

Penerapan Metode Fuzzy Q-Learning Pada Robot Berkaki Enam Pemadam Api Dalam Pencarian Jalur Tercepat Menuju Titik Api

Richa Watiasih¹, Pangki Dwi Pradana²

Prodi Teknik Elektro Fakultas Teknik Universitas Bhayangkara Surabaya^{1,2}
richa@ubhara.ac.id¹

Abstrak—Robot pemadam api merupakan alternatif pengganti petugas pemadam kebakaran dalam skala kecil. Waktu yang cepat dalam pencarian titik api sangat diperlukan agar api cepat dapat dipadamkan. Penelitian ini telah menghasilkan sebuah robot berkaki enam pemadam api menggunakan metode *Fuzzy Q-Learning* (FQL) yang bertujuan untuk memaksimalkan pencarian jalur tercepat menuju titik api sehingga dapat meminimalkan waktu tempuh robot dalam menyelesaikan misinya. Dimana penggunaan sensor Ultrasonik sebagai pendeteksi jarak robot terhadap halangan, sedangkan sensor UVTron dan TPA81 digunakan untuk mendeteksi keberadaan dan merasakan adanya api. Data dari ketiga sensor tersebut digunakan sebagai input pada FQL. Fuzzy logic digunakan untuk mempercepat pengambilan keputusan pergerakan robot, sedangkan Q-learning memberikan suatu reward dan punishment pada setiap pergerakan. Data dari Q-learning disimpan pada MicroSD supaya dapat cepat diproses untuk pertimbangan pada pergerakan selanjutnya. Robot berkaki enam pemadam api ini telah diuji coba beberapa kali sampai mendapatkan reward maksimum. Berdasarkan pengujian yang telah dilakukan sebanyak 10 kali diketahui waktu tercepat menemukan titik api ketika tanpa menggunakan metode FQL adalah 218 detik, sedangkan ketika menggunakan metode FQL diperoleh keberhasilan menemukan titik api dengan waktu tercepat 154 detik. Jika dibandingkan dengan tanpa menggunakan metode diketahui bahwa penggunaan metode FQL dapat meminimalkan waktu tempuh menuju titik api.

Kata kunci—robot berkaki enam, pemadam api, Ultrasonik, UVtron, TPA81, Fuzzy Q-learning (FQL)

Abstract—Fire extinguishing robots are an alternative replacement for firefighters on a small scale. Fast time searching for fire points is necessary so the fire can be extinguished quickly. This research has produced a six-legged fire extinguishing robot using the *Fuzzy Q-Learning* (FQL) method which aims to maximize the search for the fastest path to the fire point so that it can minimize the robot's travel time in completing its mission. The Ultrasonic sensor detects the robot's distance from obstacles, while the UVTron and TPA81 sensors are used to detect the presence and sense of fire. Data from the three sensors is used as input for FQL. Fuzzy logic speeds up decision-making for robot movements, while Q-learning provides reward and punishment for every movement. Data from Q-learning is stored on Micro SD to be processed for consideration for the next move quickly. Based on tests that have been carried out 10 times, it is known that the fastest time to find hotspots without using the FQL method is 218 seconds, whereas if you use the FQL method, success in finding hotspots is achieved with the fastest 154 seconds. Compared to not using this method, it is known that using the FQL method can minimize travel time to the fire point.

Keywords — six-legged robot, fire extinguisher, Ultrasonic, UVtron, TPA81, fuzzy Q-learning (FQL)

I. PENDAHULUAN

Kinerja petugas pemadam api dalam melakukan tugasnya dapat dibidang berbahaya. Resiko kecelakaan dan terjebak dalam bangunan

yang terbakar ataupun gangguan sesak nafas dapat terjadi [1]. Oleh karena itu tugas pemadam kebakaran dapat digantikan dengan robot pemadam api. Robot harus mampu menemukan

titik api pada suatu ruangan kemudian memadamkannya. Robot pemadam api didesain secara *autonomous* tanpa adanya campur tangan manusia ketika di jalankan, mulai dari posisi awal robot dan berakhir pada tujuan ruang api [2]. Karena kondisi lingkungan yang tidak diketahui maka harus ada perencanaan jalur menuju titik api. Dimana robot bergerak memerlukan algoritma perencanaan jalur agar dapat menemukan jalur yang aman dalam menghindari tabrakan dan optimal dalam mencapai target [3].

Dalam beberapa kasus robot pemadam api berkaki masih belum bisa bermanuver dengan baik, seperti tidak menemukan titik api, robot tersangkut di ujung dinding, maupun tidak dapat kembali ke *home* setelah menyelesaikan misi, sehingga diperlukan sebuah sistem perencanaan jalur dan kontrol pergerakan robot. Penelitian tentang robot berkaki pemadam api dengan menggunakan metode *Proporsional Integral Derivative* (PID) telah dihasilkan [4], sedangkan [5] dari penelitian yang telah dilakukan dapat diketahui bahwa robot mampu menjejak dinding namun sulit untuk mempertahankan posisi terhadap referensi ini disebabkan karena kontroler *Derivative* mempengaruhi respon sistem sehingga dibutuhkan pengembangan metode kontrol cerdas. Beberapa penelitian kecerdasan buatan yang diterapkan pada perencanaan jalur bagi robot berkaki telah dilakukan seperti penerapan fuzzy logic [6], neural network [7], Ant Colony [8] dan penggunaan metode hybrid seperti antara neural network dengan fuzzy logic [9]. Namun belum optimal karena tidak mempertimbangkan faktor pengenalan lingkungan yang dilalui robot.

