PERANCANGAN BEHAVIOR-BASED ROBOT DENGAN ALGORITMA FUZZY Q-LEARNING (FQL) PADA SISTEM NAVIGASI ROBOT OTONOM BERODA DALAM MEDAN YANG TIDAK TERSTRUKTUR

Made Santo Gitakarma[1], Gede Nurhayata[2]
[1,2]Jurusan Teknik Elektronika, Fakultas Teknik & Kejuruan
Universitas Pendidikan Ganesha
Singaraja, Indonesia

e-mail: santo@undiksha.ac.id, gede_nur@yahoo.co.id

Abstrak

Pada banyak aplikasi robotika, seperti sistem navigasi robot mandiri atau robot otonom yang bergerak dengan mandiri pada lingkungan tidak terstruktur, sangat sulit atau tidak mungkin memperoleh model matematik yang tepat dari interaksi robot dengan lingkungannya. Untuk itu diperlukan pendekatan sistem kendali robot yang dikenal dengan sistem kendali *Behavior-Based Robot* (BBR). Pada pendekatan ini, sistem diuraikan menjadi beberapa modul yang masing-masingnya bertanggung jawab untuk melakukan satu perilaku (*behavior*). Salah satu metode pembelajaran yang paling cocok untuk aplikasi robot adalah *Reinforcement Learning* (RL), dengan jenis algoritma Q-*learning*. Kombinasi Q-*learning* dengan *Fuzzy Inference System* (FIS) dikenal dengan nama *Fuzzy* Q-*Learning* (FQL). Berdasarkan percobaan yang dilakukan sebanyak 3 kali pada robot beroda dapat disimpulkan bahwa waktu rata-rata robot kembali ke *Homebase* yaitu 1 menit 10 detik. Sedangkan waktu rata-rata robot dalam mematikan api lilin adalah 2 detik. Sehingga dapat dikatakan robot yang dibuat mempunyai kinerja yang cukup baik.

Kata kunci: *fuzzy q-learning*, *reinforcement learning*, robot otonom, webots

Abstract

In many robotics applications, such as autonomous robot navigation systems or autonomous robots that move independently in unstructured environments, it is very difficult or impossible to obtain a precise mathematical model of the robot interaction with the environment. Approach is needed for the robot control system known as the control system Behavior-Based Robot (BBR). In this approach, the system is decomposed into several modules each of which is responsible for performing a behavior (behavior). One method of learning the most suitable for robotic applications is Reinforcement Learning (RL), with the type Q-learning algorithm. The combination of Q-learning to Fuzzy Inference System (FIS) is known by the name of Fuzzy Q-Learning (FQL). Based on the experiments performed 3 times on a wheeled robot can be concluded that the average time down the back to home base, which is 1 minute 10 seconds. While the average time of deadly robots in a candle flame is 2 seconds. So it can be said that robots have made a pretty good performance.

Keywords : fuzzy, q-learning, reinforcement learning, autonomous robots, webots

PENDAHULUAN

Pada banyak aplikasi robotika, seperti sistem navigasi robot mandiri atau robot otonom yang bergerak dengan mandiri pada lingkungan tidak terstruktur, sangat sulit atau tidak mungkin memperoleh model matematik yang tepat dari interaksi robot dengan lingkungannya. Bahkan jika dinamika robot dapat dijelaskan secara analitik, lingkungan dan interaksi robot melalui sensor dan aktuator sulit diperoleh model matematiknya. Ketiadaan

pengetahuan yang tepat dan lengkap mengenai lingkungannya membatasi penerapan desain sistem kendali konvensional pada domain robot mandiri. Yang diperlukan adalah sistem kendali cerdas dan sistem pembuat keputusan dengan kemampuan merespon (*reasoning*) pada kondisi tidak tentu dan kemampuan belajar dari pengalaman (Hoffman, 2003).

Untuk mewujudkan tujuan tersebut, pertama-tama diperlukan sistem kendali robot yang tidak berbasiskan model yang dikenal dengan sistem kendali *Behavior-Based Robot* (BBR). Pada pendekatan ini, sistem diuraikan menjadi beberapa modul yang masing-masingnya bertanggung jawab untuk melakukan satu perilaku (*behavior*). Tiap perilaku mengandung jalur lengkap mulai dari merasakan (*sensing*) sampai aksi. Semua modul yang mewakili satu perilaku bekerja bersama-sama (Kweon dkk, 1992).

Pada lingkungan tak terstruktur, perubahan-perubahan besar mungkin terjadi. Untuk mengatasi perubahan yang besar pada lingkungan saat robot berjalan sebagaimana juga perubahan pada misi/tugas yang terus menerus, sistem kendali robot juga harus mampu mengubah kebijakan kendalinya untuk menyesuaikan dengan semua kondisi baru. Secara umum, hal ini membutuhkan arsitektur sistem kendali adaptif (Anam, 2008). Oleh karena itulah diperlukan sistem kendali robot yang dapat mempelajari lingkungannya dalam rangka melakukan adaptasi terhadap perubahan yang terjadi pada lingkungannya.

Penggunaan pembelajaran dalam sistem navigasi disamping untuk mengatasi kondisi lingkungan yang tidak terstruktur, juga digunakan sebagai sarana untuk kemudahan dalam merancang perilaku yang baik. Dengan adanya pembelajaran, perancangan perilaku tidak harus dibuat secara lengkap dan mendetil saat merancang kode program. Dengan merancang bagaimana robot harus belajar, perilaku yang detil akan terbentuk dengan sendirinya dan akan semakin baik seiring dengan pembelajaran yang terus dilakukan.

