

IMPLEMENTASI ALGORITMA MAZE MAPPING PADA ROBOT PEMADAM API HEXAPOD

Bintang Surya Tryatmojo

Departement of Electrical Engineering, Faculty of Engineering
Universitas Muhammadiyah Yogyakarta

Kampus Terpadu UMY, JL. Brawijaya, Tamantirto, Kasihan, Bantul, Yogyakarta 55183, Indonesia.

*Corresponding author, email: bintangsurvat@gmail.com

ABSTRAK

Algoritma *maze mapping* merupakan salah satu algoritma pada robot *maze* berbasis teknik pemetaan (*mapping*), dimana digunakan untuk memecahkan atau mencari jalan keluar di dalam sebuah *maze* dalam rangka untuk mencapai targetnya. Pada penelitian ini, algoritma *maze mapping* diimplementasikan pada robot pemadam api *hexapod* menggunakan aturan KRPAI tahun 2018. Algoritma ini digunakan untuk menutupi kelemahan dari algoritma *wall following*, dimana agar robot mampu bernavigasi keluar dari jalur *maze* yang berbentuk *loop* dan mencapai target dengan jalur terpendek pada arena KRPAI tahun 2018.

Hasil dari penelitian ini adalah robot pemadam api *hexapod* telah mampu keluar dari jalur *maze* yang berbentuk *loop* dan mampu memasuki seluruh *room* api pada arena KRPAI tahun 2018. Sensor pendukung pada sistem robot mampu bekerja dengan baik. Hasil *tuning* kontrol *Proportional Derivative* pada navigasi *wall following* robot, secara empirik didapatkan performa stabil pada nilai $K_p=55$ dan $K_d=23$ dengan kecepatan rata-rata navigasi sebesar 21,8 cm/s. Hasil akhir implementasi algoritma *maze mapping* pada robot pemadam api *hexapod* memiliki tingkat keberhasilan mengikuti jalur dan menuju target *room* yang telah ditentukan sebesar 87,5% dari total 16 kali percobaan, dengan rata-rata waktu tercepat sebesar 81,8 detik dan waktu terlama sebesar 103,8 detik.

Kata kunci : Algoritma, *wall following*, *maze mapping*, *maze*, kontrol *proportional derivative* navigasi, robot *hexapod*, KRPAI.

1. Pendahuluan

Kontes Robot Pemadam Api Indonesia (KRPAI) merupakan ajang perlombaan robot tahunan tingkat mahasiswa yang diselenggarakan oleh KEMENRISTEK-DIKTI di Indonesia. Misi utama dari KRPAI yaitu robot harus mampu mencari dan memadamkan sebuah titik api yang diletakkan secara acak di dalam salah satu ruang (*room*) pada arena yang berbentuk *maze* atau labirin. Robot yang tercepat melakukan misi memadamkan api dan mendapatkan akumulasi point akhir dengan nilai terkecil akan menjadi pemenang. Pada aturan KRPAI tahun 2018, bentuk *maze* yang digunakan terdiri atas beberapa model konfigurasi, dimana model konfigurasi *maze* ini dipilih berdasarkan undian sebelum perlombaan dimulai.

Pada umumnya robot *hexapod* di KRPAI melakukan navigasi penjelajahan pencarian api pada suatu *maze* dengan menggunakan algoritma *wall following* (telusur dinding) dengan konsep dasar mengikuti dinding kanan atau mengikuti dinding kiri. Namun pada kasus tertentu, penerapan algoritma *wall following* ini memiliki kelemahan apabila robot berhadapan dengan model konfigurasi lintasan *maze* yang berbentuk *loop* atau lingkaran, karena pada model lintasan yang berbentuk *loop* ini robot cenderung akan terus berputar mengintari jalur lintasan *maze* yang sama dan akan terus memasuki ruang api yang sama. Dengan demikian, apabila posisi titik api berada di dalam ruang lain yang berbeda lintasan, maka robot akan tersesat dan tidak akan pernah

dapat menjangkau lokasi ruang api tersebut hingga waktu yang diberikan habis.

Pada penelitian tugas akhir ini akan dirancang dan diimplementasikan algoritma *maze mapping* pada robot pemadam api *hexapod* sebagai solusi untuk menutupi kelemahan dari algoritma *wall following* ketika melakukan misi pencarian ruang api pada sebuah *maze*. Pada penelitian ini juga akan dilakukan *tuning* kontrol PID pada robot agar didapatkan mobilisasi yang baik dan didapatkan *error* yang minimum ketika merealisasikan algoritma *maze mapping*.

Dengan menggunakan algoritma *maze mapping* robot *hexapod* diharapkan mampu mengatasi jalur lintasan *maze* yang berbentuk *loop*, mengitari seluruh lintasan *maze*, dan mampu menjelajahi seluruh ruang api dengan waktu yang singkat ketika melakukan misi pencarian api.

