini, dan

mengontrol sebuah game sepak bola sesuai aturan-aturan tertentu.

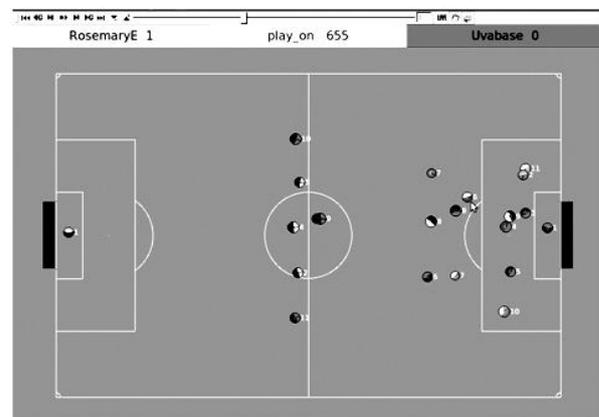
Gambar 1. Skema diagram blok simulator RoboCup



Secara garis besar, berdasarkan Gambar 1 di atas ada tiga buah sistem besar yang terlibat dalam sebuah pertandingan sepak bola virtual, yaitu: Sistem Soccer Server, Sistem Soccer Monitor, dan Sistem Soccer Client.

Selanjutnya tiga komponen utama tersebut, dijabarkan sebagai berikut.

- a) Soccer Server. Soccer Server adalah sebuah file *executable* yang mengatur hubungan antara klien-klien (monitor, klien kesebelasan), dan menghasilkan lingkungan virtual bagi klien-klien. Soccer Server juga mengatur semua peraturan permainan dan protokol-protokol komunikasi antar klien. Soccer Server ini adalah properti RoboCup, dan konfigurasinya tidak dapat diubah lagi.
- b) Soccer Monitor. Soccer Monitor adalah sebuah file *executable* yang menyediakan tampilan visual yang berupa grafis lapangan sepak bola dan tampilan visual para *agent*/pemain di atas lapangan. Informasi-informasi mengenai posisi dan aksi yang dilakukan oleh pemain berasal dari Soccer Server. Seperti halnya Soccer Server, Soccer Monitor ini adalah properti RoboCup, dimana konfigurasinya sudah permanen dan tidak dapat diubah-ubah lagi.



Gambar 2. Tampilan soccer monitor

c) Soccer Client.

Soccer Client adalah sebuah file *executable* yang menghadirkan satu set pemain sepakbola (*soccer players*) untuk bermain sepak bola dalam lingkungan Soccer Server. Soccer Client telah banyak dikembangkan oleh banyak universitas, misalnya: Carnegie Melon University, Tsing Hua University, Universiteit van Amsterdam, dan lain-lain. Masing-masing universitas tersebut telah mengimplementasikan berbagai kecakapan pada agen-agensya, sedemikian rupa sehingga para agen dapat memainkan pertandingan sepak bola melawan agen dari Soccer Client lain secara *autonomous*. Soccer Client adalah properti dari masing-masing pengembang, dan bukan properti RoboCup. Pengembang dapat dengan bebas mengembangkan sistem kliennya masing-masing selama sistem tersebut dapat terhubung dan berkomunikasi dengan Soccer Server.

Untuk menerapkan aturan-aturan pada permainan, simulator memiliki sebuah modul wasit yang mengontrol pertandingan. Wasit buatan (*artificial refferee*) ini dapat mendeteksi situasi-situasi trivial seperti misalnya ketika sebuah tim melakukan gol, atau ketika bola keluar lapangan. Wasit juga menerapkan aturan *offside*, mengontrol mode permainan (*kick off, corner kick, dan seterusnya*), dan menghentikan pertandingan ketika babak pertama dan atau babak kedua telah berakhir.

