

PENGENDALI ROBOT BERGERAK BERBASIS PERILAKU MENGUNAKAN PARTICLE SWARM FUZZY CONTROLLER

Andi Adriansyah

Fakultas Teknik, Universitas Mercu Buana
Jl. Raya Meruya Selatan, Kembangan, Jakarta Barat, 11650
andi@mercubuana.ac.id

Abstrak

Salah satu bidang penelitian yang cukup diminati dalam kategori robot bergerak adalah robot bergerak berbasis perilaku (*behavior-based mobile robots*). Pada robot ini diperlukan sebuah pengendali yang ideal. Pengendali ini diharapkan mampu menghasilkan aksi perilaku yang sempurna dalam rangka menangani kejadian-kejadian yang nampak sebagai sebuah konflik pada saat robot bergerak. Beberapa teknik pengendali telah ditawarkan untuk mengatasi problem ini, diantaranya adalah pengendali yang berbasis logika fuzzy, dikenal dengan robot berbasis perilaku dengan logika fuzzy (*fuzzy behavior-based robot*). Paper ini menawarkan teknik baru untuk menentukan fungsi keanggotaan logika fuzzy dan basis aturan fuzzy secara otomatis menggunakan *Particle Swarm Optimization (PSO)*. Pendekatan ini disebut dengan *Particle Swarm Fuzzy Controller (PSFC)*. PSO adalah metoda optimasi yang menggunakan prinsip perilaku sosial dari kumpulan partikel, seperti burung atau ikan. Pada penelitian ini, perilaku robot akan dikendalikan menggunakan PSFC untuk menghasilkan perilaku tunggal. Kemudian, sebuah pendekatan yang disebut dengan *Context Dependent Blending (CDB)* digunakan. CDB adalah sebuah aturan berbasis logika fuzzy yang menghasilkan aksi akhir untuk mengkoordinasikan perilaku-perilaku tunggal yang ada. Algoritma ini akan divalidasi menggunakan sebuah model robot bergerak dan disimulasikan menggunakan MATLAB. Hasil simulasi menunjukkan hasil yang menguntungkan untuk meningkatkan performa pengendali yang sudah ada.

Kata kunci – robot bergerak, *particle swarm optimization*, *fuzzy system*

1. PENDAHULUAN

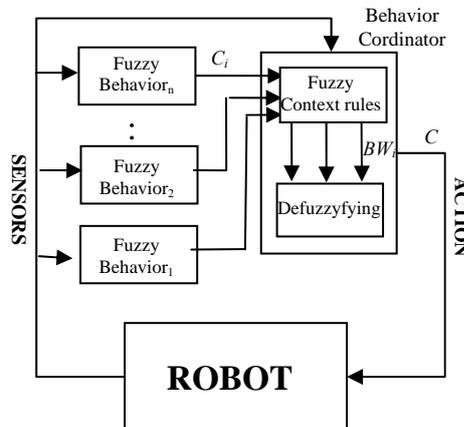
Aplikasi industri dan teknis robot bergerak terus meningkat secara signifikan, baik dari sisi keandalan, akses maupun harga. Salah satu bidang dalam robot bergerak yang menarik untuk dikaji dan sebagai alternatif terhadap robot konvensional adalah pengendali robot berbasis perilaku (Arkin, 1998). Walaupun demikian, terdapat beberapa problem pada metoda ini. Salah satu problemnya adalah bagaimana mengambil keputusan pada saat robot berhadapan dengan keadaan dimana terdapat dua perilaku yang saling bertentangan. Problem ini dikenal dengan istilah *behavior design problem* (Carreras dkk, 2000). Sejumlah sistem kontrol telah ditawarkan untuk menyelesaikan problem ini, baik metoda kontrol klasik maupun modern, seperti Logika Fuzzy, Jaringan Syaraf Tiruan dan Penghitungan Evolusi (*Evolutionary Computations*) (Hoffman dan Pfister, 1997). Sistem Logika Fuzzy termasuk teknik yang paling banyak diimplementasikan untuk mengendalikan perilaku individu robot bergerak (Saffiotti, 1997 dan Aguire dan Gonzales, 2000). Namun, terdapat kesulitan untuk mendapatkan parameter-parameter Logika Fuzzy yang sesuai secara manual. Terlebih lagi jika jumlah variabel dan aturan Fuzzy-nya terus meningkat. Oleh karena itu, diperlukan metoda untuk mendapatkan nilai-nilai tersebut secara otomatis.

