IMPLEMENTASI METODE SIMPLE MAZE WALL FOLLOWER DENGAN MENGGUNAKAN FREE RTOS PADA ROBOT MAZE

SKRIPSI

Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik

Disusun oleh:

Agung Widya Gumelar

NIM: 145150300111066



PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

JUDUL SKRIPSI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik

Disusun Oleh:
Agung Widya Gumelar
NIM: 145150300111066

Skripsi ini telah diuji dan dinyatakan lulus pada 18 Juli 2018 Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

<u>Dahnial Syauqy, S.T., M.T., M.Sc.</u> NIK: 201607 870423 1 022 Sabriansyah Rizqika Akbar, S.T., M.Eng
NIP: 19820809 201212 1 004

Mengetahui Ketua Jurusan Teknik Informatika

<u>Tri Astoto Kurniawan, S.T., M.T., Ph.D</u> NIP: 19710518 200312 1 001





PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsurunsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 21 Juli 2018

METERAI TEMPEL 94956ABF104247442
ENAM KIRU KUPIAH
6000 DJP

Agung Widya Gumelar NIM: 145150300111066

KATA PENGANTAR

Dengan menyebut nama Allah Subhanahu Wa Ta'alla yang maha pengasih lagi maha penyayang, puji syukur saya panjatkan atas ke hadirat Allah Subhanahu Wa Ta'ala, karena ridha-Nya saya bisa menyelesaikan skripsi ini dan dengan menyelesaikan skripsi ini insya'Allah saya mendapatkan rahmat-Nya. Adapun maksud penyusunan skripsi ini adalah untuk memenuhi salah satu persyaratan dalam memenuhi ujian Sarjana Fakultas Ilmu Komputer dan mengharapkan rahmat Allah Subhanahu Wa Ta'ala. Judul skripsi yang disusun adalah: "Implementasi Metode *Simple Maze Wall Follower* dengan Menggunakan *Free* RTOS pada Robot *Maze*".

Banyak cobaan dan hambatan yang saya alami dalam menyusun skripsi ini, akan tetapi semua itu teratasi dengan baik berkat ridha Allah Azza Wa Jalla, do'a orang tua yang sama seperti do'a seorang Nabi kepada umatnya dan dukungan dari banyak pihak. Berdasarkan hal tersebut, pada kesempatan ini saya mengucapkan terima kasih kepada:

- 1. Allah Subhanahu Wa Ta'ala yang telah meridhai dan memberikan segala kebaikan kepada saya untuk menyelesaikan skripsi ini dan memberikan banyak hikmah dalam proses penyelesaian skripsi.
- 2. Nabi Muhammad Shalallahu 'Alaihi Wasallam yang telah memberikan sebaik-baik contoh dalam meminta pertolongan kepada Allah Subhanahu Wa Ta'ala dan cara melewati segala kesulitan dengan ketenangan.
- 3. Kedua orang tua yang telah meridhai saya, dengan ridhanya maka ridha Allah Subhanahu Wa Ta'ala hadir di setiap proses pengerjaan skripsi saya. Dan juga do'a kedua orang tua yang banyak berperan dalam penyelesaian skripsi.
- 4. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
- 5. Bapak Heru Nurwasito, Ir., M.Kom. selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya Malang.
- 6. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
- 7. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang dan selaku Dosen Pembimbing II yang telah banyak memberikan ide dalam penyelesaian skripsi, membimbing dengan sabar dan pengarahan yang sangat bermanfaat dalam penyelesaian skripsi.
- 8. Bapak Dahnial Syauqy, S.T., M.T., M.Sc. selaku Dosen Pembimbing I yang telah banyak memberikan pengarahan dalam menyelesaikan laporan skripsi dan revisi-revisi yang diberikan sangat bermanfaat dalam keilmuan suatu penulisan.

- 9. Seluruh dosen dan karyawan Fakultas Ilmu Komputer yang telah berperan dalam proses penyelesaian skripsi ini.
- 10. Kartika Putri Widya Prawesthi selaku adik kandung saya satu-satunya yang selalu memberikan kasih sayang dan semangat kepada saya.
- 11. Nelli Nur Rahma selaku teman dekat yang selalu memberikan semangat serta mengingatkan sholat dan nanti akan menjadi pendamping hidup saya jika saya rajin dalam beribadah.
- 12. Teman Bujang Ganong, yaitu Hanif Yudha Prayoga dan Chandra Salim yang telah memberikan saran dan semangat dalam mengerjakan tugas akhir ini.
- 13. Teman kontrakan H-320 yang telah senantiasa membantu dalam penulisan tugas akhir ini.
- 14. Teman 4 sejoli, yaitu Moh. Eka Wasis, Bimo Setyo Negoro, dan Habrin Andrian Hesta Prakosa yang telah setia meluangkan waktunya untuk kumpul bersama dan sharing terkait pengerjaan tugas akhir ini.
- 15. Teman Fake Friend yang selalu memberikan hiburan saat mulai lelah dalam mengerjakan tugas akhir.
- 16. Kos 577E yang selalu setia menemani ngopi ketika butuh inspirasi.
- 17. Mas Idang Wahyudin S. selaku teman yang setia menemani membantu dalam mengerjakan penelitian ini siang dan malam.
- 18. Mas Algarado P. Sumitro selaku teman sekaligus kakak yang setia mencetak laporan dari penulis dan mentraktir makan seminggu 3 kali.
- 19. Aulia Putri Noviyani, selaku teman dekat penulis yang selalu memberikan semangat dan menemani saat mengerjakan penelitian ini.

Saya menyadari bahwa laporan skripsi ini jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun saya ucapkan terima kasih. Saya mengharapkan semoga laporan skripsi ini dapat berguna.

Malang, 2 Juli 2018

Penulis Agungwidya7790@yahoo.com

ABSTRAK

Agung Widya Gumelar, Implementasi Metode Simple Maze Wall Follower dengan Menggunakan Free RTOS pada Robot Maze

Pembimbing: Dahnial Syauqy, S.T., M.T., M.Sc. dan Sabriansyah Rizqika Akbar, S.T., M.Eng

Salah satu jenis robot yang dapat mempermudah kegiatan manusia dalam menemukan tujuan di dalam labirin adalah Robot Maze Wall Follower, robot tersebut juga dapat diikutkan kontes robot dalam kategori robot pemadam kebakaran. Robot Maze Wall Follower menggunakan metode Simple Maze, yaitu berjalan mengikuti dinding kiri sebagai acuan utama dan garis berwarna gelap sebagai titik finish. Implementasi Free RTOS dibuat untuk multitasking dalam mengukur jarak antara robot dengan dinding yang menggunakan sensor ultrasonik dan photodioda. Setiap sensor digunakan sebagai input sistem dibuatkan task yang memiliki prioritas 1 dan modul L298N Driver Motor dual H-Bridge yang digunakan sebagai output sistem dibuatkan task 1. Hasil dari pengujian pengukuran jarak dinding menggunakan sensor ultrasonik HC-SR04 yang dilakukan pengujian sebanyak 7 kali dengan jarak berbeda-beda mempunyai tingkat error 0%, dan pembacaan garis yang berada di dasar labirin yang dilakukan pengujian sebanyak 6 kali dengan warna yang berbeda-beda menggunakan modul sensor photodioda yang mempunyai tingkat error 0%. Rata-rata hasil waktu untuk menyelesaikan qoal dengan jarak antara titik mulai dan selesai sepanjang 959 cm dan melewati 23 tikungan dan 2 jalan buntu oleh Robot Maze Wall Follower yang tidak menggunakan Free RTOS adalah 75,62429 detik dan yang menggunakan Free RTOS adalah 68,36143 detik dengan 7 kali pengujian. Maka robot dengan Free RTOS lebih cepat 7,262857 detik dari robot tanpa *Free* RTOS.

Kata kunci : Robot, Robot *Maze Wall Follower*, Labirin, Pemadam Kebakaran, Simple Maze, RTOS, Free RTOS, Dual H-Bridge, Multitasking, Task, photodiode

ABSTRACT

Agung Widya Gumelar, Implementasi Metode Simple Maze Wall Follower dengan Menggunakan Free RTOS pada Robot Maze

Pembimbing: Dahnial Syauqy, S.T., M.T., M.Sc. dan Sabriansyah Rizqika Akbar, S.T., M.Eng

One type of robot that can facilitate human activities in finding the destination in the labyrinth is Maze Wall Follower Robot, that robot can also be included in contest robot in category robot firefighter. Maze Wall Follower Robot use Simple Maze method, which follows the left wall as the main reference and the dark line as the finish point. Implementation Free RTOS is made for multitasking in measuring distances between robots and walls using ultrasonic sensors and photodiodes. Each sensor is used as system inputs made tasks that have priority 1 and L298N module Driver Motor dual H-Bridge which is used as system output made task 1. Result of measurement of wall distance measurement using ultrasonic sensor HC-SR04 which tested 7 times with different distance has 0% error, and line reading at the bottom of the maze is tested 6 times with different colors using photodiode sensor module which has 0% error. Average time result to complete goal with distance between start and finish point 959 cm and pass 23 turns and 2 dead end by Maze Wall Follower Robot which does not use Free RTOS is 75,62429 second and use Free RTOS is 68,36143 second with 7 times testing. Then the robot with Free RTOS faster 7.262857 seconds from robot without Free RTOS.

Keywords: Robot, Maze Wall Follower Robot, Maze, Simple Maze, Firefighter, RTOS, Free RTOS, H-Bridge, multitasking, task, photodiode

DAFTAR ISI

PENGESAH	HAN	ii
PERNYATA	AN ORISINALITAS	iii
KATA PENC	GANTAR	iv
ABSTRAK		vi
ABSTRACT		vii
DAFTAR IS	l	viii
DAFTAR TA	4BEL	xii
	AMBAR	
BAB 1 PEN	DAHULUAN	1
	ar Belakang	
1.2 Rur	musan Masalah	2
1.3 Tuj	uan	2
	infaat	
1.5 Bat	asan Masalah	3
1.6 Sist	tematika Penulisan	3
BAB 2 LAN	DASAN KEPUSTAKAAN	5
2.1 Tinj	jauan Pustaka	5
2.1.1	Robot Maze Solving Untuk Menyelesaikan Tiga Persimpang	an5
2.1.2	Kontrol Motor DC Penggerak Roda Pada Robot Cerdas	5
2.1.3	Peningkatan Kinerja Perangkat Elektronik Berbasis Mikr AVR 8 Bit Dengan Menggunakan RTOS (<i>Real Time Operation</i>)	ng System)
2.1.4	Penerapan Metode <i>Simple Maze</i> Pada Robot <i>Wall Follow</i> Menyelesaikan Jalur Dalam Menelusuri Sebuah Labirin	
2.2 Das	sar Teori	7
2.2.1	Arduino Nano	7
2.2.2	Motor DC MM10	8
2.2.3	Modul L298N <i>Driver</i> Motor DC	9
2.2.4	Sensor Ultrasonik HC-SR04	10
2.2.5	Modul Sensor Photodioda FC-51	11
2.2.6	Buzzer	12
2.2.7	PWM (Pulse Width Modulation)	12
2.2.8	Labirin	13

2.2.9	Simp	ie waze wan fonower	14
2.2.1	0 Real	Time Operating System (RTOS)	14
BAB 3 ME	TODOL	OGI PENELITIAN	17
3.1. Ali	ir Meto	de Penelitian	17
3.1.1	Studi	dan Pengkajian Literatur	18
3.1.2	Anali	sis Kebutuhan Sistem	18
3.1.3	Perar	ncangan Sistem	19
3.1.4	Imple	ementasi Sistem	19
3.1.5	Peng	ujian dan Analisis	19
3.1.6	Kesin	npulan	19
		KEBUTUHAN	
		n Umum Sistem	
4.1.1	Persp	pektif Sistem	21
4.1.2	Ruang	Lingkup	21
4.1.3	Asum	nsi dan Ketergantungan	22
4.2. Ke	butuha	in Fungsional	22
4.3. Ke	butuha	n Perangkat	24
4.2.1	Kebu	tuhan Perangkat Keras	24
		tuhan Perangkat Lunak	
		n Non-Fungsional	
BAB 5 PER	RANCAI	NGAN DAN IMPLEMENTASI	26
5.1 Pe		gan Sistem	
5.1.1	Gaml	baran Umum Sistem	26
5.1.2	Perar	ncangan <i>Prototype</i> Alat	27
5.1.3	Perar	ncangan Perangkat Keras	28
5.1.4	Perar	ncangan Perangkat Lunak	34
5.	1.4.1	Perancangan Pembacaan Sensor Ultrasonik HC-SR04	35
5.:	1.4.2	Perancangan Pembacaan Modul Sensor Photodioda	36
5.	1.4.3	Perancangan Menentukan Putaran Motor DC pada Modul Driver Motor	
5.:	1.4.4	Perancangan Notifikasi Buzzer	38
5.:	1.4.5	Perancangan Sistem Tidak Menggunakan Free RTOS	38
5.	1.4.6	Perancangan Sistem Menggunakan Free RTOS	39
5.2 Im	plemer	ntasi Sistem	43