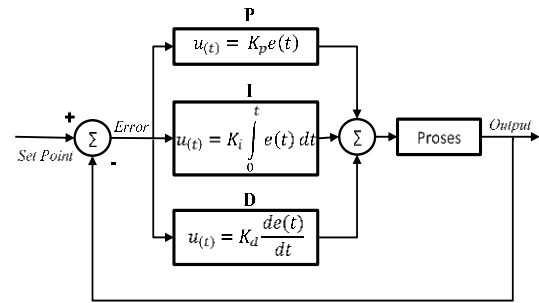
2. Dasar Teori

2.1. Robot Hexapod

Robot *hexapod* adalah salah satu jenis robot berkaki yang berbentuk menyerupai serangga yang bergerak dengan menggunakan enam buah kaki. Robot *Hexapod* dinilai memiliki fleksibilitas yang tinggi karena robot secara statis dapat berjalan dengan stabil dengan menggunakan tiga kaki secara bergantian berdasarkan ritme yang telah diatur dengan menggunakan formula *inverse kinematics*. (Khidir, 2014)

2.2. Kontrol PID

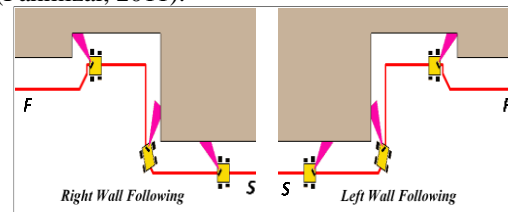
Kontrol PID merupakan sistem kendali klasik yang terdiri atas 3 elemen yaitu *Proportional*, *Integral*, dan *Derivative*. Kontrol PID bekerja dengan memanfaatkan mekanisme umpan balik atau *feed back* pada sebuah sistem kendali. Kontrol PID secara kontinyu akan menghitung nilai *error* atau nilai kesalahan sebagai beda antara nilai *set point* yang diinginkan dengan nilai *variable* terukur (*output*), supaya didapatkan nilai *error* yang seminimal mungkin.



Gambar 2.1 Blok Diagram Gabungan Kontrol P,I, dan D

2.3. Wall Following Algorithm

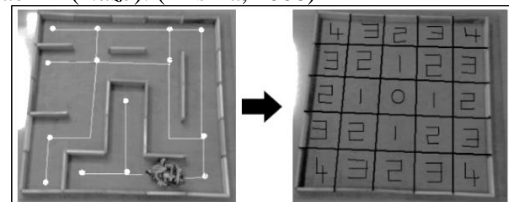
Wall following algorithm adalah salah satu algoritma untuk menyediakan orientasi gerak kepada robot dalam bernavigasi dengan metode menyusuri kontur dinding. Prinsip kerja dari sistem navigasi ini adalah dengan mengatur jarak robot dengan dinding supaya tetap konstan. (Fahmizal, 2011).



Gambar 2.2. Ilustrasi Navigasi Wall Following (Sumber: Purnama, 2017)

2.4. Maze Mapping Algorithm

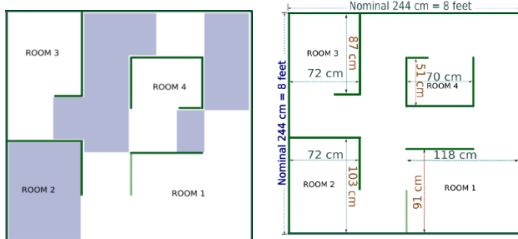
Maze mapping Algorithm atau *maze solving algorithm* adalah algoritma yang bekerja dengan cara memetakan atau menggambarkan beberapa kondisi lingkungan di dalam sebuah labirin (*maze*), dimana kemudian kondisi lingkungan yang telah dipetakan tersebut digunakan untuk menemukan jalan keluar atau jalur terpendek dari sebuah labirin (*maze*). (Mishra, 2008)



Gambar 2.3. Ilustrasi Metode Memorisasi Maze Mapping Micromouse Robot. (Sumber: Mishra, 2008)

2.5 KRPAI 2018

Gambaran singkat sistem pertandingan Kontes Robot Pemadam Api Indonesia (KRPAI) 2018 yaitu sebuah robot berkaki diletakkan pada sebuah arena berupa labirin (*maze*) yang menyerupai rumah dengan 4 buah ruangan, dimana pada salah satu ruangan terdapat sebuah lilin yang mewakili sebuah titik api. Robot harus dapat menemukan keberadaan titik api tersebut lalu kemudian memadamkannya.

