

Penerapan *Behavior Based Architecture* dan *Q Learning* pada Sistem Navigasi Otonom *Hexapod Robot*

Handy Wicaksono^{1,2}, Prihastono^{1,3}, Khairul Anam⁴, Rusdhianto Effendi², Indra Adji Sulistijono⁵,
Son Kuswadi⁵, Achmad Jazidie², Mitsuji Sampei⁶

¹ Department of Electrical Engineering, Sepuluh Nopember Institute of Surabaya, Indonesia

² Department of Electrical Engineering, Petra Christian University, Surabaya, Indonesia
(Tel : +62-31-2983115; e-mail : handy@petra.ac.id)

³ Department of Electrical Engineering, University of Bhayangkara, Surabaya, Indonesia

⁴ Department of Electrical Engineering, University of Jember, Jember, Indonesia

⁵ Department of Mechatronics, Electronics Engineering Polytechnic Institute of Surabaya, Surabaya, Indonesia

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

Abstrak

Hexapod robot banyak digunakan karena kestabilan dan fleksibilitas pola geraknya. Pada penelitian ini akan didesain hexapod dengan arsitektur behavior based yang bersifat cepat dan reaktif terhadap masukan dari dunia luar. Selain itu akan diterapkan Q learning sebagai algoritma pembelajaran robot, sehingga robot dapat mengantisipasi hal – hal tak terduga di lingkungannya.

Dari hasil simulasi nampak bahwa penerapan arsitektur behavior based dan algoritma pembelajaran Q learning berhasil digunakan untuk sistem navigasi otonom robot yang bertujuan untuk menemukan target berupa sumber cahaya.

Kata kunci : *hexapod robot, behavior based architecture, 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, menghindari, 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).

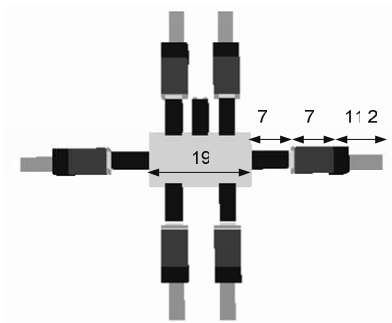
Ada berbagai metode untuk penyelesaian masalah *reinforcement learning*, salah satu yang paling populer ialah *Q Learning Algorithm* (Watkins, 1989). Kelebihan dari *Q Learning* ialah sifatnya yang *off policy* (dapat mengikuti policy apapun), algoritma yang sederhana, dan konvergen terhadap *optimal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

Pada penelitian ini akan dirancang *hexapod robot* dengan arsitektur *behavior based*. Kemudian juga akan ditambahkan *Q learning* sebagai mekanisme pembelajaran robot. Robot akan melakukan navigasi otonom untuk menghindari halangan dan menemukan target berupa sumber cahaya.

2. Metodologi Penelitian

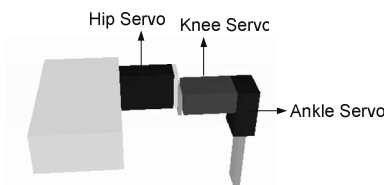
2.1 Mekanik dan Mekanisme Gerak Robot

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



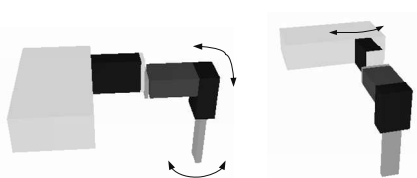
Gambar 1. Ukuran *hexapod robot*

Masing – masing kaki robot terdiri dari 3 buah motor servo yang diberi nama : *hip servo*, *knee servo* dan *ankle servo*. Konstruksi potongan salah satu kaki dapat dilihat pada gambar di bawah.



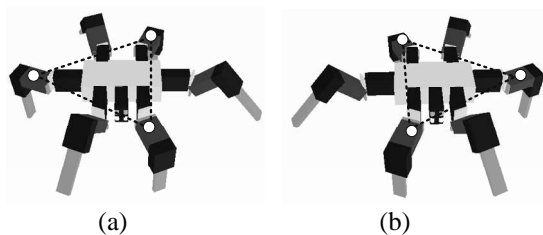
Gambar 2. Bagian – bagian pembentuk kaki robot

Arah gerakan *knee servo* dan *ankle servo* ialah rotasional dengan arah vertikal, sedang arah gerakan *hip servo* ialah rotasional dengan arah horizontal. Keduanya nampak pada gambar di bawah.



