

# Human-Friendly Arm Robot Using Interactive Genetic Algorithm

Cynthia Rachmanita<sup>1</sup>, Indra Adji Sulistijono<sup>1</sup>

<sup>1</sup>Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya

Kampus PENS-ITS Sukolilo, Surabaya

[cynthia@student.eepis-its.edu](mailto:cynthia@student.eepis-its.edu)

[indra@eepis-its.edu](mailto:indra@eepis-its.edu)

**Abstrak** Makalah ini membahas tentang lengan robot sebagai pembangkit trayektori. Berbagai metode pembangkit trayektori telah dikenalkan, tetapi robot harus dapat berinteraksi dengan lingkungan termasuk dengan manusia. Dalam makalah ini diterapkan interactive genetic algorithm sebagai pembangkit trayektori menggunakan evaluasi manusia. Dibutuhkan suatu metode untuk memperkirakan nilai evaluasi manusia, untuk itu digunakan fungsi state-value.

**Kata kunci**— Robot, trayektori, interactive genetic algorithm, evaluasi, state-value

## I. PENDAHULUAN

Dewasa ini robot human-friendly dan robot partner telah banyak dikembangkan untuk generasi selanjutnya demi kepentingan manusia [1]-[3]. Robot – robot tersebut memerlukan kemampuan kecerdasan melalui interaksi antara robot dengan manusia. Robot harus dapat menghasilkan tingkah laku yang diinginkan sebagai komunikasi dengan pemiliknya [1]-[3]. S.Kristensen dkk, mengenalkan metode pembelajaran secara langsung antara manusia dengan robot dalam aplikasi pelayan kopi [3]. Artur M.A. mengenalkan strategi – strategi untuk memperoleh kemampuan tertentu berbasis interaksi robot dengan manusia. Interaksinya dengan manusia memberikan kemampuan baru sebagai seorang aktor [4].

Algoritma – algoritma yang lain telah diterapkan untuk memecahkan masalah – masalah pencarian jalur (path-planning). Xiao mengenalkan adaptive evolutionary planner / navigator menggunakan berbagai macam operator untuk mengembangkan dan memperbaiki jalur – jalur yang akan dilalui [5]. Metode perencanaan trayektori secara hierarki pada ruang pseudo-potential untuk proses perakitan telah dikembangkan. Metode ini secara dasar mempunyai dua permukaan yaitu trajectory generator dan configuration generator menggunakan algoritma virus-evolutionary genetika [6]-[8], [5]. Sekarang, berbagai macam sistem adaptive robot telah dikenalkan untuk beradaptasi dengan lingkungan yang dinamis.

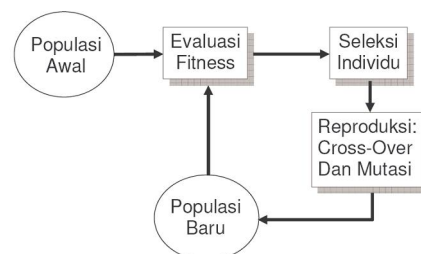
Pembelajaran dengan bantuan (reinforcement learning) telah diterapkan untuk membangun sebuah agen yang memaksimalkan penggunaan yang diharapkan, dengan memberikan hadiah (reward) atau hukuman (punishment) dari lingkungan. Yaitu, reinforcement learning tidak menggunakan sinyal pembelajaran secara langsung, tetapi menggunakan evaluasi umpan-balik yang diperoleh melalui interaksi dengan lingkungan. Pencarian akan menjadi sulit jika ruang keadaan

berukuran besar. Jika menggunakan lookup table, ruang kerja tiga dimensi diuraikan ke dalam sel – sel  $N \times N \times N$  dimana  $N$  menunjukkan ukuran dari sel – sel pada masing – masing sumbu. Selanjutnya, pencarian yang berturut – turut dari kandidat trayektori sulit dilakukan karena dinamika robot lengan. Sehingga, evaluasi manusia diperlukan untuk mewujudkan robot lengan yang human-friendly yaitu pergerakan lengan dianggap aman dan efisien oleh manusia. Efisien dengan maksud, (1) trayektori yang dihasilkan mempunyai jarak terdekat, (2) trayektori bergerak sepanjang titik – titik yang diberikan, (3) trayektori jauh dari halangan, (4) trayektori mencapai obyek dengan fungsi energi minimum dan (5) trayektori bergerak dengan kecepatan motor tinggi atau rendah. Sehingga robot akan mudah digunakan dan mudah berinteraksi.