2.3 Arsitektur Agent dan Formasi Tim

Salah satu Soccer Client yang kode sumbernya bersifat terbuka dan bebas dikembangkan adalah UvA Trilearn [4] yang dikembangkan oleh R. De Boer dan J. R. Kok [5] pada tahun 2001. Berikutnya, Safreni Candra Sari [6] melalui tesisnya pada tahun 2010 mengembangkan mekanisme seleksi aksi yang ada pada UvA Trilearn dan memberi nama timnya dengan sebutan Rosemary.

a) Arsitektur Agent

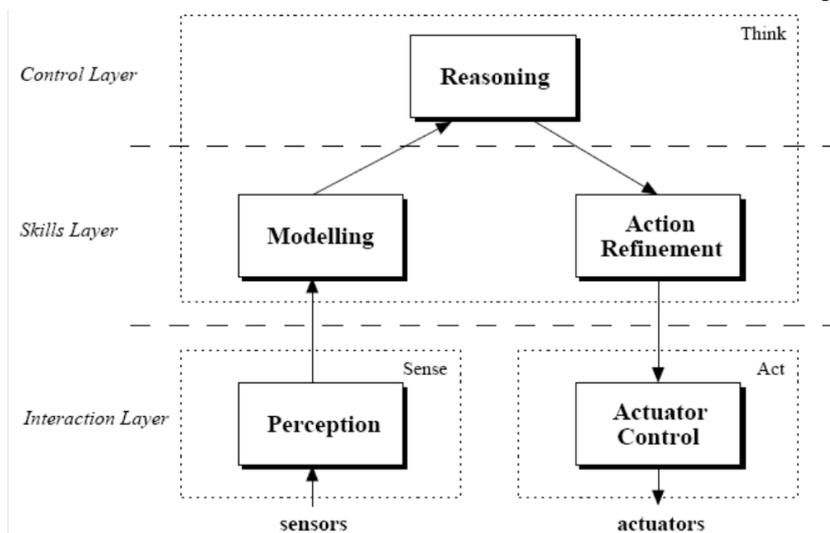
Arsitektur *agent* yang ditunjukkan pada Gambar 3 adalah hirarki, yaitu tersusun atas tiga lapisan pada berbagai tingkat abstraksi. Lapisan bawah adalah *Interaction Layer* yang menangani interaksi dengan lingkungan simulasi *soccer server*. Lapisan ini menyembunyikan detail *soccer server* sebanyak mungkin dari lapisan lain.

Lapisan tengah adalah *Skills Layer* yang menggunakan fungsionalitas yang ditawarkan oleh *Interaction Layer* untuk membuat suatu model abstrak dari dunia dan untuk menerapkan berbagai macam keterampilan (*skill*) dari setiap *agent*, seperti: mengumpan bola, menghadang laju lawan, dan lain-lain.

Lapisan tertinggi adalah *Control Layer* yang berisi komponen penalaran dari sistem. Dalam lapisan ini, tindakan terbaik yang dipilih dari *Skills Layer* adalah tergantung dari keadaan dunia atau lingkungan saat ini dan strategi bermain tim.

Selama pertandingan berlangsung, persepsi memasuki sistem melalui *Interaction Layer* dan naik ke atas melalui *Skills Layer* dimana digunakan untuk memperbaharui model dunia dari *agent*. Informasi keadaan dari dunia yang terkini untuk selanjutnya digunakan oleh *Control Layer* sebagai alasan untuk mengambil kemungkinan aksi yang terbaik. Aksi yang dipilih oleh *Control Layer* ini kemudian bekerja dalam *Skills Layer* untuk menentukan perintah aktuator yang sesuai. Perintah ini kemudian dieksekusi oleh modul kontrol aktuator pada *Interaction Layer*.

Agents UvA Trilearn memiliki kapabilitas berpersepsi, berfikir, dan bertindak. Pengaturan untuk arsitektur *agent* diberikan sedemikian rupa sehingga ketiga aktifitas ini dapat berlangsung secara bersamaan atau paralel.