5.2.1	Imple	ementasi <i>Prototype</i> Alat	43
5.2.2	Imple	ementasi Perangkat Keras	44
5.2.3	Imple	ementasi Perangkat Lunak	47
5.2	.3.1	Implementasi Pembacaan Sensor Ultrasonik HC-SR04	48
5.2	.3.2	Implementasi Pembacaan Modul Sensor Photodioda	49
5.2	.3.3	Implementasi Menentukan Putaran Motor DC pada L298N Driver Motor	
5.2	.3.4	Implementasi Notifikasi Buzzer	52
5.2	.3.5	Implementasi Sistem Tidak Menggunakan Free RTOS	52
5.2	.3.6	Implementasi Sistem Menggunakan Free RTOS	56
BAB 6 PEN	GUJIA	N DAN ANALISIS	69
6.1 Per	ngujiar	n Pembacaan Sensor Ultrasonik HC-SR04	69
6.1.1	Tujua	an Pengujian	69
6.1.2	Prose	edur Penguijan	69
		dan Analisis Pengujian	
6.2 Per		n Pembacaan Modul Sensor Photodioda	
6.2.1	Tujua	an Pengujian	71
6.2.2		edur Pengujian	
6.2.3	Hasil	dan Analisis Pengujian	71
	8N Dr	n Pemberian Nilai PWM dan arah Putaran Motor DC pada iver Motor	72
6.3.1	Tujua	an Pengujian	72
6.3.2	Prose	edur Pengujian	72
6.3.3	Hasil	dan Analisis Pengujian	73
6.4 Per	ngujiar	n Notifikasi Buzzer	73
6.4.1	Tujua	an Pengujian	73
6.4.2	Prose	edur Pengujian	74
6.4.3	Hasil	dan Analisis Pengujian	74
		n Arah Gerak Robot pada Sistem Robot <i>Maze Wall Followei</i> S dan menenggunakan <i>Free</i> RTOS	•
6.5.1	Tujua	an Pengujian	75
6.5.2	Prose	edur Pengujian	75
6.5.3		dan Analisis Pengujian	
6.6 Per	ngujiar	n Perbandingan Waktu <i>Tanpa</i> Free RTOS dan Dengan <i>Free</i>	e RTOS
			77

6.6.1	Tujuan Pengujian	78
6.6.2	Prosedur Pengujian	78
6.6.3	Hasil dan Analisis Pengujian	78
BAB 7 PEN	UTUP	. 80
7.1 Kesii	mpulan	. 80
7.2 Sara	n	. 81
DVETVB DI	Ιςτακα	82



DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino Nano	7
Tabel 2.2 Spesifikasi Motor DC MM10	9
Tabel 2.3 Spesifikasi Modul L298N <i>Driver</i> Motor DC	10
Tabel 2.4 Spesifikasi Sensor Ultrasonik HC-SR04	11
Tabel 5.1 Konfigurasi Pin Sensor HC-SR04 Kanan dengan Arduino Nano	29
Tabel 5.2 Konfigurasi Pin Sensor HC-SR04 Depan dengan Arduino Nano	29
Tabel 5.3 Konfigurasi Pin Sensor HC-SR04 Kiri dengan Arduino Nano	30
Tabel 5.4 Konfigurasi Pin Modul Sensor Photodioda dengan Arduino Nano	31
Tabel 5.5 Konfigurasi Pin Modul <i>Driver</i> Motor dengan Arduino Nano	31
Tabel 5.6 Konfigurasi Pin Buzzer dengan Arduino Nano	32
Tabel 5.7 Konfigurasi Pin Motor DC kanan dengan Modul L298N <i>Driver</i> Motor.	33
Tabel 5.8 Konfigurasi Pin Motor DC Kiri dengan Modul L298N <i>Driver</i> Motor	33
Tabel 5.9 Konfigurasi Baterai dengan Modul <i>Driver</i> Motor	
Tabel 5.10 Source Code Sensor Ultrasonik HC-SR04	
Tabel 5.11 Source Code Modul Sensor Photodioda	50
Tabel 5.12 Source Code Modul L298N Driver Motor	
Tabel 5.13 Source Code Buzzer	
Tabel 5.14 Source Code Sistem tanpa Free RTOS	52
Tabel 5.15 Potongan Program Inisialisasi Pin	57
Tabel 5.16 Potongan Program Pendefinisian <i>Task</i> RTOS	57
Tabel 5.17 Potongan Program Void Setup()	
Tabel 5.18 Potongan Program Pembuatan <i>Task</i> RTOS	58
Tabel 5.19 Potongan Program Proses Penghitungan Nilai Sensor Ultrasonik	59
Tabel 5.20 Potongan Program <i>Task</i> UltrasonikKiri	60
Tabel 5.21 Potongan Program <i>Task</i> UltrasonikDepan	60
Tabel 5.22 Potongan Program <i>Task</i> UltrasonikKanan	61
Tabel 5.23 Potongan Program <i>Task</i> Photodioda dan Buzzer	61
Tabel 5.24 Potongan Program <i>Task</i> Motor	62
Tabel 5.25 Source Code Sistem dengan Free RTOS	64
Tabel 6.1 Hasil Pengujian Sensor Ultrasonik HC-SR04	70
Tabel 6.2 Hasil Pengujian Sensor Photodioda	72
Tabel 6.3 Hasil Pengujian Modul L298N Driver Motor	73

Tabel 6.4	Hasil Pengujian Buzzer	74
Tabel 6.5	Hasil Pengujian Sistem tanpa Free RTOS dan dengan Free RTOS	76
Tabel 6.6	Hasil Penguijan Menemukan Goal	79



DAFTAR GAMBAR

Gambar 2.1 Arduino Nano	7
Gambar 2.2 PinOut Arduino Nano	8
Gambar 2.3 Motor DC MM10	9
Gambar 2.4 Modul L298N <i>Driver</i> Motor DC dan <i>PinOut</i>	LO
Gambar 2.5 Sensor Ultrasonik HC-SR04	LO
Gambar 2.6 Modul Sensor Photodioda	L1
Gambar 2.7 PinOut Modul sensor Photodioda	L1
Gambar 2.8 Buzzer	L2
Gambar 2.9 PWM (Pulse Width Modulation)	
Gambar 2.10 Labirin	L3
Gambar 2.11 Flowchart Wall Follower Kiri	L4
Gambar 2.12 Konsep Multitasking	
Gambar 2.13 Konsep Konkuren	15
Gambar 3.1 Diagram Alir Penelitian	L7
Gambar 5.1 Diagram Blok Sistem	26
Gambar 5.2 Desain Prototype Robot Tampak Atas	27
Gambar 5.3 Desain <i>Prototype</i> Robot Tampak Bawah	28
Gambar 5.4 Skematik Sensor HC-SR04 Kanan	28
Gambar 5.5 Skematik Sensor HC-SR04 Depan	29
Gambar 5.6 Skematik Sensor HC-SR04 Kiri	30
Gambar 5.7 Skematik Modul Sensor Photodioda	
Gambar 5.8 Skematik Modul L298N <i>Driver</i> Motor	31
Gambar 5.9 Skematik Buzzer	32
Gambar 5.10 Skematik Motor DC	32
Gambar 5.11 Skematik Baterai	33
Gambar 5.12 Ukuran Labirin	34
Gambar 5.13 Jarak Labirin dengan Robot	34
Gambar 5.14 Flowchart Sensor Ultrasonik HC-SR04	35
Gambar 5.15 Flowchart Sensor Photodioda	36
Gambar 5.16 Flowchart Modul L298N Motor	37
Gambar 5.17 Flowchart Buzzer	38
Gambar 5.18 Flowchart Sistem tanpa Free RTOS	39

Gambar 5.19	Flowchart Sistem dengan Free RTOS Poin A	40
Gambar 5.20	Flowchart Sistem dengan Free RTOS Poin B	41
Gambar 5.21	Flowchart Sistem dengan Free RTOS Poin C	42
Gambar 5.22	Implementasi <i>Prototype</i> Robot Tampak Atas	43
Gambar 5.23	Implementasi <i>Prototype</i> Robot Tampak Bawah	44
Gambar 5.24	Sensor Ultrasonik HC-SR04 Kanan dengan Arduino Nano	44
Gambar 5.25	Sensor Ultrasonik HC-SR04 Depan dengan Arduino Nano	45
Gambar 5.26	Sensor Ultrasonik HC-SR04 Kiri dengan Arduino Nano	45
Gambar 5.27	Modul Sensor Photodioda dengan Arduino Nano	45
Gambar 5.28	Modul L298N <i>Driver</i> Motor dengan Arduino Nano	46
Gambar 5.29	Motor DC Kanan dengan Modul L298N <i>Driver</i> Motor	46
	Motor DC Kiri dengan Modul L298N <i>Driver</i> Motor	
	Buzzer dengan Arduino Nano	
	Baterai 9V dengan Modul L298N <i>Driver</i> Motor	
Gambar 5.33	Labirin 5x6	47
Gambar 6.1 G	Grafik Pengujian dan Analisis	69
Gambar 6.2 L	abirin yang Digunakan Pengujian	77
Gambar 6.3 L	abirin yang Digunakan Pengujian Menemukan <i>Goal</i>	79

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Seiring dengan kemajuan jaman, tidak dapat dipungkiri teknologi semakin berkembang. Salah satu jenis teknologi yang semakin berkembang yaitu robot. Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu kecerdasan buatannya (Yultrisna & Sofiyan, 2013). Robot tidak hanya digunakan untuk mempermudah pekerjaan manusia. Di sisi lain, robot juga dapat digunakan sebagai perlombaan robot baik skala nasional maupun internasional. Salah satu divisi yang dipertandingkan di Kontes Robot Indonesia adalah Robot Pemadam Api. Misi dari robot yang berkategori pemadam api ini adalah berlomba secepat mungkin untuk menuju sebuah lilin yang terletak secara acak di dalam arena pertandingan yang menyerupai labirin atau *maze* kemudian memadamkan api tersebut (Sutrisna, 2014).

Maze adalah jaringan jalan yang rumit dan berliku-liku. sebuah puzzle dalam bentuk percabangan jalan yang kompleks dan memliki banyak jalan buntu menggunakan metode Simple Maze (Udani, Marhaeni, & Jampel, 2011). Jenis dari Simple Maze ada dua yaitu Line Follower dan Wall Follower. Wall Follower yaitu suatu algoritma untuk memberikan navigasi kepada robot dengan acuan dinding. Salah satu keuntungannya adalah tidak perlu adanya garis penuntun ataupun suatu tanda khusus sebagai arahan bagi robot. Cara kerjanya adalah dengan mengatur jarak dinding dengan robot tetap konstan. Bila terjadi perubahan, maka robot akan bergerak untuk kemudian menyesuaikan jarak lagi (Yultrisna & Sofiyan, 2013). Robot Maze Wall Follower merupakan robot yang digunakan sebagai media dalam penerapan metode Simple Maze.

Robot Maze Wall Follower mencari tujuan yang berada di dalam labirin dengan cara mengikuti dinding sebagai acuannya. Robot Maze Wall Follower menggunakan unit sistem pengendali yang bernama mikrokontroler, yang beroperasi dengan menjalankan program yang telah disimpan di dalamnya. Cara kerja mikrokontroler yaitu secara bergantian dalam menyelesaikan antrean tugas, yang satu diselesaikan terlebih dahulu kemudian tugas yang lainnya. Dalam hal ini mikrokontroler bekerja untuk menentukan pergerakan roda belakang pada robot dengan ketentuan kedua roda mempunyai kecepatan sama, lebih pelan kiri atau kanan setelah dilakukan pemrosesan dari data output sensor HC-SR04 dalam mendeteksi dinding. Kemudian untuk menghentikan jalan roda saat sensor Photodioda menemukan garis finish atau garis berwarna gelap.

Pada umumnya apabila tugas-tugas/ tasks yang harus dikerjakan oleh suatu perangkat elektronik jumlahnya banyak, solusi yang paling mudah dilakukan

adalah dengan cara menambah jumlah *chip* mikrokontroler yang digunakan. Tetapi menambah jumlah *chip* mikrokontroler berarti menaikkan biaya produksi dari perangkat elektronik yang akan dibuat dan konsumsi daya listrik yang diperlukan juga bertambah besar (Suntaya, 2016). Salah satu cara untuk menangani hal ini adalah dengan meningkatkan kinerja dari sebuah *chip* mikrokontroler itu sendiri dengan menggunakan RTOS (*Real Time Operating System*). RTOS adalah sebuah sistem operasi yang diperuntukkan pada perangkat sistem *embedded*. Dengan RTOS sebuah mikrokontroler bisa dioperasikan secara *multitasking* yang berarti melakukan banyak pekerjaan dalam satu waktu secara bersamaan. Konsep ini akan mengurangi kondisi di mana suatu keadaan mikrokontroler tidak melakukan aktivitas karena sedang menunggu suatu tugas (Suntaya, 2016)

Dari masalah di atas akan dibuat sebuah robot *maze* yang dapat menelusuri ruangan dengan acuan dinding kiri dengan menggunakan metode Simple Maze, kemudian setiap tugas yang dikerjakan dapat berlangsung secara bersamaan dengan menggunakan RTOS (Real Time Operating System) supaya sebuah chip mikrokontroler bisa memenuhi jangka waktu yang dibutuhkan dari tugas-tugas yang diberikan, dan diharapkan tidak akan terjadi kegagalan pada sistem. Maka diambil tema dengan judul "Implementasi Metode Simple Maze Wall Follower Dengan Menggunakan Free RTOS Pada Robot Maze".

1.2 Rumusan Masalah

Berdasarkan beberapa masalah yang telah diuraikan pada latar belakang, maka dapat dirumuskan permasalahannya sebagai berikut :

- 1. Bagaimana robot dapat mendeteksi dinding dan garis sebagai langkah awal untuk menerapkan metode Simple Maze?
- 2. Bagaimana mengimplementasikan metode Simple Maze untuk menemukan garis finish pada labirin?
- 3. Bagaimana mengimplementasikan sistem pembacaan sensor secara bersamaan dengan *Free* RTOS?
- 4. Bagaimana hasil perbandingan waktu dan efisiensi dalam menemukan *goal* untuk robot *maze* tanpa RTOS dan dengan RTOS?

1.3 Tujuan

Adapun untuk maksud dan tujuan dari studi kasus yang dibuat adalah sebagai berikut :

- 1. Untuk mendeteksi dinding dan garis sebagai langkah awal untuk menerapkan metode Simple Maze pada Robot Maze Wall Follower.
- 2. Untuk mengimplementasikan metode Simple Maze dalam menemukan garis *finish* pada labirin.