Karena model lingkungannya yang tidak terstruktur dan tidak diketahui, model pembelajaran tidak diawasi (*unsupervised learning*) lebih tepat digunakan. Salah satu metode yang banyak digunakan adalah pembelajaran pengkondisian aksi kembali atau *reinforcement learning* (RL). Pada pembelajaran RL, robot belajar melalui interaksi dengan lingkungannya. Robot menerima kondisi (*state*) lingkungannya dan memilih aksi yang tepat untuk diberikan ke lingkungannya. Kemudian keadaan lingkungan berubah dan robot menerima penghargaan (*reward*) berupa nilai dari lingkungannya. Tujuan belajar disini adalah memilih aksi yang dapat memaksimalkan penghargaan yang diterima (Sutton,1998).

Salah satu metode pembelajaran yang paling cocok dan banyak digunakan untuk aplikasi robot adalah *Reinforcement Learning* (RL), dengan jenis algoritma Q-*learning*. Algoritma ini menggunakan tabel Q untuk mencocokkan kondisi diskrit dan aksi saja. Sedangkan pada aplikasi robot dengan arsitektur kontrol waktu nyata yang ukuran kondisi dan data sensornya bersifat kontinyu, hal ini menjadi tidak praktis. Oleh karena itu, untuk memperluas algoritma Q-*learning* berkaitan dengan kondisi dan aksi yang kontinyu, L. Jouffe (1998) mengkombinasikan Q-*learning* dengan *Fuzzy Inference System* (FIS) yang dikenal dengan *Fuzzy* Q-*Learning* (FQL). Menurut Youssef (2011), algoritma FQL sangat baik digunakan pada bidang robot dan bidang kecerdasan buatan atau *Artificial Intelligence* (AI).

Dalam lomba robot beroda yang umum diadakan tiap tahun, robot yang dibuat untuk menghindari halangan dan mencapai tujuan atau target dalam suatu arena yang didesain sedemikian rupa. Sensor yang digunakan biasanya sensor jarak dan api. Dengan adanya beberapa sensor, diperlukan pengaturan yang baik.

Pada penelitian sebelumnya (Santo, 2010) terdapat kesulitan dalam melakukan implementasi salah satu pengembangan algoritma RL pada robot multiplatform. Sehingga dalam penelitian ini akan diterapkan algoritma FQL hanya pada robot mandiri beroda atau *autonomous wheeled robot* (AWR). Hasil dari penelitian ini berupa rancangan algoritma yang baik untuk diterapkan pada robot AWR. Robot akan disimulasikan dengan simulator 3D khusus robot yaitu Webots 5.5.2 sehingga pergerakan robot dapat dianalisa dan dibandingkan sebelum dan sesudah menggunakan algoritma.

Rancangan penelitian ini diharapkan dapat sebagai acuan simulasi tahap awal dalam membuat robot. Sehingga akan mengurangi kesalahan desain robot yang berakibat meningkatnya biaya yang diperlukan dalam membuat robot yang sebenarnya.

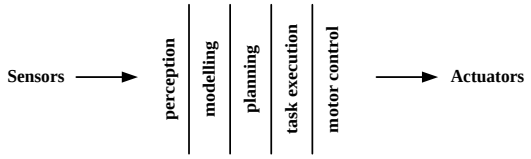
Kebutuhan Sistem Kontrol untuk Sistem Robot Mandiri

Ada beberapa aspek yang harus diperhatikan untuk membentuk sistem robot yang mandiri, diantaranya (Birk, 1998) :

a. Sistem kontrol robot bersifat dikendalikan sensor (*sensor driven*)
   Satu masalah yang dihadapi pada lingkungan yang tidak terstruktur adalah tidak mungkin memperkirakan hasil dari tiap aksi secara tepat. Kondisi ini juga menunjukkan sedikitnya kemungkinan menemukan deret aksi yang akan memecahkan tugas yang diberikan berdasarkan semua kemungkinan yang telah diperhitungkan sebelumnya.
   Oleh karena itu sistem kontrol robot harus dikendalikan sensor (*sensor driven*) dan mengijinkan robot untuk bereaksi terhadap kejadian (*event*) yang tidak diharapkan secara online. Pendekatan kontrol reaktif seperti ini dapat menjadi kokoh dengan mempertimbangkan gangguan yang terbatas dan dapat memperbaiki diri secara mandiri terhadap gangguan yang tak dapat dimodelkan tanpa memerlukan perencanaan ulang.
b. Arsitektur kontrol yang adaptif dengan kemampuan belajar online
   Untuk mengatasi perubahan yang besar pada lingkungan saat *runtime* sebagaimana juga perubahan pada misi/tugas yang terus menerus, sistem robot mandiri harus mampu mengubah kebijakan kontrolnya untuk menyesuaikan dengan semua kondisi baru. Secara umum, hal ini membutuhkan arsitektur sistem kontrol adaptif. Bergantung pada kondisi operasi, variasi yang luas dari mekanisme pembelajaran mesin dapat digunakan untuk melakukan adaptasi. Pada tugas robot yang mengijinkan pengawasan oleh operator luar dapat dilakukan denga strategi kontrol termodifikasi yang telah diprogram sebelumnya.
c. Memenuhi batasan keamanan
   Spesifikasi dasar arsitektur kontrol untuk keperluan sistem robot mandiri adalah ia memenuhi batasan keamanan tertentu untuk menghindari kegagalan besar dan tak dapat diperbaiki. Hal ini khususnya penting ketika tugas baru dilatihkan tanpa pengawasan dan ketika pengaruh perbedaan aksi harus ditentukan melalui uji coba.