Gambar 3. Arsitektur agent tim UvA Trilearn 2001

b) *Formasi Tim*

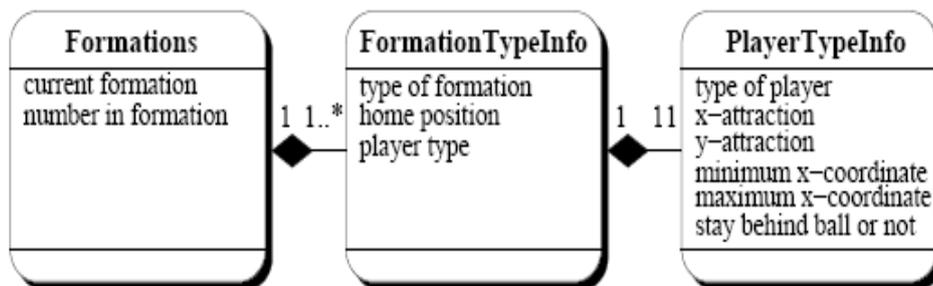
Kerjasama antar *agents* dapat diperoleh dengan menggunakan formasi tim. Formasi dibuat untuk pemetaan *agent*, untuk situasi-situasi pasif dimana para *agent* tidak mempunyai peran aktif dalam permainan. Hal ini diberikan saat *agents* tidak berada dekat bola. Pada umumnya, mekanisme pemetaan *agent* menggunakan metoda yang disebut dengan *Situation Based Strategic Positioning*^[16]. Ini berarti bahwa formasi ditentukan oleh satu set peran-peran yang terdiri dari tipe-tipe pemain dan *home position* di lapangan. Dalam formasi tersebut, seorang *agent* diberikan tugas sebuah peran yang harus dilakukan selama formasi ini digunakan. Dalam situasi pasif, seorang *agent* menentukan posisi strategisnya di lapangan dengan cara menghitung jumlah bobot dari *home position*-nya dan posisi bola aktual dimana yang juga memiliki peran sebagai titik tarik. Perhatikan bahwa daya tarik terhadap bola ditentukan berbeda-beda untuk tipe-tipe pemain yang berbeda. *Home positions* yang bermacam-macam dapat digunakan untuk menentukan pola-pola pemetaan antara *teammate* dimana daya tarik terhadap bola dalam pola-pola tersebut diterapkan di lapangan pada posisi tertentu.

mensimulasikan secara nyata pergerakan sekumpulan burung yang bagus dan sukar diprediksi. Setiap individu dalam *swarm* (kerumunan) direpresentasikan oleh sebuah vektor dalam ruang pencarian multidimensi. Vektor tersebut juga memiliki sebuah vektor yang digunakan untuk menentukan pergerakan partikel atau *agent* selanjutnya yang disebut *velocity vector* (vektor kecepatan). Algoritma PSO juga digunakan untuk menentukan bagaimana cara memperbaharui kecepatan dari sebuah partikel. Setiap partikel memperbaharui kecepatannya berdasarkan kecepatan terkini dan posisi terbaik yang telah dijelajah sejauh ini, dan juga berdasarkan posisi terbaik global yang dijelajah oleh *swarm* [8].

b) *Penentuan Posisi Strategis Agent*

Penentuan posisi strategis pada UvA Trilearn dipengaruhi beberapa faktor, yaitu: *home position*, faktor ketertarikan terhadap bola, posisi bola, kebolehan berada di belakang bola atau tidak, jarak tempuh sumbu x minimum dan maksimumnya.

Algoritma penentuan posisi strategis atau yang disebut dengan fungsi *GetStrategicPosition* yang terdapat pada tim UvA Trilearn dapat dilihat pada Gambar 5.



Gambar 4. UML class diagram untuk kelas-kelas yang berhubungan dengan formasi dan penentuan posisi

Implementasi mekanisme penentuan posisi dan formasi pada UvA Trilearn seperti terlihat pada Gambar 4 di atas adalah melibatkan tiga obyek kelas, yaitu *Formations*, *FormationTypeInfo*, dan *PlayerTypeInfo*.