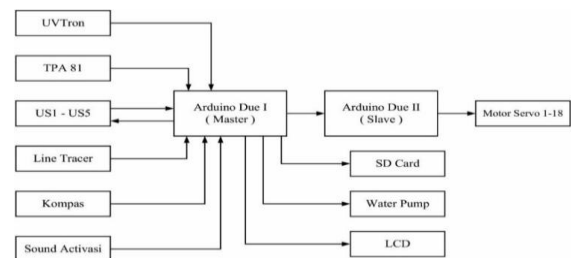
Pada area kebakaran yang tidak terduga, robot harus mampu melakukan manuver secara kompleks dan sulit diprediksi [10], dimana robot diberikan sebuah metode pembelajaran untuk mengeksplorasi lingkungan yang tidak diketahui sebelumnya [11]. Selain metode navigasi yang tepat, juga diperlukan mekanisme pembelajaran pada robot untuk mengatasi hal-hal yang tak terduga baik berupa halangan statis maupun dinamis. *Reinforcement learning* adalah metode pembelajaran untuk memetakan setiap keadaan terhadap aksi yang dipilih untuk memaksimalkan reward yang diterima secara langsung dari lingkungan [12]. Ada berbagai metode untuk penyelesaian masalah *reinforcement learning*, salah satunya adalah *Q-Learning*. Kelebihan dari *Q Learning* adalah memiliki sifat *off policy* (dapat mengikuti *policy* apapun), algoritma ini sederhana dan konvergen terhadap *optimal policy* [13].

Penelitian ini mengusulkan sebuah sistem pencarian jalur tercepat bagi robot berkaki enam yang memiliki tugas untuk mencari sumber/titik api dan memadamkan api dengan menerapkan metode *Fuzzy Q-Learning* sebagai mekanisme pembelajaran robot terhadap lingkungan yang ditemui, sehingga robot dapat meminimalkan waktu pencarian sumber/titik api.

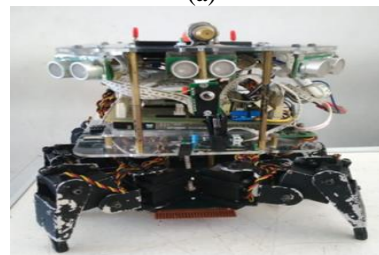
II. METODE PENELITIAN

A. Perancangan Hardware

Perancangan hardware robot berkaki enam pemadam api pada penelitian ini seperti yang ditunjukkan blok diagram pada gambar 1. Blok diagram robot ini terdiri dari bagian *Slave* dan *Master*. Pada bagian *slave* hanya mengontrol output pergerakan robot yang berupa motor servo dengan menggunakan mikrokontroler Arduino Due II. Untuk bagian *master* merupakan sistem kendali pusat dari robot berkaki enam pemadam api yang langsung terhubung dengan sensor, antara lain sensor Ultrasonik berfungsi untuk mendeteksi dinding pada lingkungan robot, dimana Pengambilan data pada sensor ultrasonik dilakukan secara kontinyu. Sensor thermal TPA81 berfungsi untuk mendeteksi radiasi panas pada benda [14]. Sensor line tracer berfungsi sebagai pendeteksi koridor dengan pintu di setiap ruang. Sensor UV-Tron berfungsi mendeteksi adanya cahaya ultraviolet yang dipancarkan api [15]. Sensor Magnetic Kompas berfungsi untuk memberikan referensi robot untuk mengetahui arah pada saat keluar ruangan. Pada bagian master juga terhubung dengan *output Water pump*, LCD dan *SD Card*. Komunikasi data dari kedua mikrokontroler menggunakan komunikasi serial.



(a)

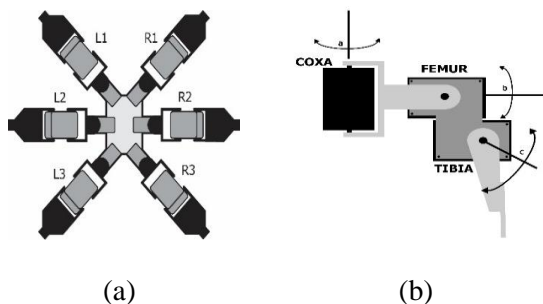


(b)

Gambar 1. Hardware robot berkaki enam; (a) Blok diagram; (b) Fisik robot berkaki enam

B. Desain Mekanik

Desain mekanik kaki-kaki robot berkaki enam ini ditunjukkan pada gambar 2. Dimana aktuator pada robot berkaki menggunakan motor servo. Robot Berkaki didesain 6 buah kaki dengan masing-masing sisi kanan dan sisi kiri berjumlah 3 kaki seperti ditunjukkan pada gambar 2(a). Setiap kaki robot memiliki sistem mekanik yang sama dan tiga derajat kebebasan (DOF) [16]. Setiap kaki terdapat 3 bagian yaitu *Coxa*, *Femur*, dan *Tibia*. Untuk setiap bagian memiliki fungsi tersendiri, *coxa* berfungsi sebagai pergerakan maju atau mundur, *femur* berfungsi untuk mengangkat atau menurunkan kaki, dan *tibia* digunakan untuk melebarkan kaki. Gambar 2(b) menunjukkan sumbu pergerakan dari setiap kaki robot.