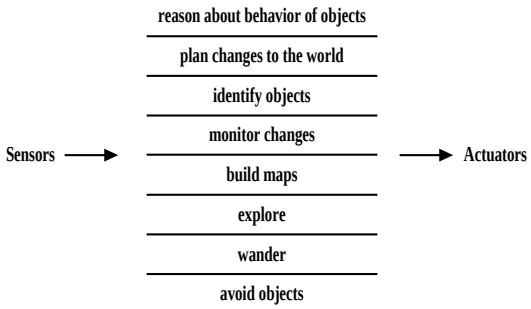
***Behavior-based Robot* (BBR)**

Pendekatan yang biasa digunakan untuk membangun sistem kontrol robot adalah dengan menguraikan setiap masalah kedalam rangkaian unit fungsional sebagaimana ditunjukkan oleh Gambar 1.



Gambar 1. Skema arsitektur deliberatif (Brooks, 1986)

Untuk mengatasi kelemahan pendekatan di atas, Brooks (1986) mengusulkan pendekatan arsitektur reaktif yang berbeda secara radikal dari paradigma arsitektur deliberatif. Secara kontras, Brooks berargumen bahwa membangun model keseluruhan yang kompleks dan menggunakan pemikiran simbolis merupakan halangan terhadap keberhasilan perkembangan *mobile robot*, dan bahwa lingkungan dimana robot bergerak faktanya ialah satu – satunya model yang dibutuhkan robot untuk dapat bergerak. Daripada menggunakan struktur vertikal dari model *sense-think-act*, ia menyarankan dekomposisi horizontal dengan istilah perilaku (*behaviors*) seperti pada Gambar 2. Pada arsitektur reaktif ini semua perilaku bekerja secara paralel, bersamaan, dan pada umumnya, secara asinkron.

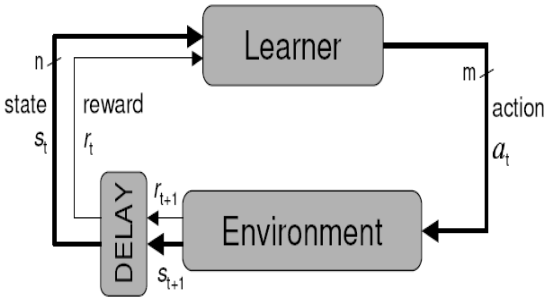


Gambar 2. Skema arsitektur reaktif (Brooks, 1986)

Reinforcement Learning

Reinforcement Learning (RL) merupakan pendekatan belajar dengan coba-coba (*trial-and-error*) untuk mencapai tujuan. Dasar belajarnya adalah interaksi antara algoritma RL dan lingkungan. Algoritma RL tidak menggunakan data respon masukan dan luaran seperti teknik belajar dengan pengawasan. Tetapi yang diperlukan adalah penghargaan yang diberikan oleh lingkungannya. Penghargaan ini menguji *state* lingkungan. Masalah RL adalah bagaimana memaksimalkan jumlah penghargaan yang akan datang.

Beberapa elemen yang muncul pada *reinforcement learning*, disebut pebelajar atau *learner*. *Learner* berinteraksi dengan lingkungan (*environment*). Lingkungan adalah segala hal diluar *learner*. *Learner* mempelajari kondisi lingkungannya dan berinteraksi dengan lingkungannya dengan memberikan aksi ke lingkungannya. Lingkungan bereaksi terhadap aksi dari learner dengan state baru dan penghargaan. Penghargaan adalah nilai skalar yang dibangkitkan oleh fungsi *reinforcement* yang menguji kondisi saat ini dan aksi terakhir. Gambar 3 menunjukkan interaksi antara pebelajar dan lingkungan.



Gambar 3. Diagram Interaksi antara Pebelajar dan Lingkungan (Carreras,2003)

Q-Learning

Dalam banyak aplikasi pembelajaran, tujuan yang ingin dicapai RL ialah mengadakan sebuah *control policy* yang memetakan ruang input diskrit ke ruang output diskrit sehingga penghargaan kumulatif maksimum dapat dicapai. Algoritma yang sederhana dan dapat diterapkan (dan paling banyak digunakan dalam aplikasi robot di atas) ialah *Q learning* (Watkins dan Dayan, 1992).

Kelebihan *Q-Learning* ialah sifatnya yang *off policy* (dapat mengikuti aturan apapun untuk menghasilkan solusi optimal), kemudahan algoritmanya, dan kemampuannya untuk konvergen pada aturan optimal. Algoritma *Q-Learning* ialah sebagai berikut (Perez, 2003).

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., $\in$-greedy)
Take action **a**, observe **r**, **s'**
Apply

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$$

s ← s';
until **s** is terminal
dimana
Q(s,a) : komponen dari tabel Q (state, action)
s : state s' : next state
a : action a' : next action
r : reward α : learning rate
γ : discount factor

Fuzzy Q-Learning

Pada *Fuzzy* Q-*learning*, pembelajaran tidak dilakukan setiap *state* pada ruang *state*. Oleh karena itu diperlukan optimasi pada beberapa *state* yang mewakili. Dengan demikian, digunakan interpolasi *fuzzy* untuk memperkirakan *state* dan aksi. (Glorennec dkk,1997)

Untuk merepresentasikan aksi dan *Q* fungsi pada *Fuzzy Inference System* (FIS), diperlukan modifikasi pada aturan dasar. Bentuk aturan-aturannya adalah :
Jika S_i **maka** aksi = a_i, $a_i \in A_i$
Jika S_i dan aksi= a_i **maka** nilai-q = $q(S_i,a_i)$
dengan *State* S_i didefinisikan dengan "x_1 adalah $S_{i,1}$ **dan** x_2 adalah $S_{i,2}$ **dan** …. X_n adalah $S_{i,n}$", $\left(S_{i,j}\right)_{j=1}^{n}$ adalah label *fuzzy* dan A_i adalah himpunan aksi yang mungkin yang dapat dipilih pada *state* S_i.