Robot akan memilih sebuah tingkah laku dan membuat suatu keputusan untuk beraksi dengan manusia. Karena itu, akan dikenalkan metode perencanaan trayektori menggunakan fungsi state-value dan interaktif algoritma genetika. Fungsi state-value digunakan untuk memperkirakan evaluasi manusia. Dalam proyek ini, jumlah kuadrat jarak antara dua konfigurasi, jumlah kuadrat beda diantara sudut masing – masing joint antara dua konfigurasi, dan jumlah dari evaluasi manusia menggunakan fungsi state-value telah dikenalkan. Robot human-friendly harus menghasilkan trayektori berbasis evaluasi manusia.

## II. ALGORITMA GENETIKA (GA)

Adalah teknik pencarian yang di dalam ilmu komputer untuk menemukan penyelesaian perkiraan untuk optimisasi dan masalah pencarian. Proses dalam Algoritma Genetika akan mengikuti siklus seperti yang diilustrasikan pada gambar 1



Gambar 1. siklus Algoritma Genetika

Beberapa definisi penting yang perlu diperhatikan dalam Algoritma Genetika adalah:

**Genotype (Gen)**, sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika, gen ini bisa berupa nilai biner, float, integer maupun karakter, atau kombinatorial.

**Allele**, nilai dari gen.

**Kromosom**, gabungan gen-gen yang membentuk nilai tertentu.

**Individu**, menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.

**Populasi**, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.

**Generasi**, menyatakan satu-satuan siklus proses evolusi.

**Nilai Fitness**, menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.

#### A. Mekanisme Algoritma Genetika

Mekanisme yang ada dalam GA yaitu hanya melibatkan penyalinan gen dan pertukaran gen. Siklus perkembangbiakan GA diawali dengan pembuatan himpunan solusi secara acak dinamakan populasi, di mana di dalamnya terdapat individu-individu yang dinamakan kromosom. Kromosom ini secara lambat laun mengalami iterasi 'perkembangbiakan' dalam sebuah generasi. Selama dalam sebuah generasi, kromosom-kromosom ini dievaluasi, dengan menggunakan rumus-rumus yang telah ditentukan dalam fungsi *fitness*.

Mekanisme dari GA tersebut yaitu:

- Pembangkitan populasi awal
- Fitness
- Seleksi
- CrossOver / KawinSilang
- Mutasi

#### B. Prosedur Algoritma Genetika

##### 1) Populasi Awal

Populasi adalah kumpulan dari individu-individu yang mempunyai solusi. Populasi awal pada algoritma GA adalah sekumpulan individu yang dibangkitkan secara acak. Yang perlu dipertimbangkan adalah jumlah populasi, jumlah populasi yang terlalu sedikit akan sulit untuk mencapai solusi terbaik. Sedangkan jika jumlah populasi terlalu banyak maka perhitungan akan menjadi kompleks dan membutuhkan waktu komputasi yang lama.

##### 2) Fitness suatu Individu

Nilai fitness adalah nilai yang menyatakan baik tidaknya suatu solusi (individu). Nilai fitness ini yang dijadikan acuan dalam mencapai nilai optimal dalam algoritma genetika. Algoritma genetika bertujuan mencari individu dengan nilai fitness yang paling tinggi.

##### 3) Seleksi

Seleksi posisi terbaik atau mencari individu yang terbaik dimulai dengan menghitung nilai fitness setiap posisi yang dilalui satu individu. Dengan membandingkan nilai fitness setiap posisi yang dilalui individu tertentu maka akan didapat best *trajectory* suatu individu. Berikut ini adalah

contoh algoritma seleksi tersebut. Induk digantikan keturunannya segera setelah kelahiran keturunan, dimana operasi genetika bersifat tidak pandang bulu, keturunan yang baik menggantikan induk yang jelek.