Tulisan ini akan mempresentasikan sebuah pendekatan untuk menyelesaikan masalah perancangan perilaku pada robot bergerak diatas dengan menggunakan suatu pengendali berbasis Logika Fuzzy dengan memanfaatkan teknik optimasi *Particle Swarm Optimization (PSO)*. PSO digunakan untuk mendapatkan parameter Logika Fuzzy yang optimal secara otomatis. Teknik tersebut dinamai dengan *Particle Swarm Fuzzy Controller (PSFC)*. Sistematika tulisan dibagi dalam beberapa bagian. Robot bergerak berbasis perilaku dan PSFC akan dibahas pada bagian kedua dan ketiga. Kemudian, perancangan metoda yang ditawarkan dijelaskan pada bagian keempat. Bagian kelima akan menampilkan beberapa hasil eksperimen dan analisisnya. Terakhir, tulisan ini akan diakhiri dengan kesimpulan pada bagian keenam.

2. ROBOT BERGERAK BERBASIS PERILAKU

Metode robot berbasis perilaku adalah suatu pendekatan yang diinspirasi dari sistem biologis, dimana suatu sistem didistribusikan dalam beberapa modul kecil yang disusun secara parallel (Arkin, 1998). Setiap modul, disebut perilaku (*behavior*), memiliki target tertentu yang harus dicapai dan berlaku seperti sebuah sistem kontrol individu. Secara keseluruhan, robot terdiri dari beberapa *behavior* yang mengambil input dari sensor-sensor yang ada dan mengirimkan output proses pengendalian ke aktuator (motor) robot. Karena terdiri dari

beberapa *behavior*, maka diperlukan koordinasi untuk menghasilkan perintah akhir yang akan dikirimkan ke aktuator, jika terdapat beberapa *behavior* yang aktif pada waktu yang sama. Robot berbasis perilaku dengan Logika Fuzzy (*Fuzzy behavior-based robot*) adalah penggunaan teknik Logika Fuzzy untuk menyelesaikan problem pengendali individual *behavior* dan koordinasinya (Saffiotti, 1999). Arsitektur sistem ini terdiri dari dua bagian besar, yaitu bagian modul *behavior* dengan Logika Fuzzy (*Fuzzy Behavior*) dan bagian koordinasi *behavior* yang disebut dengan *Context Dependent Blending* (CDB). Arsitektur ini ditampilkan pada Gambar 1 di bawah ini.



Gambar 1. Arsitektur Pengendali Robot Berbasis Perilaku dengan Logika Fuzzy

Namun, terdapat kesulitan, membosankan serta kerja coba-coba untuk menentukan fungsi keanggotaan (*Membership Function*, MF) dan basis aturan (*Rule Base*, RB) Logika Fuzzy yang sempurna secara manual. Oleh karena itu, hibridisasi sistem yang memiliki kemampuan belajar dan penalaran terhadap Logika Fuzzy banyak dihasilkan (Hoffmann, 2003), seperti: Sistem Neuro-Fuzzy (Parasuraman dkk, 2003 dan Rusu dkk, 2003) dan Fuzzy-Genetik (Hagras dkk, 2001). PSFC termasuk salah satu teknik yang akan dijelaskan pada bagian berikut ini.

3. PARTICLE SWARM FUZZY CONTROLLER

PSO adalah salah satu teknik optimasi menggunakan partikel banyak yang dibangun oleh Kennedy dan Eberhart (Kennedy and Eberhart, 1995), (Eberhart dan Kennedy, 1995). Metoda ini mendapatkan hasil yang optimal dengan mensimulasikan beberapa karakter sosial dari sekumpulan makhluk hidup kecil (partikel), seperti sekawanan burung atau ikan. Kumpulan makhluk tersebut dapat mencapai tujuan mereka secara efektif dengan memanfaatkan informasi bersama dari keseluruhan kumpulan mereka dan informasi pribadi dari masing-masing mereka.

Konsep dasar PSO dapat dijelaskan sebagai berikut: setiap solusi, disebut dengan partikel, mengetahui posisi nilai terbaik menurutnya (*pbest*) dan posisinya. Selain itu, setiap partikel juga mengetahui nilai terbaik menurut kumpulannya (*gbest*) diantara harga-harga terbaik menurut masing-masing partikel (*pbest*). Harga-harga terbaik ini didasari oleh sebuah fungsi penentu (*fitness function*, $F(.)$), bergantung kepada problem yang dihadapi. Setiap partikel akan mencoba untuk memodifikasi posisinya menggunakan kecepatan dan posisi pada saat tersebut. Kecepatan setiap partikel dapat dihitung menggunakan persamaan dibawah ini:

$$v_i^{k+1} = w_i * v_i^k + c_1 * rand * (pbest - s_i^k) + c_2 * rand * (gbest - s_i^k) \quad (1)$$

dimana v_i^k , v_i^{k+1} , and s_i^k , adalah vektor kecepatan, vektor kecepatan modifikasi dan vektor kecepatan untuk setiap partikel ke- i pada generasi ke- k . Kemudian, *pbest* dan *gbest* adalah posisi tebaik bagi partikel ke- i dan posisi terbaik bagi kumpulan partikel. Setelah itu, w_i adalah bobot inersia (*inertia weighth*). Terakhir, c_1 dan c_2 adalah koefisien kognitif dan koefisien sosial, yang mempengaruhi kecepatan partikel.