Langkah-langkah dalam algoritma *Fuzzy Q-learning* antara lain :
1. Dapatkan *state* x
2. Untuk tiap aturan, pilih konsekuen aktual menggunakan seleksi *ε-greedy*
3. Hitung konsekuen global menggunakan Persamaan

$$a(x) = \frac{\sum_{i=1}^{N} \alpha_i(x) \text{ x } a(i,i^\circ)}{\sum_{i=1}^{N} \alpha_i(x)}$$

dan q global didapat dari

$$Q(x,a) = \frac{\sum_{i=1}^{N} \alpha_i(x) \text{ x } q(i,i^\circ)}{\sum_{i=1}^{N} \alpha_i(x)}$$

4. Terapkan aksi global a dan dapatkan *state* baru
5. Dapatkan fungsi *reinforcement* r
6. Perbarui nilai q

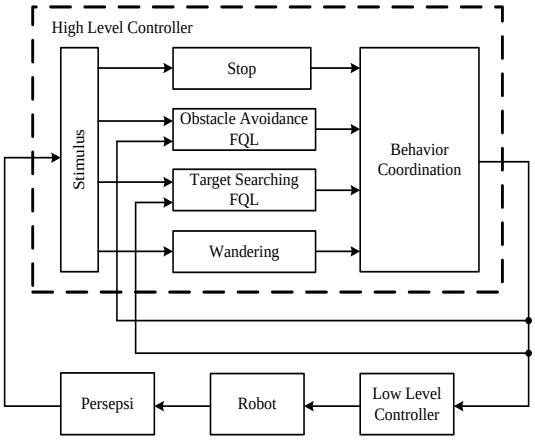
METODE PENELITIAN

Robot yang akan dibuat bertujuan untuk dapat melakukan navigasi secara otonom. Karena itu robot didesain memiliki perilaku – perilaku berikut :
- Menghindari halangan (*Obstacle Avoidance*)
- Mencari target (*Target Searching*)
- Berkeliling (*Wandering*)
- Berhenti (*Stop*)

Dari keempat perilaku tersebut, dipilih perilaku menghindari halangan dan mencari target yang akan menjadi tujuan pembelajaran dengan algoritma FQL.

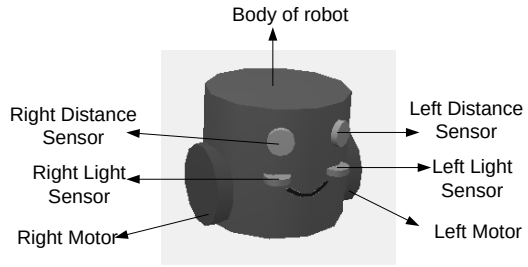
Robot akan dikendalikan secara langsung oleh pengendali tingkat bawah (*low level controller*) berupa kontroler P yang berfungsi untuk mengendalikan pergerakan robot dengan beberapa motor servo yang ada. Robot juga akan mengirim sinyal persepsi (melalui sensor – sensor yang dimilikinya) pada pengendali tingkat atas (*high level controller*). Kemudian sinyal tersebut menjadi masukan stimulus yang akan mengaktifkan perilaku-perilaku tertentu (yang dikoordinasi oleh bagian *behavior*

coordination). Output dari *behavior coordination* diberikan pada *low level controller* untuk pengendalian robot. Diagram blok keseluruhan robot terlihat pada Gambar 4 berikut ini.



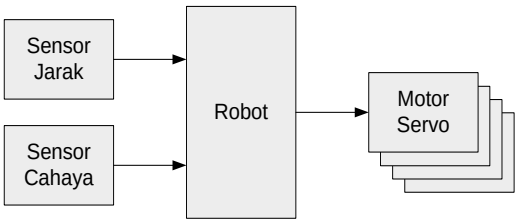
Gambar 4. Diagram blok robot

Robot yang digunakan di sini ialah robot beroda. Robot beroda (wheeled robot) merupakan robot yang sangat umum ada saat ini, bahkan telah ada rumusan kinematika robot beroda. robot beroda sederhana contoh dari Webots dapat dilihat pada Gambar 5.



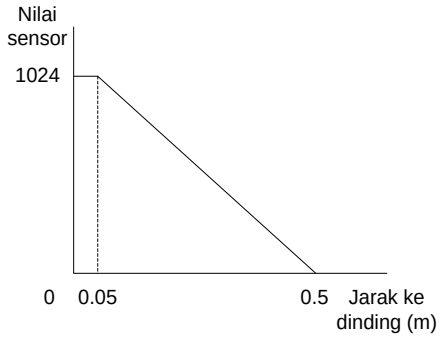
Gambar 5. Contoh robot dari Webots

Robot yang akan dirancang dilengkapi dengan 2 jenis sensor yaitu sensor jarak dan sensor cahaya. Sensor jarak untuk mengetahui jarak dari halangan yang ada di depan robot sedangkan sensor cahaya untuk menemukan target cahaya. Berikut ini diagram blok dari sensor dan aktuator yang dimiliki robot.