##### 4) CrossOver

*CrossOver* adalah menggabungkan/mengawinkan dua individu dan menghasilkan dua individu baru. Pindah silang adalah operator genetic yang utama. Operator ini bekerja mengambil 2 individu dan memotong string kromosom mereka pada posisi yang terpilih secara acak.

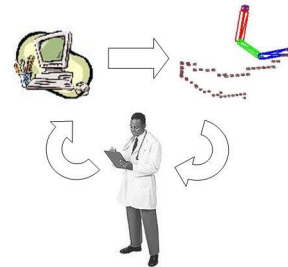
Operasi ini tidak selalu dilakukan pada semua individu yang ada. Individu dipilih secara acak untuk dilakukan crossing dengan Pc antara 0,6 - 0,95. Jikapindah silang tidak dilakukan maka nilai dari induk akan diturunkan kepada keturunannya.

##### 5) Mutasi

Mutasi digunakan untuk melakukan modifikasi satu atau lebih nilai gen dalam individu yang sama. Mutasi memastikan bahwa probabilitas untuk pencarian pada daerah tertentu dalam persoalan tidak akan pernah nol dan mencegah kehilangan total materi genetik setelah pemilihan dan penghapusan.

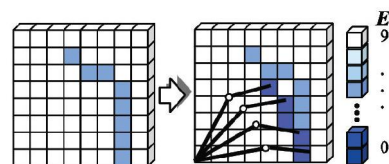
### III. RANCANGAN SISTEM

Trayektori yang dibangkitkan oleh CPU akan dikirim ke rangkaian hardware dan melalui evaluasi manusia akan dinilai trayektori – trayektori yang terbaik pada saat itu. Gambar berikut adalah blok diagram keseluruhan sistem.



Gambar 2. Blok diagram sistem.

Evaluasi manusia yang diberikan akan digunakan untuk memperbaiki fungsi state-value. Gambar berikut ini adalah contoh cara pemberian nilai dan cara memperbaiki setiap trayektori yang dihasilkan, dengan memisalkan kotak berbentuk dua dimensi.



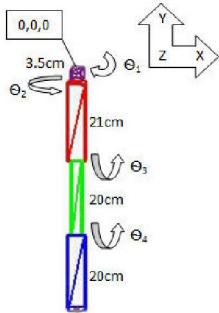
Gambar 3. Sel – sel fungsi state-value.

Nilai fitness merupakan ukuran yang menentukan seberapa baik setiap individu yang dibangkitkan. Tujuan utama dari keseluruhan proses ini adalah untuk membangkitkan trayektori yang mempunyai jarak minimum dari konfigurasi awal sampai konfigurasi akhir dan mempunyai nilai evaluasi manusia tertinggi.

$$fit_i = w_1 f_p + w_2 f_d + w_3 f_v \dots\dots\dots(1)$$

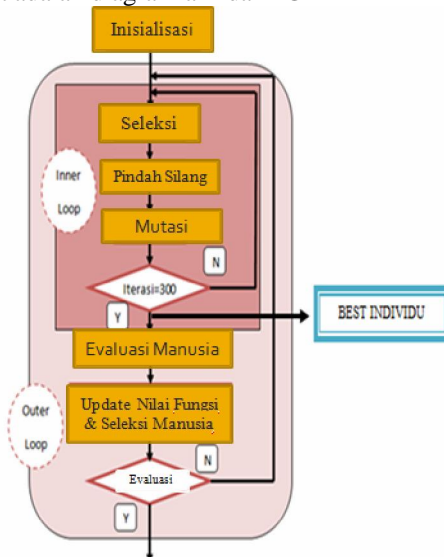
$$f_v = \sum_{(x,y,z) \in P} value(x, y, z) \dots\dots\dots(2)$$

$w_1, w_2, w_3$  adalah koefisien jarak,  $f_p$  adalah jumlah kuadrat dari jarak antara 2 konfigurasi,  $f_d$  adalah jumlah kuadrat dari sudut antara 2 konfigurasi pada masing – masing joint, dan  $f_v$  adalah jumlah nilai evaluasi manusia yang menggunakan fungsi *state - value*.



Gambar 4. Pergerakan dan ukuran robot lengan.