Gambar 3. Arah gerakan *knee servo* & *ankle servo*

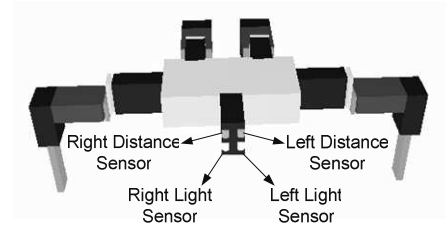
Mekanisme gerak dari *hexapod robot* ini menggunakan *tripod gait*, yang mendasarkan kestabilan robot pada tiga titik segitiga berupa kaki – kaki robot yang berada di tanah dalam 1 waktu. Pada gambar berikut nampak 2 buah konfigurasi segitiga saat robot berjalan. Sehingga saat robot berjalan, selalu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



Gambar 4. Segitiga kestabilan robot

2.2 Sensor – Sensor pada Robot

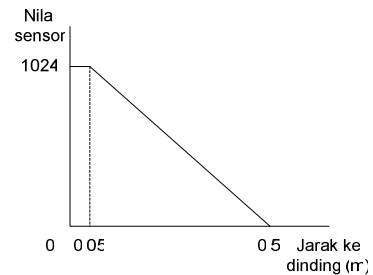
Sensor yang digunakan dalam sistem navigasi otonom *hexapod robot* ini ialah dua buah sensor jarak dan dua buah sensor cahaya. Konfigurasi sensor dapat dilihat pada gambar di bawah.



Gambar 5. Peletakan sensor pada robot

Dari gambar di atas nampak bahwa sensor yang digunakan ialah 2 buah sensor jarak (kiri dan kanan) serta 2 buah sensor cahaya (kiri dan kanan).

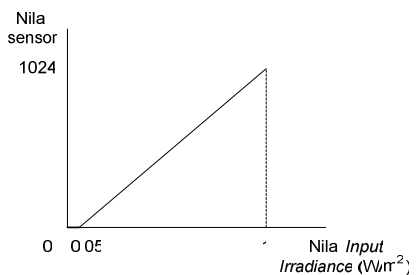
Karakteristik dari sensor jarak nampak pada grafik berikut.



Gambar 6. 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*.

Sedang karakteristik dari sensor cahaya nampak pada grafik berikut.



Gambar 7. Grafik karakteristik sensor cahaya

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

2.3 Macam – Macam *Behavior* pada Robot

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

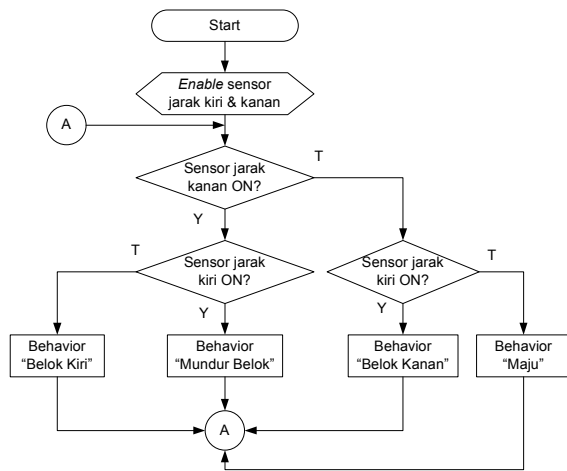
1. Berkeliling (*wandering*)
2. Hindari halangan (*obstacle avoidance*)

3. Cari target (*search target*)
4. Stop (*find target*)

Berikut ini penjelasan masing – masing penjelasan *behavior*.

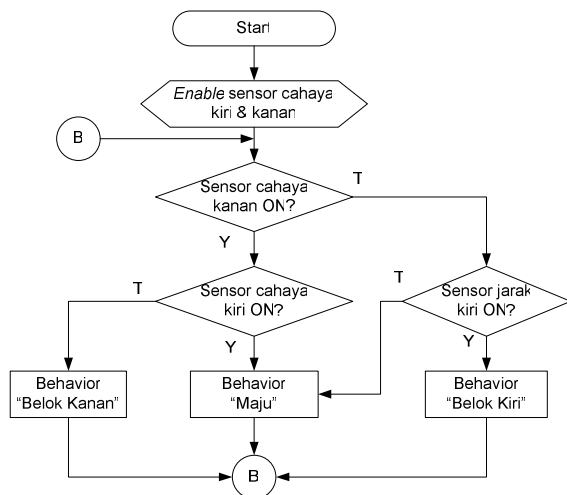
Karena robot perlu berkeliling di arena untuk menemukan target, maka *wandering behavior* sangat diperlukan. Pada prinsipnya, jika tidak ada halangan dan target, maka robot akan bergerak maju, belok kiri, maju dan belok kanan selama waktu tertentu. Tanpa *behavior* ini, robot hanya akan berjalan maju, ataupun menyusuri dinding arena saja.