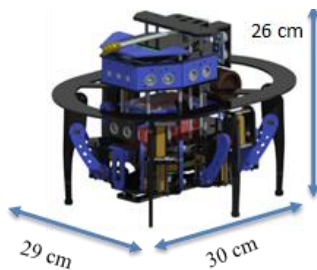


Gambar 2.4. Bentuk Labirin Arena KRPAI 2018

3. Perancangan Sistem

3.1. Perancangan Mekanik Robot

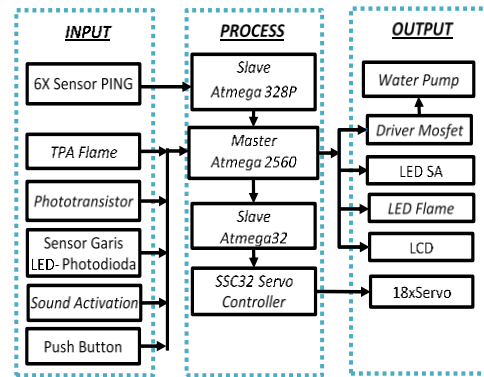
Perancangan mekanik robot pada penelitian ini disesuaikan berdasarkan ketentuan aturan KRPAI tahun 2018 dimana ukuran maksimal dimensi robot tidak boleh melebihi panjang=31cm, lebar=31cm, dan tinggi=27 cm, serta 1 kaki robot minimal memiliki 2 derajat kebebasan gerak.



Gambar 3.1. Perancangan Mekanik Robot Hexapod dalam Bentuk 3d

3.2. Perancangan Hardware Utama

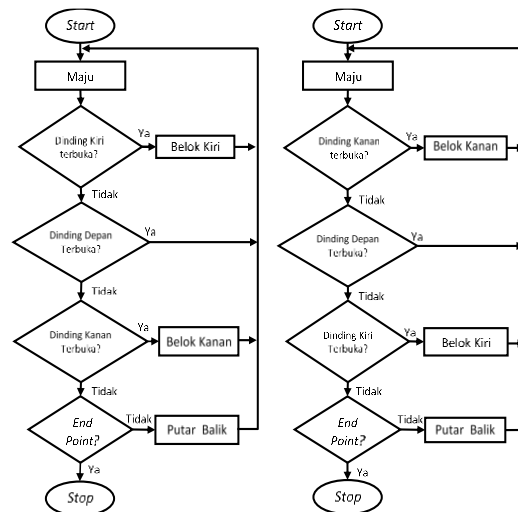
Perancangan *hardware* utama merupakan tahap perancangan berupa piranti-piranti elektronik. Piranti elektronik pada sistem ini digunakan sebagai suatu pengendalian dan pengukuran utama sebuah sistem. Pada gambar 3.2 adalah blok diagram perancangan sistem *hardware* robot pemadam api *hexapod*.



Gambar 3.2. Blok Diagram Sistem Hardware Utama Robot Pemadam Api Hexapod

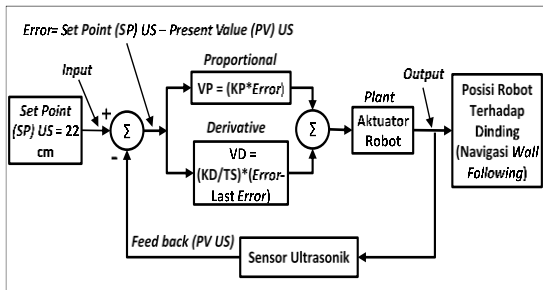
3.3 Perancangan Program Navigasi Wall Following dengan Kontrol PD

Flowchart program navigasi *wall following* ditunjukkan pada gambar 3.3.



Gambar 3.3 Flowchart Navigasi Wall Following

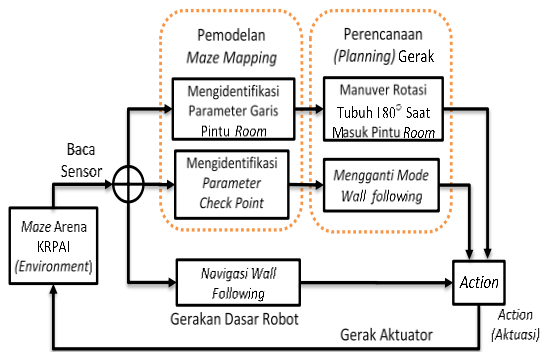
Sistem kontrol yang dirancang untuk memperoleh mobilisasi yang baik ketika bernavigasi, digunakan metode kontrol *proportional derivative*, dengan *setpoint* yang diberikan dari hasil pembacaan sensor ultrasonik kepada komparator yakni sebesar 22 cm dari dinding, seperti yang ditunjukkan pada gambar 3.4.



Gambar 3.4 Perancangan Sistem Kontrol *Proportional Derivative*

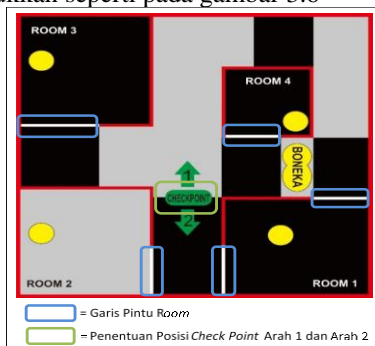
3.4 Konsep Perancangan Algoritma *Maze Mapping*

Pemodelan *maze mapping* yang digunakan sebagai perencanaan gerak sistem aktuator robot dijelaskan pada blok diagram seperti pada gambar 3.5 berikut.