Gambar diatas adalah pergerakan dan ukuran lengan robot yang nanti digunakan dalam percobaan. Titik awal robot adalah titik (0,0,0), memanjang ke arah sumbu y negatif. Gambar berikut adalah diagram alir dari IGA



Gambar 5. Flowchart IGA

Terdapat dua bagian utama dalam flowchart diatas yaitu inner loop dan outer loop. Pada inner loop terdapat proses IGA sedangkan human - evaluation dan update state - value terdapat di outer loop. Proses pertama adalah inisialisai dimana individu-individu baru akan dibangkitkan secara acak. Dari setiap individu tersebut akan dicari nilai terbaik kemudian mengevaluasi trayektori individu tersebut melalui fungsi evaluasi fitness.

#### IV. HASIL PERCOBAAN

Pengujian lengan robot dilakukan pada beberapa parameter yaitu.

TABEL I  
PARAMETER IGA, POP\_SIZE 50

Population Size	50
Num Configuration	10
Local Evaluations (Inner loop times)	10
Space Size	22
Prob_Crossover	0.7
Prob_Mutation	1.0

TABEL II  
PARAMETER IGA YANG DIBUAT VARIABEL

Coba ke	SUM OF ANGLE	SUM OF DISTANCES	SUM OF ESTIMATED EVALUATION
1.	0,8	0,1	0,1
2.	0,1	0,8	0,1
3.	0,1	0,1	0,8
4.	0,3	0,1	0,6

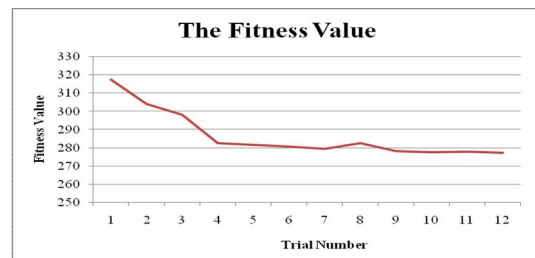
#### Percobaan ke-1:

Percobaan pertama dilakukan dengan menset nilai pemberat  $w_1$  dengan nilai yang ekstrim terhadap parameter pemberat yang lain. Parameter  $w_1$  diset 0.8 karena penjumlahan ketiga parameter harus satu dengan perubahan satu digit dibelakang koma. Hal ini dilakukan untuk mengetahui pengaruhnya terhadap trayektori yang dihasilkan robot.

$$w_1 = 0.8 // \text{weight sum of angle}$$

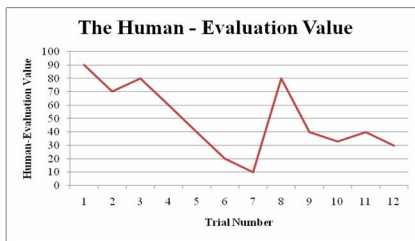
$$w_2 = 0.1 // \text{weight sum of distance}$$

$w_3 = 0.1 // \text{weight sum of human evaluation}$ , didapatkan kurva seperti gambar di bawah ini:



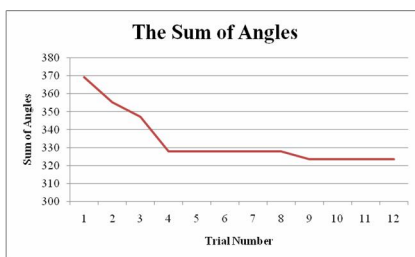
Gambar 6. Grafik nilai fitness  $w_1=0.8, w_2=0.1, w_3=0.1$

Pada grafik dapat dilihat bahwa untuk mendapatkan trayektori yang baik maka membutuhkan beberapa trial, untuk nilai fitness trial ke-12 didapatkan nilai 275.



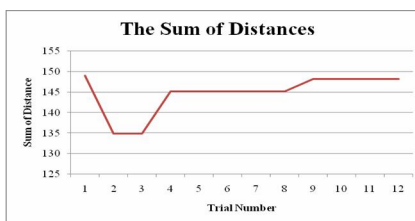
Gambar 7. Grafik evaluasi manusia  $w_1=0.8, w_2=0.1, w_3=0.1$

Jika grafik dilihat dari evaluasi manusia makin lama makin tidak baik terjadi tahanan yang menyolok karena kurang bisa untuk menemukan solusi yang diharapkan setelah. Sehingga saat trial ke 8, nilai evaluasi manusia mengalami kenaikan grafik hal ini dikarenakan trayektori yang dihasilkan hanya mengandalkan jumlah nilai sudut yang terkecil.