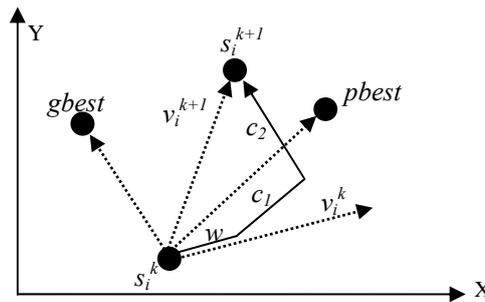
Kemudian, posisi partikel terbaru akan dihitung menggunakan persamaan di bawah ini:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (2)$$

Proses perubahan kecepatan dan posisi untuk setiap partikel dapat digambarkan seperti yang tampak pada Gambar 2.

Algoritma untuk mendapatkan vektor posisi terbaik dari PSO menggunakan n partikel, dapat disimpulkan sebagai berikut:

1. Vektor posisi awal $S[n]$ dan vektor vektor kecepatan $V[n]$ didapat dengan cara acak, dimana $s_i = [s_i^1, s_i^2, \dots, s_i^n]$ dan $v_i = [v_i^1, v_i^2, \dots, v_i^n]$.
2. Vektor kecepatan v_i^{k+i} dari partikel ke- i dihitung menggunakan Persamaan (1).
3. Vektor posisi baru dari partikel ke- i didapat dengan menggunakan Persamaan (2).
4. Jika harga $F(s_i^k)$ yang didapatkan lebih baik dari harga $F(pbest_i)$, vektor posisi s_i^k diset menjadi $pbest$. Namun jika harga $F(pbest_i)$ lebih baik dari harga $F(gbest)$, vektor posisi $gbest$ diset menjadi $pbest$.
5. Jika jumlah generasi yang ditentukan telah dicapai, proses pencarian akan dihentikan. Jika belum, proses pada Step ke-2 akan dijalankan kembali.



Gambar 2. Kecepatan dan Posisi Terbaru per Partikel pada PSO

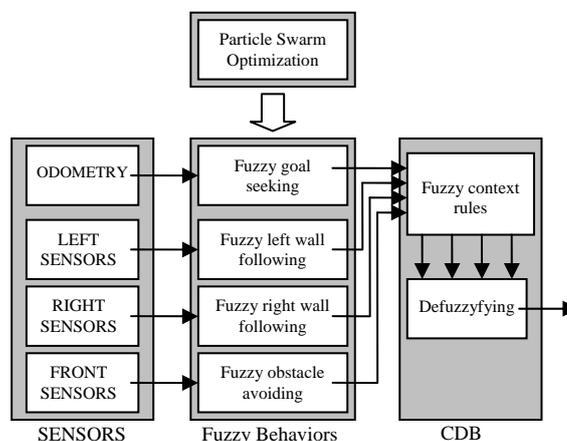
Dalam rangka upaya untuk mendapatkan keseimbangan yang baik antara eksplorasi dan eksploitasi pada proses pencarian, konsep *inertia weight*, w , diperkenalkan, sebagaimana terdapat pada Persamaan (1). Pemanfaatan *inertia weight* pertama kali dilaporkan pada (Eberhart dan Shi, 2000) dimana w diturunkan secara linear dari 0.9 hingga 0.4. dan ternyata mendapatkan hasil peningkatan performa optimasi. Tulisan ini menggunakan prinsip penurunan harga w secara sigmoid (*Sigmoid Decreasing Inertia Weight*, SDIW) (Adriansyah, 2006).

Sedangkan PSFC adalah sebuah system Logika Fuzzy yang ditambahkan dengan mekanisme optimasi PSO. PSO dioperasikan untuk mendapatkan parameter Logika Fuzzy yang optimal untuk suatu sistem dengan criteria optimasi tertentu (Esmindkk, 1998 dan Esmindkk, 2002).

Parameter Logika Fuzzy yang biasa dioptimasi adalah fungsi keanggotaan Logika Fuzzy atau Basis Aturan Logika Fuzzy atau kedua-duanya. Sejauh ini, aplikasi PSFC pada robot bergerak masih jarang sekali. Oleh karena itu, tulisan ini mengimplementasikan PSFC untuk masing-masing modul *behavior* dan koordinasi *behavior*.

4. PERANCANGAN SISTEM

Arsitektur pengendali robot bergerak berbasis perilaku menggunakan PSFC ditunjukkan pada Gambar 3 di bawah ini. Robot bergerak terdiri dari empat *behavior*, yaitu: pencarian target (*goal seeking*), pengikut tembok kiri (*left wall following*), pengikut tembok kanan (*right wall following*) dan penghindar rintangan (*obstacle avoiding*).