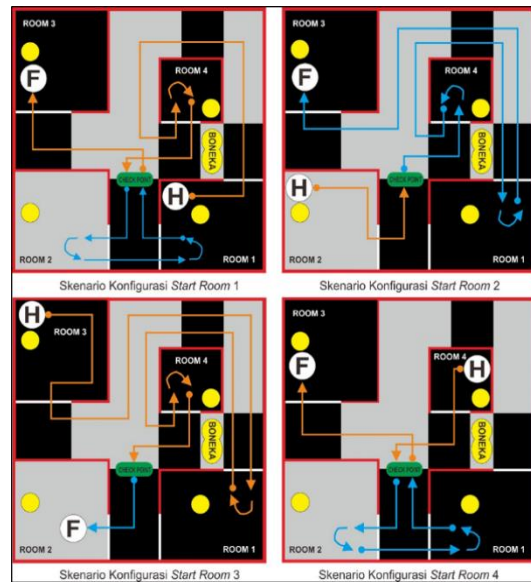
Gambar 3.5 Sistem Gerak Robot Berbasis *Maze Mapping*

Ilustrasi pemodelan *maze mapping* pada arena KRPAI 2018 dalam mengidentifikasi parameter checkpoint dan parameter garis pintu *room* api yang dirancang pada penelitian ini ditunjukkan seperti pada gambar 3.6



Gambar 3.6 Penentuan Parameter *Maze Mapping*.

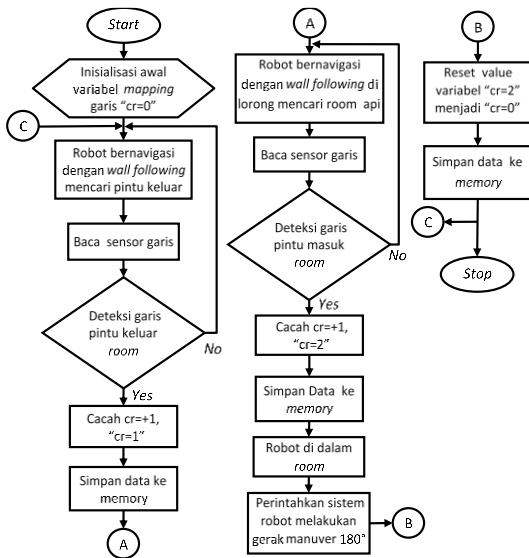
Dari hasil penentuan parameter *maze mapping*, maka kemungkinan-kemungkinan arah gerak navigasi dan manuver robot pada perancangan ini dapat ditunjukkan seperti gambar 3.7 berikut. Garis warna *orange* menunjukkan robot menggunakan mode telusur kiri, dan garis warna *biru* menunjukkan robot menggunakan mode telusur kanan.



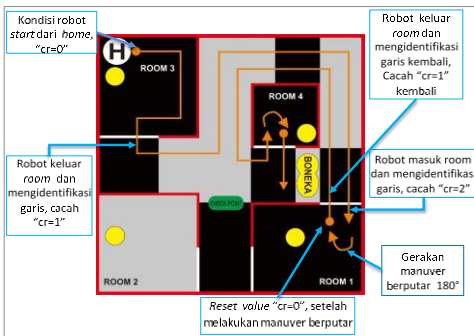
Gambar 3.7 Penentuan Parameter *Maze Mapping*.

3.5 Perancangan Program Algoritma *Maze Mapping*

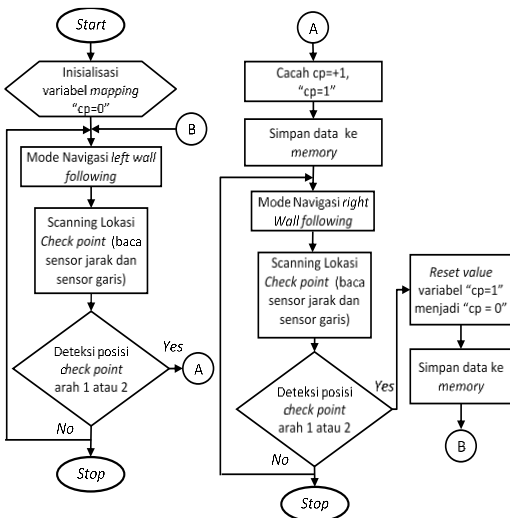
Pada bahasa pemrograman, algoritma algoritma *maze mapping* dirancang dengan menggunakan memorisasi variabel sebagai indikator hasil *mapping* parameter.