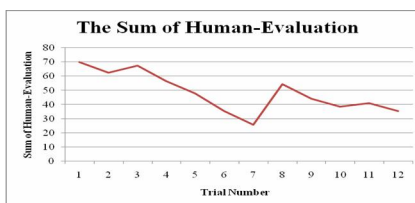
Gambar 8. Grafik nilai jumlah sudut  $w_1=0.8, w_2=0.1, w_3=0.1$

Data yang pertama dengan  $w_1 = 0.8$  menunjukkan grafik jumlah sudut yang semakin lama semakin menurun karena nilai sudut yang berpengaruh dalam hal ini sehingga hasilnya baik.



Gambar 9. Grafik nilai jumlah jarak  $w_1=0.8, w_2=0.1, w_3=0.1$

Data yang pertama dengan  $w_1 = 0.1$  menunjukkan grafik jumlah jarak yang semakin lama semakin menurun tapi sempat naik karena nilai sudut yang berpengaruh dalam hal ini sehingga hasilnya untuk jumlah jarak kurang begitu bagus.



Gambar 10. Grafik jumlah evaluasi manusia  $w_1=0.8, w_2=0.1, w_3=0.1$

Data diatas dengan menunjukkan grafik jumlah evaluasi manusia yang semakin lama semakin menurun walaupun

kurang begitu baik, karena kurang bisa untuk menemukan solusi yang diharapkan.

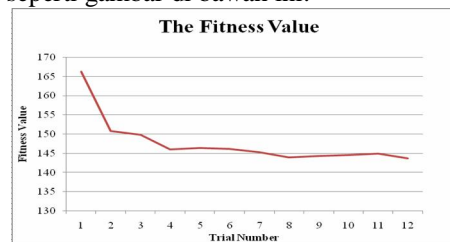
### Percobaan ke-2 :

Percobaan kedua dilakukan dengan menset nilai  $w_2$  lebih tinggi dari parameter yang lain namun tetap jumlah ketiga parameter adalah 1. Hasil yang diperoleh adalah gerakan lengan robot terlalu acak, kurang halus bahkan sampai melebihi batas – batas yang telah ditentukan.. Berikut ini adalah grafik *sum of distance* pada pengujian ini.

$$w_1 = 0.1 // \text{weight sum of angle}$$

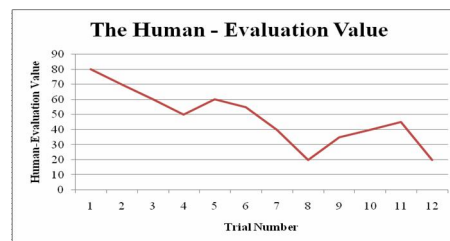
$$w_2 = 0.8 // \text{weight sum of distance}$$

$w_3 = 0.1 // \text{weight sum of human evaluation,}$  didapatkan kurva seperti gambar di bawah ini:



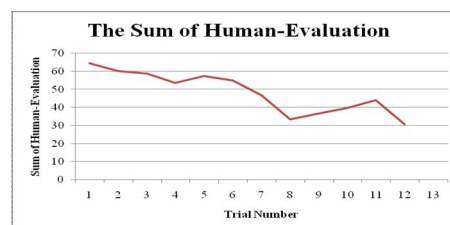
Gambar 11. Grafik nilai fitness  $w_1=0.1, w_2=0.8, w_3=0.1$

Pada grafik dapat dilihat bahwa untuk mendapatkan trayektori yang baik maka membutuhkan beberapa trial, untuk nilai fitness trial ke-12 didapatkan nilai 143.