Gambar 2. Desain mekanik; (a) struktur kaki-kaki; (b) Sumbu pergerakan setiap kaki

C. Perancangan Software

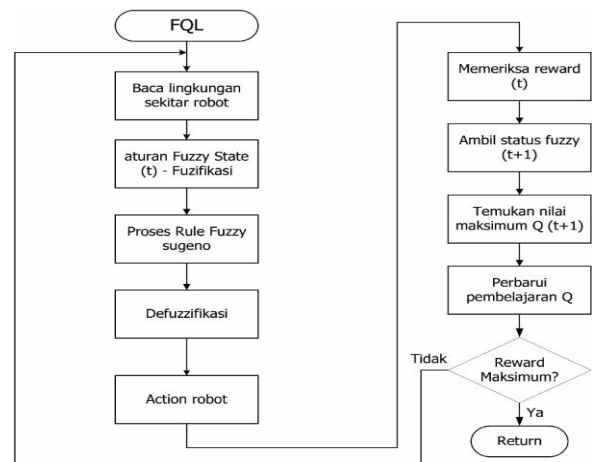
Untuk menghasilkan perencanaan jalur tercepat maka diperlukan algoritma supaya robot dapat menghindari segala halangan yang ditemui selama perjalanan yaitu dengan menerapkan algoritma *obstacle avoiding* berbasis FQL dan menerapkan algoritma *target searching* berbasis FQL supaya robot dengan cerdas dapat mendeteksi keberadaan sumber/titik api.

FQL adalah pengembangan dari metode *Reinforcement learning*, dimana *Reinforcement learning* merupakan salah satu metode *Machine Learning*. FQL merupakan pembelajaran kompetisi dimana ada agen-agen dengan kemampuan belajar untuk beraksi secara optimal dengan mengevaluasi konsekuensi dari aksi-aksinya. *Q-learning* yang bersifat konvergen dan *off policy* membuat *Q-learning* untuk aplikasi *real time*. *Q-learning* melakukan *update* terhadap *action-value function* seperti pada persamaan (1).

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha (R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)) \quad (1)$$

Dimana α menyatakan *learning rate*, $0 < \alpha \leq 1$ menentukan ukuran laju dimana nilai yang lama digantikan dengan nilai yang baru. Untuk γ menyatakan *Discount rate*, $0 < \gamma \leq 1$ menentukan nilai *reward* masa depan, semakin kecil nilai γ , maka agen akan semakin mementingkan *reward* dekat, bukan *reward* di masa depan. Untuk R_t merupakan sinyal *reinforcement learning* dimana ketika robot mendapatkan *reward* akan bernilai +1 dan Ketika mendapatkan *punishment* robot bernilai -1 [17].

Pembelajaran yang dilakukan oleh FQL tidak dilakukan setiap *state*. Oleh karena itu diperlukan optimasi pada beberapa *state* yang mewakili. Dengan demikian digunakan interpolasi *fuzzy* bekerja sebagai universal aproksimator dan kandidat yang baik untuk menyimpan nilai *Q*. Penambahan metode *Fuzzy* digunakan agar dapat bekerja pada *state* dan aksi yang kontinu. Untuk diagram alir dari FQL dapat dilihat pada gambar 3.



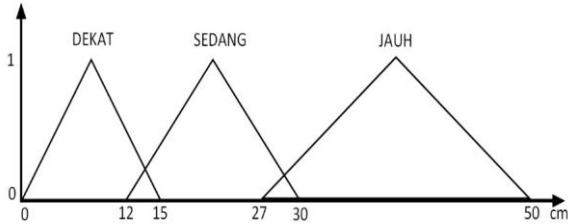
Gambar 3. Flowchart dari FQL

D. Obstacle Avoiding-FQL

Obstacle Avoiding dipandang sebagai sebuah *behavior* yang dibangkitkan oleh FQL. *Behavior* ini memiliki tugas untuk menghindari setiap halangan yang ditemui pada saat menyusuri ruangan-ruangan yang dilalui dan dideteksi oleh sensor ultrasonik.

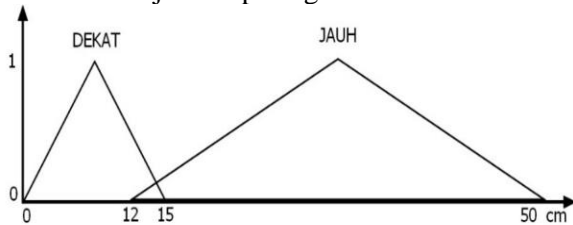
FQL adalah sistem *fuzzy inference* yang bagian konsekuennya memiliki lebih dari satu kemungkinan. Oleh karena itu memiliki struktur seperti sistem *fuzzy inference* pada umumnya dengan beberapa pengecualian. *Obstacle Avoiding-FQL* menggunakan input berupa data jarak dari 5 sensor ultrasonik. Pada proses fuzifikasi terdiri dari 2 input. *Input* jarak pertama

berupa data jarak dari sensor ultrasonik US1, US4, dan US5 dengan masing-masing menggunakan 3 *membership function* yaitu: dekat, sedang, dan jauh. *Membership function* dari input jarak pertama ditunjukkan pada gambar 4.



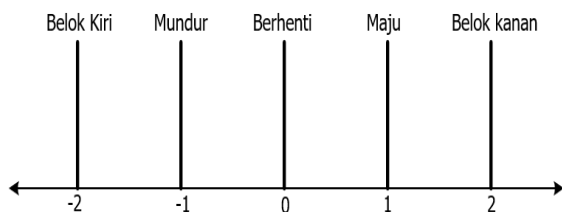
Gambar 4. Membership function input jarak pertama

Sedangkan untuk *membership function* untuk input jarak kedua berupa data jarak dari sensor ultrasonik US2 dan US3 dengan masing-masing menggunakan 2 *membership function* yaitu: dekat dan jauh. *Membership function* dari input jarak kedua ditunjukkan pada gambar 5.