- 3. Untuk mengimplementasikan sistem pembacaan sensor secara bersamaan dengan *Free* RTOS.
- 4. Untuk mengetahui hasil perbandingan waktu dan efisiensi dalam menemukan *goal* untuk robot *maze* tanpa RTOS dan dengan RTOS.

1.4 Manfaat

Manfaat yang didapat dari studi kasus ini adalah sebagai berikut :

- 1. Dapat menemukan tujuan yang terletak di dalam labirin yang sangat rumit dengan menggunakan acuan dinding kiri.
- 2. Dapat membantu pemadam kebakaran dalam menemukan sumber kebakaran dalam suatu gedung yang memiliki banyak ruangan.
- 3. Dapat digunakan untuk robot pengangkut barang di dalam sebuah ruangan bila robot dibuat dalam model yang lebih besar.

1.5 Batasan Masalah

Batasan di sini menggunakan supaya permasalahan yang dibahas tidak terlalu luas dan lebih terfokus pada 1 titik. Pada studi kasus yang dibahas ini memiliki batasan maslah sebagai berikut :

- 1. Pada penelitian ini metode Simple Maze yang dibuat hanya untuk metode penelusuran mengikuti dinding sebelah kiri pada robot Maze Wall Follower.
- 2. Menggunakan dinding dan garis berwarna gelap sebagai acuan robot dalam menemukan *goal*.
- 3. Menggunakan RTOS untuk multitasking.

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini adalah sebagai berikut :

BAB 1 Pendahuluan

Bab ini berisi tentang latar belakang dilakukannya penelitian, rumusan masalah mengapa penelitian ini layak untuk dilakukan, tujuan dari penelitian, manfaat dari penelitian, batasan masalah pada penelitian yang dilakukan, dan sistematika penulisan untuk laporan dari penelitian yang dilakukan.

BAB 2 Landasan Kepustakaan

Bab ini membahas tinjauan pustaka dari penelitian sebelumnya yang memiliki kekurangan, sehingga perlu dikembangkan supaya lebih baik lagi dan dasar teori yang mendukung dalam pembuatan sistem ini.

BAB 3 Metode Penelitian

Bab ini membahas tentang langkah-langkah dalam melakukan penelitian, antara lain studi literatur, analisis kebutuhan yang akan digunakan pada sistem

sistem, perancangan sistem, implementasi sistem, pengujian dan analisis, dan kesimpulan.

BAB 4 Rekayasa Kebutuhan Sistem

Bab ini membahas tentang langkah-langkah dalam melakukan penelitian yang sedang dilakukan, antara lain gambaran umum sistem, kebutuhan fungsional dan kebutuhan non-fungsional.

BAB 5 Perancangan dan Implementasi Sistem

Bab ini membahas tentang perancangan perangkat keras dan perangkat lunak, serta implementasi sistem perangkat keras dan perangkat lunak terkait penelitian Implementasi Metode *Simple Maze Wall Follower* Dengan Menggunakan *Free* RTOS Pada Robot *Maze*.

BAB 6 Pengujian dan Analisis

Pada bab ini dibahas tentang pengujian yang dilakukan terhadap sistem untuk mengetahui apakah sistem sudah berjalan sesuai yang diinginkan yaitu meliputi tujuan pengujian, prosedur pengujian, dan hasil dan analisa pengujian sistem.

BAB 7 Penutup

Bab ini akan membahas tentang kesimpulan dari penelitian yang sudah dilakukan. Serta saran yang bertujuan untuk pengembangan sistem lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Dalam landasan kepustakaan ini membahas tentang kajian pustaka dan dasar teori yang diperlukan dalam penelitian. Landasan kepustakaan bertujuan untuk membahas tentang penelitian yang sudah ada dan berkaitan dengan topik yang dibahas, sedangkan dasar teori akan membahas tentang berbagai teori dan kebutuhan alat yang diperlukan dalam penelitian yang sedang dilakukan.

2.1 Tinjauan Pustaka

Di sini akan membahas tentang penelitian yang sudah ada dan berkaitan dengan studi kasus yang akan dibuat.

2.1.1 Robot Maze Solving Untuk Menyelesaikan Tiga Persimpangan

Penelitian ini membahas bagaimana cara untuk membuat robot *maze solving* untuk menyelesaikan tiga jalur. Cara kerja robot ini adalah robot melewati garis yang sudah dipersiapkan dengan adanya persimpangan robot bisa memilih jalur sesuai tujuan yang ditentukan. Robot *maze* Menggunakan mikrokontroler untuk mengatur kecepatan laju robot dan sebagai alat untuk mengendalikan sensor garis yang digunakan (Saputro & Haryanto, 2015).

Pada penelitian ini dikatakan bahwa robot *maze* akan mengikuti *line* yang ada sesuai dari hasil masukan dari sensor Photodioda yang digunakan. Supaya robot dapat bergerak maju maka kecepatan motor DC sebelah kanan dibuat sama dengan kecepatan motor sebelah kiri, agar robot dapat bergerak ke kiri maka kecepatan motor sebelah kanan dibuat lebih tinggi dengan kecepatan motor sebelah kiri agar robot dapat bergerak ke kanan maka arah putaran roda kiri ke depan dan arah putaran roda kanan ke belakang dengan kecepatan yang sama.

Pada penelitian tersebut memilih menggunakan robot *Maze Solving* karena sebelumnya sudah menggunakan *line follower* dan memiliki kekurangan yaitu hanya bisa mengikuti jalur lurus saja atau melengkung. Maka apabila ada pertigaan atau persimpangan maka robot belum tepat dalam menyelesaikannya.

2.1.2 Kontrol Motor DC Penggerak Roda Pada Robot Cerdas

Penelitian ini membahas bagaimana cara menggerakkan roda pada robot cerdas dengan menggunakan motor DC sebagai alat penggerak roda. Pengontrolan motor DC menggunakan proses yaitu dengan memasukkan data PWM pada mikrokontroler dapat dijalankan oleh motor DC sesuai data yang dimasukkan baik nila PWM rendah sampai nilai PWM tinggi dan mengubah arah putar motor DC, tinggi rendahnya nilai PWM mempengaruhi konsumsi tegangan yang dibutuhkan oleh motor DC untuk mengoptimalkan putaran motor,

sedangkan untuk arus adalah sama tergantung spesifikasi dari motor (Aribowo & Herwanto, 2011).

Dalam penelitian ini data yang akan diberikan ke mikrokontroler dapat ditampilkan pada LCD. Semua instruksi yang diberikan oleh mikrokontroler dan dilakukan tindakan oleh motor DC dengan nilai yang tetap atau berubah dari data yang diberikan seperti naik turunnya PWM dan arah putar motor dapat diketahui dari LCD yang telah diprogram.

2.1.3 Peningkatan Kinerja Perangkat Elektronik Berbasis Mikrokontroler AVR 8 Bit Dengan Menggunakan RTOS (*Real Time Operating System*)

Penelitian ini membahas tentang bagaimana cara melakukan tugas-tugas yang banyak tanpa menambah jumlah *chip* mikrokontroler yang akan digunakan. Karena menggunakan banyak *chip* akan membutuhkan lebih banyak biaya serta konsumsi listrik yang akan digunakan dalam membuat suatu perangkat cerdas (Sutaya, 2015).

Pada penelitian ini akan dilakukan pengujian *multitasking* yang dilakukan oleh mikrokontroler. Pengujian dilakukan secara berulang kali dengan cara menambah jumlah *multitasking* pada setia step yang dilakukan. Keuntungan dari penelitian ini adalah dengan menggunakan RTOS maka juga bisa mengetahui jumlah maksimum proses atau pekerjaan yang akan dilakukan oleh sebuah mikrokontroler.

2.1.4 Penerapan Metode Simple Maze Pada Robot Wall Follower Untuk Menyelesaikan Jalur Dalam Menelusuri Sebuah Labirin

Penelitian ini menerapkan algoritma Simple Maze Wall Follower pada robot Wall Follower dalam menemukan rute terpendek dalam maze atau labirin. Peneliti melakukan beberapa modifikasi dari penelitian sebelumnya, yaitu penerapan metode yang awalnya pada robot Line Follower kemudian diterapkan pada Wall Follower, yang tadinya hanya sensor photodioda menjadi sensor Ultrasonik dan yang tadinya menggunakan acuan garis menjadi ruangan sebagai acuannya.

Terdapat dua jenis cara mudah Simple Maze Wall Follower yaitu menelusuri bagian kanan atau kiri. Menelusuri bagian kanan yaitu robot menyimpan jalur dengan menelusuri dinding bagian kanan dengan sensor, sedangkan menelusuri bagian kiri yaitu robot menyimpan jalur dengan menelusuri dinding bagian kiri. Sensor yang digunakan adalah sensor Ultrasonik HC-SR04 (Shofa & Andrasto, 2015).

Dari *paper* yang dibahas di atas dapat disimpulkan bahwa penggunaan sensor Ultrasonik dan Photodioda yang dapat mengikuti acuan dinding bagian kanan atau kiri, serta dapat menemukan titik *finish* yang ditentukan. Akan tetapi dari *paper* di

atas belum ada yang menggunakan sensor Ultrasonik dan sensor Photodioda secara bersamaan. Dalam hal ini sensor Ultrasonik untuk menentukan jalur dan sensor photodioda dapat menemukan titik *finish* yang ditentukan.

2.2 Dasar Teori

2.2.1 Arduino Nano

Arduino Nano adalah mikrokontroler yang berbasis *chip* ATmega328P dengan bentuk yang sangat kecil. Secara fungsi tidak ada bedanya dengan Arduino Uno. Perbedaan utama terletak pada tidak adanya *jack power* DC dan penggunaan konektor Mini-B USB (Arduino, 2006). Untuk bentuk dari Arduino Nano dapat dilihat pada gambar 2.1.



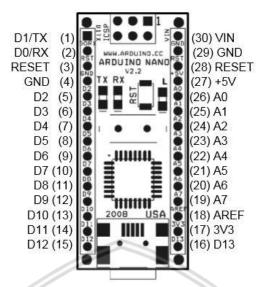
Gambar 2.1 Arduino Nano Sumber : (Arduino, 2006)

Arduino Nano memiliki spesifikasi yang hampir sama dengan Arduino Uno. Untuk spesifikasi dari Arduino Nano dapat dilihat pada tabel 2.1.

Tabel 2.1 Spesifikasi Arduino Nano

2.1 Spesifikasi Aldullo Nalio
ATmega328P
5V
7V - 12V
14 buah
6 buah
40 mA
32 KB, 0.5 KB telah digunakan untuk bootloader
2 KB
1 KB
16 Mhz
45 mm x 18 mm
5 g

Sumber: (Arduino, 2006)



Gambar 2.2 PinOut Arduino Nano

Sumber : (Arduino, 2006)

Beberapa pin power pada Arduino Nano:

- 1. GND adalah ground.
- 2. VIN adalah pin yan digunakan untuk masukan dari sumber tegangan.
- 3. Pin 5V adalah pin output di mana pada pin tersebut mengalir tegangan 5V.
- 4. REF. Ini adalah pin yang menyediakan referensi tegangan mikrokontroler.

Arduino Nano memiliki 14 buah digital pin yang dapat digunakan sebagai input dan output, dengan menggunakan fungsi pinMode(), digitalWrite(), dan digitalRead(). Pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus sebesar 20mA. Nilai maksimum adalah 40mA, yang sebisa mungkin dihindari supaya tidak terjadi kerusakan chip mikrokontroler (Arduino, 2006).

Beberapa pin memiliki fungsi khusus:

- 1. Serial, terdiri dari 2 pin yaitu pin 0 (RX) dan pin 1 (TX) yang digunakan untuk menerima (RX) dan mengirim (TX) data serial.
- 2. External Interrup, yaitu pin 2 dan pin 3 yang dapat digunakan untuk interupsi.
- 3. PWM yaitu pin 3, 5, 6, 9, 10, dan 11 menyediakan *output* PWM 8-bit dengan menggunakan fungsi *analogWrite()*.
- 4. SPI yaitu pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) mendukung komunikasi SPI dengan menggunakan SPI *library*.

2.2.2 Motor DC MM10

Motor Listrik DC adalah perangkat yang mengubah energi listrik menjadi energi kinetik atau gerakan. Motor DC memiliki dua terminal dan memerlukan tegangan

arus searah (*Direct Current*) untuk dapat menggerakkannya. Untuk bentuk roda dan motor DC yang digunakan dapat dilihat pada gambar 2.4.



Gambar 2.3 Motor DC MM10

Sumber: (Multicomp, 2012)

Motor DC banyak digunakan sebagai penggerak dalam berbagai peralatan, baik kecil maupun besar yang berkecepatan lambt maaupun cepat. Motor DC juga banyak dipakai karena dapat disesuaikan secara ideal menerima pulsa digital untuk kendali kecepatan. Cara pengendalian motor DC ini bisa secara PWM (*Pulse Width Modulation*). Pemilihan cara pengendalian akan tergantung dari kebutuhan terhadap gerakan motor DC itu sendiri (Ariwibowo & Herwanto, 2011). Dalam hal ini membutuhkan motor DC untuk menggerakkan roda suatu robot *maze*. Untuk spesifikasi dari Motor DC beserta Roda dapat dilihat pada tabel 2.2.

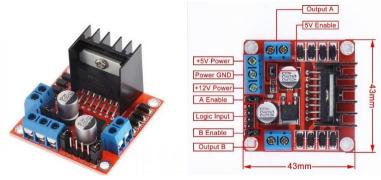
Tabel 2.2 Spesifikasi Motor DC MM10

Sumber Tegangan	3V - 12V (rekomendasi 9V)	
Rasio <i>gearbox</i>	1:48	
Berat	80gr	
Ukuran motor dan <i>gearbox</i>	70x22x18mm³	
Ukuran roda dengan ban	diameter 5,4 cm	

Sumber: (Multicomp, 2012)

2.2.3 Modul L298N Driver Motor DC

Modul L298N *Driver* Motor adalah jenis IC yang dapat mengendalikan arah putaran dan kecepatan motor DC. Mampu mengeluarkan *output* tegangan untuk Motor DC hingga 50 V. IC L298N terdiri dari transistor-transistor logis (TTL) dengan gerbang *Nand* yang memudahkan dalam menentukan arah putaran suatu motor DC dan motor *stepper*. Pada L298N menggunakan *dual H-Bridge* sehingga dapat digunakan untuk menggerakkan dua motor DC secara bersamaan sekaligus (Boxall, 2014) Untuk bentuk dari Modul L298N *Driver* Motor DC beserta pin nya dapat dilihat pada gambar 2.5.