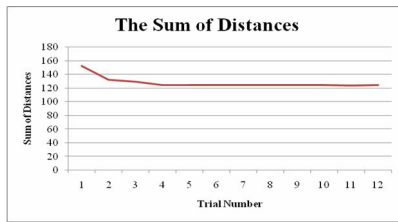
Gambar 12. Grafik nilai evaluasi manusia,  $w_1=0.1, w_2=0.8, w_3=0.1$

Nilai evaluasi manusia mengalami kenaikan saat trial 4 menuju trial 5 dan 8 ke 9 (gerakan semakin memburuk). Hal ini menunjukkan trayektori yang dihasilkan hanya mengacu pada jumlah jarak, gerakan lengan robot menjadi tidak beraturan tidak sesuai untuk jabat tangan.



Gambar 13. Grafik jumlah evaluasi manusia  $w_1=0.1, w_2=0.8, w_3=0.1$

Data diatas dengan menunjukkan grafik jumlah evaluasi manusia yang semakin lama semakin menurun walaupun kurang begitu baik, karena kurang bisa untuk menemukan solusi yang diharapkan.



Gambar 14. Grafik jumlah jarak  $w_1=0.1, w_2=0.8, w_3=0.1$

Data yang kedua yaitu parameter  $w_2=0.8$  yang lebih besar nilainya dibanding parameter yang lain, grafik jumlah jarak menunjukkan grafik yang membaik tetapi tidak menurut nilai evaluasi manusia.

### Percobaan ke-3 :

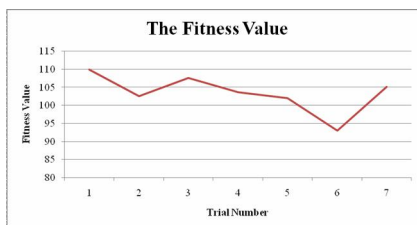
Percobaan ketiga dilakukan dengan menset  $w_3$  dengan nilai yang lebih tinggi dari parameter yang lain. Namun tetap masih dengan jumlah ketiga parameter sama dengan satu.

$$w_1 = 0.1 // \text{weight sum of angle}$$

$$w_2 = 0.1 // \text{weight sum of distance}$$

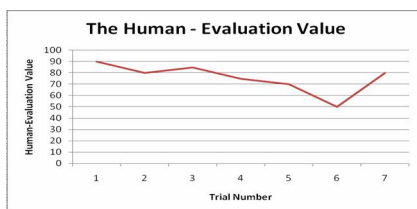
$$w_3 = 0.8 // \text{weight sum of human evaluation, didapatkan}$$

kurva seperti gambar di bawah ini:



Gambar 15. Grafik nilai fitness  $w_1=0.1, w_2=0.1, w_3=0.8$

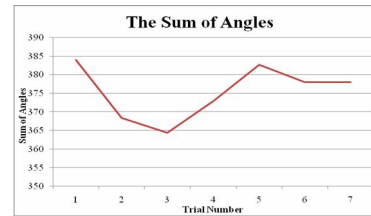
Pada grafik dapat dilihat bahwa untuk mendapatkan trayektori yang baik maka membutuhkan beberapa trial, untuk nilai fitness trial ke-7 didapatkan nilai 105.



Gambar 16. Grafik nilai evaluasi manusia  $w_1=0.1, w_2=0.1, w_3=0.8$

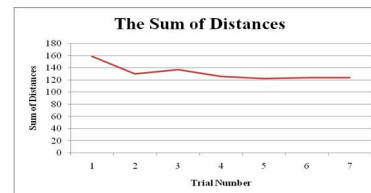
Data yang ketiga untuk human evaluation, meskipun menunjukkan keseluruhan grafik yang semakin baik (grafik menurun) tetapi kurang bisa untuk menemukan solusi yang diharapkan ditunjukkan dengan grafik nilai evaluasi manusia yang agak datar tapi masih terjadi tanjakan. Hal ini

menunjukkan gerakan robot seperti terperangkap dalam daerah tertentu.



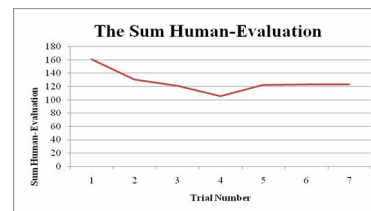
Gambar 17. Grafik nilai jumlah sudut  $w_1=0.1, w_2=0.1, w_3=0.8$

Data yang pertama dengan  $w_1 = 0.1$  menunjukkan grafik jumlah sudut yang semakin lama semakin menurun karena nilai sudut yang tidak berpengaruh dalam hal ini sehingga hasilnya kurang baik.