Gambar 5. Membership function input jarak kedua

Berbeda dengan sistem logika *fuzzy* biasa, *output* dari FQL adalah nilai estimasi dari fungsi q yang besarnya tidak pasti. Akan tetapi masing-masing nilai estimasi disesuaikan dengan aksi kendali yang akan diberikan ke robot. Dengan demikian, *membership function* dari *ouput* diwakili dengan *Membership function* aksi *obstacle avoiding-FQL* bukan *Membership function* fungsi q . *Membership function* aksi *obstacle avoiding-FQL* yang digunakan adalah *singleton* seperti ditunjukkan oleh Gambar 6.



Gambar 6. Membership function untuk output aksi *obstacle avoiding-FQL*

Pemilihan aksi dari 5 aksi yang ada di tiap *rule* menggunakan *e-greedy selection*. Sebuah aksi dipilih jika nilai fungsi q yang disesuaikan bernilai maksimal dengan probabilitas *e-greedy* adalah $1 - \epsilon$. Sebuah aksi dipilih secara acak jika probabilitas *e-greedy* adalah ϵ . Nilai ϵ adalah nilai acak *real* kurang dari 1.

Fungsi *reinforcement* untuk *Obstacle Avoiding-FQL* diperoleh dari definisi tugas yaitu bergerak bebas tanpa menabrak halangan. Sinyal *reinforcement* r menghukum robot dengan nilai -1 setiap kali robot menabrak halangan atau salah satu nilai sensornya lebih besar atau sama dengan 1000. Jika jarak robot ke halangan lebih dari nilai *threshold* tertentu, $d_k = 300$, nilai penaltinya adalah 0. Sedangkan jika nilai salah satu sensor kurang dari nilai d_k maka robot akan mendapatkan penghargaan/*reward* 1. Fungsi *reinforcement* yang mengajari robot untuk menjauh dari halangan adalah menggunakan persamaan (2).

$$r = \begin{cases} -1 & \text{jika tabrakan atau } d_s \geq 1000 \\ 0 & d_k \leq d_s < 1000 \\ 1 & d_s < d_k \end{cases} \quad (2)$$

dengan d_s adalah jarak terpendek dari pembacaan 5 sensor ultrasonik.

Untuk Rule Base dari *obstacle avoidance - FQL* menghasilkan 108 *rule* seperti yang ditunjukkan pada tabel 1.

Tabel 1. Rule Base *Obstacle Avoidance-FQL*

Input Jarak 1			Input Jarak 2			
			Us2			
			Dekat	Dekat	Jauh	Jauh
			Us3			
Us1	Us4	Us5	Dekat	Jauh	Dekat	Jauh
Dekat	Dekat	Dekat	Ber	Mu	Mu	B_K a
Dekat	Dekat	Sedan	B_K a	B_K a	B_K a	B_K a
Dekat	Dekat	Jauh	B_K a	B_K a	B_K a	B_K a
Dekat	Sedan	Dekat	B_K i	B_K i	B_K i	B_K i
Dekat	Sedan	Sedan	B_K i	B_K i	B_K i	B_K i
Dekat	Sedan	Jauh	B_K a	B_K a	B_K a	B_K a
Dekat	Jauh	Dekat	B_K i	B_K i	B_K i	B_K i
Dekat	Jauh	Sedan	B_K i	B_K i	B_K i	B_K i

Dekat	Jauh	Jauh	B_K i	B_K i	B_K i	B_K i
Sedan g	Dekat	Dekat	Ma	B_K a	B_K i	Ma
Sedan g	Dekat	Sedan g	B_K a	B_K a	Ma	Ma
Sedan g	Dekat	Jauh	B_K a	B_K a	Ma	Ma
Sedan g	Sedan g	Dekat	B_K i	Ma	B_K i	Ma
Sedan g	Sedan g	Sedan g	Ma	Ma	Ma	Ma
Sedan g	Sedan g	Jauh	Ma	Ma	Ma	Ma
Sedan g	Jauh	Dekat	B_K i	Ma	B_K i	Ma
Sedan g	Jauh	Sedan g	Ma	Ma	Ma	Ma
Sedan g	Jauh	Jauh	Ma	Ma	Ma	Ma
Jauh	Dekat	Dekat	B_K a	B_K a	B_K i	Ma
Jauh	Dekat	Sedan g	B_K a	B_K a	Ma	Ma
Jauh	Dekat	Jauh	B_K a	B_K a	Ma	Ma
Jauh	Sedan g	Dekat	B_K i	Ma	B_K i	Ma
Jauh	Sedan g	Sedan g	Ma	Ma	Ma	Ma
Jauh	Sedan g	Jauh	Ma	Ma	Ma	Ma
Jauh	Jauh	Dekat	B_K i	Ma	B_K i	Ma
Jauh	Jauh	Sedan g	Ma	Ma	Ma	Ma
Jauh	Jauh	Jauh	Ma	Ma	Ma	Ma

Ket: BKa=Belok Kanan, BKi=Belok Kiri,
Ma=Maju, Mu=Mundur

Setelah proses fuzzy inference yang menghasilkan *rule base*, maka proses Defuzzifikasi dilakukan untuk mengubah kembali variabel *fuzzy* menjadi variabel nyata, atau dengan kata lain aksi kontrol *fuzzy* yang masih berupa himpunan, dirubah menjadi nilai nyata yang berupa nilai tunggal. Pada penelitian ini menggunakan fuzzy model sugeno maka menggunakan *Weighted Average (WA)* yaitu menggunakan persamaan (3) [18].