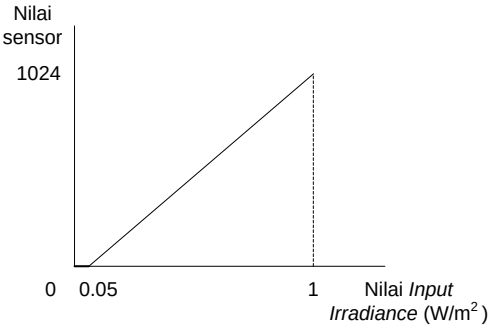
Gambar 6. Diagram blok robot dan sensor-aktuatornya

Karakteristik dari sensor jarak nampak pada grafik berikut.



Gambar 7. 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.

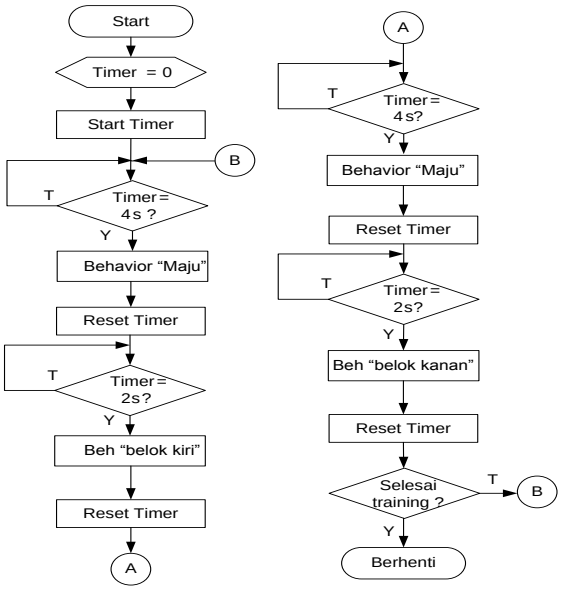


Gambar 8. Grafik karakteristik sensor cahaya

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

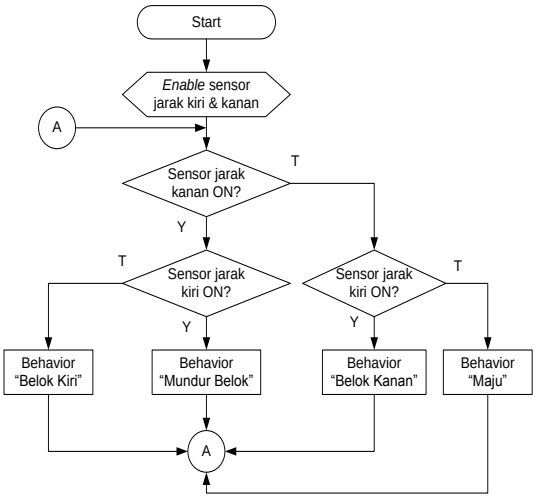
Robot yang akan dibuat bertujuan untuk dapat melakukan navigasi secara otonom. Karena itu robot didesain memiliki perilaku – perilaku berikut :

1. Berkeliling (*Wandering*)
   Wandering Behavior adalah prilaku robot saat berkeliling (*wandering*) di arena untuk menemukan target. Tanpa perilaku ini, robot hanya akan berjalan maju, ataupun menyusuri dinding (*wall following*) arena saja.



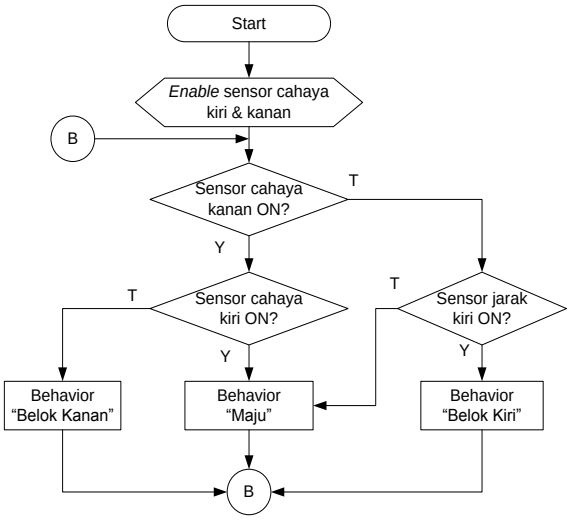
Gambar 9. Diagram alir dari *wandering behavior*

2. Menghindari Halangan
   Obstacle Avoidance Behavior adalah perilaku robot untuk menghindari halangan. Perilaku ini dilakukan berdasarkan pendeteksian obyek oleh 2 buah sensor jarak milik robot.



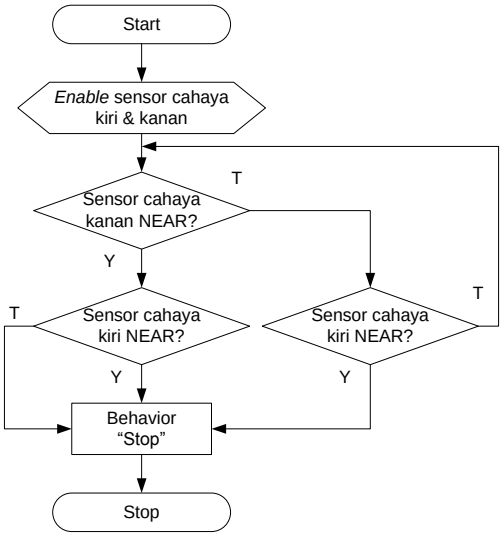
Gambar 10. Diagram alir dari obstacle avoidance behavior

3. Mencari target (*Target searching*)
   Search Target behavior adalah perilaku dalam mencari target. Jika sensor cahaya robot mendeteksi target berupa sumber cahaya, maka *search target behavior* akan aktif dan robot akan bergerak mendekati sumber cahaya itu.