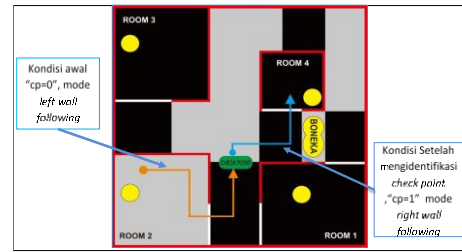
Gambar 3.8 Program Mapping Parameter Garis



Gambar 3.9 Ilustrasi Gerakan Manuver Untuk Memangkas Waktu Jelajah



Gambar 3.10 Program Mapping Parameter Checkpoint



Gambar 3.11 Ilustrasi Mode Navigasi Robot Setelah Mengidentifikasi Check Point

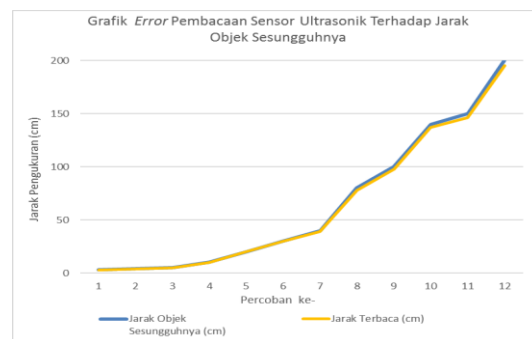
4. Data Hasil Pengujian dan Pembahasan

4.1. Pengujian Sensor Ping Ultrasonik

Pengujian pengukuran jarak sensor PING ultrasonik pada robot dengan range 3 cm hingga 200 cm memiliki total rata-rata error sebesar 1.2 % terhadap jarak sesungguhnya.

Tabel 4.1 Data Hasil Pengujian Sensor Ultrasonik PING

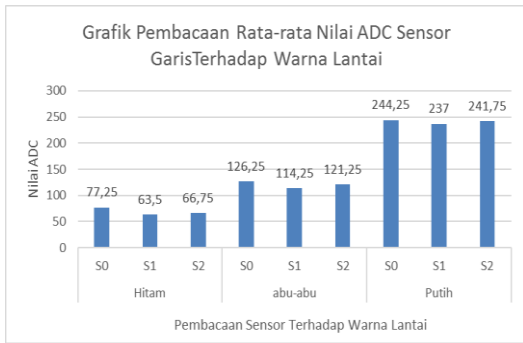
No	Jarak Objek Sesungguhnya (cm)	Pembacaan Ultrasonik (cm)	Error (%)
1	3	3	0
2	4	4	0
3	5	5	0
4	10	10	0
5	20	20	0
6	30	30	0
7	40	39	2,5
8	80	78	2,5
9	100	98	2
10	140	137	2,15
11	150	146	2,7
12	200	195	2,5
Rata-rata Error (%)			1,20%



Gambar 3.12 Grafik Error Pembacaan Sensor Ping Ultrasonik

4.2. Pengujian Sensor Garis Led-Potodiada

Hasil pengujian sensor garis Led-Photodiada pada bidang warna hitam, abu-abu, dan putih ditunjukkan seperti pada blok diagram gambar 3.10



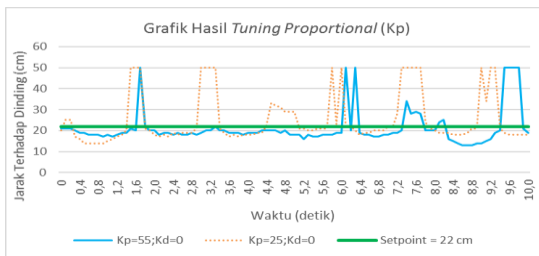
Gambar 3.13 Grafik Pembacaan Sensor Garis Led-Photodioda Terhadap Warna Lantai

4.3. Pengujian Tuning Kontrol Proportional Derivative

Pengujian Tuning Kp dan Kd dilakukan pada navigasi wall following robot.

Tabel 4.2 Data Hasil Tuning Eksperimen Parameter Kp

Kp	Kd	Kondisi Robot Saat Bernavigasi
25	0	Respon robot terlihat tampak sangat lambat, beresilasi sedang dan tidak stabil
35	0	Respon robot terlihat tampak meningkat, beresilasi rendah dan masih tidak stabil
55	0	Respon robot terlihat tampak cepat dan stabil, namun masih mengalami osilasi
60	0	Respon robot terlihat tampak responsif, tidak stabil dan osilasi sangat tinggi

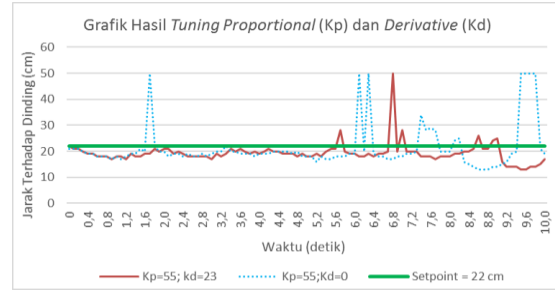


Gambar 3.14 Grafik Pengaruh Hasil Tuning Kp

Pemberian Kp=25, akibat sistem yang tidak responsif robot tampak memiliki lonjakan pembacaan jarak yang overshoot cenderung lebih lama set point dibandingkan dengan pemberian Kp=55.