Gambar 2.4 Modul L298N Driver Motor DC dan PinOut

Sumber: (Boxall, 2014)

Untuk pemasangan Modul L298N *Driver* Motor ini dibutuhkan 6 buah pin mikrokontroler. 2 buah untuk pin *Enable* yang masing-masing satu untuk tiap motor DC. 4 buah untuk mengatur kecepatan motor DC tersebut. Spesifikasi dari Modul L298N *Driver* Motor dapat dilihat pada tabel 2.3.

Tabel 2.3 Spesifikasi Modul L298N Driver Motor DC

Tipe	Dual H-Bridge
chip	ST L298N
Tegangan logika	5V DC
Tegangan	5-35V DC
Arus Logika	0mA-36mA
Arus Putaran	2A (maksimal 1 motor)
Temperatur	-20 C – 135 C
Power maksimal	25W
Berat	30g
Ukuran	43 x 43 x 27mm

Sumber: (Boxall, 2014)

2.2.4 Sensor Ultrasonik HC-SR04

HC-SR04 adalah Sensor Ultrasonik yang memiliki dua elemen, yaitu elemen Pendeteksi gelombang Ultrasonik, dan juga elemen Pembangkit gelombang Ultrasonik. Ultrasonik adalah sensor yang dapat mendeteksi gelombang suara yang memiliki frekuensi Ultrasonik atau frekuensi di atas kisaran frekuensi pendengaran manusia (Ozdisan, 2018). Untuk gambar dari sensor ultrasonik HC-SR04 dapat dilihat pada gambar 2.6.



Gambar 2.5 Sensor Ultrasonik HC-SR04

Sumber: (Jones, 2018)

Dalam penelitian yang dilakukan sensor Ultrasonik yang digunakan sebanyak 3 unit untuk mendeteksi dinding bagian kanan, kiri, dan depan dari robot *maze*. Untuk spesifikasi dari sensor Ultrasonik HC-SRO4 dapat dilihat pada tabel 2.4.

Tabel 2.4 Spesifikasi Sensor Ultrasonik HC-SR04

Jangkauan deteksi secara teori	2cm sampai 400 -500cm
Jangkauan deteksi secara praktik	2cm sampai 80cm
Jangkauan sudut deteksi	15 derajat
Sumber tegangan	5V DC
Frekuensi Ultrasonik	40 Hz
Dapat dihubungkan langsung ke kaki	Bisa
mikrokontroler	

Sumber: (Ozdisan, 2018)

2.2.5 Modul Sensor Photodioda FC-51

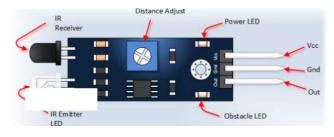
Photodioda adalah sensor yang akan mengubah perubahan intensitas cahaya yang diterima menjadi perubahan konduktans pada terminal sensor tersebut. Photodioda merupakan sensor cahaya yang akan mengalirkan arus listrik satu arah saja di mana akan mengalirkan arus listrik dari kaki anoda ke kaki katoda pada saat menerima intensitas cahaya. Photodioda sering digunakan pada aplikasi sensor pembaca garis pada robot *line follower* atau *line tracer* (Anjaswati, 2013). Untuk gambar dari sensor photodioda dapat dilihat pada gambar 2.7.



Gambar 2.6 Modul Sensor Photodioda

Sumber: (BoxElectronica, 2018)

Photodioda ini dapat dikonfigurasikan untuk memberikan logika *HIGH* atau *LOW* tergantung dari konfigurasi rangkaian yang digunakan. Untuk *pinout* dari modul sensor photodioda dapat dilihat pada gambar 2.8.



Gambar 2.7 PinOut Modul sensor Photodioda

Sumber: (BoxElectronica, 2018)

Keterangan:

- 1. LED inframerah berfungsi sebagai pemancar.
- 2. LED photodioda sebagai penerima.
- 3. IC komparasi berfungsi sebagai pembanding tegangan *input* dengan tegangan dari photodioda.
- 4. LED Power sebagai indikator adanya tegangan *input* yang masuk ke dalam modul.
- 5. Trimpot digunakan untuk mengatur jarak antara LED dengan objek.
- LED status sensor sebagai indikator saat pengukuran sesuai dengan pengaturan nilai trimpot maka menyala, sedangkan kondisi normal akan dalam kondisi mati.
- 7. Untuk kaki pin dari modul sensor yaitu antara lain GND, VCC, dan Digital output.

2.2.6 Buzzer

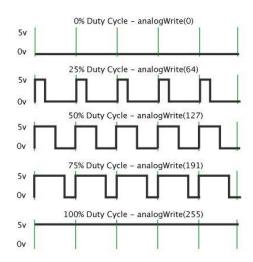
Buzzer adalah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Buzzer terdiri dari kumparan yang terpasang pada diafragma dan kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer Biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm) (Sulistyowati & Febriantoro, 2012). Untuk gambar dari buzzer yang digunakan dapat dilihat pada gambar 2.9.



Gambar 2.8 Buzzer Sumber: (Kingstate, 2011)

2.2.7 PWM (Pulse Width Modulation)

Pulse Width Modulation (PWM) adalah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam satu periode, untuk mendapatkan tegangan ratarata yang berbeda. Beberapa contoh aplikasi PWM adalah modulasi data untuk telekomunikasi, pengontrolan daya atau tegangan yang masuk ke beban, regulator tegangan, efek audio dan penguatan, serta aplikasi-aplikasi lainnya (Arduino, 2018). Untuk cara kerja dari PWM dapat dilihat pada gambar 2.10.



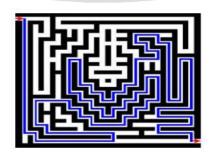
Gambar 2.9 PWM (Pulse Width Modulation)

Sumber: (Arduino, 2018)

Aplikasi PWM berbasis mikrokontroler berupa pengendalian kecepatan motor DC, pengendalian motor *servo*, dan pengaturan nyala terang LED. Oleh karena itu diperlukan pemahaman terhadap konsep PWM itu sendiri. PWM yang membuat motor DC dapat mempercepat putaran dan memperlambat putarannya. Nilai PWM dipengaruhi oleh nilai *duty cycle*.

2.2.8 Labirin

Labirin adalah jaringan jalan yang ruwet dan berliku-liku. Kita diharuskan menemukan jalan mana yang harus dilalui untuk sampai pada jalan keluar. Labirin dalam dunia robotika disebut dengan *maze* (Pratama, 2007). *Wall maze* merupakan jalur rumit dengan bentuk ruangan yang mempunyai dinding tanpa adanya atap, sedangkan *Line maze* merupakan jalur rumit berbentuk garis. *Wall maze* atau labirin merupakan tempat yang penuh dengan jalan dan lorong yang berliku-liku dan simpang siur atau sesuatu yang sangat rumit dan berbeli-belit. Untuk contoh bentuk labirin dapat dilihat pada gambar 2.11.



Gambar 2.10 Labirin

Sumber: (Pratama, 2007)

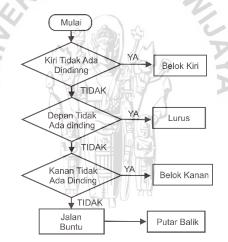
Dalam hal keperluan untuk robot *maze* dapat digambarkan bahwa labirin adalah sebuah gedung yang memiliki banyak jalur, persimpangan dan ruangan.

Dalam kehidupan, labirin dapat ditemukan pada susunan jalan kecil ,gang-gang di kawasan perumahan, serta tempat-tempat pariwisata.

2.2.9 Simple Maze Wall Follower

Simple Maze merupakan metode yang digunakan untuk menemukan tujuan yang berada dalam labirin jalan keluar dari labirin. "Ada dua jenis algoritma yang digunakan yaitu mengikuti dinding kiri atau kanan" (Saman & Abdramane, 2013).

Simple maze mempunyai 2 algoritma, yaitu line maze follower atau wall maze follower. Untuk line maze follower terdapat 2 pilihan yaitu mengikuti garis sebelah kiri atau garis sebelah kanan. Untuk Wall Maze Follower juga terdapat 2 pilihan yaitu mengikuti dinding sebelah kiri atau kanan. Simple Wall maze follower Jika mengikuti dinding sebelah kiri maka robot akan memprioritaskan pengecekan terhadap dinding sebelah kiri tanpa memedulikan di bagian depan dan kanan ada dinding atau tidak. Dan jika robot memilih wall follower kanan maka robot akan memprioritaskan dinding sebelah kanan tanpa memedulikan di bagian depan dan kiri ada dinding atau tidak.



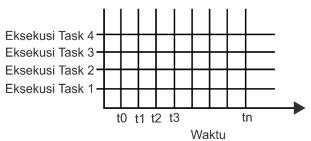
Gambar 2.11 Flowchart Wall Follower Kiri

Gambar 2.12 adalah Wall follower kiri, memprioritaskan dinding kiri ketika akan melakukan belok kiri, jalan lurus, belok kanan, dan putar balik.

2.2.10 Real Time Operating System (RTOS)

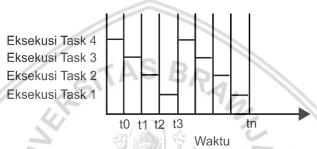
Real Time Operating System (RTOS) adalah sistem operasi yang dikembangkan untuk aplikasi dengan waktu nyata yang berasa di sekitar prosesor atau pengontrol. RTOS Memungkinkan prioritas untuk diubah secara instan dan data yang akan diproses cukup cepat sehingga hasilnya dapat digunakan sebagai tanggapan terhadap proses lain yang terjadi pada saat yang sama. RTOS memiliki kemampuan untuk segera merespons dengan cara yang telah ditentukan dan dapat diprediksi terhadap peristiwa eksternal (Chandane, 2016). RTOS digunakan pada sistem karena dapat mempercepat dari sensing jarak HC-SR04, pembacaan

sensor photodioda, dan output sistem dua Motor DC. Salah satu RTOS yang dapat digunakan secara gratis dan mempunyai contoh pada *library* Arduino IDE adalah *Free* RTOS.



Gambar 2.12 Konsep Multitasking

Sumber: (Jatmiko, et al., 2015)



Gambar 2.13 Konsep Konkuren

Sumber: (Jatmiko, et al., 2015)

Pada Gambar 2.13 menunjukkan bagaimana konsep cara kerja sistem secara multitasking. Di gambar terdapat empat task yang akan berjalan dalam waktu dan eksekusi yang sama. Akan tetapi pada memori konvensional konsep multitasking belum dapat dijalankan karena keterbatasan sumber daya prosesor dan memori. Namun task yang ada secara bergantian akan dijalankan dengan konkurensi. Konsep konkurensi dapat dilihat pada Gambar 2.14. Sistem operasi sudah mengetahui task mana saja yang akan dijalankan dalam rentan waktu tertentu. Selanjutnya sistem operasi membagi rentan waktu yang ada untuk mengeksekusi beberapa task dan mengatur task pada memori untuk menjalankan suatu task. Prosesor mengambil state terakhir dari task tersebut kemudian melanjutkan eksekusi task tersebut. Proses perpindahan antara eksekusi task dilakukan dengan cepat dan konkuren sehingga seolah - olah terlihat prosesor melakukan beberapa task dalam rentan waktu bersamaan. Embedded sistem dapat menggunakan RTOS untuk menjalankan beberapa task secara konkuren. Pada akhirnya sistem tertanam seolah-olah dapat bekerja secara *multitasking*. Alah satu jenis dari RTOS adalah Free RTOS.

Free RTOS (Free Real Time Operating System) digunakan pada embeded devices atau mikrokontroler untuk kebutuhan sistem operasinya. Free RTOS didistribusikan secara gratis lewat GPL dengan beberapa pengecualian untuk kalangan industri tertentu. Keunggulan dari Free RTOS adalah.

- 1. Rutinitas dalam *Free* RTOS sangat sederhana dan memiliki *stack* untuk *task* ringan dan terbatas.
- 2. Kecil dan sederhana. Hanya 3 file Sumber kode. Konsumsi ROM dan RAM yang minimal. *Binary* kernel RTOS tipikalnya akan menghabiskan area sebesar 4 9 KB.
- 3. Gratis untuk digunakan pada projek komersial.
- 4. Terdapat contoh yang bisa langsung dijalankan.



BAB 3 METODOLOGI PENELITIAN

3.1. Alir Metode Penelitian

Alir metodologi penelitian yang dilakukan secara umum dapat dilihat dari diagram alir pada gambar berikut :



Gambar 3.1 Diagram Alir Penelitian

Berdasarkan diagram alir pada gambar 3.1 dapat dilihat semua proses dilakukan secara berurutan dimulai dari studi dan pengkajian literatur sebagai pre-riset sampai dengan kesimpulan. Dari beberapa tahapan di atas ada satu tahap yang memiliki syarat untuk melanjutkan ke tahap selanjutnya. Syarat ini memiliki pengertian apabila pada suatu tahapan tidak memiliki hasil atau *output* seperti yang diharapkan, maka tahapan tersebut akan diulang. Ketika tahapan yang dijalankan sesuai dengan apa yang diinginkan oleh pengguna, maka akan dilanjutkan ke tahapan selanjutnya. Hal ini terlihat pada tahapan implementasi sistem, ketika sesuai akan dilanjutkan ke tahap selanjutnya yaitu tahap analisis, ketika hasil tidak sesuai maka tahapan implementasi sistem akan diulang sampai memiliki hasil seperti yang diinginkan.

