

# Perancangan Sistem Navigasi Otonom pada *Behavior Based Hexapod Robot*

Handy Wicaksono<sup>1,2)</sup>, Prihastono<sup>1,3)</sup>, Khairul Anam<sup>4)</sup>, Rusdhianto Effendi<sup>2)</sup>, Indra Adji Sulistijono<sup>5)</sup>, Son Kuswadi<sup>5)</sup>, Achmad Jazidie<sup>2)</sup>, Mitsuji Sampei<sup>6)</sup>

<sup>1</sup> Department of Electrical Engineering, Sepuluh Nopember Institute of Surabaya, Indonesia

<sup>2</sup> Department of Electrical Engineering, Petra Christian University, Surabaya, Indonesia

<sup>3</sup> Department of Electrical Engineering, University of Bhayangkara, Surabaya, Indonesia

<sup>4</sup> Department of Electrical Engineering, University of Jember, Jember, Indonesia

<sup>5</sup> Department of Mechatronics, Electronics Engineering Polytechnic Institute of Surabaya, Surabaya, Indonesia

<sup>6</sup> Department of Mechanical and Environmental Informatics,

Tokyo Institute of Technology, Tokyo, Japan

E-mail : handy@petra.ac.id

## ABSTRAK

Robot berkaki enam (*hexapod*) memiliki kelebihan dibanding robot beroda dalam hal kemampuannya melewati daerah tidak rata. Pada penelitian ini, mobilitas pergerakan *hexapod* akan diuji untuk mengetahui performanya dalam melewati balok dan tangga. Supaya dapat bereaksi dengan cepat, maka arsitektur *behavior based* akan digunakan pada *hexapod*. Aplikasi navigasi otonom dipilih untuk menunjukkan bahwa arsitektur tersebut berjalan dengan baik. Dari hasil simulasi nampak bahwa *behavior based hexapod robot* memiliki mobilitas yang baik (mampu melewati halangan setinggi maksimal 10 cm) dan dapat menyelesaikan tugasnya untuk menghindari halangan dan menemukan sumber cahaya.

**Kata kunci:** mobilitas, *hexapod robot*, *behavior based architecture*, sistem navigasi otonom

## ABSTRACT

*Six legged robot (hexapod) has advantage over wheeled robot in its capability to walk over rough terrain. In this paper, hexapod mobility will be tested in order to measure its performance in walk through beam and stair. Behavior based architecture will be used in hexpod, so it can react quickly. Autonomous navigation application has been chosen here in order to prove that the architecture is running well. From simulation result, it can be seen that behavior based hexapod robot has good mobility (it can walk through obstacle that has 10 cm height) and it can accomplish its task to avoid the obstacles and find the light source.*

**Keywords:** mobility, *hexapod robot*, *behavior based architecture*, autonomous navigation system

## 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 [1].

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* [2].

*Hexapod* telah digunakan untuk berbagai aplikasi, misalnya COMET, *hexapod* dengan dua buah lengan tambahan, digunakan untuk menjinakkan ranjau [3]. Contoh lain ialah DANTE, *hexapod* yang digunakan untuk penjelajahan gunung berapi Mount Spur, Alaska, untuk mengumpulkan dan mengirimkan data

---

**Catatan:** Diskusi untuk makalah ini diterima sebelum tanggal 1 Desember 2008. Diskusi yang layak muat akan diterbitkan pada Jurnal Teknik Elektro volume 9, nomor 1, Maret 2009.

ilmiah pada operator dan ahli gunung berapi di lokasi yang berjauhan [4].

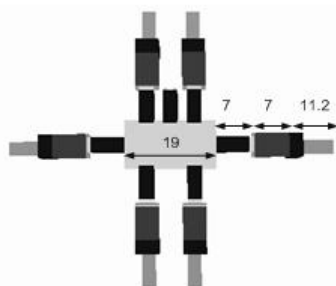
Dalam banyak aplikasi robot, sering kali dibutuhkan reaksi yang cepat dari robot. Arsitektur *behavior based control* merupakan arsitektur robot yang cocok karena memiliki struktur *behavior* horizontal yang bekerja bersama secara paralel, bersamaan dan asinkronus [5]. *Hexapod* pertama yang digunakan dengan arsitektur *behavior based* ialah Genghis [6].

