

Penerapan *Fuzzy Q Learning* pada Navigasi Otonom *Behavior Based Hexapod Robot*

Handy Wicaksono¹⁾, Prihastono^{2),3)}, Khairul Anam⁴⁾, Rusdhianto Effendi²⁾, Indra Adji Sulistijono⁵⁾,
Son Kuswadi⁵⁾, Achmad Jazidie²⁾, Mitsuji Sampei⁶⁾

¹⁾ Jurusan Teknik Elektro, Universitas Kristen Petra

(Tel : +62-31-2983115; e-mail : handy@petra.ac.id)

²⁾ Jurusan Teknik Elektro, Institut Teknologi Sepuluh Noverber

³⁾ Jurusan Teknik Elektro, Universitas Bhayangkara

⁴⁾ Jurusan Teknik Elektro, Universitas Jember

⁵⁾ Jurusan Mekatronika, Politeknik Elektronika Negeri Surabaya

⁶⁾ Department of Mechanical and Environmental Informatics, Tokyo Institute of Technology

Abstrak - *Hexapod robot* banyak digunakan karena kestabilan dan fleksibilitas pola gerakannya. Pada penelitian ini akan didesain *hexapod* dengan arsitektur *behavior based* yang bersifat cepat dan reaktif terhadap masukan dari dunia luar. Selain itu akan diterapkan *fuzzy Q learning* sebagai algoritma pembelajaran robot yang mampu menangani *state/action* yang kontinyu, sehingga robot dapat mengantisipasi hal – hal tak terduga di lingkungannya dengan lebih tepat.

Dari hasil simulasi nampak bahwa penerapan arsitektur *behavior based* dan algoritma pembelajaran *fuzzy Q learning* berhasil digunakan untuk sistem navigasi otonom robot yang bertujuan untuk menghindari halangan dan menemukan target berupa sumber cahaya. Jumlah rata – rata *reward* yang diterima robot dalam 1 iterasi adalah 0.62 satuan *reward*.

Kata kunci : *hexapod robot, behavior based architecture, fuzzy q learning, navigasi otonom*

1. Pendahuluan

Robot beroda merupakan jenis robot yang banyak digunakan dalam berbagai aplikasi karena kecepatannya dalam melintasi bidang yang rata, dan kemudahan dalam desain dan implementasi. Namun dalam dunia nyata sering terdapat kondisi bidang tidak rata, sehingga digunakan robot berkaki. Keuntungan robot ini ialah mampu bergerak di daerah yang halus ataupun kasar, memanjat tangga, menghindar, dan melangkah di atas halangan (Bekey, 2005).

Salah satu jenis robot berkaki yang paling banyak digunakan ialah robot berkaki enam (*hexapod*). *Hexapod* memiliki banyak keuntungan, di antaranya ialah fleksibilitas dalam pola gerakan, karena robot memiliki kestabilan statis jika tiga atau lebih kakinya ada di tanah. *Hexapod* juga memiliki kecepatan yang lebih tinggi dari robot berkaki empat saat menggunakan *statically stable gait* (Billah dkk, 2008).

Hexapod telah digunakan untuk berbagai aplikasi, misalnya COMET, *hexapod* dengan dua buah lengan tambahan, digunakan untuk menjinakkan ranjau (Nonami dkk, 2000). Contoh lain ialah DANTE, *hexapod* yang digunakan untuk penjelajahan gunung berapi Mount Spur, Alaska, untuk mengumpulkan dan mengirimkan data ilmiah pada operator dan ahli gunung berapi di lokasi yang berjauhan (Bares dkk, 1999).

Dalam banyak aplikasi robot, sering kali dibutuhkan reaksi yang cepat dari robot. Arsitektur *behavior based control* merupakan arsitektur robot yang cocok karena memiliki struktur *behavior horizontal* yang bekerja bersama secara paralel,

bersamaan dan asinkronus (Brooks, 1986). *Hexapod* pertama yang digunakan dengan arsitektur *behavior based* ialah Genghis (Brooks, 1989)

Selain arsitektur yang tepat, juga diperlukan mekanisme pembelajaran yang tepat pada robot untuk mengatasi hal – hal tak terduga. *Reinforcement learning* adalah metode *unsupervised learning* yang dapat belajar dari kritik/*reward* secara langsung (*online*) dari lingkungan, sehingga cocok untuk aplikasi robot. (Glennec, 2000).