3.1.1 Studi dan Pengkajian Literatur

Studi literatur perlu dilakukan untuk menunjang tahap perancangan dan implementasi pada penelitian ini. Literatur di sini digunakan untuk teori penguat dan juga sebagai landasan dalam penelitian. Beberapa literatur yang digunakan dapat berupa sebagai berikut.

- 1. Teori mengenai robot maze.
- 2. Mikrokontroler Arduino Nano, Motor DC, PWM (*Pulse Width Modulation*), Sensor Ultrasonik HC-SR04, Sensor Photodioda, Buzzer, Modul L298N *Driver* Motor *dual H-bridge*.
- 3. Penerapan metode Simple Maze.
- 4. Menggunakan RTOS untuk mulltitasking sistem.

3.1.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem digunakan untuk menganalisis apa saja kebutuhan dari sebuah sistem yang dikerjakan dalam penelitian ini. Analisis ini juga bertujuan agar dapat mengidentifikasi kebutuhan sistem. kebutuhan tersebut dapat berupa kebutuhan perangkat keras dan kebutuhan perangkat lunak.

Perangkat keras yang dibutuhkan dalam membangun sistem ini mempunyai spesifikasi sebagai berikut.

- 1. Perangkat keras yang dapat diunggah suatu program untuk membaca nilai sensor yang masuk kemudian mengirimkan data tersebut secara serial ke komputer untuk dilakukan proses perhitungan dengan metode Simple Maze.
- 2. Perangkat keras yang dapat membaca dan menjadi masukan Mikrokontroler yang digunakan.
- Perangkat keras yang dapat diunggah suatu program untuk membaca nilai sensor yang masuk kemudian dilakukan perhitungan PWM (*Pulse Width Modulation*) dan dilakukan tindakan oleh motor DC yang berperan sebagai penggerak roda belakang.

Sedangkan perangkat lunak yang dibutuhkan dalam membangun sistem ini mempunyai spesifikasi sebagai berikut.

- 1. Perangkat lunak yang dapat digunakan untuk membuat program dan dapat diunggah ke Mikrokontroler yang digunakan.
- 2. Perangkat lunak yang dapat menyimpan data Base berupa program yang kan ditentukan.
- 3. Perangkat lunak yang dapat membuat sistem melakukan *multitasking* secara bersamaan.

3.1.3 Perancangan Sistem

Perancangan sistem ini bertujuan untuk merancang sistem dalam penelitian ini berjalan secara terstruktur. Diawali dengan merancang sensor Ultrasonik dan sensor Photodioda untuk mendeteksi objek yang sesuai dan akurat. Kemudian membuat rancangan Modul L298N *Driver* motor DC *dual H-bridge* untuk menggerakkan roda dari robot dan kecepatan dapat berubah sesuai kondisi yang terjadi. Kemudian sebagai notifikasi robot berhenti berbelok atau berjalan menggunakan buzzer. Dan untuk melakukan tugas dari ke empat sensor dan Modul L298N *Driver* Motor DC secara bersamaan digunakan *Free* RTOS. Terakhir dilakukan penggabungan dari rancangan tersebut menjadi satu buah robot dan dilakukan pengujian dari sistem yang akan dijabarkan lebih lanjut pada bab 5.

3.1.4 Implementasi Sistem

Tahap ini diawali dengan pembuatan alat yang digunakan untuk menentukan tindakan robot *maze* oleh ketiga sensor Ultrasonik HC-SR04 dan satu sensor Photodioda, kemudian nilai dari sensor akan dimasukkan ke dalam perhitungan dengan menggunakan metode Simple Maze dan Mikrokontroler akan mengambil hasil yang akan dijadikan *output*. Pada tahap implementasi sistem perangkat keras serta perangkat lunak akan diatur sedemikian rupa agar sesuai dengan tahap perancangan sistem yang telah dibuat. Perangkat keras dan perangkat lunak harus bergabung dan berjalan beriringan agar dapat membentuk sebuah robot *maze* yang mengikuti dinding sebelah kiri dalam menemukan tujuannya.

3.1.5 Pengujian dan Analisis

Pada tahap ini dilakukan pengujian dan analisis. Pengujian dilakukan sebanyak 7 kali untuk masing-masing sensor Ultrasonik HC-SR04, 6 kali pengujian pada sensor Photodioda, 2 kali pengujian pada buzzer, 5 kali pengujian pada Modul L298N *Driver* Motor DC. Hal ini dilakukan agar terdapat hubungan yang baik dan benar pada setiap komponen yang terpasang.

3.1.6 Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan mulai dari perancangan, implementasi, dan pengujian serta analisis sistem telah selesai dilakukan. Kesimpulan ini diambil dari hasil pengujian dan analisis terhadap sistem yang dibuat. Tahap kesimpulan perlu dilakukan guna untuk mengetahui tingkat akurasi dan ketepatan dalam pembacaan oleh ke tiga ultrasonik HC-SR04 dan satu sensor photodioda tertanam pada alat ini secara keseluruhan.

Selain berguna untuk mendapatkan kesimpulan, tahap ini juga berisi saran yang digunakan untuk pengembangan dan penelitian lebih lanjut serta memperbaiki

kesalahan sistem yang sudah ada. Saran juga berguna untuk menambahkan ide untuk pengembangan sistem pada penelitian selanjutnya, sehingga penelitian ini dapat dilanjutkan dengan baik dan dapat berguna dalam membantu pekerjaan manusia atau dapat diikutkan dalam kontes robot dalam kategori pemadam kebakaran atau dalam menemukan jalur terdekat antara titik mulai dan tujuan setelah menemukan *goal* .



BAB 4 REKAYASA KEBUTUHAN

Bab 4 Rekayasa Kebutuhan ini akan dijelaskan mengenai semua kebutuhan dari sistem yang dibangun agar bisa memenuhi tujuan yang diinginkan. Kebutuhan tersebut meliputi gambaran umum sistem, kebutuhan fungsional, kebutuhan perangkat dan kebutuhan non fungsional. Dengan adanya rekayasa kebutuhan diharapkan sistem yang akan dibuat dapat bekerja dengan baik.

4.1 Gambaran Umum Sistem

Sistem yang akan dibuat adalah robot *maze* yang dapat menyelesaikan suatu labirin yang sudah ditentukan. Dalam sistem ini menggunakan tiga sensor Ultrasonik HC-SR04 dan satu sensor photodioda. Penempatan sensor ultrasonik yaitu, satu di bagian depan robot, satu di bagian samping kanan robot, dan satu di bagian kiri robot. Sedangkan sensor photodioda diletakkan di bagian depan di bawah sensor ultrasonik dan sensor ini menghadap ke bawah. Dalam hal ini sensor ultrasonik bertugas untuk mendeteksi dinding pada labirin sedangkan sensor photodioda bertugas untuk menemukan garis *finish*.

Masukan dari sistem ini adalah *output* dari sensor Ultrasonik yaitu dinding dan dari sensor photodioda berupa garis. Kemudian data akan diolah dalam Arduino Nano dengan ketentuan program yang telah dibuat. Sedangkan Keluaran dari sistem ini adalah kecepatan berputar roda akan berubah seiring dengan hasil pengolahan data. Apabila robot berjalan lurus maka kecepatan kedua roda sama, jika robot akan belok kiri maka kecepatan roda bagian kiri dan kanan sama tetapi arah putaran roda kiri ke belakang dan roda kanan ke depan, kemudian jika robot akan belok kanan maka kecepatan roda kanan dan kiri sama tetapi arah putaran roda kiri ke depan dan roda kanan ke belakang, untuk putar balik roda kiri berputar ke depan dan roda kanan ke belakang dengan kecepatan yang sama dan jika robot akan berhenti maka roda berhenti berputar.

4.1.1 Perspektif Sistem

Sistem ini dikatakan berjalan dengan seharusnya apabila sesuai dengan yang diharapkan yaitu sistem mampu membaca dinding labirin dan garis yang berbentuk persegi berwarna gelap yang ada dan melakukan tindakan yang telah ditentukan oleh program.

4.1.2 Ruang Lingkup

Ruang lingkup pada sistem ini yaitu robot *maze* untuk menemukan lokasi *finish* pada suatu labirin dengan mengikuti dinding bagian sebelah kiri. Dalam sistem ini menggunakan 4 sensor yang terdiri dari 3 sensor Ultrasonik yang berguna mendeteksi dinding yang ada pada labirin dan 1 sensor photodioda yang berguna

untuk mendeteksi garis warna gelap atau terang. Ketika robot telah menemukan garis *finish* maka buzzer akan berbunyi sebagai notifikasi bahwa robot telah berhasil menemukan tujuannya.

4.1.3 Asumsi dan Ketergantungan

Asumsi dan ketergantungan yang terdapat pada sistem ini antara lain:

- 1. Data sensor Ultrasonik sebelah kanan dari robot dapat dibaca dengan benar apabila letaknya minimal 3,5 cm dari objek atau dinding labirin.
- 2. Data sensor Ultrasonik sebelah depan dari robot dapat dibaca dengan benar apabila letaknya minimal 2 cm dari objek atau dinding labirin.
- 3. Data sensor Ultrasonik sebelah kiri dari robot dapat dibaca dengan benar apabila letaknya minimal 3,5 cm dari objek atau dinding labirin.
- 4. Data sensor photodioda dapat dibaca dengan benar apabila letaknya 3 cm di atas alas labirin dan objek berada tepat di tengah robot atau tepat di bawah dari sensor.
- Sumber tegangan untuk Modul L298N Driver Motor harus 9V, supaya pergerakan dari kedua Motor DC sesuai PWM yang dimasukkan dalam program.
- 6. Ukuran lebar labirin yaitu 33 cm supaya sensor dapat membaca dengan akurat dan robot bisa tetap berada di tengah labirin.
- 7. Sebagai prioritas utama Free RTOS adalah tiga sensor Ultrasonik dan satu sensor Photodioda yang memiliki tingkatan yang sama supaya saat memberikan input sistem cepat. Serta Modul L298N Driver Motor juga mempunyai tingkatan yang sama supaya cepat dalam menentukan output sistem yaitu pergerakan dari kedua Motor DC.

4.2. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang harus dipenuhi dalam sebuah sistem. Supaya sistem dapat berfungsi dan bekerja dengan baik sesuai yang diinginkan. Berikut beberapa kebutuhan fungsional dari sistem.

1. Sensor Ultrasonik HC-SR04 bagian kanan robot dapat mendeteksi dinding labirin dengan akurasi 3,5 cm.

Fungsi ini berguna untuk mendapatkan data *input* dari sensor HC-SR04 yang berupa letak dinding dari robot bagian sebelah kanan. Kemudian akan dilakukan tindakan sesuai program yang sudah ditentukan, yaitu robot akan berhenti, belok kanan, belok kiri, putar balik, dan jalan lurus.

2. Sensor Ultrasonik HC-SR04 bagian depan robot dapat mendeteksi dinding labirin dengan akurasi 2 cm.

Fungsi ini berguna untuk mendapatkan data *input* dari sensor HC-SR04 yang berupa letak dinding dari robot bagian sebelah depan. Kemudian akan dilakukan tindakan sesuai program yang sudah ditentukan, yaitu robot akan berhenti, belok kanan, belok kiri, putar balik, dan jalan lurus.

3. Sensor Ultrasonik HC-SR04 bagian kiri robot dapat mendeteksi dinding labirin dengan akurasi 3,5 cm.

Fungsi ini berguna untuk mendapatkan data *input* dari sensor HC-SR04 yang berupa letak dinding dari robot bagian sebelah kiri. Kemudian akan dilakukan tindakan sesuai program yang sudah ditentukan, yaitu robot akan berhenti, belok kanan, belok kiri, putar balik, dan jalan lurus.

4. Sensor Photodioda dapat mengenali garis gelap yang diletakkan di alas/ dasar labirin dengan jarak 3 cm dari sensor.

Fungsi ini berguna untuk menentukan garis *finish* dari labirin. Di mana robot akan berhenti ketika sensor Photodioda mendeteksi warna gelap di alas/ dasar pada labirin.

5. Motor DC sebelah kanan berputar sesuai kondisi yang ada.

Fungsi ini berguna untuk menentukan arah bergerak dari robot *maze*. Ketika data yang masuk dari sensor HC-SR04 dan Photodioda sudah masuk maka akan dilakukan tindakan atau *output* oleh motor DC sebelah kanan. Untuk jalan lurus maka kecepatan Motor DC kanan dengan kecepatan motor DC kiri sama dan arah putaran roda ke depan, untuk berbelok kanan maka motor DC sebelah kanan berputar ke arah belakang dengan jeda 0,8 detik, untuk berbelok kiri maka motor DC sebelah kanan berputar ke arah depan dengan jeda 0,8 detik, untuk putar balik motor DC kanan berputar ke arah depan dengan jeda 1,2 detik, dan untuk robot berhenti maka motor DC sebelah kanan tidak berputar.

6. Motor DC sebelah kiri dapat berputar sesuai kondisi yang ada.

Fungsi ini berguna untuk menentukan arah bergerak dari robot *maze*. Ketika data yang masuk dari sensor HC-SR04 dan photodioda sudah masuk maka akan dilakukan tindakan atau *output* oleh motor DC sebelah kiri. Untuk jalan lurus maka kecepatan Motor DC kiri dengan kecepatan motor DC kanan sama dan arah putaran roda ke depan, untuk berbelok kiri maka motor DC sebelah kiri berputar ke arah belakang dengan jeda 0,8 detik, untuk berbelok kanan maka motor DC sebelah kiri berputar ke arah depan dengan jeda 0,8 detik, untuk putar balik motor DC kiri berputar ke arah belakang dengan jeda 1,2 detik, dan untuk robot berhenti maka motor DC sebelah kiri tidak berputar.