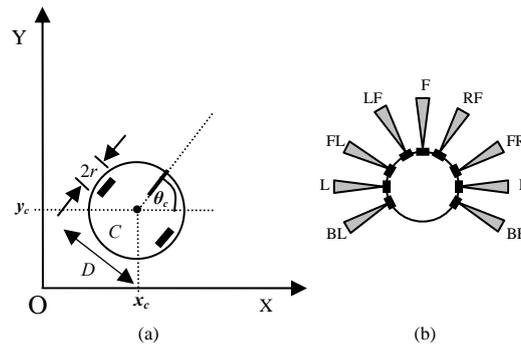


Gambar 3. Arsitektur Pengendali Robot

Setiap *behavior* Fuzzy memiliki akses penuh mendapatkan bacaan sensor dan memprosesnya sendiri untuk menghasilkan output berupa pengendalian arah robot secara individu. Seluruh parameter Logika Fuzzy dihasilkan dari proses PSO. Kemudian, semua pengendalian aksi individual ini dikoordinasikan untuk menghasilkan pengendalian aksi akhir robot menggunakan CDB.

4.1. Model Robot Bergerak

Percobaan menggunakan robot MagellanPro sebagai model. Robot ini merupakan robot berbentuk sirkular dengan dua motor mengendalikan dua roda serta sebuah roda bebas sebagai penunjang. Robot memiliki dimensi sebagai berikut: $D = 40.6$ cm, $H = 25.4$ cm, $r = 5.7$ cm dan $W = 36$ cm, dimana D adalah diameter, H adalah tinggi, r adalah radius roda dan W adalah jarak antara roda, sebagaimana digambarkan pada Gambar 4(a).



Gambar 4. Model Robot Bergerak: (a) Dimensi dan Posisi dan (b) Konfigurasi Sensor pada Robot

Robot dialokasikan pada daerah kerja berbentuk dua dimensi dengan koordinat kartesius. Robot memiliki tiga derajat kebebasan yang direpresentasikan dengan posisi $p_c = (x_c, y_c, \theta_c)$, dimana (x_c, y_c) mengindikasikan posisi spasial robot pada koordinat dan θ_c adalah sudut haapan robot.

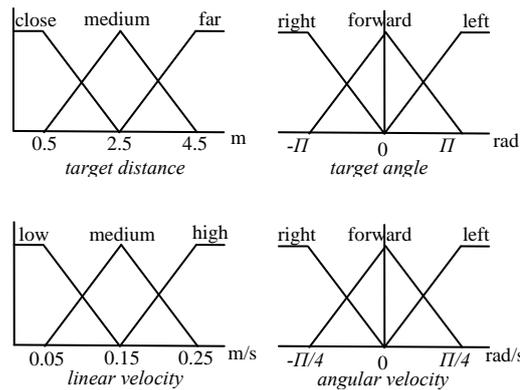
4.2. Perancangan Behavior Fuzzy

Setiap *behavior* memiliki struktur Logika Fuzzy yang sama, yaitu menggunakan fungsi trapesium dan segitia, baik bagi input maupun outputnya. Input bagi tiap *behavior* merupakan sinyal sensor (x_1, x_2, \dots, x_n) dan outputnya merupakan aksi pengontrolan (y_1, y_2) . Output dihasilkan dari proses Logika Fuzzy menggunakan metoda defuzifikasi centroid, dengan persamaan sebagai berikut:

$$y_o = \frac{\sum_{i \in I} \alpha_i \cdot C_{ox} D_{ox}}{\sum_{i \in I} \alpha_i \cdot D_{ox}} \quad (3)$$

dimana C_{ox} dan D_{ox} adalah parameter pusat dan lebar fungsi keanggotaan, α_i adalah hasil perkalian dari setiap derajat keanggotaan dari input untuk aturan ke- i dan i adalah banyaknya aturan Logika Fuzzy.

Beberapa *behavior* memiliki tiga buah input, kecuali *goal seeking behavior* memiliki dua input. Pada *goal seeking behavior*, inputnya adalah jarak target (d) dan sudut target (θ). Sedangkan *behavior* lain, inputnya adalah sebagai berikut: jarak kiri depan (FL), jarak kiri (L) dan jarak kiri belakang (BL) untuk *left wall following*, jarak kanan depan (FR), jarak kanan (R) dan jarak kanan belakang (BR) untuk *right wall following*. Terakhir, untuk *obstacle avoiding behavior* inputnya adalah jarak depan kiri (LF), jarak depan (F) dan jarak depan kanan (RF). Konfigurasi sensor ditampilkan pada Gambar 4(b) diatas. Setiap input memiliki tiga klasifikasi bahasa, yaitu: CLOSE, MEDIUM dan FAR untuk jarak, serta RIGHT, FORWARD dan LEFT untuk sudut. Semua *behavior* memiliki dua output, yaitu *linear velocity* (v) and *angular velocity* (ω). Sebagai contoh, fungsi keanggotaan *goal seeking behavior* diperlihatkan pada Gambar 5.