Sedang *obstacle avoidance behavior* berguna untuk menghindari halangan yang dideteksi oleh 2 buah sensor jarak milik robot. Berikut ini *flow chart* dari *behavior* ini.



Gambar 8. *Flow chart* dari *obstacle avoidance behavior*

Jika sensor cahaya robot mendeteksi target berupa sumber cahaya, maka *search target behavior* akan aktif dan robot akan bergerak mendekati sumber cahaya itu. Jika jarak robot sudah “dekat” dengan target, maka *stop behavior* akan aktif dan robot akan berhenti. Berikut ini *flow chart* dari *search target behavior*.

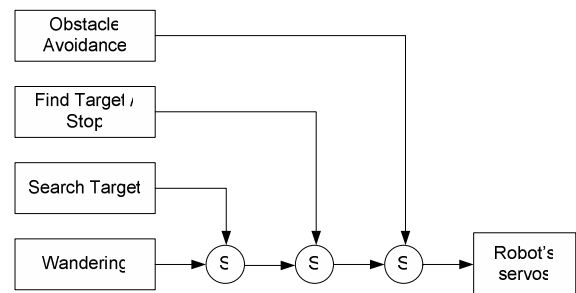


Gambar 9. *Flow chart* dari *search target behavior*

2.4 Behavior Coordination pada Robot

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). Berikut ini struktur koordinasi *behavior* dari robot ini.

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 10. *Subsumption Architecture* untuk robot dengan navigasi otonom

2.5 Algoritma Q Learning pada Robot

Untuk melakukan pembelajaran pada robot digunakan *Q learning*, salah satu jenis *reinforcement learning* yang paling populer. Sifatnya yang sederhana, konvergen dan *off policy* membuat *Q learning* cocok untuk aplikasi *real time* seperti robot. Algoritma *Q Learning* ialah sebagai berikut.