Pada penelitian ini akan dirancang model simulasi *hexapod robot* dengan mobilitas yang lebih baik dari robot Genghis di atas. *Hexapod* ini memiliki kemampuan untuk melewati halangan berbentuk balok dan tangga. Arsitektur *behavior based* akan digunakan supaya robot dapat bereaksi dengan cepat terhadap input dari lingkungan. Robot akan dilengkapi dengan beberapa *behavior* sehingga dapat melakukan navigasi otonom untuk menghindari halangan dan menemukan target berupa sumber cahaya.

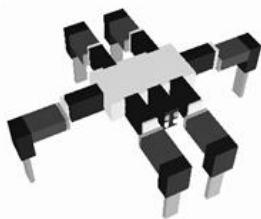
## PERANCANGAN SISTEM

### Mekanik dan Mekanisme Gerak Robot

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



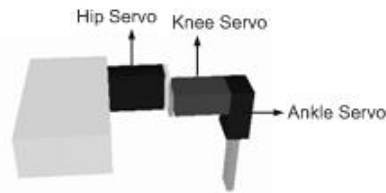
Gambar 1. Ukuran bagian *hexapod robot*



Gambar 2. *Hexapod robot* dalam kondisi inisial

Jika *hexapod robot* diatur dalam kondisi inisial (berdiri), maka bentuknya akan seperti gambar 2.

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 3.



Gambar 3. Bagian-bagian pembentuk kaki robot

Arah gerakan *knee servo* dan *ankle servo* ialah rotasional dengan arah vertikal seperti tampak pada gambar 4.

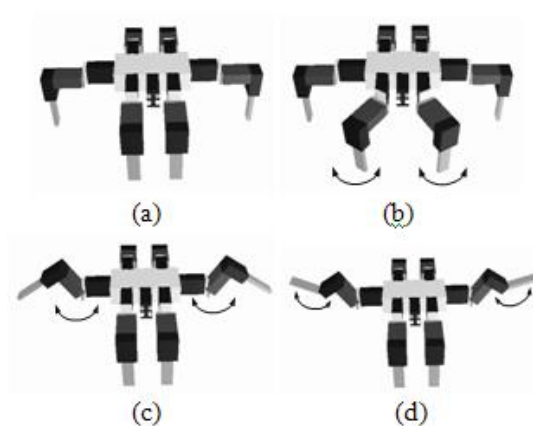


Gambar 4. Arah gerakan *knee servo* & *ankle servo*

Sedang arah gerakan *hip servo* ialah rotasional dengan arah horizontal seperti pada gambar 5.



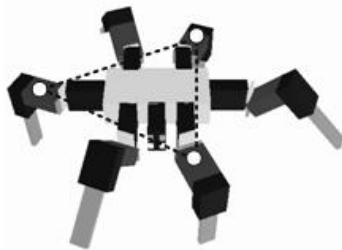
Gambar 5. Arah gerakan *hip servo*



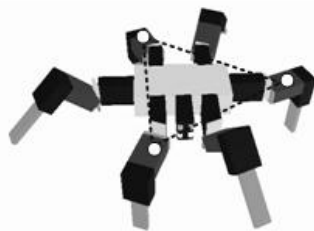
Gambar 6. Pergerakan servo dalam robot: (a) posisi robot mula-mula; (b) pergerakan *hip servo*; (c) pergerakan *knee servo*; (d) pergerakan *ankle servo*

Gerakan *hip servo*, *knee servo* dan *ankle servo* dalam robot secara keseluruhan dapat dilihat pada gambar 6.

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 7 dan gambar 8 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.



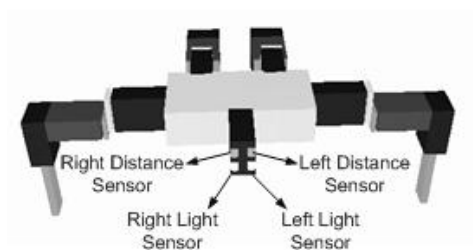
Gambar 7. Segitiga kestabilan pertama pada robot



Gambar 8. Segitiga kestabilan ke dua pada robot

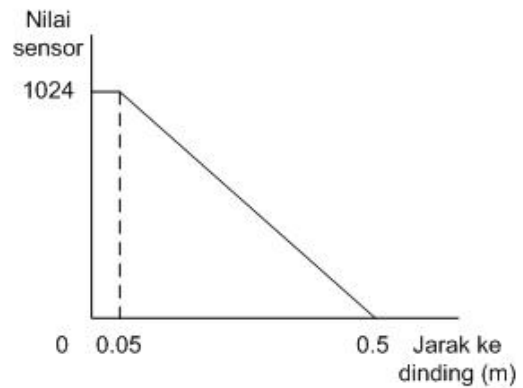
### Sensor-Sensor pada Robot

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 9.