Gambar 5. Fungsi Keanggotaan untuk Input dan Output bagi Goal Seeking Behavior

Terakhir, basis aturan Logika Fuzzy bagi koordinasi *behavior* menggunakan CDB dengan bentuk sebagai berikut:

IF *obstacle_distance_close* THEN *obstacle_avoiding*
 IF $\sim(\textit{obstacle_distance_close})$ AND *left_distance_close* THEN *left_following*
 IF $\sim(\textit{obstacle_distance_close})$ AND *right_distance_close* THEN *right_following*
 IF $\sim(\textit{obstacle_distance_close})$ AND $\sim(\textit{left_distance_close})$ AND $\sim(\textit{right_distance_close})$ THEN *goal_seeking*

Berdasarkan perancangan Sistem Logika Fuzzy inilah, PSFC kemudian diaplikasikan untuk mendapatkan parameter-parameter Fuzzy yang dibutuhkan.

4.3. Perancangan Particle Swarm Fuzzy

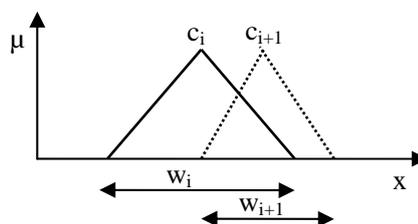
Dikarenakan PSO bekerja menggunakan variabel yang dikodekan, maka semua parameter yang akan dicari harga optimalnya harus dikodekan terlebih dahulu dalam bentuk *string* dengan panjang tertentu. Kode *string* tersebut merupakan penggabungan semua parameter yang akan dianggap sebagai partikel PSO.

Pertama, untuk mendapatkan basis aturan Fuzzy, setiap aturan dikodekan dalam bentuk integer berdasarkan jumlah urutan klasifikasi bahasa dari keluaran fungsi keanggotaannya. Misalnya, '1', '2', dan '3' untuk LOW, MEDIUM dan HIGH bagi masing-masing output *linear velocity* (*v*), and RIGHT, FORWARD, and LEFT untuk *angular velocity* (*ω*).

Setelah itu, untuk menala bentuk fungsi keanggotaan digunakan persamaan di bawah ini:

$$\begin{aligned} C_x &= C_x + k_i \\ W_x &= W_x + j_i \end{aligned} \tag{4}$$

dimana k_i dan j_i adalah koefisien yang ditala, C_x , dan W_x adalah sejumlah pusat dan lebar setiap fungsi keanggotaan. Ini berarti k_i mengatur pergeseran letak pusat segitiga (ke kiri atau ke kanan) dan j_i mengatur lebar fungsi keanggotaan untuk semakin lebar atau semakin sempit. Prinsip ini digambarkan pada Gambar 6.



Gambar 6. Prinsip Penalaan Fungsi Keanggotaan Logika Fuzzy

Proses PSO dimulai dengan membangkitkan populasi awal berdasarkan pengkodean diatas secara acak. Kemudian, setiap populasi akan dievaluasi dan dihubungkan dengan persamaan penentu untuk mendapatkan harga terbaiknya, *pbest* and *gbest*. Fungsi penentu ini bergantung kepada pencapain yang diharapkan masing-masing *behavior*, yaitu:

$$f_{goal} = \sum_{i=0}^I \sum_{k=0}^K (100e_{\theta}^2(k) + e_d^2(k) + 100/v(k)) \quad (5)$$

$$f_{wall} = \sum_{i=0}^I \sum_{k=0}^K (100e_d^2(k) + 100\omega^2(k) + 0.1/v(k)) \quad (5)$$

$$f_{obs} = \sum_{i=0}^I \sum_{k=0}^K (100\omega^2(k) + 0.5/v(k) + 100c(k)) \quad (7)$$

dimana I adalah jumlah total posisi awal robot, K adalah jumlah step simulasi yang dijalankan untuk setiap posisi awal, e_{θ} adalah kesalahan sudut, e_d adalah kesalahan jarak, $\omega(k)$, dan $v(k)$ adalah *linear velocity* and *angular velocity* pada saat k , serta c is konstanta untuk mengecek tabrakan robot dengan rintangan. Secara umum, tujuan dari fungsi-fungsi penentu di atas adalah menghasilkan pergerakan robot yang dapat menghindari rintangan, memiliki kecepatan yang tinggi, bergerak secara kontiniu, mengatur jarak dengan dinding dan mencapai target dengan tepat.