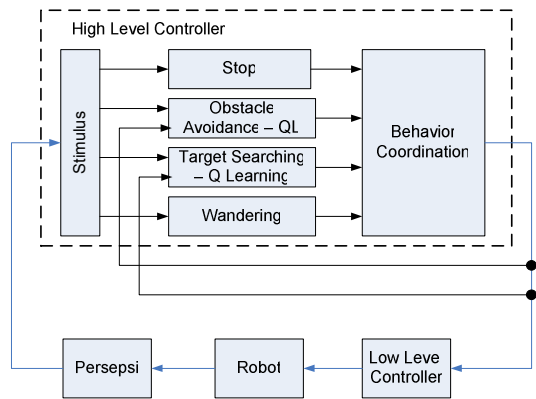
```

Initialize Q(s,a) arbitralily
Repeat (for each episode) :
  Initialize s
  Repeat (for each step of episode):
    Choose a from s using policy
      derived from Q (e.g.,
    ε-greedy)
    Take action a, observe r, s'
    Apply
    Q(s,a) ← Q(s,a) + α[r + γ max_{a'} Q(s',a') - Q(s,a)]
    s ← 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

Algoritma ini akan diterapkan pada dua *behavior*: *obstacle avoidance* dan *search target*. Berikut ini diagram blok lengkap dari robot dengan penerapan keduanya.

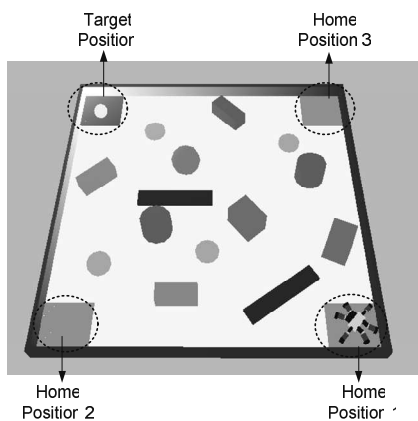


Gambar 11. Diagram blok robot beserta *Q learning behavior*

3. Hasil dan Pembahasan

3.1 Simulasi penerapan *behavior based architecture*

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

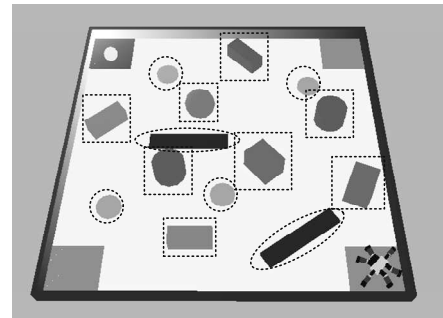


Gambar 12. Arena simulasi beserta *home position* dan *target position*

Target yang digunakan berupa lampu (sumber cahaya). Sedang 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 nampak bahwa *LO* ditandai dengan lingkaran (atau elips) putus – putus, sedang *HO* ditandai dengan kurva persegi empat putus – putus.



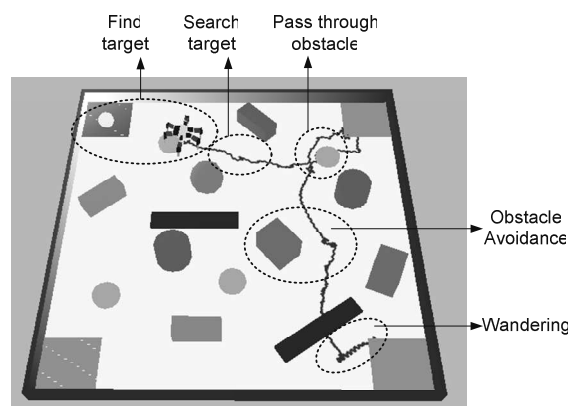
Gambar 13. Gambar arena dan jenis – jenis *obstacle* yang digunakan

3.1.1 Simulasi masing – masing *behavior*

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. Berikut ini hasil simulasi yang menunjukkan hal tersebut.



Gambar 14. Perilaku robot berjalan dengan baik

Dari hasil di atas dapat dilihat bahwa *wandering behavior* akan bekerja saat robot jauh dari halangan dan target. Keuntungan dari *wandering behavior* ialah robot dapat :

- Mencari target lebih efektif
- Melepaskan diri saat robot tersangkut pada *obstacle*

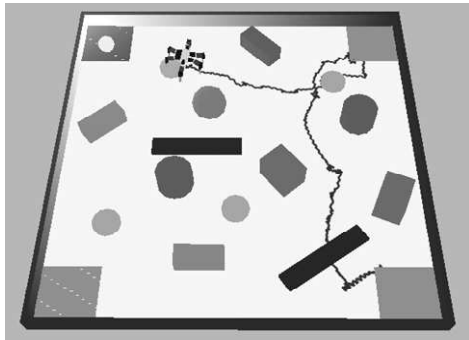
Jika ada *high obstacle* maka *obstacle avoidance behavior* yang akan aktif. Jika ada *low obstacle* maka robot hanya akan melewatinya saja.

Sedang jika robot sudah mendekati robot dan mendeteksi adanya cahaya, maka *search target behavior* akan aktif dan robot akan bergerak menuju sumber cahaya. Jika posisi sudah dekat, maka robot akan berhenti karena telah menemukan robot.

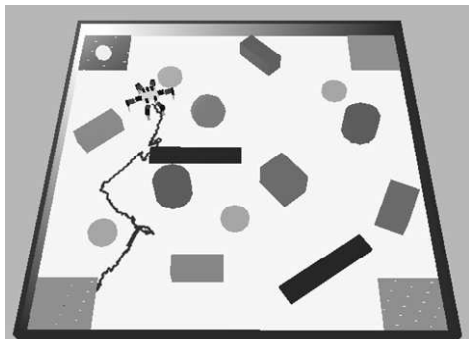
3.1.2 Simulasi keseluruhan *behavior*

Pada simulasi ini akan diuji apakah robot dapat menemukan target secara otonom sekaligus

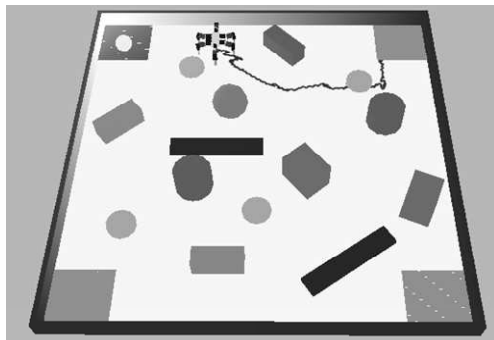