Gambar 9. Peletakan sensor pada robot

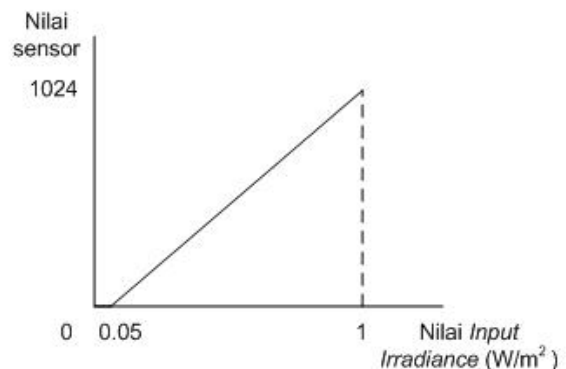
Karakteristik dari sensor jarak nampak pada grafik berikut.



Gambar 10. Grafik karakteristik sensor jarak

Grafik tersebut didapatkan dari karakteristik ideal sensor yang terdapat pada panduan dari program simulator yang digunakan. Sedangkan pengaturan nilai sensor dilakukan pada program yang sama. Dari Gambar 10 nampak bahwa jangkauan terjauh sensor ialah 0.5 m. Dalam simulasi ini, sensor diasumsikan ideal dan bebas *noise*.

Sedangkan karakteristik dari sensor cahaya (*infra red*) nampak pada grafik berikut.



Gambar 11. Grafik karakteristik sensor cahaya

Data yang digunakan sama dengan sensor jarak pada bagian sebelumnya. Dalam simulasi ini, sensor juga diasumsikan ideal dan bebas *noise*.

### Macam-Macam *Behavior* pada Robot

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

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

Berikut ini penjelasan masing–masing *behavior*.

Wandering behavior

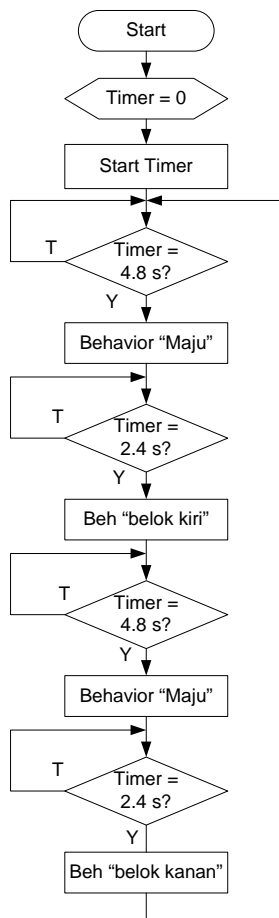
Karena robot perlu berkeliling di arena untuk menemukan target, maka *behavior* ini sangat diperlukan. Tanpa *behavior* ini, robot hanya akan berjalan maju, ataupun menyusuri dinding arena saja. Adapun *flow chart* dari *wandering behavior* dapat dilihat pada gambar 12.

Obstacle Avoidance Behavior

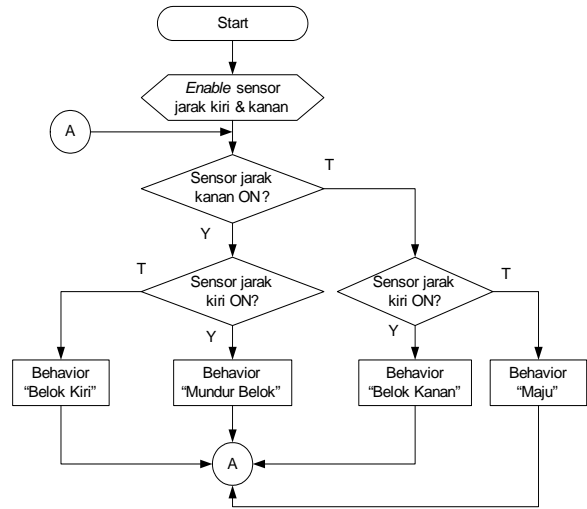
*Behavior* ini berguna untuk menghindari halangan yang dideteksi oleh 2 buah sensor jarak milik robot. *Flow chart* dari *behavior* ini dapat dilihat pada gambar 13.