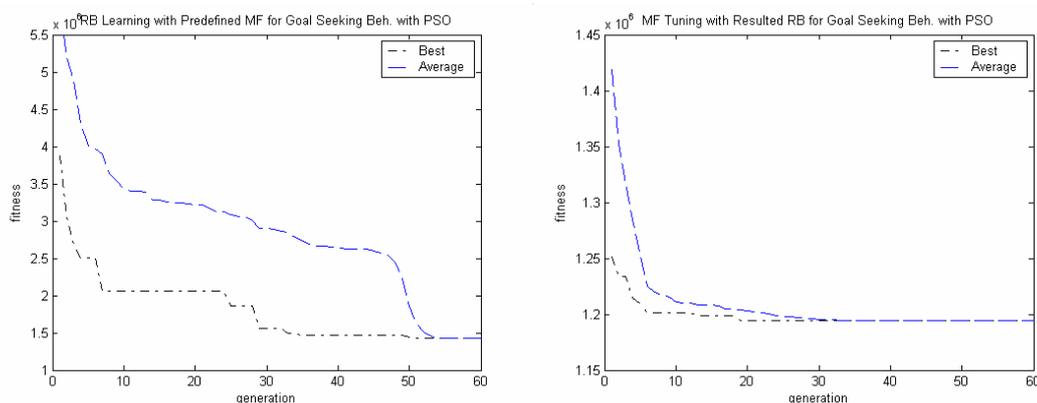
Setelah itu, kecepatan dan posisi partikel akan diperbaharui berdasarkan harga optimum dari masing-masing fungsi penentu. Prosedur ini diulang-ulang, sehingga kondisi yang diharapkan terpenuhi atau jumlah generasi tercapai.

5. HASIL SIMULASI

Simulasi komputer untuk penentuan parameter-parameter Logika Fuzzy bagi masing-masing *behavior* dan animasi pergerakan robot dibangun menggunakan perangkat lunak MATLAB Versi 6.5 Release 13. Untuk mendapatkan parameter Logika Fuzzy terbaik, beberapa proses PSO telah dilakukan. Basis Aturan Logika Fuzzy terbaik dicari terlebih dahulu dengan menggunakan harga fungsi keanggotaan yang ditentukan. Setelah itu, berdasarkan basis aturan Logika Fuzzy terbaik yang telah dihasilkan, dicari fungsi keanggotaannya.

Secara umum, setiap eksperimen terdiri dari 40 partikel individu dan diproses sebanyak 60 generasi. Untuk *goal seeking behavior* terdapat 12 parameter fungsi keanggotaan dan 18 parameters basis aturan Logika Fuzzy yang dicari. Sedangkan untuk *wall following behavior* terdapat 6 parameter fungsi keanggotaan dan 54 parameter basis aturan. Terakhir, untuk *obstacle avoiding behavior* terdiri dari 6 parameter fungsi keanggotaan dan 54 parameter basis aturan.

Proses PSO untuk *goal seeking behavior*, sebagai contoh, ditunjukkan pada Gambar 7. Gambar tersebut mengilustrasikan proses pencarian fungsi keanggotaan dan basis aturan Logika Fuzzy dengan menunjukkan harga fungsi penentunya. Dari gambar tersebut dapat dilihat bahwa harga fungsi penentunya menurun sebanding dengan jumlah generasi menuju harga optimumnya.

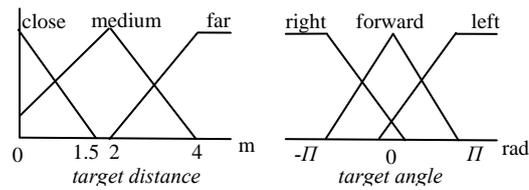


Gambar 7. Proses PSO bagi Goal Seeking Behavior

Fungsi keanggotaan dan basis aturan Logika Fuzzy yang dihasilkan untuk *goal seeking behavior* masing-masing ditampilkan pada Tabel 1 dan Gambar 8 di bawah ini.