7. Buzzer dapat berbunyi sesuai kondisi yang ada.

Fungsi ini berguna untuk membunyikan suara dari buzzer sebagai notifikasi. Ketika robot dalam mendeteksi warna terang dan tetap berjalan maka buzzer diam. Sedangkan Ketika robot dalam mendeteksi warna gelap dan berhenti maka buzzer berbunyi.

8. Multitasking Free RTOS

Pada fungsi ini sistem mengolah penjadwalan untuk tiap *task* yang diperuntukkan untuk masing-masing sensor Ultrasonik dan sensor *photodioda* yang digunakan. Satu sensor memiliki satu *task* untuk mengatur proses penjadwalan kapan sensor tersebut diaktifkan dan juga memproses data yang

diberikan oleh sensor sehingga bisa ditampilkan ke serial monitor Arduino IDE. Setiap sensor memiliki prioritas yang sama. Untuk sensor Ultrasonik berupa angka nilai jarak yang terdeteksi oleh sensor dengan satuan centimeter. Dan untuk sensor photodioda berupa keterangan sudah menemukan garis finish atau belum.

4.3. Kebutuhan Perangkat

Kebutuhan perangkat di sini meliputi perangkat keras dan lunak. Kebutuhan perangkat keras yang dibutuhkan harus sesuai dengan sistem dan memiliki kemampuan dan memenuhi kriteria dari sistem. Dan perangkat lunak yang digunakan harus memenuhi kriteria sistem, yaitu untuk membuat source code dan bisa digunakan secara multitasking.

4.2.1 Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan ada 6. Untuk perangkat keras yang digunakan pada sistem ini adalah sebagai berikut.

1. Sensor Ultrasonik HC-SR04

Sensor Ultrasonik HC-SR04 adalah dapat membaca jarak kurang lebih 2 cm hingga 80cm. Tegangan pada sensor ini adalah 5 V. Cara kerja sensor ini adalah dengan memancarkan gelombang suara Ultrasonik sesaat dan kemudian akan menghasilkan *output* berupa data yang sesuai dengan waktu pantulan dari gelombang suara Ultrasonik yang dipancarkan sesaat kemudian kembali menuju sensor. Dalam sistem ini HC-SR04 digunakan untuk mendeteksi dinding labirin.

2. Sensor Photodioda

Sensor photodioda adalah sensor yang peka terhadap cahaya, sensor photodioda akan mengalami perubahan resistansi pada saat menerima intensitas cahaya dan akan mengalirkan arus listrik secara *forward*. Dalam sistem ini sensor photodioda digunakan untuk mendeteksi garis *finish*. Garis *finish* yang dimaksud adalah ditandai dengan warna yang gelap. Jarak sensor dengan garis adalah 2,5 cm, dan ukuran garis adalah sama dengan lebar dari labirin yaitu 38 cm.

3. Arduino Nano

Arduino Nano adalah mikrokontroler yang berbasis *chip* ATmega328P dengan bentuk yang kecil. Secara fungsi tidak ada bedanya dengan Arduino Uno, namun pada Arduino Nano tidak ada *jack power* DC dan penggunaan konektor Mini-B USB.

4. Motor DC

Motor DC adalah suatu perangkat yang mengubah energi listrik menjadi energi kinetik atau gerakan. Dalam sistem ini Motor DC digunakan untuk menggerakkan roda dari robot. Motor DC yang digunakan sebanyak 2 unit.

5. Modul L298 Driver Motor

Modul L298 *Driver* Motor adalah jenis IC *driver* motor yang dapat digunakan untuk mengendalikan arah putaran dan kecepatan kedua motor DC dengan mengatur nilai PWM. Modul L298 *Driver* Motor yang digunakan adalah jenis *dual H-Bridge*.

6. Buzzer

Buzzer berfungsi sebagai indikator suara. Dalam sistem yang akan dijalankan buzzer hanya akan berbunyi ketika robot telah menemukan garis finish, yaitu ketika sensor photodioda telah mendeteksi alas/ dasar labirin yang berwarna gelap.

4.2.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang dibutuhkan dalam penelitian ini adalah.

- 1. Software Arduino IDE, yaitu untuk membuat program, menentukan perintah yang akan digunakan dan kemudian akan dimasukkan ke dalam Arduino Nano.
- 2. Free RTOS library yang digunakan untuk menangani implementasi RTOS pada Arduino. Di antara fungsi dari Free RTOS yang digunakan adalah:
 - a. xTaskCreate() fungsi ini digunakan untuk membuat sebuah tugas baru dan menambahkan ke daftar tugas yang siap dijalankan.
 - b. vTaskDelay() fungsi ini digunakan untuk memberikan sebuah jeda pada tugas-tugas dengan nilai satuan tick. Yaitu pemberian deadline untuk setiap task, dan apabila tidak terpenuhi akan ditinggal dan menuju ke task selanjutnya.

4.4. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang diterapkan untuk mendukung fungsi dasar dari sistem supaya mampu bekerja lebih optimal. Pada penelitian ini kebutuhan non-fungsional yang digunakan adalah menggunakan metode Simple Maze dan Free RTOS. Simple Maze berguna untuk menentukan pergerakan dari robot berjalan yaitu mengikuti dinding kiri dengan algoritma Wall Follower. Selanjutnya, bila peta yang sudah dibuat tersebut dijalankan. Free RTOS yaitu berguna untuk melakukan proses multitasking pada robot untuk melakukan eksekusi atau pekerjaan yang teratur.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

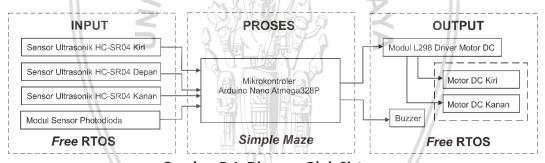
Pada bab ini menjelaskan mengenai proses perancangan dan implementasi sistem secara terperinci baik perancangan dan implementasi dari sisi perangkat keras maupun dari sisi perangkat lunak untuk Robot *Maze Wall Follower* menemukan *finish* denan menggunakan metode *Simple Maze Wall Follower* dan *multitasking* sistem menggunakan *Free* RTOS.

5.1 Perancangan Sistem

Perancangan Robot *Maze Wall Follower* dengan menggunakan *Free* RTOS pada robot *maze* ini dimulai dari perancangan *prototype* alat, perancangan perangkat keras dan perancangan perangkat lunak.

5.1.1 Gambaran Umum Sistem

Perancangan sistem ini terdapat 3 bagian yaitu *input* sistem berupa *output* sensor, proses dari sistem yang sudah ditentukan menggunakan Arduino Nano, dan *output* dari sistem berupa pergerakan dari roda dan bunyi dari buzzer. Gambar 5.1 merupakan diagram blok dari sistem yang dirancang.

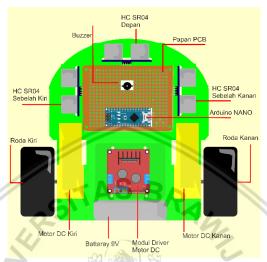


Gambar 5.1 Diagram Blok Sistem

Diaram blok pada gambar 5.1 merupakan gambaran umum terkait rancangan sistem yang dibuat, terdapat tiga bagian beruap *input*, proses dan *output*. Pada bagian *input* terdapat 3 sensor HC-SR04 untuk mendeteksi jarak yang terletak pada bagian kiri, depan, dan kanan robot dan menggunakan 1 sensor photodioda untuk menemukan garis *finish*. Setiap hasil pengukuran dan pembacaan dari setiap sensor diteruskan pada mikrokontroler Arduino Nano untuk diproses berdasarkan program yang ada dan kemudian dilanjutkan pada proses *output*. Pada proses *output* terdapat buzzer sebagai notifikasi saat garis finis ditemukan, pada Modul L298N *Driver* Motor dilakukan tindakan pemberian nilai PWM sesuai kondisi yang ada kemudian diteruskan ke Motor DC kiri dan kanan sebagai tahap *output* akhir.

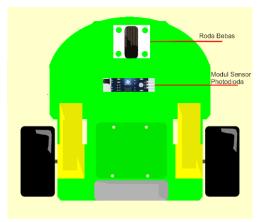
5.1.2 Perancangan Prototype Alat

Dalam mendesain *prototype* alat harus diperhatikan dengan seksama letak setiap komponen supaya dalam implementasi dapat diterapkan sesuai dengan yang diharapkan. Pada perancangan *prototype* ini hanya desain Robot *Maze Wall Follower* secara keseluruhan, yaitu pada gambar 5.2 dan gambar 5.3.



Gambar 5.2 Desain Prototype Robot Tampak Atas

Pada Gambar 5.2 adalah desain *prototype* dari Robot *Maze Wall Follower* tampak dari atas. Terdapat tiga sensor ultrasonik HC-SR04 yang diletakkan di bagian depan robot, 1 di bagian sebelah kanan, 1 di bagian sebelah kiri, dan 1 di bagian depan. Ada Dua Motor DC yang diletakkan di bagian belakang bawah robot, satu sebelah kanan, satu sebelah kiri robot. Arduino Nano diletakkan dibagikan atas robot di depan Modul L298N *Driver* Motor. Modul L298N *Driver* Motor diletakkan di bagian atas robot di belakang Arduino Nano. Buzzer sebagai notifikasi suara yang diletakkan bagian atas robot di depan Arduino Nano. Sedangkan untuk sumber daya atau baterai diletakkan di bagian tengah belakang robot yang berada di bawah papan akrilik. Papan akrilik sebagai bodi dari robot untuk menempatkan komponen-komponen yang ada, juga sebagai bentuk dari robot. Terdapat juga papan PCB yang digunakan sebagai tempat memasukkan pin dari ke tiga sensor HC-SR04, buzzer, dan Arduino Nano supaya mudah dalam penempatannya. Ada dua roda yang sebagai penggerak utama dari robot, yaitu diletakkan sebelah kanan dan kiri di bagian belakang robot.

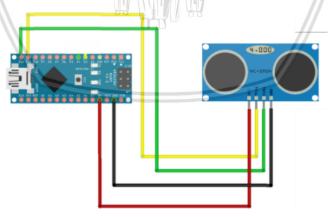


Gambar 5.3 Desain Prototype Robot Tampak Bawah

Pada Gambar 5.3 adalah desain *prototype* dari Robot *Maze Wall Follower* tampak dari bawah. Di bagian paling depan robot ada roda bebas 360 derajat yang digunakan untuk berbelok. Dan di belakang roda bebas ada modul sensor Photodioda yang digunakan untuk mendeteksi warna gelap atau terang. Sensor ini menghadap ke bawah karena objek berada di alas dari labirin yang akan dilintasi oleh robot.

5.1.3 Perancangan Perangkat Keras

Pada sub bab ini menjelaskan tentang perancangan sistem dari segi perangkat keras yaitu meliputi penjelasan skematik dari sensor ultrasonik kanan, ultrasonik depan, ultrasonik kiri, modul sensor photodioda, Modul L298N *Driver* Motor, Buzzer, Motor DC kanan, Motor DC kiri dan skematik baterai. Untuk skematik dari sensor ultasonik HC SR-04 sebelah kanan dapat dilihat pada gambar 5.4.



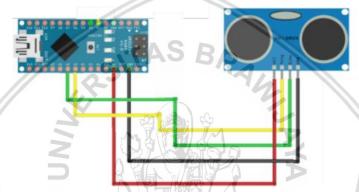
Gambar 5.4 Skematik Sensor HC-SR04 Kanan

Gambar 5.4 adalah skematik dari sensor ultrasonik HC-SR04 dengan Arduino Nano yang diletakkan di bagian kanan. Penjelasan terkait pin yang akan dihubungkan dari HC-SR04 bagian kanan menuju Arduino Nano dapat dilihat pada tabel 5.1.

Tabel 5.1 Konfigurasi Pin Sensor HC-SR04 Kanan dengan Arduino Nano

Pin Sensor Ultrasonik HC-SR04	Pin Arduino Nano
VCC	5V
Trigger	D12
Echo	D11
GND	GND

Tabel 5.1 menunjukkan konfigurasi pin sensor HC-SR04 yang terletak di bagian kanan robot dengan Arduino Nano secara terperinci. Sensor Ultrasonik HC-SR04 memiliki 4 buah pin yaitu VCC, Trigger, Echo, dan GND. Konfigurasinya yaitu pin VCC akan terhubung dengan pin 5V Arduino Nano, Trigger akan dihubungkan dengan pin Digital 11 Arduino Nano, Echo akan terhubung dengan pin Digital 12 Arduino Nano, dan GND akan terhubung dengan pin GND pada Arduino Nano.



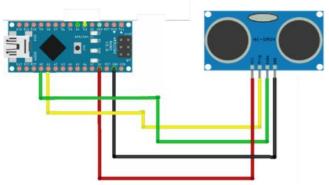
Gambar 5.5 Skematik Sensor HC-SR04 Depan

Gambar 5.5 adalah skematik dari sensor ultrasonik HC-SR04 dengan Arduino Nano yang diletakkan di bagian depan. Penjelasan terkait pin yang akan dihubungkan dari HC-SR04 bagian depan menuju Arduino Nano dapat dilihat pada tabel 5.2.

Tabel 5.2 Konfigurasi Pin Sensor HC-SR04 Depan dengan Arduino Nano

Pin Sensor Ultrasonik HC-SR04	Pin Arduino Nano
VCC	5V
Trigger	A3
Echo	A2
GND	GND

Tabel 5.2 menunjukkan konfigurasi pin sensor HC-SR04 yang terletak di bagian depan robot dengan Arduino Nano secara terperinci. Sensor Ultrasonik HC-SR04 memiliki 4 buah pin yaitu VCC, Trigger, Echo, dan GND. Konfigurasinya yaitu pin VCC akan dihubungkan dengan pin 5V Arduino Nano, Trigger akan terhubung dengan pin Analog 3 Arduino Nano, Echo akan terhubung dengan pin Analog 2 Arduino Nano, dan GND akan terhubung dengan pin GND pada Arduino Nano.