Ada berbagai metode untuk penyelesaian masalah *reinforcement learning*, salah satu yang paling populer ialah *Q Learning Algorithm* (Watkins, 1989). Kelebihan dari *Q Learning* ialah sifatnya yang *off policy* (dapat mengikuti policy apapun), algoritma yang sederhana, dan konvergen terhadap *optimal policy* (Perez, 2003).

Supaya *Q learning* dapat digunakan untuk *continuous state/action* perlu dilakukan generalisasi menggunakan *function approximator*, di antaranya : *Cerebellar Mode Action Controller* (CMAC), *neural networks*, Kohonen's *self organizing map*, dan lain – lain. Namun *function approximator* di atas hanya mengasumsikan *piecewise continue action*, bukan *continuous action* sepenuhnya (Deng dkk., 2004b).

Sedang *Fuzzy Inference System* (FIS) dapat digunakan untuk generalisasi pada *state space* dan dapat menghasilkan *continuous action* (Jouffle, 1998). Terdapat beberapa struktur *Fuzzy Q Learning* (FQL) seperti yang dibuat oleh Glennec (1997) yang kemudian dimodifikasi lagi oleh Deng (2004a, 2004b).

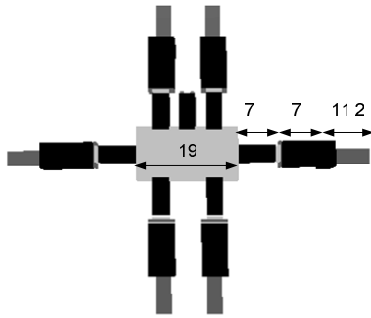
Pada penelitian ini akan dirancang *hexapod robot* dengan arsitektur *behavior based*. Kemudian juga akan

diterapkan algoritma *fuzzy Q learning* (penyempurnaan dari *Q learning*) sebagai mekanisme pembelajaran robot. Robot akan melakukan navigasi otonom untuk menghindari halangan dan menemukan target berupa sumber cahaya.

2. Metodologi Penelitian

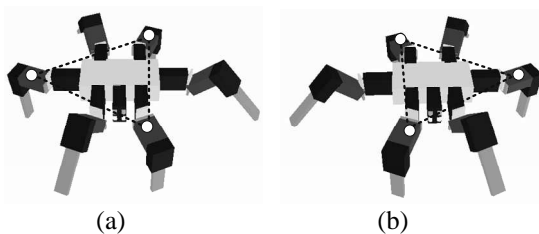
2.1 Mekanik dan Mekanisme Gerak Robot

Berikut ini ukuran robot hexapod (dalam satuan cm). Semua kaki robot direntangkan supaya mempermudah pemberian ukuran.



Gambar 1. Ukuran *hexapod robot*

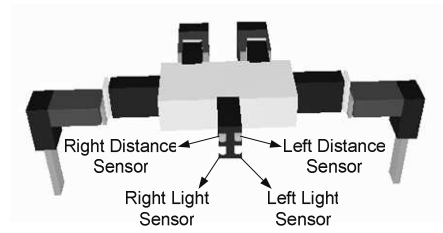
Mekanisme gerak dari *hexapod robot* ini menggunakan *tripod gait*, yang mendasarkan kestabilan robot pada tiga titik segitiga berupa kaki – kaki robot yang berada di tanah dalam 1 waktu. Pada gambar berikut nampak 2 buah konfigurasi segitiga saat robot berjalan. Sehingga saat robot berjalan, selalu ada tiga kaki yang sedang berada di tanah. Gambar 2 menunjukkan gambaran masing – masing segitiga kestabilan.



Gambar 2. Segitiga kestabilan robot

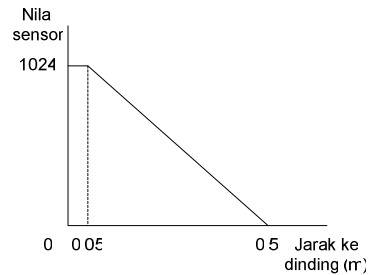
2.2 Sensor – Sensor pada Robot

Sensor yang digunakan dalam sistem navigasi otonom *hexapod robot* ini ialah dua buah sensor jarak dan dua buah sensor cahaya. Konfigurasi sensor dapat dilihat pada Gambar 3. Dari gambar tersebut nampak bahwa sensor yang digunakan ialah 2 buah sensor jarak (kiri dan kanan) serta 2 buah sensor cahaya (kiri dan kanan).