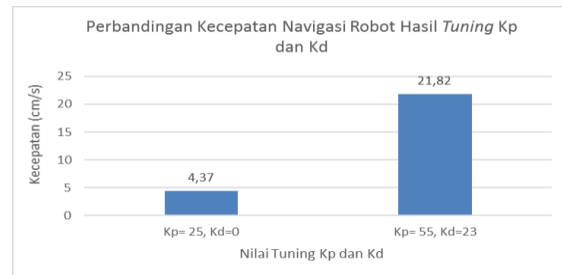
Tabel 4.3 Data Hasil Tuning Eksperimen Parameter Kp dan Kd

Kp	Kd	Kondisi Robot Saat Bernavigasi
55	8	Respon robot terlihat tampak sangat cepat dan terkadang robot masih beresilasi
55	23	Respon robot terlihat tampak cepat, osilasi jauh berkurang dan stabil
55	40	Respon robot terlihat tampak semakin melambat dan stabil
55	50	Respon robot terlihat tampak semakin melambat dan sedikit beresilasi



Gambar 3.15 Grafik Pengaruh Hasil Tuning Kp dan Kd

Berdasarkan data grafik pada gambar 3.15 dapat diketahui bahwa navigasi robot didapatkan kondisi yang stabil dan responsif pada Kp=55, dan Kd=23;



Gambar 3.16 Grafik Perbandingan Kecepatan Navigasi Hasil Tuning Kp dan Kd

Rata-rata kecepatan navigasi tuning optimal pada Kp=55 dan Kd=23 didapatkan kecepatan navigasi sebesar 21,8 cm/s.

4.4. Pengujian Indikator Maze Mapping.

Pada pengujian ini robot telah berhasil menunjukkan indikator-indikator nilai variabel sesuai dengan program algoritma yang telah dirancang.

Tabel 4.4 Hasil Pengujian Indikator Mapping Garis Pintu Room

No	Keluar Room		Masuk Room	
	Indikator Mapping "cr=1"	Indikator Mapping "cr=2"	Gerakan Manuver Berputar 180°	Reset Indikator Mapping "cr=0"
1	✓	✓	✓	✓
2	✓	✓	✓	✓
3	✓	✓	✓	✓
4	✓	✓	✓	✓
5	✓	✓	✓	✓

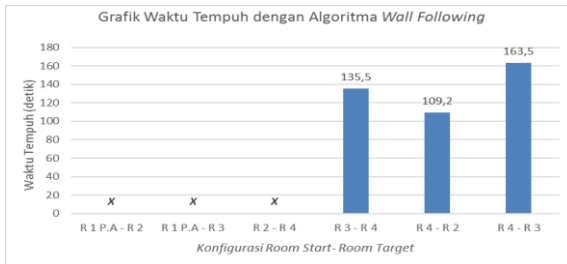
Tabel 4.5 Data Pengujian Indikator Mapping Check Point Arah 1

No	Sebelum Melewati Check Point Arah 1		Setelah Melewati Check Point Arah 1	
	Variabel "cp"	Perilaku Navigasi	Variabel "cp"	Perilaku Navigasi
	1	0	Left Wall Follow	1
2	0	Left Wall Follow	1	Right Wall Follow
3	1	Right Wall Follow	0	Left Wall Follow
4	1	Right Wall Follow	0	Left Wall Follow

Tabel 4.6 Data Pengujian Indikator Mapping Check Point Arah 2

No	Sebelum Melewati Check Point Arah 2		Setelah Melewati Check Point Arah 2	
	Variabel "cp"	Perilaku Navigasi	Variabel "cp"	Perilaku Navigasi
	1	0	Left Wall Follow	1
2	0	Left Wall Follow	1	Right Wall Follow
3	1	Right Wall Follow	0	Left Wall Follow
4	1	Right Wall Follow	0	Left Wall Follow

4.5. Pengujian dengan Algoritma Wall Following.
 Dengan Algoritma Wall Following robot masih belum mampu menjelajahi beberapa room api



Gambar 3.17 Grafik Waktu Tempuh dengan Algoritma Wall Following

4.6. Pengujian dengan Algoritma Maze Mapping
 Pengujian algoritma Maze Mapping dilakukan percobaan pada 4 buah konfigurasi dengan total sebanyak 16 kali .