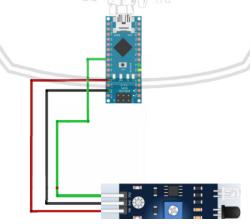
Gambar 5.6 Skematik Sensor HC-SR04 Kiri

Gambar 5.6 adalah skematik dari sensor ultrasinik HC-SR04 dengan Arduino Nano yang diletakkan di bagian kiri. Penjelasan terkait pin yang akan dihubungkan dari HC-SR04 bagian kiri menuju Arduino Nano dapat dilihat pada tabel 5.3.

Tabel 5.3 Konfigurasi Pin Sensor HC-SR04 Kiri dengan Arduino Nano

Pin Sensor Ultrasonik HC-SR04	Pin Arduino Nano
VCC	5V
Trigger	A1
Echo	A0
GND	GND

Tabel 5.3 menunjukkan konfigurasi pin sensor HC-SR04 yang terletak di bagian kiri robot dengan Arduino Nano secara terperinci. Sensor Ultrasonik HC-SR04 memiliki 4 buah pin yaitu VCC, Trigger, Echo, dan GND. Konfigurasinya yaitu pin VCC akan terhubung dengan pin 5V Arduino Nano, Trigger akan dihubungkan dengan pin Analog 1 Arduino Nano, Echo akan terhubung dengan pin Analog 0 Arduino Nano, dan GND akan terhubung dengan pin GND pada Arduino Nano.



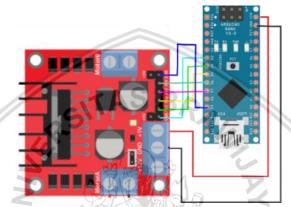
Gambar 5.7 Skematik Modul Sensor Photodioda

Gambar 5.7 adalah skematik dari Modul sensor Photodioda dengan Arduino Nano yang diletakkan di bagian bawah robot. Penjelasan terkait pin yang akan dihubungkan dari Modul sensor Photodioda menuju Arduino Nano dapat dilihat pada tabel 5.4.

Tabel 5.4 Konfigurasi Pin Modul Sensor Photodioda dengan Arduino Nano

Pin sensor Photodioda	Pin Arduino Nano
OUT	A5
GND	GND
VCC	5V

Tabel 5.4 menunjukkan konfigurasi pin Modul sensor Photodioda dengan Arduino Nano secara terperinci. Modul sensor Photodioda memiliki 3 buah pin yaitu OUT, GND, dan VCC. Konfigurasinya yaitu pin OUT akan terhubung dengan pin Analog 4 Arduino Nano, pin GND akan terhubung dengan pin GND Arduino Nano, dan pin VCC akan terhubung dengan pin 5V pada Arduino Nano.



Gambar 5.8 Skematik Modul L298N Driver Motor

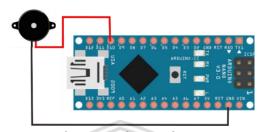
Gambar 5.8 adalah skematik dari Modul L298N *Driver* Motor dengan Arduino Nano. Penjelasan terkait pin yang akan dihubungkan dari Modul L298N Motor menuju Arduino Nano dapat dilihat pada tabel 5.5.

Tabel 5.5 Konfigurasi Pin Modul *Driver* Motor dengan Arduino Nano

Pin Modul L298N Driver Motor	Pin Arduino Nano
ENA	D3
IN1	D4
IN2	D5
IN3	D7
IN4	D6
ENB	D9
VCC 5V	5V
GND	GND

Tabel 5.5 menunjukkan konfigurasi pin Modul L298N *Driver* Motor dengan Arduino Nano secara terperinci. Modul L298N *Driver* Motor memiliki total 13 pin, namun yang akan dihubungkan dengan Arduino Nano ada 8 yaitu ENA, IN1, IN2, IN3, IN4, ENB, VCC 5V, dan GND. Konfigurasinya yaitu pin ENA akan terhubung dengan pin Digital 3 Arduino Nano, pin IN1 akan terhubung dengan pin Digital 4 Ardino Nano, pin IN2 akan terhubung dengan pin Digital 5 Ardino Nano, pin IN3 akan terhubung dengan pin Digital 7 Ardino Nano, pin IN4 akan terhubung dengan

pin Digital 6 Ardino Nano, pin ENB akan terhubung dengan pin Digital 9 Ardino Nano, pin VCC 5V akan terhubung dengan pin 5V Ardino Nano, dan pin GND akan terhubung dengan pin GND pada Ardino Nano. ENA digunakan untuk mengatur kecepatan motor kiri dan ENB digunakan untuk mengatur kecepatan motor kanan. Sedangkan IN1 dan IN2 digunakan untuk mengatur arah putaran dari motor kiri, kemudian IN3 dan IN4 digunakan untuk mengatur kecepatan motor kanan.



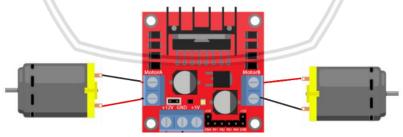
Gambar 5.9 Skematik Buzzer

Gambar 5.9 adalah skematik dari Buzzer dengan Arduino Nano. Penjelasan terkait pin yang akan dihubungkan dari Buzzer menuju Arduino Nano dapat dilihat pada tabel 5.6.

Tabel 5.6 Konfigurasi Pin Buzzer dengan Arduino Nano

1	Pin Buzzer	Pin Arduino Nano
2	Positif	D10
	Negatif	GND

Tabel 5.6 menunjukkan konfigurasi pin buzzer dengan Arduino Nano secara terperinci. Ada 2 buah pin pada buzzer yang akan dihubungkan dengan Arduino Nano, yaitu pin positif dan negatif. Pin positif akan terhubung dengan pin Digital 10 di Ardino Nano. Dan pin negatif akan terhubung dengan pin GND pada Arduino Nano.



Gambar 5.10 Skematik Motor DC

Gambar 5.10 adalah skematik dari Motor DC dengan Modul L298N *Driver* Motor DC. Ada 2 Motor DC yang dihubungkan dengan Modul L298N *Driver* Motor yaitu sebelah kiri dan kanan. Kemudian tiap Motor DC akan dipasang roda untuk berjalan. Penjelasan terkait pin yang akan dihubungkan dari Motor DC sebelah kanan menuju Modul L298N *Driver* Motor dapat dilihat pada tabel 5.7. Dan pin yang akan dihubungkan dari Motor DC sebelah kiri menuju Modul L298N *Driver* Motor dapat dilihat pada tabel 5.7.

Tabel 5.7 Konfigurasi Pin Motor DC kanan dengan Modul L298N Driver Motor

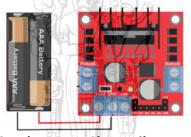
Pin Motor DC	Pin Modul L298N <i>Driver</i> Motor
Positif	OUT 4
Negatif	OUT 3

Tabel 5.7 menunjukkan konfigurasi pin Motor DC sebelah kanan dengan Modul L298N *Driver* Motor secara terperinci. Ada 2 buah pin pada Motor DC yang akan dihubungkan dengan Modul L298N *Driver* Motor, yaitu pin positif dan negatif. Pin positif akan terhubung dengan pin OUT 4 Modul *Driver* Motor DC. Dan pin negatif akan terhubung dengan pin OUT 3 pada Modul L298N *Driver* Motor.

Tabel 5.8 Konfigurasi Pin Motor DC Kiri dengan Modul L298N Driver Motor

Pin Motor DC	Pin Modul L298N <i>Driver</i> Motor
Positif	OUT 2
Negatif	OUT 1

Tabel 5.8 menunjukkan konfigurasi pin Motor DC sebelah kiri dengan Modul L298N *Driver* Motor secara terperinci. Ada 2 buah pin pada Motor DC yang akan dihubungkan dengan Modul L298N *Driver* Motor, yaitu pin positif dan negatif. Pin positif akan terhubung dengan pin *OUT* 2 Modul L298N *Driver* Motor. Dan pin negatif akan terhubung dengan pin *OUT* 1 pada Modul L298N *Driver* Motor DC.



Gambar 5.11 Skematik Baterai

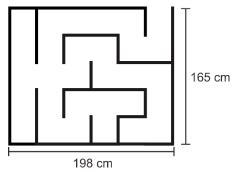
Gambar 5.11 adalah skematik dari Baterai dengan Modul L298N *Driver* Motor. Baterai yang digunakan di sini adalah 9V yang disusun secara paralel. Penjelasan terkait pin yang akan dihubungkan dari Baterai menuju Modul L298N *Driver* Motor dapat dilihat pada tabel 5.9.

Tabel 5.9 Konfigurasi Baterai dengan Modul Driver Motor

Pin baterai 9V	Pin Modul L298N <i>Driver</i> Motor
Positif	VCC +12V
Negatif	GND

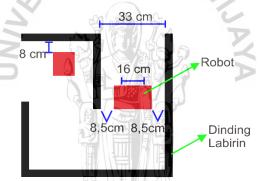
Tabel 5.9 menunjukkan konfigurasi pin baterai dengan Modul L298N *Driver* Motor secara terperinci. Ada 2 buah pin pada baterai yang akan dihubungkan dengan Modul L298N *Driver* Motor, yaitu pin positif dan negatif. Pin positif akan terhubung dengan pin VCC +12V Modul L298N *Driver* Motor. Dan pin negatif akan terhubung dengan pin GND pada Modul L298N *Driver* Motor.

Perancangan untuk labirin yang digunakan pada sistem yaitu menggunakan labirin 5x6 dengan jarak antar dinding yaitu 33 cm. Untuk gambar labirin yang digunakan dalam sistem dan ukurannya dapat dilihat pada gambar 5.12.



Gambar 5.12 Ukuran Labirin

Pada gambar 5.12 ukuran dari labirin yaitu 165x 198 cm dan kelilingnya yaitu 726 cm. Sedangkan tinggi dari dinding labirin dibuat lebih tinggi dari robot yaitu 12 cm. Untuk jarak robot supaya tetap di tengah labirin dapat dilihat pada gambar 5.13.



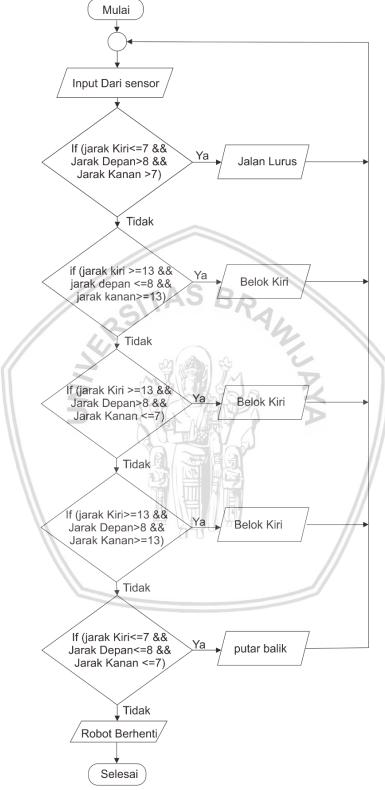
Gambar 5.13 Jarak Labirin dengan Robot

Gambar 5.13 yaitu jarak robot dengan dinding labirin supaya robot tetap berada di tengah labirin. Untuk jarak antara robot dengan dinding kanan dan kiri yaitu 8,5 cm. Dan jarak depan robot dengan dinding supaya robot tidak menabrak saat berbelok yaitu 8 cm. Maka untuk kiri kanan dibuat jarak minimal dengan dinding yaitu 7 cm supaya ada spasi sedikit robot berada di tengah. Untuk kondisi tidak ada dinding yaitu ketika jarak dinding dengan robot lebih dari 13 cm.

5.1.4 Perancangan Perangkat Lunak

Pada sub bab ini, menjelaskan tentang perancangan sistem dari segi perangkat lunak yang meliputi cara pembacaan sensor ultrasonik, pembacaan sensor photodioda, menentukan putaran motor DC, notifikasi buzzer, sistem tanpa *Free* RTOS dan sistem dengan *Free* RTOS.

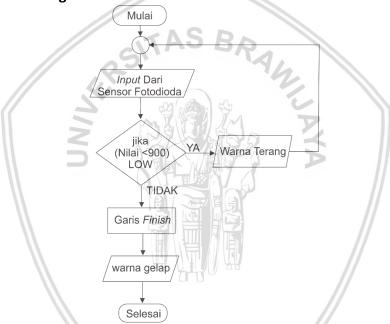
5.1.4.1 Perancangan Pembacaan Sensor Ultrasonik HC-SR04



Gambar 5.14 Flowchart Sensor Ultrasonik HC-SR04

Sensor Ultrasonik HC-SR04 diletakkan di bagian kiri, depan dan kanan robot. Sensor Ultrasonik HC-SR04 menghadap menuju dinding. Untuk acuan dalam menentukan jarak antara labirin dengan robot supaya tetap berada di tengah, yaitu berdasarkan perancangan labirin yang telah dilakukan. Saat jarak dinding bagian kiri lebih dari 13 cm maka robot akan langsung belok kiri tanpa perlu mengecek dinding bagian depan dan kanan. Sedangkan saat jarak dinding bagian kiri kurang dari atau sama dengan 7 cm maka dicek dinding depan apabila lebih dari 8 cm maka robot akan berjalan lurus. Kemudian saat jarak dinding kiri kurang dari atau sama dengan 7 cm dan dinding depan kurang dari atau sama dengan 8 cm maka dicek dinding kanan apabila lebih dari 13 cm maka robot akan belok kanan. Dan saat jarak dinding bagian kiri kurang dari 7 cm, depan kurang dari 7 dan kanan kurang dari 7 cm maka robot akan berputar balik dengan haluan kanan. Untuk pengukuran menggunakan satuan cm karena mengacu pada penelitian sebelumnya, dan satuan cm sudah cukup akurat untuk mengukur jarak.