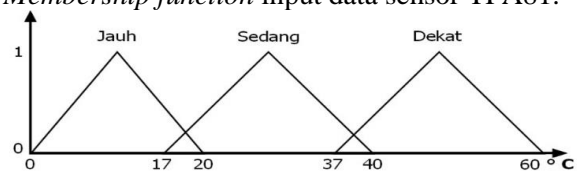
$$WA = \frac{\mu(k1) \times k1 + \mu(k2) \times k2 + \dots}{\mu(k1) + \mu(k2) + \dots} \quad (3)$$

E. Target Searching-FQL

Target Searching-FQL adalah *behavior* yang dibangkitkan oleh *Fuzzy Q-Learning* untuk mencari dan menuju target. Tujuannya adalah untuk mendeteksi sumber cahaya dan temperatur dari api. Dimana sensor UVTron untuk

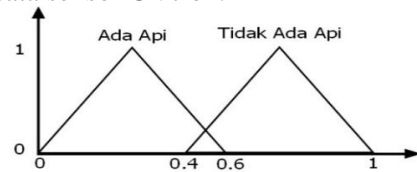
mendeteksi cahaya api dan sensor TPA81 untuk mendeteksi cahaya dari api.

Target Searching-FQL menggunakan 2 input fuzzy yaitu berupa data sensor TPA81 dan data Sensor UVtron. Sensor TPA81 memiliki 3 *Membership function* yaitu dekat, sedang, dan jauh. Karena sensor ini mendeteksi panas maka di representasikan melalui jarak dengan mendeteksi suhu, ketika suhu rendah maka di jarak robot dari api jauh sedangkan jika suhu tinggi maka jarak robot dan api dekat. Pada gambar 7 merupakan *Membership function* input data sensor TPA81.



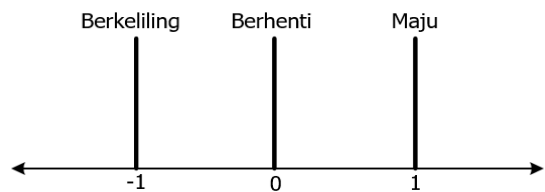
Gambar 7. Membership function untuk input data sensor TPA81

Sensor UVtron digunakan sebagai pendeteksi cahaya dari api pada sebuah ruangan. Ketika ada api uvtron akan bernilai 1 dan jika tidak ada maka bernilai 0. Untuk *Membership function* dari sensor UVtron yaitu Ada Api dan Tidak Ada Api. Dapat dilihat pada gambar 8 *Membership function* input data sensor UVtron.



Gambar 8. Membership function untuk input data sensor Uvtron

Output dari *Fuzzy Q-Learning* untuk *target searching* ini adalah nilai estimasi dari fungsi *q* yang besarnya tidak pasti. Akan tetapi masing-masing nilai estimasi bersesuaian dengan aksi kontrol yang diberikan ke robot. Dengan demikian, *membership function* dari output diwakili dengan *membership function* aksi *target searching-FQL*. Bentuk kurva *membership function* aksi *target searching-FQL* yang digunakan adalah *singleton* seperti ditunjukkan oleh Gambar 9.



Gambar 9. Membership function untuk output aksi target searching-FQL

Pemilihan aksi dari 2 aksi yang ada di tiap *rule* menggunakan *ε-greedy selection*. Sebuah aksi dipilih jika nilai fungsi q yang bersesuaian bernilai maksimal dengan probabilitas *ε-greedy* adalah $1-\epsilon$, dimana sebuah aksi dipilih secara acak jika probabilitas *ε-greedy* adalah ϵ [13]. Nilai ϵ adalah nilai acak real kurang dari 1.

Fungsi *reinforcement* untuk *Target Searching-FQL* diperoleh dari definisi tugas yaitu bergerak berkeliling dan maju menuju target. Robot mendapatkan *reward* ketika robot berjalan menghadap dan dekat terhadap sumber/titik api dan robot akan mendapatkan hukuman jika robot berada jauh dari sumber/titik api. Fungsi *reinforcement* di jabarkan oleh persamaan (4).

$$r = \begin{cases} -1 & \text{jika } d_s < 300 \\ 0 & \text{jika } 300 \leq d_s < 800 \\ 1 & \text{jika } d_s \geq 800 \end{cases} \quad (4)$$

Dimana d_s adalah nilai tertinggi dari pembacaan sensor UVTron dan sensor TPA81.

Setelah proses fuzifikasi, selanjutnya menentukan *Rule base*. *Rule base* adalah sebuah proses pemetaan *input* terhadap *output* dengan menggunakan logika *fuzzy*. Proses ini melibatkan *membership function* operator logika *fuzzy*, dan aturan *IF-THEN*. Untuk *rule base* dari *searching target* menghasilkan 6 rule yang dapat dilihat pada tabel 2.