Gambar 11. Diagram alir dari *search target behavior*

4. Berhenti (Stop)
   Jika jarak robot sudah "dekat" dengan target, maka *stop behavior* akan aktif dan robot akan berhenti.

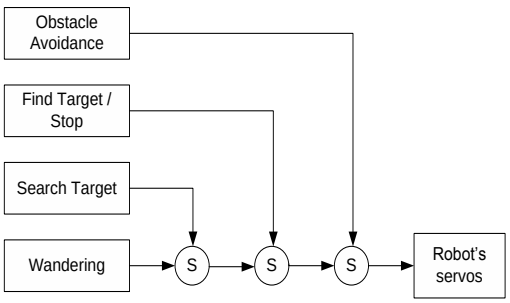


Gambar 12. Diagram alir dari *stop behavior*

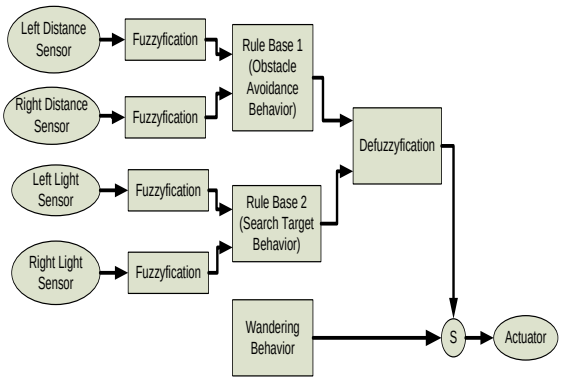
Dari keempat perilaku tersebut, dipilih perilaku obstacle avoidance dan target searching yang akan menjadi target pembelajaran dengan FQL.

Simulasi dilakukan dengan program Webots 5.5.2 untuk mengetahui performa beberapa metode koordinasi behavior berikut :

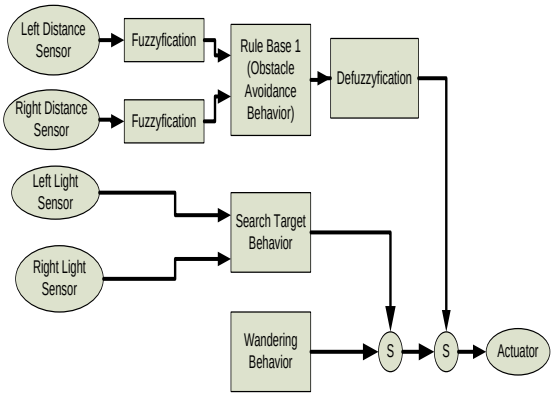
1. *Subsumption Architecture* (SA)
2. *Fuzzy Behavior Coordinator* (FBC)
3. *Modified Fuzzy Behavior Coordinator* (MFBC)



Gambar 13. Metode *subsumption architecture* untuk navigasi robot



Gambar 14. Skema *fuzzy behavior coordination*



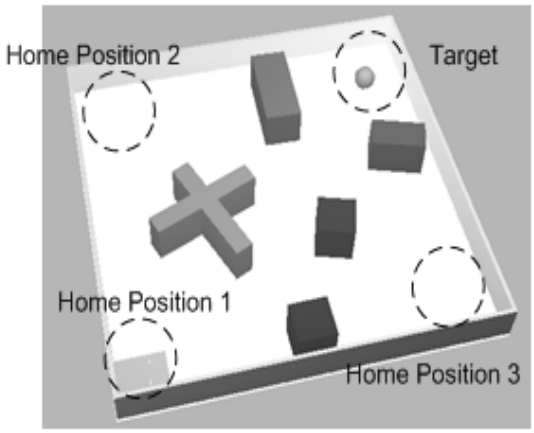
Gambar 15. Skema *modified fuzzy behavior coordination* (Handy, 2009)

HASIL DAN PEMBAHASAN

Pada penelitian ini akan dilakukan pengujian untuk melihat performa robot mandiri beroda (AWR) yang meliputi :

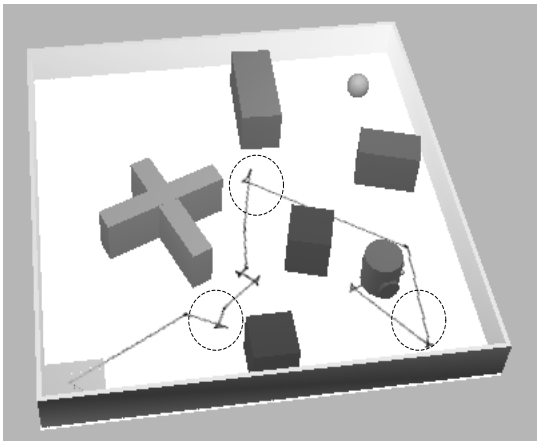
- Simulasi metode *behavior coordination* pada AWR
- Simulasi Pencarian Target
- Simulasi Q-*Learning* pada AWR
- Simulasi FQL pada AWR
- Perbandingan performansi masing-masing algoritma

Ada 3 posisi start yang berbeda di arena robot. Arena beserta halangan, posisi start dan posisi target dapat dilihat pada gambar di bawah.