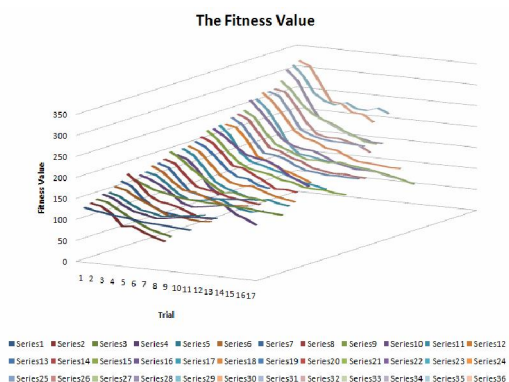
Gambar 18. Grafik nilai jumlah jarak  $w_1=0.1, w_2=0.1, w_3=0.8$

Data yang pertama dengan  $w_1 = 0.1$  menunjukkan grafik jumlah jarak yang semakin lama semakin menurun karena nilai jarak kurang begitu pengaruh dalam hal ini sehingga hasilnya untuk jumlah jarak kurang begitu bagus.

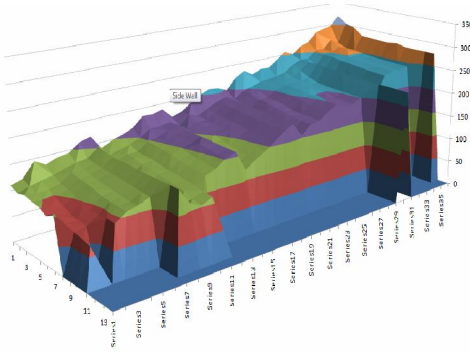


Gambar 19. Grafik jumlah evaluasi manusia  $w_1=0.1, w_2=0.1, w_3=0.8$

Data diatas dengan menunjukkan grafik jumlah evaluasi manusia yang semakin lama semakin menurun walaupun kurang begitu baik, karena kurang bisa untuk menemukan solusi yang diharapkan.



Gambar 20. Grafik nilai fitness, pop size=50.



Gambar 21. Grafik 3D nilai *fitness*, pop size=50.

TABEL III

PARAMETER PEMBERAT PERCOBAAN IGA

Konstanta			End of Trial	Konstanta			End of Trial
w1	w2	w3		w1	w2	w3	
0.1	0.1	0.8	6↑	0.4	0.1	0.5	10↑
0.1	0.2	0.7	9	0.4	0.2	0.4	10
0.1	0.3	0.6	7	0.4	0.3	0.3	10↑
0.1	0.4	0.5	10↑	0.4	0.4	0.2	10
0.1	0.5	0.4	10↑	0.4	0.5	0.1	10↑
0.1	0.6	0.3	10↑				
0.1	0.7	0.2	10↑	0.5	0.1	0.4	10↑
0.1	0.8	0.1	12↑	0.5	0.2	0.3	10↑
				0.5	0.3	0.2	10↑
0.2	0.1	0.7	12↑	0.5	0.4	0.1	14
0.2	0.2	0.6	12				
0.2	0.3	0.5	10↑	0.6	0.1	0.3	10↑
0.2	0.4	0.4	10↑	0.6	0.2	0.2	10↑
0.2	0.5	0.3	10↑	0.6	0.3	0.1	10↑
0.2	0.6	0.2	10				
0.2	0.7	0.1	10↑	0.7	0.1	0.2	10↑
				0.7	0.2	0.1	10
0.3	0.1	0.6	10				
0.3	0.2	0.5	10	0.8	0.1	0.1	12
0.3	0.3	0.4	10↑				
0.3	0.4	0.3	10				
0.3	0.5	0.2	10↑				

## V. KESIMPULAN

Jumlah pop size atau ukuran populasi yang semakin tinggi akan cepat menemukan solusi yang terbaik dan dapat mencegah terjadinya konvergensi pada wilayah lokal tetapi komputasi akan semakin berat. Diatas 100 jumlah populasi akan cepat menemukan solusinya.