Tabel 2. *Rule Base* untuk *Target Searching-FQL*

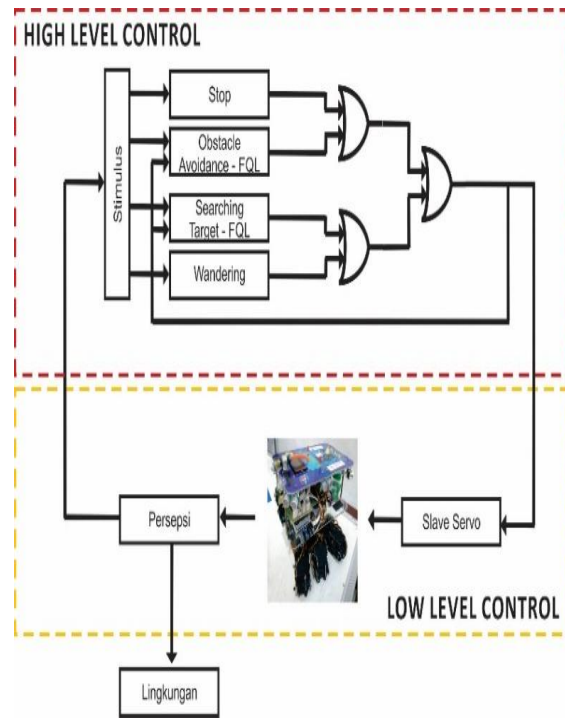
Uvtron			
	Dekat	Sedang	Jauh
Tidak Ada Api	Berkeliling	Berkeliling	Berkeliling
Ada Api	Berhenti	Maju	Maju

Setelah proses *rule base* dihasilkan, maka selanjutnya proses Defuzzifikasi dilakukan untuk mengubah kembali variabel *fuzzy* menjadi variabel nyata dengan menggunakan persamaan (3).

F. Sistem Kontrol

Sistem kontrol yang dirancang pada penelitian ini terdiri dari dua level kontroller utama yaitu *High Level Controller* dan *Low Level Controller*. *Low level Controller* berfungsi sebagai pengendalian pergerakan robot. Data sensor yang didapat dari pembacaan lingkungan dikirim ke *High Level Controller*. Kemudian data tersebut

menjadi masukan untuk *behaviour* tertentu. Keluarannya menjadi *setting point* untuk *low level controller*.



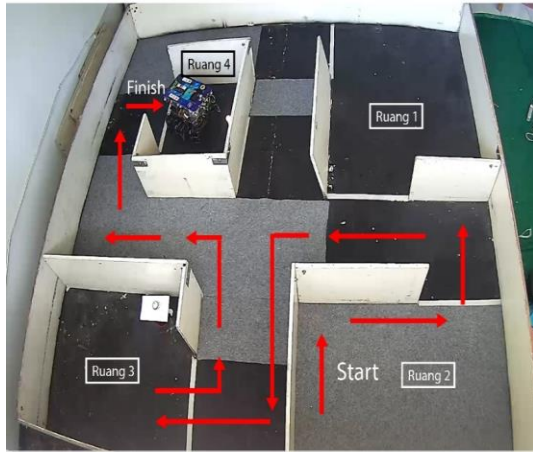
Gambar 10. Blok kontroler FQL

High Level Controller terdiri dari empat *behaviour*, dari empat *behaviour* terdapat dua *behaviour* yang dibuat saat desain yaitu *stop* dan *wandering*, dan dua *behaviour* yang dibentuk melalui proses belajar *on line* FQL, *Obstacle Avoiding-FQL* dan *Target Searching-FQL*. *Low Level Controller* digunakan untuk mengirimkan sinyal kontrol dari luaran kontroler level atas ke aktuator slave servo. Blok kontroler metode FQL ditunjukkan pada gambar 10.

III. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Tanpa Metode

Pengujian tanpa metode dilakukan dengan program *Wall Following* dimana robot menelusuri dinding dari start sampai menemukan titik api. Pada awal *start* robot akan menelusuri ruangan untuk menemukan jalan keluar dari ruang 2. Kemudian robot bergerak menelusuri ruangan dengan *scan* kiri sampai menemukan ruang selanjutnya yaitu ruang 3. Setelah sampai pada ruang 3 robot membaca apakah ada titik api atau tidak dan dilanjutkan pada ruang selanjutnya sampai menemukan sumber/titik api. Gambar 11 menunjukkan jalur tempuh robot tanpa menggunakan metode.



Gambar 11. Jalur tempuh robot tanpa metode

Pengujian tanpa metode dilakukan sebanyak 10 kali dengan ruang start dan api yang sama. Waktu tempuh robot tanpa metode ditunjukkan pada tabel 3.

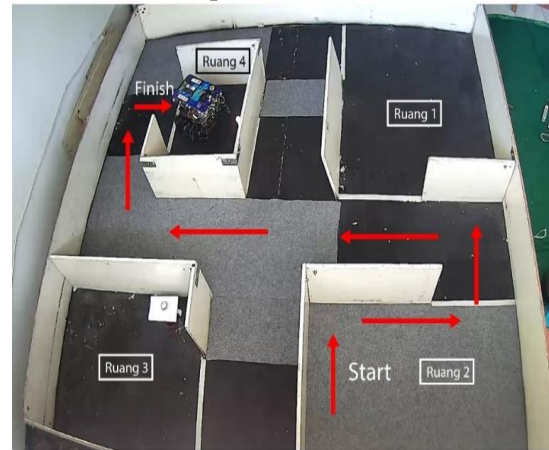
Tabel 3. Waktu tempuh robot menemukan sumber/titik api tanpa metode

Percobaan ke	Waktu tempuh untuk sampai target (detik)	Tingkat Keberhasilan
1	225	berhasil
2	218	berhasil
3	222	berhasil
4	219	berhasil
5	223	berhasil
6	230	berhasil
7	225	berhasil
8	224	berhasil
9	218	berhasil
10	220	berhasil

B. Hasil Pengujian FQL

Pengujian ini dilakukan untuk mengetahui kehandalan metode FQL untuk pencarian jalur tercepat menemukan titik api sehingga waktu tempuh akan menjadi lebih singkat. Pengujian dilakukan beberapa kali sampai mendapat hasil yang maksimal. Pada pengujian robot harus menuju target berupa api dengan jalur yang tercepat. Robot *start* pada ruang 2 dan titik api pada ruang 4. Pada awal percobaan robot gagal menemukan titik api. Setelah beberapa percobaan robot dapat menemukan sumber/titik api, dengan semakin sering melakukan percobaan waktu tempuh juga semakin singkat. Untuk jalur tempuh robot dengan *metode* FQL dapat dilihat pada gambar 12. Saat pengujian data aksi, *punishment* dan *reward* disimpan pada SD card. Data tersebut

disimpan saat awal robot di jalankan sampai menemukan titik api.