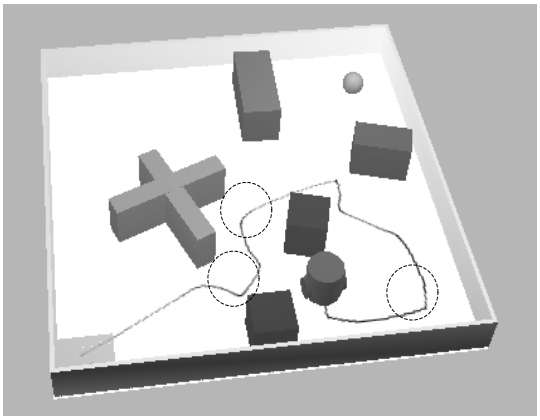


Gambar 16. Arena untuk simulasi pencarian target

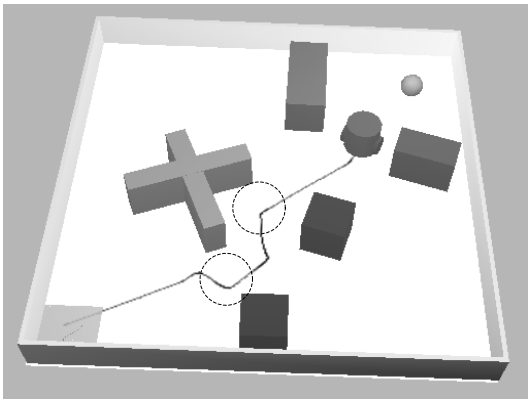
Simulasi metode *behavior coordination* pada AWR akan dilakukan untuk mengetahui berapa lama waktu yang dibutuhkan robot mulai dari posisi awal sampai robot menemukan target. Pada simulasi ini, lintasan yang dilewati robot untuk mencapai target akan diamati untuk memahami setiap pergerakan robot di dalam arena yang telah dirancang sebelumnya. Berikut ini hasil simulasi dari pergerakan robot dengan metode SA, FBC, dan MFBC.



Gambar 17. Hasil simulasi pergerakan robot dengan metode SA



Gambar 18. Simulasi pergerakan robot dengan metode FBC



Gambar 19. Simulasi pergerakan robot dengan metode MFBC

Tabel 1. Hasil simulasi pencarian target

Posisi	Posisi *start* Robot (x,y,z,teta)	Waktu menemukan target (menit)		
		SA	FBC	MFBC
1	(-0.42, 0, 0.41, 2.02)	1:07 m	0:23 m	0:14 m
2	(0.41, 0, 0.43, 0.92)	1:04 m	0:12 m	0:14 m
3	(-0.43, 0,-0.45, 4.11)	1:43 m	1:22 m	0:28 m

Dari tabel di atas dapat dilihat bahwa SA membutuhkan waktu paling lama (1:07 m, 1:04 m, 1:43 m) untuk mencapai target

dari masing – masing posisi. Hal ini terjadi karena SA hanya melakukan 1 jenis *behavior* dalam 1 waktu, sehingga *search target behavior* akan diabaikan saat *behavior* yang lain aktif. Selain itu, respon robot dengan metode SA relatif kasar sehingga dapat memperlama waktu pencarian target.

FBC memberikan hasil yang lebih baik (0:23 m, 0:12 m, 1:22 m) karena setiap *behavior* memberikan kontribusi pada keputusan yang dibuat robot. Namun demikian hal tersebut juga dapat berakibat pada waktu berpikir robot yang lebih lama karena setiap perilaku akan diproses oleh *fuzzy inference engine*. MFBC menghasilkan hasil yang sedikit lebih baik dari FBC (0:14 m, 0:14 m, 0:28 m), karena salah satu perilaku (*search target*) tidak diproses dengan *fuzzy inference engine* sehingga pergerakan robot lebih cepat. Perilaku tersebut dianggap lebih utama dari yang lain.

Penerapan algoritma pembelajaran pada robot beroda dapat memperlama proses eksekusi program oleh pengendali robot. Penambahan algoritma Q-*Learning* dan kemudian *Fuzzy* Q-*Learning* makin memperpanjang waktu yang dibutuhkan oleh robot. Kehadiran Fuzzy Q-Learning termodifikasi akan sedikit mengurangi waktu tersebut di atas. Berikut ini perbandingan dari ketiga algoritma pembelajaran di atas jika dilihat dari jumlah penghargaan yang diterima oleh perilaku *obstacle avoidance behavior* pada robot.

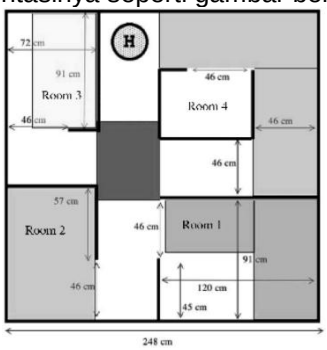
Tabel 2. Perbandingan jumlah reward yang diterima robot

Algoritma Pembelajaran	Jumlah Reward dalam 500 iterasi
Q Learning	107
Fuzzy Q Learning	322
Modified Fuzzy Q Learning	258

Robot yang digunakan dalam penelitian ini adalah robot beroda seperti pada gambar berikut.

Gambar 20. Robot Ganesh tahun 2012

Arena yang digunakan untuk implementasinya seperti gambar berikut.