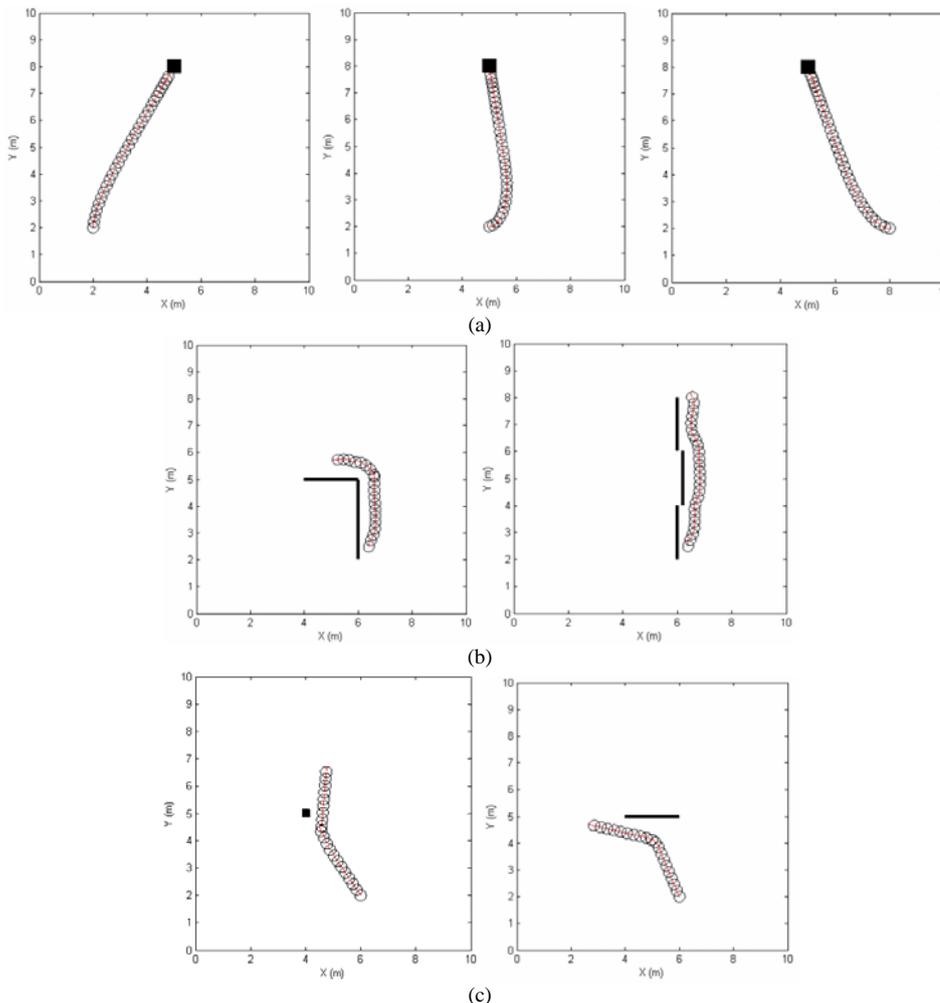
Tabel 1. Basis Aturan untuk Goal Seeking Behavior

INPUT		OUTPUT	
jarak target	jarak target	angular velocity	linear velocity
close	right	right	low
close	forward	forward	medium
close	left	left	low
medium	right	right	low
medium	forward	forward	medium
medium	left	left	low
far	right	right	low
far	forward	forward	high
far	left	left	low



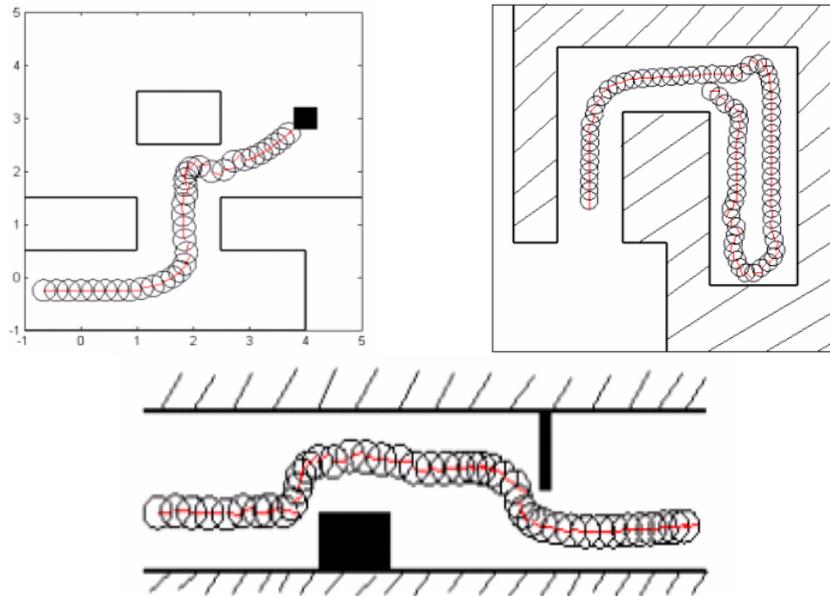
Gambar 8. Fungsi Keanggotaan untuk Goal Seeking Behavior

Pergerakan robot secara individu dari posisi awal yang berbeda diperlihatkan pada Gambar 9. Gambar-gambar tersebut menunjukkan bahwa pengendalian masing-masing *behavior* yang menggunakan PSFC memiliki sifat yang baik dan dapat diandalkan.



Gambar 9. Pergerakan Robot pada Behavior Individu dengan Kondisi Berbeda (a) Goal Seeking Behavior, (b) Left Wall Following, dan (c) Obstacle Avoiding

Terakhir, sebuah ruangan diatur sedemikian rupa sehingga dapat digunakan untuk menguji semua *behavior* dan pengkoordinasiannya. Gambar 10 menunjukkan pergerakan robot pada situasi yang berbeda untuk mencapai target dan keluar dari daerah yang sulit tanpa menabrak rintangan ataupun dinding. Gambar tersebut juga menunjukkan pergerakan robot yang kontiniu serta transisi yang halus dari sebuah *behavior* ke *behavior* lainnya. Transisi antara *behavior* ini membuat robot bergerak dalam jalur yang tercepat dan waktu yang minimum.