Search target dan stop behavior

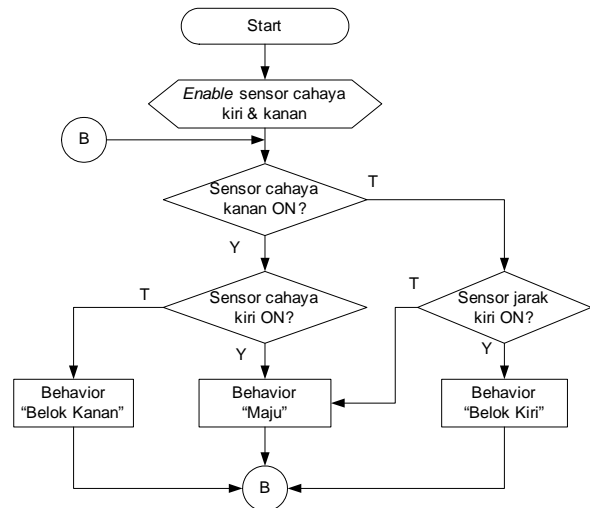
Jika sensor cahaya robot mendeteksi target berupa sumber cahaya, maka *search target behavior* akan aktif dan robot akan bergerak mendekati sumber cahaya itu. Jika jarak robot sudah “dekat” dengan target, maka *stop behavior* akan aktif dan robot akan berhenti. *Flow chart* dari kedua *behavior* tersebut dapat dilihat pada Gambar 14 dan 15.



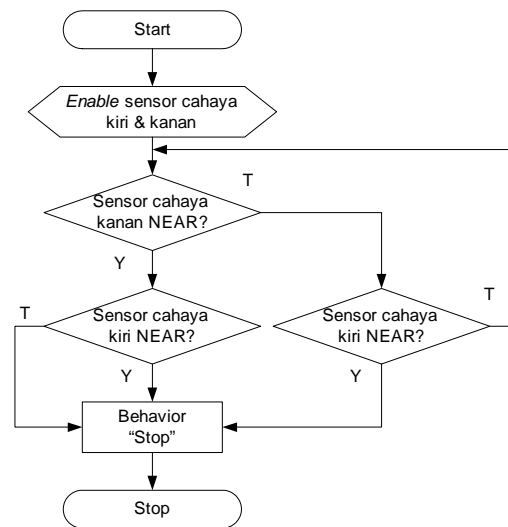
Gambar 12. *Flow chart* dari *wandering behavior*



Gambar 13. *Flow chart* dari *obstacle avoidance behavior*



Gambar 14. *Flow chart* dari *search target behavior*

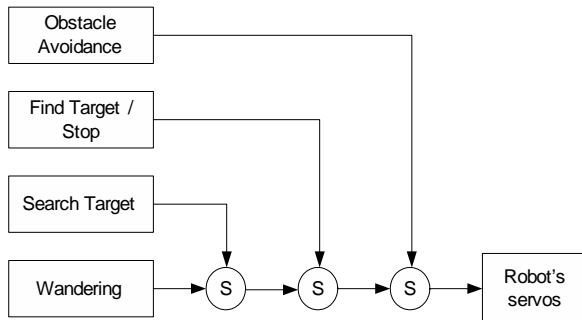


Gambar 15. *Flow chart* dari *find target behavior*

### 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[5]. Gambar 16 menunjukkan struktur koordinasi *behavior* dari robot ini.

Dari Gambar 16 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. Dengan kata lain masing – masing *behavior* akan aktif secara bergantian.



Gambar 16. *Subsumption Architecture* untuk robot dengan navigasi otonom

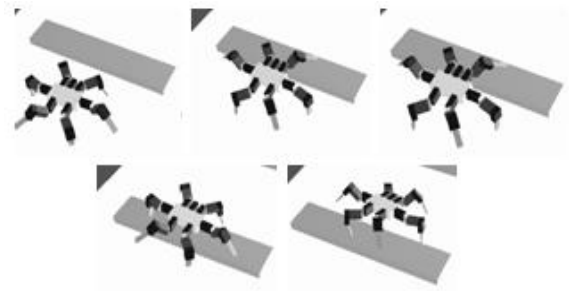
Mekanisme pergerakan robot dengan menggunakan *tripod gait* seperti yang telah dibahas pada bagian mekanik dan mekanisme gerak robot.