menghindari *high obstacle* dalam prosesnya, jika robot diletakkan dari *home position* yang berbeda – beda. Berikut ini hasil simulasi dengan 3 *home position* yang berbeda.



Gambar 15. Hasil simulasi navigasi otonom robot dari *home position 1*



Gambar 16. Hasil simulasi navigasi otonom robot dari *home position 2*



Gambar 17. Hasil simulasi navigasi otonom robot dari *home position 3*

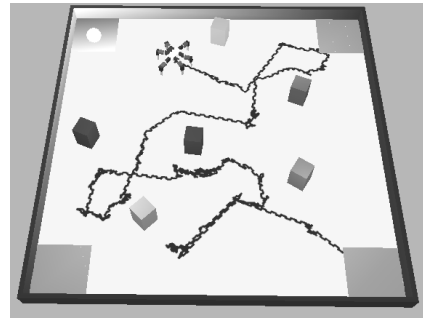
Dari gambar di atas nampak bahwa robot mampu menyelesaikan tugasnya untuk menemukan target dari 3 *home position* yang berbeda. Selain itu robot juga berhasil melewati *low obstacle* dan menghindari *high obstacle*.

3.2 Simulasi penerapan algoritma *Q learning* pada robot

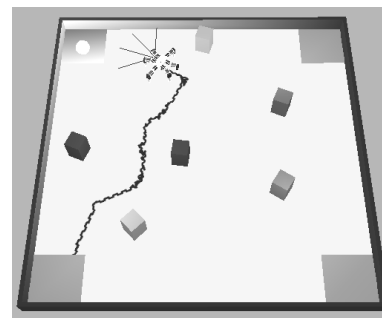
Pada simulasi ini, diterapkan algoritma *Q learning* hanya pada dua *behavior* pada robot : *obstacle avoidance* dan *search target*. Di sini robot tidak mengikuti suatu perintah untuk melakukan sesuatu,

robot hanya memperoleh informasi tentang *reward* yang diterima saat robot melakukan aksi tertentu.

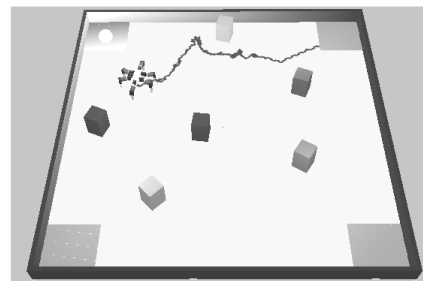
Berikut ini hasil simulasi yang diberikan jika robot diletakkan pada tiga posisi awal yang berbeda.



Gambar 18. Hasil simulasi penerapan *Q learning* dari *home position 1*



Gambar 19. Hasil simulasi penerapan *Q learning* dari *home position 2*



Gambar 20. Hasil simulasi penerapan *Q learning* dari *home position 3*

Dari gambar di atas nampak bahwa robot dapat menemukan target, meski diletakkan pada posisi yang berpindah – pindah. Keuntungan dari penggunaan *Q learning* ialah menambahkan kemampuan belajar dan mengantisipasi hal – hal tak terduga di lingkungan robot. Sedang kelemahannya ialah waktu pencapaian yang relatif lebih lama akibat kompleksnya algoritma yang berpengaruh pada waktu yang diperlukan robot untuk mengeksekusi program.

4. Kesimpulan dan Saran

Pada penelitian ini telah diterapkan algoritma *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 *Q learning* dapat membantu robot dalam mengantisipasi hal – hal tak terduga di lingkungan robot, meski juga memperlama robot untuk mencapai tujuannya.

Sebagai pengembangan ke depan, akan dieksplorasi pemanfaatan *Q learning* pada aplikasi – aplikasi lain misalnya penentuan jarak terpendek. Algoritma ini juga bisa dikembangkan dengan menggabungkannya dengan *fuzzy logic (fuzzy q learning)* untuk mendapatkan hasil pembelajaran yang lebih akurat.

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.