Gambar 3. Peletakan sensor pada robot

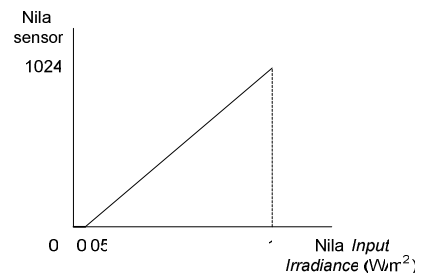
Karakteristik dari sensor jarak nampak pada Gambar 4.



Gambar 4. Grafik karakteristik sensor jarak

Dari grafik tersebut nampak bahwa jangkauan terjauh sensor ialah 0.5 m. Dalam simulasi ini, sensor diasumsikan ideal dan bebas *noise*.

Sedangkan karakteristik dari sensor cahaya nampak pada Gambar 5.



Gambar 5. Grafik karakteristik sensor cahaya

Dalam simulasi ini, sensor juga diasumsikan ideal dan bebas *noise*.

2.3 Behavior Coordination pada Robot

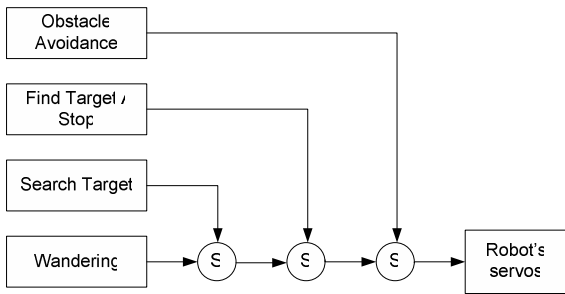
Supaya robot dapat melakukan navigasi secara otonom, maka robot harus memiliki *behavior* – *behavior* berikut :

1. Berkeliling (*wandering*)
2. Hindari halangan (*obstacle avoidance*)
3. Cari target (*search target*)
4. Stop (*find target*)

Sekumpulan *behavior* yang telah disebut di atas harus dikoordinasikan supaya dapat bekerja dengan sinkron pada robot. Metode koordinasi yang digunakan pada penelitian ini adalah *Subsumption Architecture* yang telah dikemukakan oleh Brooks (1986). Gambar 6 menunjukkan struktur koordinasi *behavior* robot.

Dari gambar nampak bahwa *Wandering* merupakan *behavior* dengan level terendah, sehingga

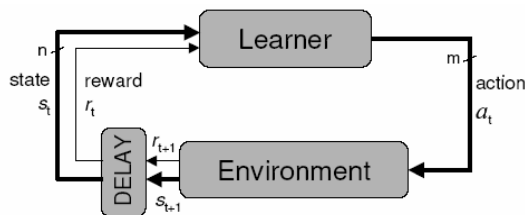
jika ada *behavior* lain yang aktif, maka *wandering* tidak akan aktif. *Behavior* dengan level prioritas tertinggi ialah *obstacle avoidance*. Hal ini berarti jika *behavior* tersebut aktif, maka semua *behavior* lain akan non aktif.



Gambar 6. *Subsumption Architecture* untuk robot dengan navigasi otonom

2.4 Algoritma Fuzzy Q Learning pada Robot

Reinforcement learning ialah metode pembelajaran yang bersifat *unsupervised* dengan mengandalkan *reward* yang diperoleh *agent* (misal : robot) dari lingkungannya. Metode ini sederhana dan efektif untuk proses yang berlangsung secara *online* dan cepat seperti yang terjadi pada robot. Gambar 7 menunjukkan skema dasar *reinforcement learning*.



Gambar 7. Skema dasar *reinforcement learning* (Perez, 2003)

Salah satu jenis *reinforcement learning* yang paling populer ialah *Q learning* yang bersifat sederhana, konvergen dan *off policy*. Hal ini membuatnya cocok untuk aplikasi *real time* seperti robot. Algoritma *Q Learning* ialah sebagai berikut.

```

Initialize  $Q(s, a)$  arbitrarily
Repeat (for each episode) :
  Initialize  $s$ 
  Repeat (for each step of episode):
    Choose  $a$  from  $s$  using policy
    derived from  $Q$  (e.g.,
     $\epsilon$ -greedy)
    Take action  $a$ , observe  $r, s'$ 
    Apply
     $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ 
     $s \leftarrow s'$ 
  until  $s$  is terminal
  