Gambar 12. Jalur tempuh robot dengan metode FQL

Tabel 4. Waktu tempuh robot menemukan sumber/titik api menggunakan metode FQL

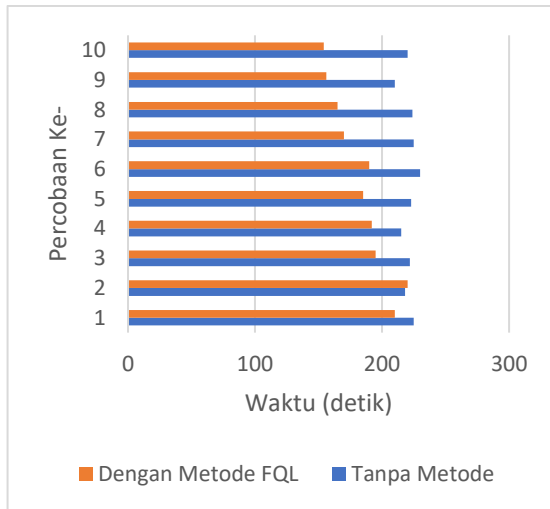
Percobaan Ke	Waktu tempuh untuk sampai target (detik)	Tingkat Keberhasilan
1	210	Tidak berhasil
2	220	Tidak berhasil
3	189	berhasil
4	188	berhasil
5	185	berhasil
6	190	berhasil
7	170	berhasil
8	165	berhasil
9	156	berhasil
10	154	berhasil

Pergerakan robot didapat dari sistem *fuzzy* yang diberikan untuk mempercepat keputusan dari aksi robot. Dari aksi tersebut maka menjadi pertimbangan nilai *reward* atau *punishment* yang akan diberikan. Ketika jalan yang dilalui itu merupakan jalur menuju titik api maka robot akan di berikan *reward*. Begitu sebaliknya apabila jalan yang dilalui robot itu salah maka mendapatkan *punishment*. Percobaan metode FQL sebagai pencarian jalur tercepat dilakukan sebanyak 10 kali. Berdasarkan pengujian tersebut didapatkan hasil yang ditunjukkan pada tabel 4.

C. Analisa

Jalur yang didapat untuk pengujian tanpa metode dapat dilihat pada gambar 11 dan jalur pengujian dengan metode *FQL* dapat dilihat pada gambar 12. Untuk ruangan pengujian, *start* berada pada ruang 2 dan titik api pada ruang 4. Pada pengujian tanpa metode robot *start* kemudian menyusuri dinding dengan menggunakan *scan*

kiri menuju ruang 3 terlebih dahulu. Ketika tidak ada api maka akan melanjutkan perjalanan ke ruang 4. Untuk jalur dengan menggunakan FQL robot bergerak dari ruang 2 langsung menuju ke ruang 4 tanpa berbelok pada ruang 3. Dari data tersebut dapat disimpulkan bahwa robot dengan menggunakan FQL dapat menemukan jalur tercepat tanpa berbelok pada ruang yang tidak terdapat titik api.



Gambar 13. Perbandingan waktu tempuh pengujian tanpa metode dengan menggunakan metode (FQL)

Untuk waktu tempuh robot tanpa menggunakan metode FQL dapat dilihat pada tabel 3 dan waktu tempuh dengan menggunakan metode FQL dapat dilihat pada tabel 4. Dari dua tabel tersebut didapat grafik perbandingan pada gambar 13. Dari grafik perbandingan waktu tempuh dapat diketahui bahwa waktu tempuh robot tanpa metode FQL berkisar antara 210–230 detik dengan ketidakpastian waktu pada setiap percobaan. Terkadang robot dapat melakukan dengan waktu yang baik terkadang juga dengan waktu yang lama. Berbeda dengan waktu tempuh robot dengan menggunakan metode FQL semakin sering robot ini berlatih maka semakin lebih cepat waktu yang dibutuhkan dalam menemukan sumber/titik api. Untuk waktu tercepat yang dicapai robot dalam menemukan sumber/titik api yaitu 154 detik. Pengujian dengan menggunakan metode FQL lebih cepat 56 detik dari pengujian tanpa metode.

IV. KESIMPULAN

Perencanaan jalur tercepat pada robot berkaki enam pemadam api menggunakan metode *Fuzzy Q-Learning* (FQL) ini didapat dari input data

sensor. Data yang didapat kemudian diproses pada metode fuzzy sehingga *output* yang didapat kemudian dikirim ke *slave* sebagai perintah pergerakan. Semakin sering melakukan uji coba pada robot, maka semakin terlatih dan lebih cepat dalam menemukan sumber/titik api. Dari percobaan sebanyak 10 kali, tingkat keberhasilan dari metode *Fuzzy Q-Learning* (FQL) ini sebesar 80%. Dengan rincian 8 kali keberhasilan dalam menemukan sumber/titik api dan 2 kali gagal dalam menemukan sumber/titik api. Jika dibandingkan data waktu tempuh robot antara tanpa menggunakan metode FQL dengan menggunakan metode FQL diketahui bahwa penggunaa metode FQL dapat meminimalkan waktu tempuh menuju titik api.