2.4 Desain Penentuan Posisi Agent

Bahasan dalam bagian ini mencakup *particle swarm optimization*, penentuan posisi strategis *agent*, dan factor-faktor penentu pergerakan.

a) *Particle Swarm Optimization*

Particle swarm optimization (PSO) didesain dan dikenalkan oleh Eberhart dan Kennedy [7]. PSO merupakan algoritma pencarian berbasis populasi yang berdasar pada simulasi kelakuan dari burung-burung, lebah atau sekumpulan ikan. Algoritma ini pada dasarnya dimaksudkan untuk

```

getStrategicPosition( $\vec{q}$ )
// $\vec{q}$ =current ball position

get home position ( $p_x, p_y$ )
and ball attraction factors
(attr_x, attr_y)
get x-coordinate range [ $min_x, max_x$ ]
and boolean BehindBall
( $s_x, s_y$ ) = ( $p_x, p_y$ ) + (attr_x, attr_y) .
( $q_x, q_y$ )
if BehindBall == true and  $s_x > q_x$  then
     $s_x = q_x$ 
end if
if  $s_x > max_x$  then
     $s_x = max_x$ 
else if  $s_x < min_x$  then
     $s_x = min_x$ 
end if
return ( $s_x, s_y$ )
    
```

Gambar 5. Algoritma penentuan posisi strategis *agent*

Persamaan yang digunakan untuk memperbaharui kecepatan (*velocity*) yang mempengaruhi pergerakan *agent* adalah

$$\vec{v}_i(t) = W\vec{v}_i(t-1) + C_1r_1(\vec{x}_{pbest} - \vec{x}_i(t-1)) + C_2r_2(\vec{x}_{gbest} - \vec{x}_i(t-1)) \quad (1)$$

dengan $\vec{v}_i(t)$ adalah jarak dari sebuah *agent* atau partikel ke-*i* yang harus ditempuh pada waktu ke-*t*, *W* adalah faktor inersia, *C*₁ dan *C*₂ adalah konstanta faktor *cognitive* dan sosial, *r*₁ dan *r*₂ adalah bilangan acak antara 0 dan 1, \vec{x}_{pbest} adalah posisi *agent* terbaik yang pernah dilaluinya, \vec{x}_{gbest} adalah posisi *agent* terbaik yang pernah dilaluinya dibanding dengan kesemua *agent*.

Sedangkan, untuk menentukan posisi baru *agent* adalah mengikuti formula

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (2)$$

dengan $\vec{x}_i(t)$ adalah posisi tujuan baru dari *agent* atau partikel ke-*i* pada waktu ke-*t*.

Perlu diperhatikan bahwa penentuan posisi terbaik adalah diperoleh dengan menghitung jarak terdekat antara *agent* dengan gawang lawan ketika menyerang, dan dengan gawang sendiri ketika bertahan.

```

Bangkitkan posisi inisial swarm secara acak
Repeat
  For setiap partikel i do
    If f(xi) < f(pi) Then pi ← xi
    pg = min(pkeseluruhan)
    Update velocity (1)
    Update posisi (2)
  End For
Until kriteria pemutusan tercapai
    
```

Gambar 6. Algoritma PSO

Pada Gambar 6 di atas merupakan algoritma penggunaan persamaan PSO sebelumnya untuk menentukan posisi *agent*.

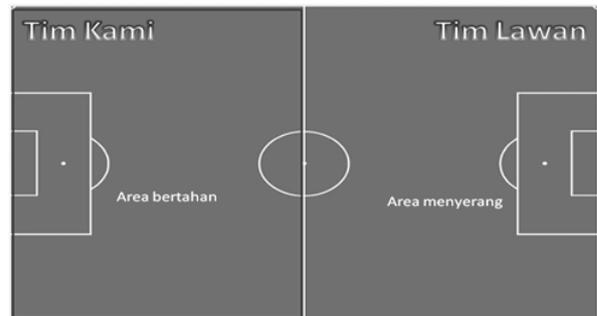
c) Faktor-Faktor Penentu Pergerakan

Penggunaan algoritma PSO dalam penentuan posisi *agent* akan memperhatikan beberapa faktor adalah sebagai berikut.

- Status penguasaan bola. Status penguasaan bola digunakan untuk mengetahui apakah bola dekat atau sedang dibawa oleh salah satu pemain kita atau bukan. Hal ini digunakan untuk memastikan bahwa hanya pemain tanpa bola saja yang bergerak sesuai dengan algoritma PSO. Begitu juga digunakan untuk mengetahui apakah bola dikuasai lawan atau tim kita.
- Area. Informasi area yang sedang ditempati *agent* digunakan untuk memilih melakukan serangan atau bertahan.