```

dimana :

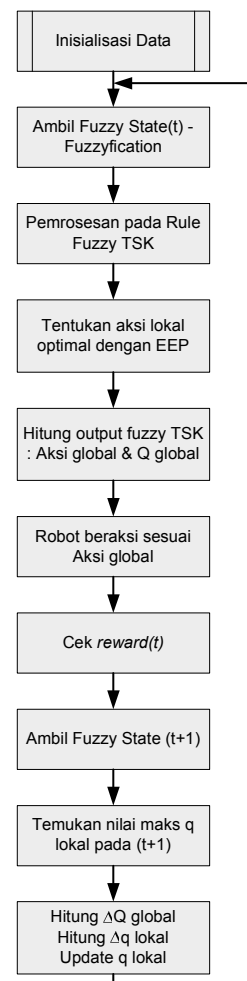
$Q(s, a)$: component of Q table (state, action)
 s : state s' : next state

a : action a' : next action
 r : reward α : learning rate
 γ : discount factor

Generalisasi *Q learning* dilakukan ketika bekerja dengan *state* dan aksi kontinyu seperti pada robot. Dalam kasus ini tabel fungsi Q akan membesar untuk menampung pasangan *state* aksi yang baru. Pada kasus ini, pembelajaran memerlukan waktu yang sangat lama dan keperluan memori yang besar. Akibatnya, metode ini sulit diterapkan. Penggunaan logika *fuzzy* untuk memodifikasi *Q-learning* agar dapat bekerja pada *state* dan aksi yang kontinyu didasarkan pada :

- *Fuzzy Inference System* merupakan aproksimator yang universal dan kandidat yang bagus untuk menyimpan nilai Q
- pengetahuan awal dapat dimasukkan dalam aturan *fuzzy*, sehingga dapat mengurangi waktu pelatihan

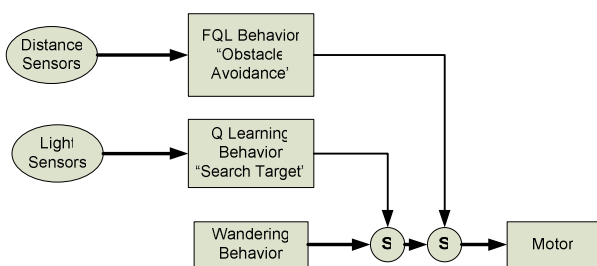
Pada *Fuzzy Q-learning*, pembelajaran tidak dilakukan setiap *state* pada *state space*. Oleh karena itu diperlukan optimasi pada beberapa *state* yang mewakili. Dengan demikian, digunakan interpolasi *fuzzy* untuk memperkirakan *state* dan aksi. (Glennedk, 1997). Gambar 8 menunjukkan *flow chart* dari algoritma FQL.



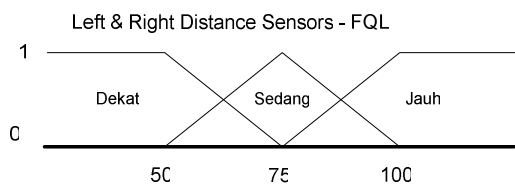
Gambar 8. *Flow chart* umum dari *fuzzy Q learning*

Penerapan *Fuzzy Q Learning* (FQL) pada penelitian ini hanya akan digunakan pada *obstacle avoidance behavior* saja. Hal ini dikarenakan *search target behavior* memiliki kecenderungan untuk bersifat acak dalam mencari target dan tidak memerlukan pembelajaran yang kompleks. Gambar 9 menunjukkan skema penerapan FQL *behavior* dengan arsitektur *subsumption architecture*.

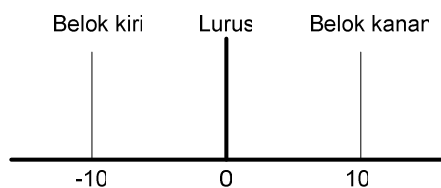
Berikutnya akan dilakukan pengaturan dari *membership function* sensor jarak kiri dan kanan. Akan digunakan MF segitiga yang digambarkan pada Gambar 10. Sedangkan kemungkinan aksi yang dihasilkan hanya ada 3 macam yaitu belok kiri, lurus, dan belok kanan, yang tampak pada Gambar 11.