Nilai pemberat  $w_1$  (pemberat *sum of angle*) diatas 0.3 mengakibatkan gerakan robot yang kurang baik menurut evaluasi manusia, gerakan lengan robot menjadi lebih halus namun tidak menunjukkan jabat tangan yang baik.

Nilai pemberat  $w_2$  (pemberat *sum of distance*) diatas 0.4 dengan  $w_1$  kurang dari 0.3 dan diatas 0.3 mengakibatkan gerakan robot menjadi acak atau tak menentu atau bahkan melebihi batas – batas jabat tangan yang diperbolehkan.

Nilai pemberat  $w_3$  (pemberat *sum of human-evaluation*) diatas 0.4 mengakibatkan robot terperangkap pada daerah tertentu saja, tidak bisa menemukan daerah solusi yang terbaik.

Perpaduan ketiga parameter  $w_1, w_2, w_3$  akan menemukan solusi yang terbaik (yaitu saat  $w_1 = 0.3, w_2 = 0.2, w_3 = 0.5$ ).

Terlihat pada data hasil percobaan bahwa IPSO dapat menemukan trayektori lebih cepat (yaitu dengan jumlah *trial* 5) daripada IGA (yaitu dengan jumlah *trial* 10).

## DAFTAR PUSTAKA

- [1] O.Khatib, K.Yokoi, O.Brock, K-S.Chang, and A.Casal, "Robots in Human Environments", In Archives of Control Sciences, *Special Issue on Granular Computing*. Vol.11 (XLVII), No. 3-4, pp.123-128, 2002.
- [2] N.Kubota, D.Hisajima, F.Kojima, and T.Fukuda, "Fuzzy and Neural Computing for Communication of a Partner Robot", *Journal of Multi-Valued Logic and Soft Computing*, Vol.9, No.2, pp.221-239, 2003.
- [3] S.Kristensen, S.Horstmann, J.Klandt, F.Lohnert and A.Stopp, "Human-Friendly Interaction for Learning and Cooperation", *Proc. of the 2001 IEEE ICRA*, pp.2590-2595, 2001.
- [4] Artur M. Arsenio, Embodied Vision - Perceiving Objects from Actions, *Proceeding of the 12th IEEE Workshop Robot and Human Interactive Communication RO-MAN 2003*, pp.365-371, 2003.
- [5] J.Xiao, Z.Michalewicz, L.Zhang, K.Trojanowski, "Adaptive Evolutionary Planner/Navigator for Mobile Robots", *IEEE Trans. Evolutionary Computation*, Vol.1, No.1, 1998, pp.18-28.
- [6] T.Fukuda, N.Kubota, and T.Arakawa, "GA Algorithms in Intelligent Robot", *Fuzzy Evolutionary Computation*, Kluwer Academic Publishers, 1997, pp.81-105.
- [7] T.Arakawa and T.Fukuda, "Natural Motion Trajectory Generation of Biped Locomotion Robot Using Genetic Algorithm through Energy Optimization", in *Proc. of The IEEE International Conference on Systems, Man, and Cybernetics*, 1996, pp.1495-1500.
- [8] N.Baba and N.Kubota, "Collision Avoidance Planning of a Robot Manipulator by Using Genetic Algorithm-A Consideration for the Problem in which Moving Obstacles and/or Several Robots Are in the Workspace", *Proc. of The First IEEE Conference on Evolutionary Computation*, 1994, pp.714-719.
- [9] J.H. Holland, "Adaptation in Natural and Artificial Systems", Cambridge, Massachusetts, MIT Press, 1995.
- [10] S.J.Russell, P.Norvig, "Artificial Intelligence", Prentice Hall, Inc., 1995.
- [11] N.Kubota, I.Adji S., F.Kojima, "Interactive Genetic Algorithm for Trajectory Generation of a Robot Manipulator", *Proceeding (CD ROM) of 4 th Asia-Pasific Conference On Simulated Evolution And Learning (SEALO2)*, pp.146-150, Singapore, 18-22 November, 2002.
- [12] Shi Yuhui, "Particle Swarm Optimization", IEEE Neural Networks Society, February 2004.
- [13] A.P. Engelbrecht, "Fundamentals of Computational Swarm Intelligence", John Wiley & Sons, 2005.