### HASIL PENGUJIAN

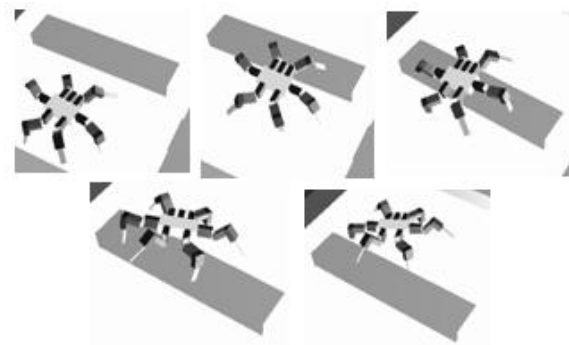
#### Simulasi robot melewati halangan

Seluruh simulasi robot pada penelitian ini dilakukan dengan program Webots dari Cyberbotics [7]. Pada simulasi ini, robot akan melewati "halangan rendah" dengan tinggi yang bervariasi. Halangan rendah di sini ialah halangan yang dapat dilewati robot, sehingga tidak perlu dihindari. Hal ini penting untuk menentukan berapa tinggi maksimal suatu halangan yang dapat dilewati robot.

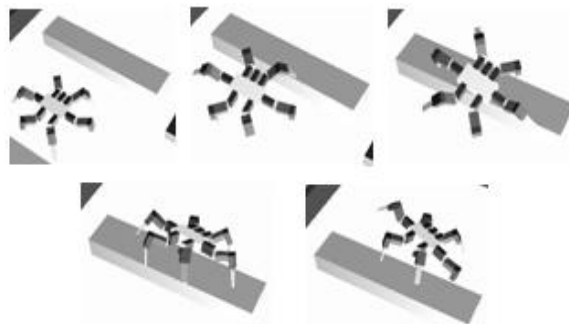
Robot akan diuji untuk melewati halangan setinggi 5, 10, dan 12 cm. Hasilnya nampak pada gambar 17, 18 dan 19.



Gambar 17. Simulasi robot melewati halangan rendah setinggi 5 cm



Gambar 18. Simulasi robot melewati halangan rendah setinggi 10 cm



Gambar 19. Simulasi robot melewati halangan rendah setinggi 12 cm

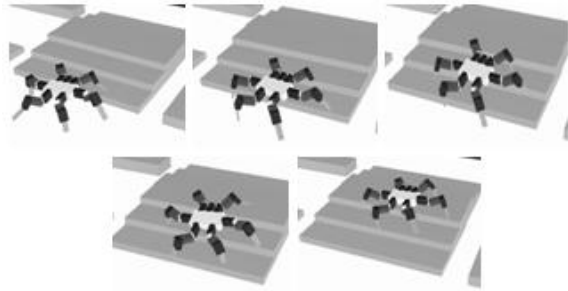
Pada ketiga gambar di atas, nampak bahwa robot berhasil melewati semua halangan. Namun untuk melewati halangan setinggi 12 cm, robot sering kali gagal. Dalam 10 kali percobaan, robot hanya berhasil melewatinya 1 kali saja. Hal ini berkaitan dengan keterbatasan kemampuan robot untuk mengangkat kakinya yang berkisar sekitar 10 cm.

Dengan demikian dapat disimpulkan bahwa robot mampu melewati dengan baik (selalu berhasil) halangan dengan tinggi maksimal 10 cm, di atas itu robot akan sering mengalami kegagalan.

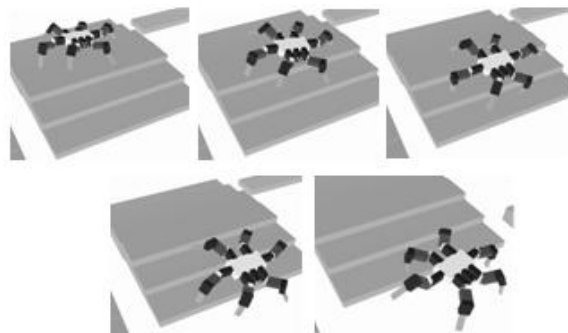
### Simulasi robot menaiki - menuruni tangga

Pada simulasi di atas, halangan yang digunakan ialah halangan tunggal. Pada simulasi berikut akan digunakan halangan berbentuk tangga.