Gambar 9. Skema *FQL behavior* pada robot dengan *subsumption architecture*



Gambar 10. *Membership function* dari sensor jarak kiri dan kanan – FQL



Gambar 11. Tiga kemungkinan aksi FQL

Sedang *fuzzy TSK rule base* yang digunakan dijelaskan berupa uraian 9 *rule*. Dalam bentuk tabel sederhana, 9 *rule* tersebut dapat diwujudkan sebagai berikut.

Tabel 1. *Rule Base* sederhana *fuzzy TSK*

	NF1	NF2	NF3
MF1	1	2	3
MF2	4	5	6
MF3	7	8	9

3. Hasil dan Pembahasan

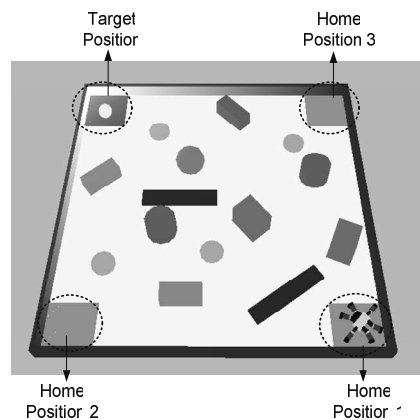
3.1 Simulasi *Behavior based architecture* pada *hexapod*

Untuk mensimulasikan kemampuan robot dalam melakukan navigasi otonom dibutuhkan arena dan halangan yang tepat. Gambar 12 menunjukkan arena beserta *obstacle* yang akan digunakan dalam simulasi. Pada gambar nampak tiga buah *home position* yang berbeda, juga 1 buah *target position* pada arena.

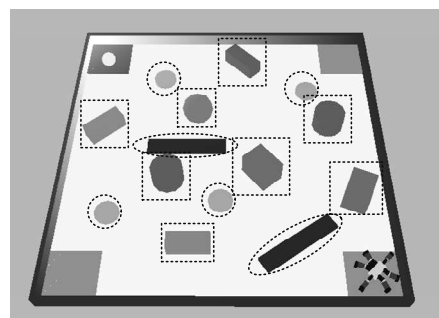
Target yang digunakan berupa lampu (sumber cahaya). Sedangkan hambatan (*obstacle*) yang digunakan pada simulasi ini di bedakan menjadi 2 macam :

- *Low obstacle (LO)* : *obstacle* yang rendah dan dapat dilewati/dilangkahi oleh robot.
- *High obstacle (HO)* : *obstacle* yang tinggi dan harus dihindari oleh robot.

Pada Gambar 13 nampak bahwa *LO* ditandai dengan lingkaran (atau elips) putus – putus, sedang *HO* ditandai dengan kurva persegi empat putus – putus.



Gambar 12. Arena simulasi beserta *home position* dan *target position*



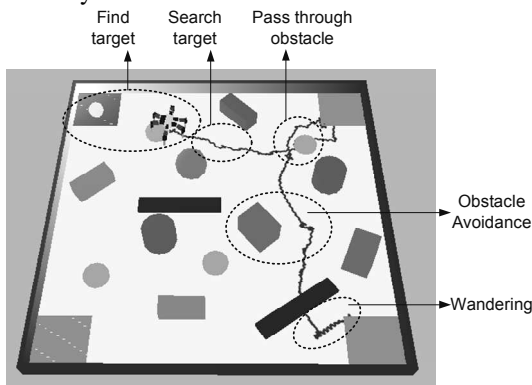
Gambar 13. Gambar arena dan beberapa jenis *obstacle*

Seperti telah dijelaskan sebelumnya, berikut ini *behaviors* yang dimiliki robot :

- Berkeliling (*wandering*)
- Hindari halangan (*obstacle avoidance*)
- Cari target (*search target*)
- Stop (*find target*)

Jika keempat *behavior* tersebut dijalankan secara terpisah, maka masing – masing akan dapat berjalan

dengan baik. Gambar 14 menunjukkan hasil simulasinya.