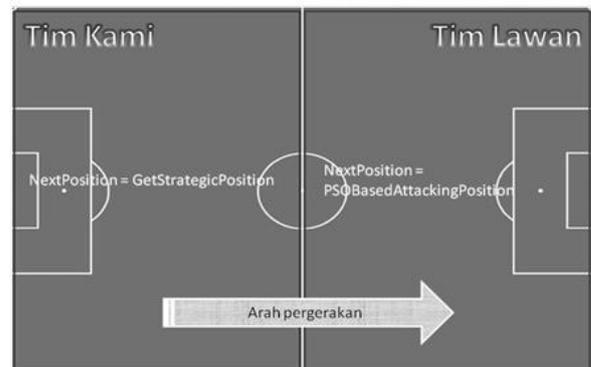
- Status permainan. Pergerakan pemain secara normal hanya bisa dilakukan saat bola sedang bergulir atau *play on*. Jika bola sedang tidak bergulir, seperti: *free kick*, *goal kick*, dan seterusnya, maka algoritma PSO tidak digunakan.
- Arah bergulirnya bola. Arah bergulirnya bola hanya digunakan ketika tim sedang diserang oleh tim lawan.

Pada Gambar 7 ditunjukkan sebuah pembagian lapangan virtual menjadi dua bagian, yaitu diasumsikan sebelah kiri (disebut area bertahan) tempat dimana tim kita berada, dan sebelah kanan untuk tim lawan (disebut area menyerang).



Gambar 7. Pembagian lapangan virtual

Untuk keperluan menyerang, skema penyerangan dapat dilihat pada Gambar 8.



Gambar 8. Penggunaan algoritma PSO ketika dalam keadaan menyerang

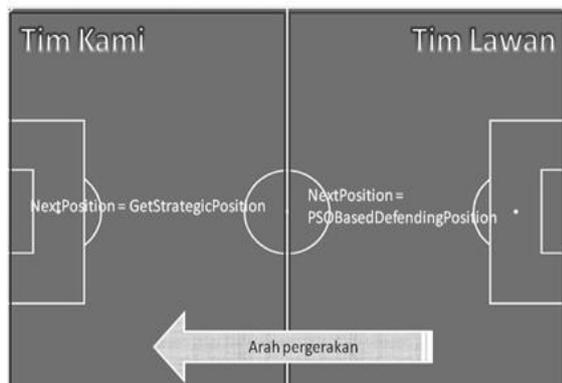
Algoritma yang digunakan adalah sebagai berikut.

```

if not isBallKickable then
  if isBallInOurPossesion then
    if NOT isDeadBallUs then
      if PosisiAgent di area bertahan then
        NextPos=GetStrategicPosition
      else
        NextPos=PSOBasedAttackingPosition
      end if
    end if
  end if
end if
    
```

Gambar 9. Algoritma penentuan posisi *agents* saat menyerang

Sedangkan untuk keperluan bertahan, skema pergerakan dapat dilihat pada Gambar 10.



Gambar 10. Penggunaan algoritma PSO ketika dalam keadaan bertahan

Adapun algoritma penentuan posisi yang digunakan untuk bertahan dapat dilihat pada Gambar 11.

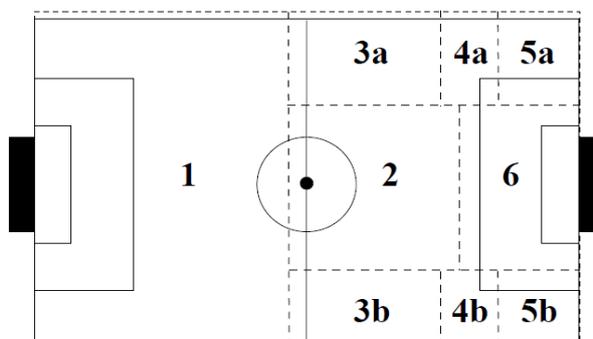
```

if not isBallKickable then
  if isBallInOurPossession then
    if NOT isDeadBallUs then
      if PosisiAgent di area menyerang then
        if isBallHeadingToGoal then
          NextPos=PSOBasedDefendingPosition
        else
          NextPos=GetStrategicPosition
        end if
      end if
    end if
  end if
end if
end if
    
```