Mula-mula robot diuji untuk melewati tangga dengan tinggi 5 cm dan lebar anak tangga sebesar 20 cm. Berikut gambar 20 dan 21 menunjukkan hasil simulasinya.

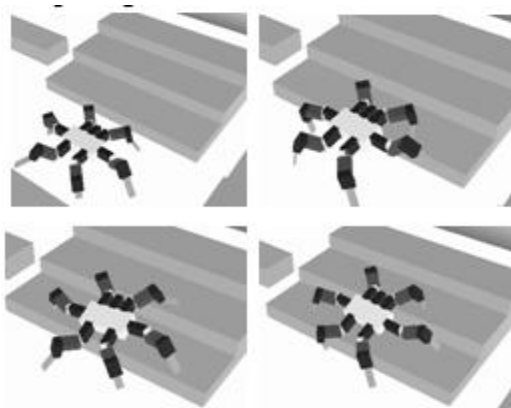


Gambar 20. Simulasi robot menaiki tangga dengan tinggi 5 cm dan lebar 20 cm



Gambar 21. Simulasi robot menuruni tangga dengan tinggi 5 cm dan lebar 20 cm

Pada uji ini robot berhasil menaiki dan menuruni tangga tersebut.

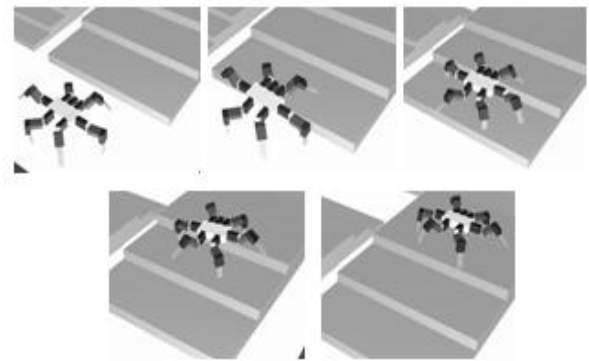


Gambar 22. Simulasi robot menaiki tangga dengan tinggi 8 cm dan lebar 25 cm

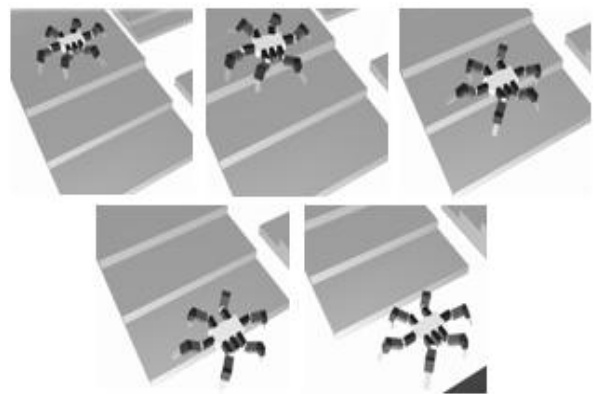
Jika ukuran tangga diubah menjadi tinggi 8 cm dan lebar 25 cm, maka hasil simulasinya akan nampak seperti gambar 22.

Dari gambar 22 di atas nampak bahwa robot gagal menaiki tangga karena jangkauan kaki yang kurang tinggi. Lebar anak tangga diperbesar menjadi 25 cm untuk mempermudah robot, namun hal tersebut tidak membawa perubahan.

Jika tinggi dikurangi menjadi 7 cm dan lebar ditambah menjadi 40 cm, maka hasil simulasinya nampak pada gambar 23 dan 24 berikut.



Gambar 23. Simulasi robot menaiki tangga dengan tinggi 7 cm dan lebar 40 cm

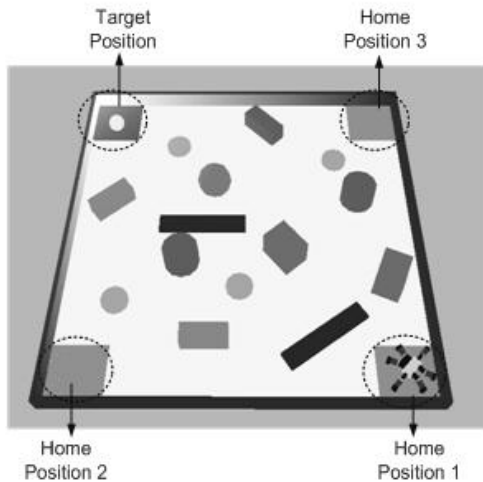


Gambar 24. Simulasi robot menuruni tangga dengan tinggi 7 cm dan lebar 40 cm

Dari gambar 23 dan 24 di atas, nampak bahwa robot berhasil menaiki dan menuruni tangga tersebut. Sehingga dapat kita simpulkan bahwa untuk menaiki tangga diperlukan lebar anak tangga yang cukup sehingga robot memiliki landasan untuk ke anak tinggi berikutnya. Adapun lebar anak tangga yang disarankan untuk ketinggian 7 cm ke atas (maksimal 10 cm) ialah minimal 40 cm.