Gambar 21. Arena robot untuk lomba standar KRCI

Gambar 22. Implementasi robot

Pengujian dilakukan sebanyak 3 kali dan kinerja robot dapat ditampilkan pada tabel berikut ini :

Tabel 3. Kinerja robot diukur dari waktu mencapai tiap ruangan

Percobaan	Waktu mencapai ruangan (menit)				
	R1	R2	R3	R4	Home base
Percobaan1	0:03	0:31	0:42	Lewat	1:10
Percobaan2	0:04	0:28	0:51	Lewat	1:11
Percobaan3	0:03	0:27	0:43	1:08	1:09

Tabel 4. Kinerja robot diukur dari waktu mematikan api lilin

Percobaan	Posisi Lilin	Waktu mematikan api lilin (detik)
Percobaan 1	Room 1	1
Percobaan 2	Room 2	3
Percobaan 3	Room 3	2

Berdasarkan percobaan di atas dapat disimpulkan bahwa waktu rata-rata robot dalam menyusuri keempat Room yang ada di arena dan kembali ke Homebase yaitu 1 menit 10 detik.

Sedangkan waktu rata-rata robot dalam mematikan api lilin adalah 2 detik. Sehingga dapat dikatakan robot yang dibuat mempunyai kinerja yang cukup baik dan mampu disiapkan untuk berlomba dalam lomba-lomba yang menuntut medan seperti ini.

SIMPULAN DAN SARAN

Penerapan *Fuzzy* Q-*Learning* memberikan hasil yang relatif lebih baik dari Q-*Learning* dalam jumlah penghargaan yang diterima dan performa navigasi robot mandiri, meski algoritma FQL membutuhkan waktu yang lebih lama untuk dieksekusi. *Fuzzy* Q-*Learning* yang termodifikasi dapat menghemat memori robot tidak memberikan penurunan signifikan pada performa robot, sehingga MFQL masih dapat digunakan sebagai media simplifikasi. Untuk penelitian selanjutnya disarankan untuk menerapkan algoritma pembelajaran pada aplikasi lain yang lebih kompleks, misal : menghindari halangan bergerak, mencari jarak terdekat, dan sebagainya.

UCAPAN TERIMAKASIH

Terima kasih kepada tim robot Undiksha yang telah ikut merampungkan proyek robot ini dan terimakasih kepada Lembaga Penelitian sebagai penyandang dana hingga terselesainya penelitian ini.

DAFTAR PUSTAKA

Anam, K. (2008). *Skema Behavior-Based Control dengan Pembelajaran Fuzzy Q-Learning untuk Sistem Navigasi Autonomous Mobile Robot*. Tesis Master, Institut Teknologi Sepuluh Nopember, Surabaya.

Birk, A.(1998). *Behavior-based Robotic, its scope dan its prospect*. Proceedings of The 24th Annual Conference of IEEE Industrial Electronics Society, IECON'98, Vol. 4, hal. 2180 – 2185.

Brooks, R. (1986). *A robust layered control system for a mobile robot*, IEEE Journal of Robotics and Automation, Vol. 2, No. 1, hal. 14–23.

Carreras, M. (2003). *A Proposal of a Behavior-based control architecture with reinforcement learning for an autonomous Underwater Robot*, Tesis PhD, Dept. of Electronic, Informatic & Automation, University of Girona, Girona

Glorennec, P. Y. (2000), *Reinforcement Learning : An Overview*, Proceedings of European Symposium on Intelligent Techniques, Aachen, Germany.

Handy W. (2009), "*Implementasi Compact Fuzzy Q Learning Untuk Navigasi Robot Otonom Berkaki*", Tesis Master, Institut Teknologi Sepuluh Nopember, Surabaya.

Hoffman, F.(2003), *An Overview on Soft Computing in Behavior-based robotics*, The Proceeding of The 10th International Fuzzy System Association, IFSA, Eds: Bilgic, T. et al., Istanbul, Turki, hal. 544 – 551.

Jouffe, L. (1998), *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.

Kweon, I, Kuno, Y, Watanabe, M, Onoguchi, K.(1992). *Behavior Based Mobile Robot Using Active Sensor Fusion*, IEEE Journal of Robotics and Automation, Vol. 2, hal. 1675 – 1682.

Manzanedo, A. dan Alvarez, J. (1998), *Fuzzy Control Applications Development System*, Mechatronics 98 Workshop, Skovde, Sweden.

Perez, Marc C. (2003). *A Proposal of Behavior Based Control Architecture with Reinforcement Learning for an Autonomous Underwater Robot*. Tesis Ph.D., University of Girona, Girona.

Santo Made (2010), *Pembelajaran Robot dengan Fuzzy Q-Learning pada Sistem Navigasi Robot Otonom Multiplatform*, Proceedings 11th SITIA, ISSN: 2087-331X, Jurusan Teknik Elektro ITS Surabaya.

Sutton, R.S., Barto,A.G. (1998). *Reinforcement Learning, an introduction*. MIT Press.

Watkins, C. dan Dayan, P. (1992). *Q-learning,Thechnical Note*. Machine Learning, Vol 8, hal. 279-292.

Youssef S. G. Nashed, Darryl N. Davis (2011), *Fuzzy Q-Learning for First Person Shooters*, The European Multidisciplinary Society for Modelling and Simulation Technology - EUROSIS-ETI, GAMEON-NA'2011, September 28-30, 2011, Rensselaer Polytechnic Institute, Troy, NY, USA.