DAFTAR PUSTAKA

- [1] Hidayat, S. and Nasution, I., “*Pelayanan Pencegahan dan Pemadam Kebakaran Pemerintah bagi Masyarakat dalam Mengatasi Musibah*”, *Jurnal Ilmu Pemerintahan dan Sosial Politik UMA (Journal of Governance and Political Social UMA)*, 1(2), 2013, pp. 176–191.
- [2] Siregar, L., “*Design and Implementation of Hardware’s Firefighting Legged Robot*”, *IOP Conference Series: Materials Science and Engineering*, 417(1). 2018. doi: 10.1088/1757-899X/417/1/012014.
- [3] Cherroun, L. and Boumehraz, M., “*Intelligent systems based on reinforcement learning and fuzzy logic approaches, application to mobile robotic*”, 2012 International Conference on Information Technology and e-Services, ICITeS, (1), 2012. doi: 10.1109/ICITeS.2012.6216661.
- [4] Latif, A., Shankar, K. and Nguyen, P. T., “*Legged Fire Fighter Robot Movement Using PID*”, *Journal of Robotics and Control (JRC)*, 1(1), pp. 15–18, 2020. doi: 10.18196/jrc.1104.
- [5] Nashrullah, E., and Triwiyatno, A., “*Pengendalian Robot Berkaki Enam Penjejak Dinding Menggunakan Metode Kontrol Proporsional Derivatif*”, *Transient: Jurnal Ilmiah Teknik Elektro*, vol. 2, no. 2, pp. 308-315, Jul. 2013., pp. 1–8.
- [6] Chen, C.H., Wang, C.C., Wang, Y.T., dan Wang, P. T., “*Fuzzy Logic Controller Design for Intelligent Robots*”, *Mathematical Problems in Engineering*, 2017. doi: 10.1155/2017/8984713.

- [7] Harper, M. Y., “*Learning and Motion planning for Gait-Based Legged Robots*”, Dissertation, Florida State University, College of Arts and Sciences, 2018.
- [8] Rajput, U. dan Kumari, M., “*Mobile robot Path Planning with Modified Ant Colony optimization*”, *International Journal of Bio-Inspired Computation*, 9(2), 2017, pp. 106-113. doi: 10.1504/IJBIC.2017.083133.
- [9] Rao, A. M., Ramji, K., Rao, B.S.S.S., Vasu, V, Puneeth, C., “*Navigation of Non-Holonmic Mobile Robot using Neuro-Fuzzy Logic with Integrated Safe Boundary Algorithm*”, *International Journal of Automation and Computing*, Vol. 14, Issue 3, June 2017, pp. 285-294. doi.org/10.1007/s11633.
- [10] Gitakarma, M. S. dan Nurhayata, G., “*Perancangan Behavior-Based Robot Dengan Algoritma Fuzzy Q-Learning (FQL) Pada Sistem Navigasi Robot Otonom*”, *Jurnal Sains dan Teknologi*, 2(1), 2013, pp. 139–150.
- [11] Lei, T. dan Ming, L., “*A Robot Exploration Strategy Based on Q-Learning Network*”, *Proceeding of Conference on Real-time Computing and Robotics*, June 6-9, 2016, pp. 57–62.
- [12] Arifin, S., Tandy, H. A. and Kristian, Y., “*Pencarian Rute Line Follower Mobile Robot Pada Maze Dengan Metode Q Learning*”, *Jurnal Otomasi Kontrol dan Instrumentasi*, 8(1), 2016, p. 55. doi: 10.5614/joki.2016.8.1.5.
- [13] James, S., “*3D Simulated Robot Manipulation Using Deep Reinforcement Learning*”, Individual Project MEng (Thesis), Imperial College London, June 2016.
- [14] Alfith, “*Perancangan Robot Cerdas Pemadam Api Dengan Sensor Thermal Array Tpa 81 Berbasis Microcontroller Arduino Mega 2560*”, *Jurnal Teknik Elektro ITP*, 5(2), 2016, pp. 95–102.
- [15] Sutawati, L. A., Kumara, I. N. S. dan Widiadha, W., “*Pengembangan Three Degree of Freedom Hexapod sebagai Robot Pemadam Api dengan Sensor UVTron Hamamatsu*”, *Majalah Ilmiah Teknologi Elektro*, 17(3), 2018, p. 417. doi: 10.24843/mite.2018.v17i03.p17.
- [16] Karakurt, T., Durdu, A. dan Yilmaz, N., “*Design of Six-Legged Spider Robot and Evolving Walking Algorithms*”, *International Journal of Machine Learning and Computing*, Vol. 5, No. 2, April 2015, 5(2), pp. 96–100. doi: 10.7763/IJMLC.2015.V5.490.
- [17] Prihastono, P. Santoso, A. Jazzidie, A., “*Autonomous Five Legs Robot Navigation Using Behaviour Based Control and Fuzzy Q-Learning*”, *International Journal of Academic Research*, Vol. 4, No. 2, March 2012, ISSN: 2075-4124, E-ISSN: 2075-7107.
- [18] Aye, Y. Y., “*Design of an Image-based Fuzzy Controller for Parking Problems of a Car-like Mobile Robot*”, Dissertation of Doctor of Philosophy in Engineering, Department of Intelligent Mechanical Systems Okayama University, 2017.