Gambar 14. Perilaku robot berjalan dengan baik

3.2 Simulasi algoritma fuzzy Q learning pada hexapod

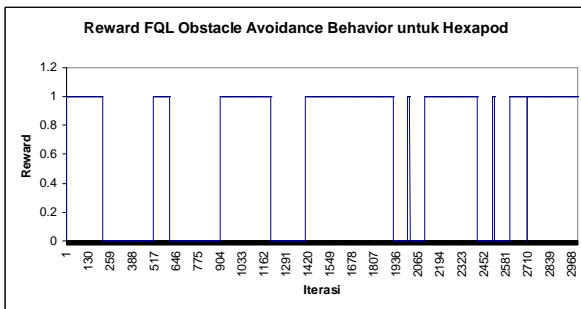
Pada simulasi ini akan ditunjukkan pengaruh fuzzy Q learning dalam memaksimalkan reward di behavior yang diproses (obstacle avoidance) dan penerapannya dalam behavior keseluruhan dari robot berkaki.

3.2.1 Simulasi FQL Obstacle Avoidance Behavior

Berikut ini hasil simulasi dari FQL obstacle avoidance behavior pada hexapod robot untuk 3000 kali iterasi. Adapun perancangan reward ialah sebagai berikut :

- $r = 1$, jika sensor kiri ≤ 100 dan sensor kanan ≤ 100
- 0 , jika sensor kiri ≤ 100 dan sensor kanan > 100 atau sensor kanan ≤ 100 dan sensor kiri > 100
- -1 , jika sensor kiri > 100 dan sensor kanan > 100

Hasil simulasi nampak pada Gambar 15.

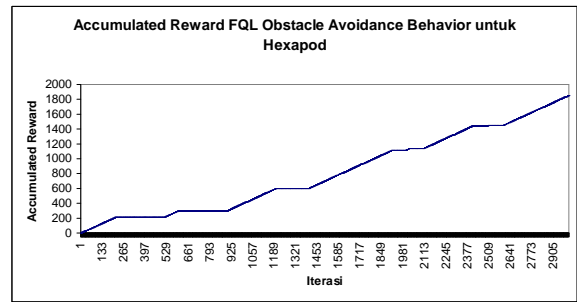


Gambar 15. Grafik reward yang diterima robot untuk FQL obstacle avoidance behavior

Dari gambar nampak bahwa robot senantiasa menerima reward positif. Hal ini berarti robot tidak pernah mengalami kondisi rawan tabrakan.

Gambar 16 menunjukkan grafik akumulasi reward yang diterima robot untuk obstacle avoidance behavior. Dari gambar nampak bahwa akumulasi reward yang diterima robot terus meningkat seiring waktu. Dalam 3000 iterasi robot memperoleh sekitar

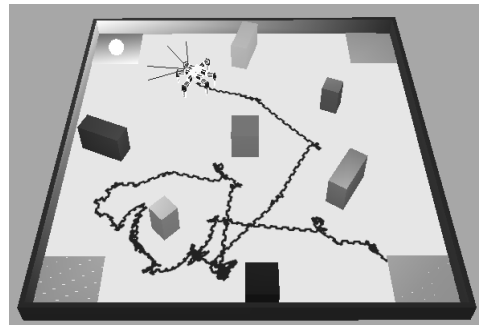
1852 satuan reward, sehingga dalam 1 iterasi rata – rata robot menerima 0.62 satuan reward.



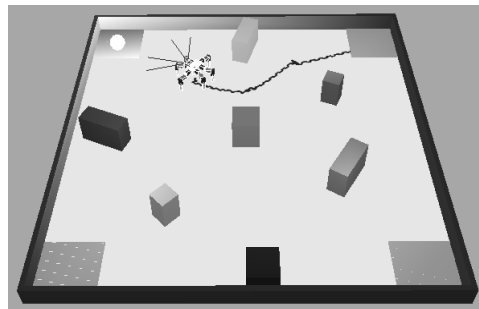
Gambar 16. Grafik accumulated reward yang diterima robot untuk FQL obstacle avoidance behavior

3.2.2 Simulasi FQL pada Hexapod Robot

Untuk menguji keseluruhan behavior pada robot, maka akan digunakan aplikasi navigasi otonom pada robot. Gambar 17 sampai 19 menunjukkan hasil simulasi yang diberikan jika robot dengan FQL behavior diletakkan pada tiga posisi awal yang berbeda.