### Simulasi robot dengan navigasi otonom

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 25 nampak tiga buah *home position* yang berbeda, juga 1 buah *target position* pada arena.

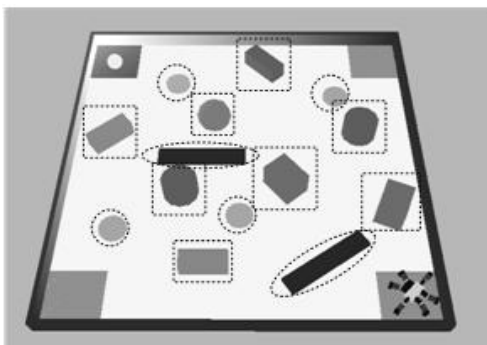


Gambar 25. Arena simulasi beserta *home position* dan *target position*

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 26 nampak bahwa *LO* ditandai dengan lingkaran (atau elips) putus-putus, sedang *HO* ditandai dengan kurva persegi empat putus-putus.



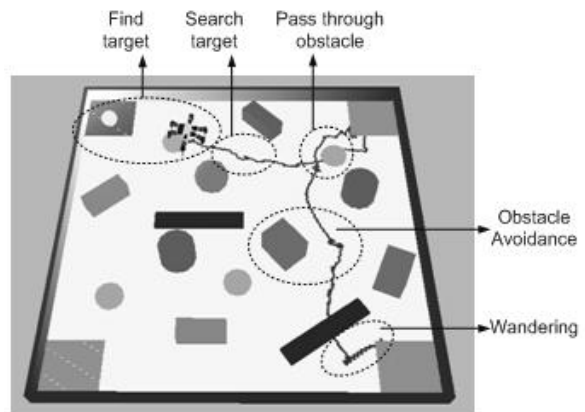
Gambar 26. Gambar arena dan jenis-jenis *obstacle* yang digunakan

### 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 gambar 27 memperlihatkan hasil simulasi yang menunjukkan hal tersebut.



Gambar 27. 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*

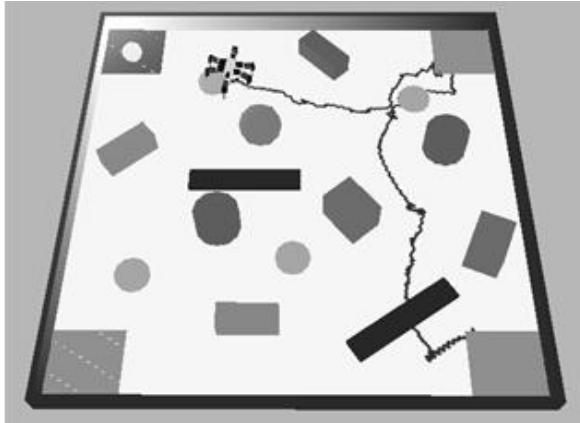
Jika ada *high obstacle* maka *obstacle avoidance behavior* yang akan aktif. Jika ada *low obstacle* maka robot hanya akan melewatinya saja.

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.