Tabel 4.7 Hasil Pengujian Konfigurasi Start (Home) Room 1-P.A

No	Room Tujuan								
	R4		R2		R1-P.B		R3		
	Status	Waktu(s)	Status	Waktu(s)	Status	Waktu(s)	Status	Waktu(s)	
1	✓	33,9	✓	56,0	✓	65,0	✓	86,9	
2	✓	35,1	✓	56,9	✓	66,7	✓	87,8	
3	✓	33,5	✓	55,7	✓	66,5	✗	✗	
4	✓	34,1	✓	56,8	✓	65,9	✗	✗	
Rata-rata Waktu Tempuh Room Akhir (Finish) (s)									87,4

Tabel 4.8 Hasil Pengujian Konfigurasi Start (Home) Room 3

No	Room Tujuan								
	R4		R1-P.A		R3		R1-P.B		
	Status	Waktu(s)	Status	Waktu(s)	Status	Waktu(s)	Status	Waktu(s)	
1	✓	24,9	✓	54,6	✓	85,3	✓	104,4	
2	✓	24,6	✓	54,9	✓	85,1	✓	104,3	
3	✓	24,6	✓	53,8	✓	83,1	✓	101,7	
4	✓	24,6	✓	54,5	✓	85,2	✓	104,7	
Rata-rata Waktu Tempuh Room Akhir (Finish) (s)									103,8

Tabel 4.9 Hasil Pengujian Konfigurasi Start (Home) Room 3

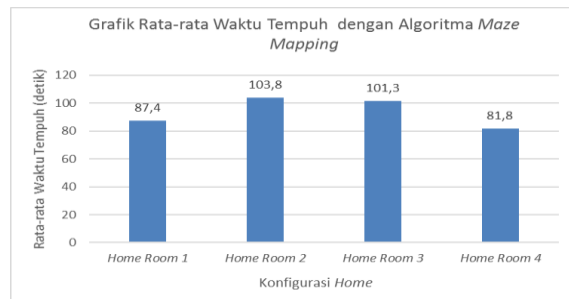
No	Room Tujuan								
	R1-P.A		R4		R2		R1-P.B		
	Status	Waktu(s)	Status	Waktu(s)	Status	Waktu(s)	Status	Waktu(s)	
1	✓	34,8	✓	68,9	✓	90,6	✓	99,9	
2	✓	35,4	✓	67,9	✓	89,9	✓	98,9	
3	✓	35,9	✓	69,9	✓	91,6	✓	100,9	
4	✓	36,0	✓	70,5	✓	93,3	✓	105,3	
Rata-rata Waktu Tempuh Room Akhir (Finish) (s)									101,3

Tabel 4.10 Hasil Pengujian Konfigurasi Start (Home) Room 4

No	Room Tujuan								
	R2		R1-P.B		R3		R1-P.A		
	Status	Waktu(s)	Status	Waktu(s)	Status	Waktu(s)	Status	Waktu(s)	
1	✓	19,8	✓	28,9	✓	49,0	✓	79,9	
2	✓	21,6	✓	30,0	✓	50,7	✓	81,9	
3	✓	20,3	✓	29,8	✓	49,8	✓	81,7	
4	✓	19,9	✓	30,9	✓	51,1	✓	83,7	
Rata-rata Waktu Tempuh Room Akhir (Finish) (s)									81,8

Pada Pengujian dengan algoritma maze mapping diperoleh persentase keberhasilan sebesar :

$$\begin{aligned} \% \text{ keberhasilan} &= (\text{jumlah keberhasilan} / \text{banyak percobaan}) \times 100 \% \\ &= (14/16) \times 100\% \\ &= 87,5\% \end{aligned}$$



Gambar 3.18 Grafik Rata-rata Waktu Tempuh dengan Algoritma Maze Mapping.

Pada Implementasi algoritma maze mapping diperoleh waktu rata-rata tempuh tercepat dalam menapai target sebesar 81,8 detik dan waktu terlama sebesar 103,8 detik.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, maka dapat ditarik beberapa simpul penting diantaranya:

1. Perancangan algoritma *maze mapping* telah berhasil dilakukan dan telah mampu memberikan solusi untuk menutupi kelemahan dari algoritma *wall following*.
2. Sensor PING ultrasonik dan sensor garis LED-Photodiode mampu bekerja dengan baik untuk mengidentifikasi dan melakukan *mapping* parameter pada *maze* KRPAI 2018.
3. Hasil tuning eksperimen Parameter *Proportional Derivative* untuk stabilisasi robot saat bernavigasi didapatkan nilai optimal menggunakan $K_p=55$ $K_d=23$, dengan kecepatan rata-rata robot yang dihasilkan sebesar 21,8 cm/s.
4. Hasil unjuk kerja akhir dari implementasi algoritma *maze mapping* pada robot pemadam api *hexapod*, memiliki tingkat keberhasilan mengikuti dan menuju target yang telah ditentukan sebesar 87,5% dari total 16 kali percobaan, dengan rata-rata waktu tercepat sebesar 81,8 detik dan waktu terlama sebesar 103,8 detik.

5.2. Saran

Beberapa saran untuk pengembangan penelitian tugas akhir ini adalah sebagai berikut :

1. Hendaknya dapat dipadukan dengan logika atau metode memadamkan api yang sesungguhnya.
2. Hendaknya dapat dipadukan dengan metode *SLAM*.
3. Sensor navigasi yang digunakan pada robot dapat diganti dengan sensor yang lebih baik lagi seperti sensor laser.
4. Sensor garis yang digunakan pada robot dapat diganti dengan sensor yang lebih responsif lagi seperti sensor *optical* atau *fiber optic*.
5. Pada sistem kontrol PID robot dapat ditambahkan kontrol *integral* agar dapat menghilangkan kondisi respon *steady state error* pada sistem, dan metode *tuning* parameter PID dapat digunakan metode yang lebih baik lagi seperti menggunakan metode *tuning* menurut kaidah *Ziegler Nichols*.
6. Untuk membantu navigasi *maze mapping* hendaknya dapat ditambahkan sensor kompas.