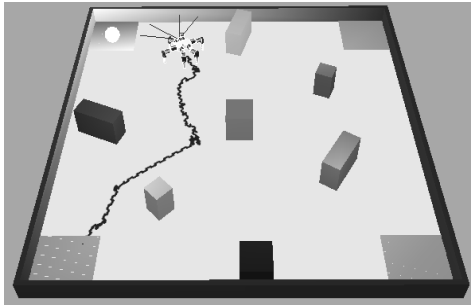


Gambar 17. Hasil simulasi FQL pada hexapod dari home position 1



Gambar 18. Hasil simulasi FQL pada hexapod dari home position 2

Dari ketiga gambar tersebut nampak bahwa hexapod robot dapat melakukan tugasnya dengan baik. Hanya saja pada Gambar 17, trayektori yang ditempuh robot cukup kasar akibat banyaknya halangan yang harus dihindari.



Gambar 19. Hasil simulasi FQL pada *hexapod* dari *home position 3*

4. Kesimpulan

Pada penelitian ini telah diterapkan algoritma *fuzzy Q learning* pada *hexapod robot* yang dirancang berdasar arsitektur *behavior based robot*. Dari hasil simulasi nampak bahwa *behavior coordination* dari robot telah berjalan dengan baik dan dapat mencapai tujuannya untuk menghindari halangan dan menemukan target.

Adapun penerapan *fuzzy Q learning* dapat membantu robot dalam mengantisipasi hal – hal tak terduga di lingkungan robot, meski juga memperlama robot untuk mencapai tujuannya. Algoritma ini memperbaiki kemampuan *q learning* dalam menghadapi situasi *state/action* yang kontinyu dan lebih kompleks. Jumlah rata – rata *reward* yang diterima robot dalam 1 iterasi adalah 0.62 satuan *reward*.

5. Acknowledgement

Penelitian ini didukung oleh *Japan International Cooperation Agency (JICA)* melalui *Technical Cooperation Project for Research and Education Development on Information and Communication Technology* di Institut Teknologi Sepuluh Nopember (PREDICT - ITS).

Daftar Pustaka

- [1] Bekey G. A., *Autonomous Robot : From Biological Inspiration to Implementation and Control*, MIT Press, 2005.
- [2] Billah M. M., Ahmed, M., Farhana, S., "Walking Hexapod Robot in Disaster Recovery : Developing Algorithm for Terrain Negotiation and Navigation", *Proc. of World Academy of Science, Engineering and Technology*, vol. 32, pp. 334 – 339, 2008.
- [3] Nonami K, Huang QJ, Komizo D, Shimoi N, Uchida H., "Humanitarian Mine Detection Six-Legged Walking Robot", *Proc. of The Third International Conference on Climbing and Walking Robots*, p. 861–868, 2000.
- [4] Bares, J.E., Wettergreen, D.S., "Dante II : Technical Description, Results, and Lesson Learned", *The International Journal of Robotic Research*, vol. 18, no. 7, pp. 621 – 649, 1999.
- [5] Brooks R., "A Robust Layered Control System For a Mobile Robot", *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14 – 23, 1986.

- [6] Brooks R., "A Robot that Walks : Emergent Behaviors from Carefully Evolved Network", *Neural Computation*, vol. 1, no. 2, pp. 253 – 262, 1989.
- [7] Glorennec, P. Y., "Reinforcement Learning : An Overview", *Proc. of European Symposium on Intelligent Techniques*, 2000.
- [8] Watkins, C., Dayan, P., "Q-learning, Technical Note", *Machine Learning*, Vol 8, pp.279-292, 1992.
- [9] Perez M.C., "A Proposal of Behavior Based Control Architecture with Reinforcement Learning for an Autonomous Underwater Robot", *Tesis Ph.D.*, University of Girona, 2003.
- [10] Deng, C., Er, M.J. dan Xu, J., "Dynamic Fuzzy Q-learning and Control of Mobile Robots", *8th International Conference on Control, Automation, Robotics and Vision*, Kunming, China, 2004a.
- [11] Deng, C. dan Er, M.J., "Real Time Dynamic Fuzzy Q-learning and Control of Mobile Robots", *Proceedings of 5th Asian Control Conference*, vol. 3, hal. 1568- 1576, 2004b.
- [12] Jouffle, L., "Fuzzy Inference System Learning by Reinforcement Methods", *IEEE Transactions on System, Man, and Cybernetics – Part C : Applications and Reviews*, Vol. 28, No. 3, hal. 338 – 355, 1998.
- [13] Glorennec, P.Y. dan Jouffle, L., "Fuzzy Q-learning", *Proceeding of the sixth IEEE International Conference on Fuzzy Sistem*, Vol. 2, No. 1, hal. 659 – 662, 1997.