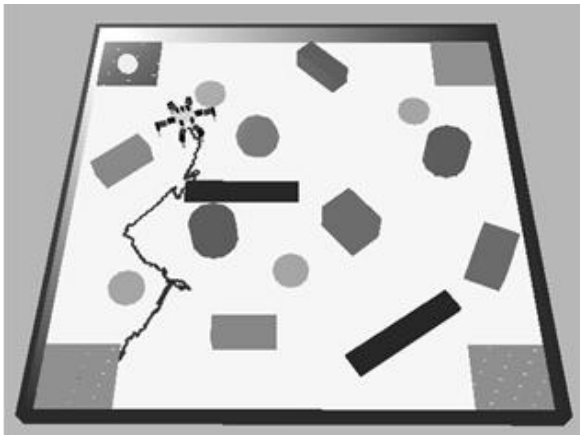
### Simulasi keseluruhan *behavior*

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

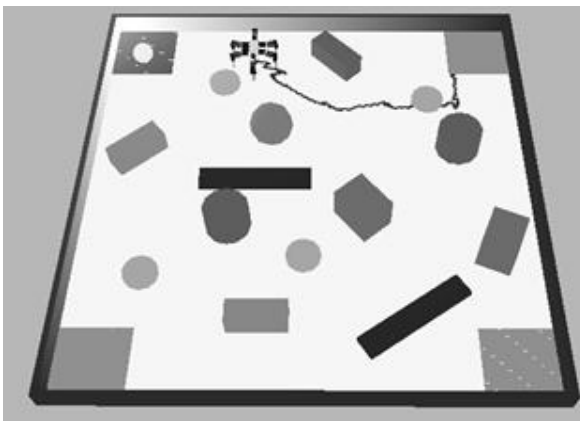
berbeda-beda. Berikut ini hasil simulasi dengan 3 *home position* yang berbeda (gambar 28, 29 dan 30).



Gambar 28. Hasil simulasi navigasi otonom robot dari *home position 1*



Gambar 29. Hasil simulasi navigasi otonom robot dari *home position 2*

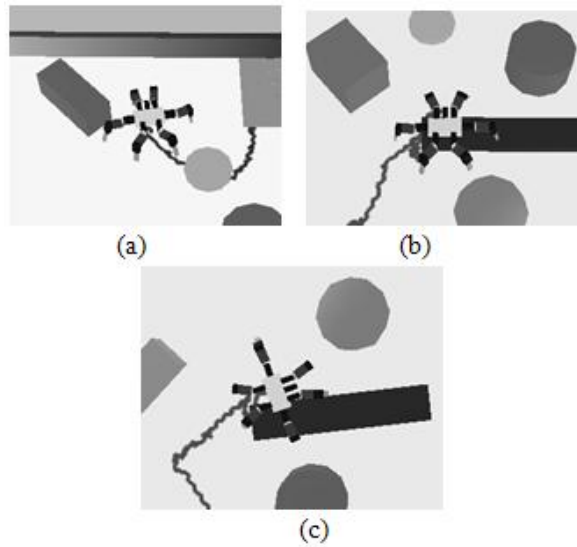


Gambar 30. Hasil simulasi navigasi otonom robot dari *home position 3*

Dari gambar 28 sampai 30 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*.

Namun demikian, robot tidak selalu berhasil dalam menjalankan tugasnya. Kadang-kadang robot akan mengalami terjepit (*stack*) pada kondisi-kondisi tertentu. Berikut ini contoh posisi robot saat mengalami *stack* (gambar 31).



Gambar 31. Berbagai posisi robot saat *stack*

Dari gambar 31 di atas nampak bahwa robot mengalami *stack* karena beberapa alasan :

- Robot terjebak di antara *high obstacle* yang membentuk cerukan dan dinding, sehingga robot tidak dapat menentukan ke arah mana harus berjalan
- Robot terjebak di antara 2 buah *high obstacle* yang berada di sebelah kanan dan kiri, dan pada waktu yang bersamaan 2 kaki belakang robot tersangkut pada *low obstacle*.
- 3 kaki robot terjebak pada *low obstacle* sedang di sebelah kiri depan robot terdapat *high obstacle*.

## KESIMPULAN DAN SARAN

Pada penelitian ini dirancang model simulasi *hexapod robot* berdasar arsitektur *behavior based robot*. Dari hasil simulasi nampak bahwa *hexapod* dapat melewati halangan maksimal setinggi 10 cm. Jika robot akan melewati tangga, maka lebar tangga minimal 40 cm. Selain itu dengan metode *behavior coordination* jenis *Subsumption Architecture*, navigasi otonom robot untuk menghindari halangan dan menemukan target dapat dicapai dengan baik dari berbagai posisi *start*.



Meski demikian, terkadang robot masih mengalami *stuck* (terjepit) pada hambatan tinggi dan rendah, sehingga perlu dilakukan langkah perbaikan di sisi perangkat keras robot. Selain itu sebagai pengembangan ke depan, penggunaan pembelajaran pada robot bisa ditambahkan supaya robot dapat mengantisipasi hal-hal yang tidak terduga.

#### 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] Cyberbotics, *Webots User Guide: Release 5.4.2*, 2007.