Gambar 11. Penggunaan algoritma PSO ketika dalam keadaan menyerang

2.5 Implementasi dan Hasil Pengujian

Tujuan semula penerapan algoritma PSO adalah untuk memperoleh posisi strategis sehingga membantu memperbesar peluang pemain dalam mencetak gol dan bergerak cepat ke daerah bertahan untuk memperkecil peluang lawan dalam mencetak gol.



Gambar 12. Pembagian area lapangan untuk mekanisme seleksi aksi

Selain penempatan *agent* pada posisi strategis, mekanisme seleksi aksi juga harus tepat. Dalam hal ini, salah satu hal yang berpengaruh pada mekanisme seleksi aksi adalah area. Diadopsi dari tim UvA Trilearn yang menggunakan *Situation Based Strategic Positioning*, maka area dibagi ke dalam beberapa bagian.

Untuk mencapai tujuan awal, algoritma PSO akan diterapkan pada tim Rosemary dan tim karakterisasi [9].

Beberapa rencana pengujian yang dilakukan adalah untuk:

- menguji algoritma PSO secara fungsional dalam penentuan posisi *agent* ketika menyerang dan bertahan,
- penentuan parameter-parameter dalam PSO yang paling optimal untuk diterapkan pada tim,
- mempertandingan tim Rosemary dan tim karakterisasi yang sudah menggunakan PSO dengan tim lain.

Tabel 1. Daftar Rencana Pengujian secara Fungsional

Kondisi	Tahap Pengujian	Deskripsi
Menyerang	Pemain bergerak maju	Seluruh <i>agents</i> , kecuali kiper, bergerak maju ke arah wilayah pertahanan tim lawan
	Pemain tengah bagian sayap melebar dan mendekat ke gawang lawan	<i>Agents</i> yang berposisi sebagai pemain tengah, khususnya wing midfielder, bergerak melebar mendekati garis pinggir kemudian semakin mendekati ke gawang lawan
	Pemain depan mencari posisi di dekat gawang lawan	Pemain depan berusaha masuk ke daerah 6, sedekat mungkin dengan gawang lawan
	Pemain tengah mengumpan bola	Pemain tengah yang sedang membawa bola akan berusaha mengumpan ke pemain yang paling dekat dengan gawang lawan, terutama pemain depan
	Pemain depan menendang bola ke gawang lawan	Pemain depan ketika berada di wilayah 6, setelah menerima bola umpan langsung menendangnya ke gawang lawan
Diserang	Pemain bergerak mundur	Seluruh <i>agents</i> bergerak mundur ke daerah pertahanan tim
	Pemain mencari posisi dekat dengan gawang	Pemain akan mencari posisi dekat dengan gawang
	Pemain melakukan marking	Pemain berusaha mengejar dan menghalangi pemain lawan yang sedang membawa bola dan berusaha merebutnya

Dari pengujian yang telah dilakukan menunjukkan bahwa rencana pengujian fungsional seperti yang tertera pada Tabel 1 berhasil dilakukan.

Beberapa parameter yang digunakan dalam formula PSO, seperti: W (faktor inersia), C_1 (faktor *cognitive*), dan C_2 (faktor sosial) telah dilakukan pengujian dengan kriteria sebagai berikut.

Tabel 2. Daftar Parameter yang Diujikan

Nomor Uji	Faktor Inersia (w)	C_1	C_2
1	1	2	2
2	0,7	3	1
3	0,4	3	1

Pengujian dilakukan dengan mempertandingkan tim tesis melawan tim Iran sebanyak masing-masing 5 kali.

Tabel 3. Hasil Uji

NO	GM	GK	M	S	K	N
1	8	3	3	2	0	11
2	7	30	0	0	5	0
3	8	16	2	0	3	6

Keterangan:

1. GM: Gol memasukkan;
2. GK: Gol kemasukan;
3. M: Jumlah menang;
4. S: Jumlah seri;
5. K: Jumlah kalah;
6. N: Nilai (menang dapat poin 3, seri dapat poin 1, dan kalah tidak dapat poin).