Gambar 10. Pergerakan Robot dalam Situasi Berbeda

6. KESIMPULAN

Paper ini mempresentasikan sebuah pengendali yang disebut dengan Particle Swarm Fuzzy Controller (PSFC). PSFC adalah sistem pengendali dengan Logika Fuzzy dimana fungsi keanggotaan dan basis aturannya ditala secara otomatis menggunakan PSO. Pengendali tersebut digunakan untuk menghasilkan sinyal aksi untuk pengendali *behavior* bagi aktuator robot bergerak. Kemudian, sebuah basis aturan Logika Fuzzy yang lain digunakan untuk mengkoordinasikan keseluruhan *behavior* dengan mengaplikasikan metode Context-Dependent Blending (CDB). Dari eksperimen yang dilakukan menunjukkan bahwa sistem yang ditawarkan mampu menghasilkan parameter Logika Fuzzy yang optimal. Hasil terbaik tersebut didapat dari proses pencarian dua tahap, pencarian basis aturan dan pencari fungsi keanggotaan. Pengaplikasian pengendali hasil pencarian pada robot bergerak menunjukkan peningkatan performa pergerakan robot. Hal ini dibuktikan dengan pergerakan robot yang andal, menjaga jarak tertentu dari dinding, mencapai tujuan dengan tepat, menghindari rintangan dan transisi antar *behavior* yang halus.

7. DAFTAR PUSTAKA

- A. Adriansyah, "Analytical and Empirical Study of Particle Swarm Optimization with a Sigmoid Decreasing Inertia Weight", *Proceeding of 1st Regional Postgraduate Conference on Engineering and Science (RPCES 2006)*, July 2006.
- A. Bonarini, G. Invernizzi, T. Halva, and M. Matteucci, "An architecture to coordinate fuzzy behaviors to control an autonomous robot", *Journal Fuzzy Sets and Systems*, Vol. 134 (1), pp. 101-115, 2003.
- A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation", *Soft Computing*, Vol. 1, Springer-Verlag, pp. 180-197, 1997.
- A. Saffiotti, E. H. Ruspini, K. Konolige, K., "Using fuzzy logic for mobile robot control", In: *Practical Application of Fuzzy Technologies*, H-J. Zimmermann (Ed), Kluwer Academic Publisher, pp. 185-206, 1999.
- A.A.A. Esmin, A.R. Aoki, and G.Lambert-Torres, "Fitting Fuzzy Membership Function using Particle Swarm Optimization", *Proc. of the IEEE Int'l Conf. on Evolutionary Computation*, May 1998.
- A.A.A. Esmin, A.R. Aoki, and G.Lambert-Torres, "Particle Swarm Optimization for Fuzzy Membership Function Optimization", *Proc. of the IEEE Int'l Conf. on System, Man and Cybernetics*, pp. 108-113, 2002.
- E. Aguirre, and A. Gonzales, "Fuzzy behaviors for mobile robot navigation : design, coordination and fusion". *International Journal of Approximate Reasoning*, Vol 25, Elsevier Science Inc., pp. 255-289, 2000.

- F. Hoffmann, "An overview on soft computing in Behavior Based Robotics", *Proc. on International Fuzzy System Association World Congress IFSA*, pp. 544-551, 2003.
- F. Hoffmann, and G. Pfister, "Evolutionary design of a fuzzy knowledge base for a mobile robot", *International Journal of Approximate Reasoning*, Vol. 17, pp. 447-469, 1997.
- H. Hagra, V. Callaghan, and M. Colley, "Outdoor mobile robot learning and adaptation", *IEEE Robotics & Automation Magazines*, pp. 53-69, September 2001.
- M. Carreras, J. Batle, P. Ridao, and G. N. Roberts, "An overview on behavior-based methods for AUV control", *Proc. of Conference on Manufacturing and Control of Marine Crafts*, August, 2000.
- P. Rusu, E. M. Petriu, T. E. Whalen, A. Cornell, and H. J. W. Spoelder, "Behavior-based neuro-fuzzy controller for mobile robot navigation", *IEEE Transaction on Instrumentation and Measurement*, Vol. 52, No. 4, pp. 1335-1340, August 2003.
- R. C. Arkin, *Behavior-based Robotics*, The MIT Press, 1998.
- R. C. Eberhart and Y. Shi, "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization", *Proc. Congress on Evolutional Computation*, San Diego, pp. 84-88, 2000.
- S. Parasuraman, V. Ganapathy, and B. Shrinzadeh, "Fuzzy decision mechanism combined with neuro-fuzzy controller for behavior-based robot navigaton", pp. 2410-2416, 2003.