7. Pada penelitian selanjutnya, hendaknya implementasi algoritma *maze mapping* ini juga dapat diterapkan pada misi *back to home* KRPAI.

Daftar Pustaka

Fahmizal, Effendi, A.K., Iskandar, R. (2011). *Implementasi Sistem Navigasi Behavior Based dan Kontroler PID pada Manuver Robot Maze*. Institut Teknologi Sepuluh Noverber. Surabaya.

Fatchurrohman, A.F. (2014). *Robot Line Follower PID Sebagai Media Pembelajaran Aplikasi Mikrokontroler Di Jurusan Pendidikan Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta*. Universitas Negeri Yogyakarta. Yogyakarta

Hidayatullah, A.H., Handayani, A.N., dan Fuady, M.J. (2017). *Performance Analysis of A* Algorithm to Determine Shortest Path of Fire Fighting Robot*. IEEE

Jepry. (2010). *Perancangan Pengendali PID Pada Proportional Valve*. Universitas Indonesia. Depok

Khidir, Malik. (2013). *Sistem Kontrol Gerak Robot Hexapod Menggunakan Algoritme Tigerpod Gait Berbasis Inverse Kinematic*. Fakultas MIPA Universitas Gadjah Mada. Yogyakarta.

Mishra, Swati. (2008). *Maze Solving Algorithm for Micro Mouse*. IEEE International Conference on Signal Image Technology and Internet Based Systems.

Nugraha, I.M. , A., Hendriawan, A., Akbar, R. (2011). *Penerapan Algoritma Maze Mapping untuk Menyelesaikan Maze pada Line Follower Robot*. EEPIS-ITS. Surabaya.

Primanio. (2007). *Pencarian Jalan Keluar Labirin dengan Metode Wall Follower*. Institut Teknologi Bandung. Bandung

Purnama, H.S. (2017). *Implementasi PID Wall Following pada Robot Hexapod untuk Kontes Robot Pemadam Api Indonesia*. Universitas Ahmad Dahlan. Yogyakarta

- Rahman, Abdullah. (2010). *Penerapan Algoritma Flood Fill untuk Menyelesaikan Maze pada Line Follower Robot*. EEPIS-ITS . Surabaya
- Qomaruddin, M.C., Alasiry A.H., & Tamami, N..(2017). *Routing Algorithm in Legged Robot with Dynamic Programming and Monte Carlo Localization*. IEEE
- Sitepu, N.M . (2013). *Implementasi Navigasi Berbasis Maze Mapping Pada Robot Pemadam Api Beroda*. Universitas Kristen Maranatha. Bandung
- Suryatini, F., Kustija, J., & Haritman,E. (2013). *Robot Cerdas Pemadam Api Menggunakan PING Ultrasonic Range Finder dan Uvtron Flame Detector Berbasis Mikrokontroler Atmega 128*. Universitas Pendidikan Indonesia. Bandung
- Sutrisna, R.Y. (2014). *Algoritma Runut Balik Robot Pemadam Api*. Institut Teknologi Bandung. Bandung
- Pratama, H., Haritman, E., & Gunawan, T. (2012). *Akuisisi Data Kinerja Sensor Ultrasonik Berbasis Sistem Komunikasi Serial Menggunakan Mikrokontroler Atmega 32*. Universitas Pendidikan Indonesia. Bandung
- Bejo, A. (2008). *C & AVR Rahasia Kemudahan Bahasa C dalam Mikrokontroler ATMEGA8535*. Yogyakarta. Graha Ilmu.
- Knuth, Donal. (1968). *The Art of Computer Programming*. Amerika-Serikat. Addison-Wesley.
- Ogata, Katsuhiko (1993). *Teknik Kontrol Automatik-terjemahan : Ir. Edi Laksono*. Erlangga. Jakarta.
- Pitowarno, Endra. (2006). *Robotika Desain, Kontrol, dan Kecerdasan Buatan*. C.V Andi Offset. Yogyakarta.
- Data Sheet PING Ultrasonic Sensor*.
- DIKTI (2018). *Panduan Kontes Robot Pemadam Api Indonesia (KRPAI) Berkaki 2018*.
<https://robotdyn.com/mega-2560-pro-mini-atmega2560-16au.html>. [Diakses: 23 Agustus 2018].
- <https://robotdyn.com/nano-v3-atmega-328-usb-ttl-ch340g-micro-usb.html>. [Diakses: 23 Agustus 2018].
- www.immersa-lab.com/pengenalan-codevision-avr.htm. [Diakses: 23 Agustus 2018].