Berdasarkan hasil pengujian pada Tabel 3, diperoleh nilai parameter optimal adalah parameter pada nomor uji 1, dimana nilai W , C_1 , C_2 secara berturut-turut adalah 1, 2, 2.

Sedangkan untuk pengujian performa tim secara menyeluruh, rencana pengujian melibatkan 6 tim, di antaranya adalah:

1. UvA Trilearn,
2. Rosemary;
3. Rosemary dengan PSO (disingkat RosemaryPSO),
4. Karakterisasi (diberi nama GalihRio),
5. Iran,
6. dan Helios (juara RoboCup 201).

Hasil klasemen dari lima kali pertandingan untuk setiap dua tim adalah sebagai berikut.

Tabel 4. Klasemen Akhir Pertandingan

No	Tim	Main	M	S	K	GM	GK	N
1	Helios 2010	25	25	0	0	470	3	75
2	GalihRio	25	16	2	7	53	98	50
3	Iran 2010	25	15	2	8	79	46	47
4	Rosemary PSO	25	12	0	13	55	155	36
5	Rosemary	25	3	2	20	7	179	11
6	UvA Trilearn	25	0	2	23	12	195	2

3. PENUTUP

Hasil desain penentuan posisi *agent* menggunakan algoritma PSO yang diterapkan pada tim yang belum menggunakan PSO secara umum memiliki performa lebih baik ketimbang tim asli yang belum menggunakannya.

Namun jika dibandingkan dengan tim Helios sang juara, performa tim masih kalah jauh. Hal ini disebabkan oleh kemampuan *agent* secara individu (*basic skill*) yang masih tertinggal jauh dibandingkan dengan kemampuan *agents* pada tim Helios. Sehingga strategi penentuan posisi *agent* tidak terlalu berpengaruh ketika melawan tim Helios. Dalam kasus ini, *basic skill* pemain seperti: *dribbling*, *passing*, dan lain-lain perlu ditingkatkan.

Pemanfaatan PSO dapat lebih ditingkatkan performanya dengan memperhatikan faktor-faktor lain seperti: stamina, jumlah halangan di depan pemain, kecepatan lari pemain, kecepatan lari bola, posisi teman lainnya, posisi lawan, dan lain-lain.

DAFTAR PUSTAKA

- [1]. Chen, M., et. al., *RoboCup Soccer Server Users Manual for Soccer Server Version 7.07 and later*, The RoboCup Federation, 2003.
- [2]. *RoboCup Regulation and Rules*, <http://www.robocup.org/about-robocup/regulations-rules/>, 18 Mei 2011, 16:00 WIB.
- [3]. H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup : the robot world cup initiative," in *Proc. of IJCAI-95 Workshop on Entertainment and AI/Alife*, pp. 19-24, 1995.
- [4]. *RoboCup*, http://staff.science.uva.nl/~jellekok/robocup/index_en.html, 18 Mei 2011, 16.00 WIB
- [5]. R. de Boer, and J. R. Kok, "The incremental development of a synthetic multi-agent system: the UvA Trilearn 2001 robotic soccer simulation team," Master thesis, University of Amsterdam, The Netherlands, February 2002.
- [6]. Safreni Candra Sari, *Perancangan dan Implementasi Algoritma dan Strategi Tim Kesebelasan Sepak Bola Virtual Pada*

- RoboCupTM Soccer Simulator*, Master tesis, Institut Teknologi Bandung, 2010.
- [7]. J. Kennedy, and R. Eberhart, "Particle Swarm Optimization," *IEEE Conference on Neural Networks*, pp. 1942-1948, (Perth, Australia), Piscataway, NJ, IV, 1995.
- [8]. A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, Wiley, 2005.
- [9]. Galih Hermawan dan R. Priyo Hartono Adji, *Dokumen Simulasi RoboSoccer – B300*, LSKK STEI-ITB, 2011.