5.1.4.2 Perancangan Pembacaan Modul Sensor Photodioda

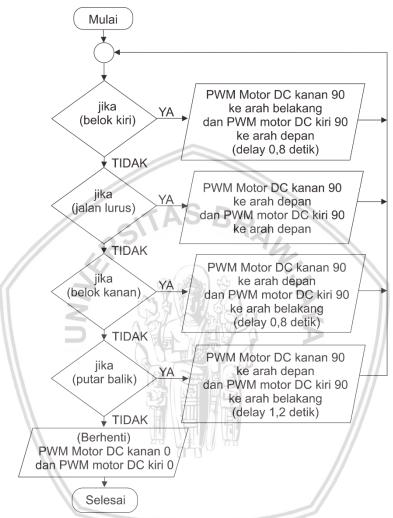


Gambar 5.15 Flowchart Sensor Photodioda

Modul sensor Photodioda yang digunakan satu unit. Sensor photodioda diletakkan di bagian bawah robot dan menghadap ke arah bawah menuju alas/dasar labirin. Jarak antara sensor dan alas yang akurat yaitu 3 cm. Alas labirin berwarna gelap yang merupakan titik garis *finish* berukuran 30x23,5 cm. Menggunakan pin Analog pada Arduino Nano. Pengecekan kondisi jika tidak setelah mendapatkan *output* dari sensor photodioda yang dikategorikan kurang dari 900 dan lebih dari 900. Nilai 900 didapatkan dari percobaan dengan melakukan perhitungan pengelompokan warna, rata-rata nilai warna yang berkategori terang yaitu di bawah 900, dan yang berkategori gelap di atas 900. Saat kondisi alas labirin terang maka nilai dari pembacaan sensor bernilai kurang dari 900 dan dalam kondisi *LOW* yang artinya bukan garis *finish*, dan saat kondisi

alas gelap maka nilai dari pembacaan sensor lebih dari 900 dan dalam kondisi *HIGH* yang artinya robot sudah menemukan garis *finish* dan robot akan berhenti.

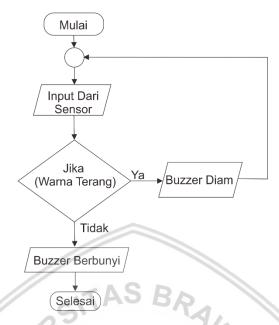
5.1.4.3 Perancangan Menentukan Putaran Motor DC pada Modul L298N *Driver*Motor



Gambar 5.16 Flowchart Modul L298N Motor

Modul L298N *Driver* Motor yang digunakan satu unit, dengan tipe *dual H-Bridge*. Modul ini merupakan alat untuk mengontrol pergerakan dari kedua motor DC. Saat kondisi jalan lurus maka kedua motor DC memiliki kecepatan yang sama. Saat berbelok kanan maka motor DC sebelah kanan lebih pelan dari motor DC sebelah kiri. Saat berbelok kiri maka motor DC sebelah kanan lebih cepat dari motor DC sebelah kiri. Saat akan putar balik dengan haluan kiri maka motor DC sebelah kanan berputar dan motor DC sebelah kiri berhenti dengan jeda tertentu. Dan ketika saat robot berhenti maka kedua motor DC berhenti berputar.

5.1.4.4 Perancangan Notifikasi Buzzer

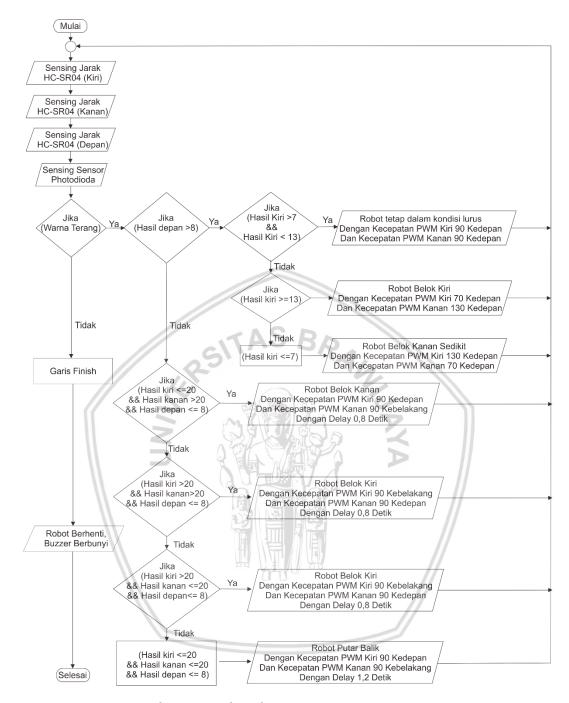


Gambar 5.17 Flowchart Buzzer

Untuk diagram buzzer sama dengan diagram alur pada sensor Photodioda, namun pada saat sensor mendeteksi warna gelap maka ditambahkan poin buzzer berbunyi. Buzzer yang digunakan ada 1 unit, sebagai notifikasi area yang berwarna gelap pada labirin. Yaitu sensor Photodioda bernilai 1 maka buzzer ini akan berbunyi sebagai notifikasi telah menemukan garis *finish*.

5.1.4.5 Perancangan Sistem Tidak Menggunakan Free RTOS

Pada poin ini akan dijelaskan tentang perancangan sistem yang tidak menggunakan *Free* RTOS. Untuk jalannya sistem sama dengan gambar 5.13 *flowchart* sistem, tetapi pada perancangan sistem tanpa RTOS lebih spesifik lagi. Yang pertama kali dilakukan adalah ketiga sensor ultrasonik dan satu sensor photodioda melakukan *sensing*. setelah semua data *output* dari sensor sudah didapatkan, maka yang pertama kali di cek adalah apakah alas/ dasar labirin berwarna gelap atau terang. Apabila dasar dari labirin berwarna terang maka akan menuju ke kondisi selanjutnya yaitu untuk menentukan robot akan jalan lurus, belok kanan sedikit, belok kiri, belok kanan, dan putar balik. Sedangkan ketika dasar dari labirin berwarna gelap maka robot akan langsung berhenti dan buzzer akan berbunyi sebagai notifikasi bahwa robot telah menemukan goal atau garis *finish*. Untuk *flowchart* sistem tidak menggunakan *Free* RTOS ditampilkan pada gambar 5.18.



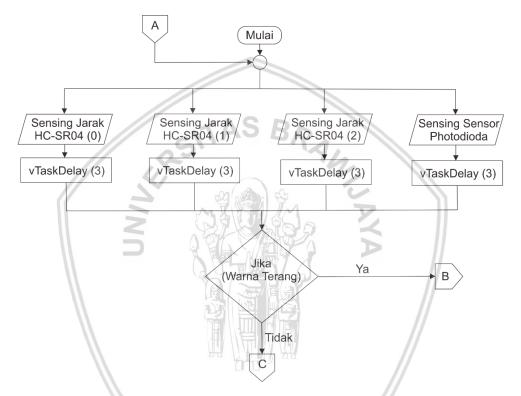
Gambar 5.18 Flowchart Sistem tanpa Free RTOS

Pada gambar 5.18 adalah perancangan sistem tanpa *Free* RTOS. Jarak supaya robot jalan lurus adalah antara 8 cm sampai 12 cm dari dinding labirin. Hal ini bertujuan supaya robot tetap berada di tengah labirin dan akurat saat akan berbelok atau putar balik.

5.1.4.6 Perancangan Sistem Menggunakan Free RTOS

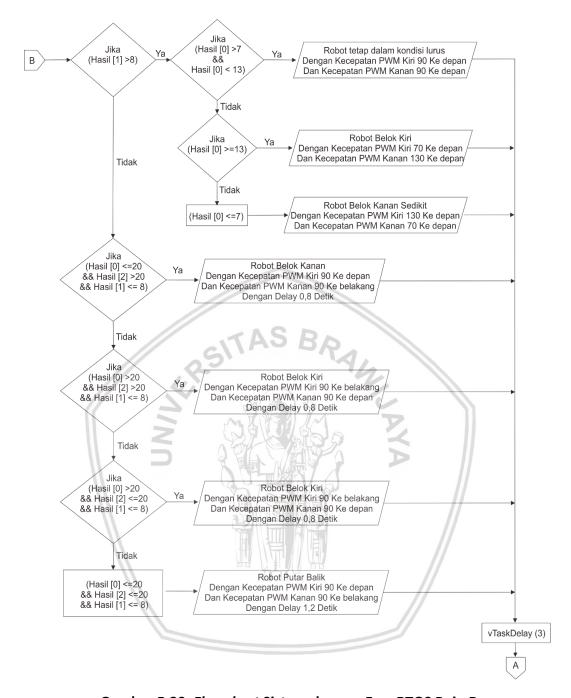
Free RTOS di sini digunakan untuk melakukan tugas-tugas dari ke 3 sensor Ultrasonik dan 1 sensor Photodioda supaya terlihat bekerja secara bersamaan.

Untuk semua modul memiliki prioritas 1, supaya cepat saat eksekusi. Faktor lama eksekusi tugas tergantung dari jarak yang terdeteksi oleh sensor, semakin jauh jarak yang terdeteksi maka semakin lama waktu eksekusi yang dibutuhkan, sebaliknya jika jarak yang terdeteksi dekat maka waktu eksekusi tugas yang dibutuhkan lebih cepat. Pada proses perancangan *Free* RTOS terdiri dari beberapa tahapan yaitu menentukan berapa tugas yang akan digunakan, menentukan tugas tiap tugas, menentukan prioritas tiap tugas, mengatur *stack size* pada tugas, dan mengatur *deadline* tiap tugas. Untuk *flowchart* sistem poin A dengan menggunakan *Free* RTOS ditampilkan pada gambar 5.19.



Gambar 5.19 Flowchart Sistem dengan Free RTOS Poin A

Pada gambar 5.19 sensor ultrasonik HC-SR04 sebelah kiri, depan, kanan, dan sensor photodioda masing-masing melakukan sensing dengan vTaskDelay (3) tick atau setara dengan 0,45 detik. Yang pertama dilakukan adalah kondisi jika warna alas dari labirin terang maka akan lanjut menuju poin B, dan apabila warna gelap maka akan menuju poin C. Untuk flowchart sistem poin B dengan menggunakan Free RTOS ditampilkan pada gambar 5.20.

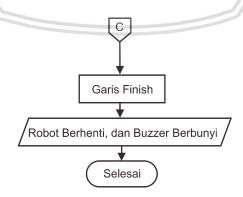


Gambar 5.20 Flowchart Sistem dengan Free RTOS Poin B

Pada gambar 5.20 yang pertama kali dilakukan adalah jika hasil jarak sensor ultrasonik HC-SR04 sebelah depan lebih dari 8 cm maka akan menuju ke kondisi untuk menyeimbangkan letak robot di tengah dan untuk robot belok kiri. Apabila hasil jarak sensor ultrasonik kiri lebih dari 7 cm dan kurang dari 13 cm maka robot akan tetap berada di tengah dengan memberikan nilai PWM motor DC kiri 90 ke depan dan motor DC kanan 90 ke depan. Kemudian jika hasil jarak sensor ultrasonik kiri lebih dari 13 cm maka robot akan kiri untuk menemukan dinding kiri dengan memberikan nilai PWM motor DC kiri 70 ke depan dan motor DC kanan

130 ke depan, untuk nilai PWM motor DC kanan ditingkatkan dari nila standar supaya robot jalan dengan cepat dan segera menemukan dinding kiri. Kondisi selanjutnya jika hasil jarak ultrasonik kiri kurang dari atau sama dengan 7 cm maka robot akan sedikit belok ke kanan untuk menjaga posisi robot tetap di tengah labirin dengan memberikan nilai PWM motor DC kiri 130 ke depan dan motor DC kanan 70 ke depan. Jika pada kondisi pertama yaitu hasil jarak sensor depan tidak lebih dari 8 cm maka akan menuju ke kondisi kedua.

Pada kondisi kedua yaitu jika hasil jarak sensor kiri kurang dari atau sama dengan 20 cm, kanan lebih dari 20 cm, dan depan kurang dari atau sama dengan 8 cm, maka robot belok kanan dengan memberikan nilai PWM motor DC kiri 90 ke depan dan motor DC kanan 90 ke belakang dengan jeda 0,8 detik. Apabila kondisi kedua tidak terpenuhi maka menuju kondisi ketiga yaitu hasil jarak sensor kiri lebih dari 20 cm, kanan lebih dari 20 cm, dan depan kurang dari atau sama dengan 8 cm, maka robot belok kiri dengan memberikan nilai PWM motor DC kiri 90 ke belakang dan motor DC kanan 90 ke depan dengan jeda 0,8 detik. Apabila kondisi ketiga tidak terpenuhi maka menuju kondisi keempat yaitu hasil jarak sensor kiri lebih dari 20 cm, kanan kurang dari atau sama dengan 20 cm, dan depan kurang dari atau sama dengan 8 cm, maka robot belok kiri dengan memberikan nilai PWM motor DC kiri 90 ke belakang dan motor DC kanan 90 ke depan dengan jeda 0,8 detik. Apabila kondisi keempat tidak terpenuhi maka menuju kondisi kelima yaitu hasil jarak sensor kiri kurang dari atau sama dengan20 cm, kanan kurang dari atau sama dengan 20 cm, dan depan kurang dari atau sama dengan 8 cm, maka robot putar balik dengan memberikan nilai PWM motor DC kiri 90 ke depan dan motor DC kanan 90 ke belakang dengan jeda 1,2 detik. Pada gambar 5.19 adalah tugas yang dilakukan pada Modul L298N Driver Motor DC yang memiliki vTaskDelay 3 tik atau 0,45 detik. Kemudian jika sudah dilakukan setiap output maka akan kembali menuju poin A. Untuk flowchart sistem poin C dengan menggunakan Free RTOS ditampilkan pada gambar 5.21.



Gambar 5.21 Flowchart Sistem dengan Free RTOS Poin C

Pada gambar 5.21 adalah jika kondisi jika hasil deteksi dari sensor photodioda berwarna gelap. Apabila sensor photodioda telah mendeteksi alas dari labirin yang

berwarna gelap berarti robot sudah menemukan *goal* atau garis *finish*. Ketika *goal* sudah tercapai maka robot berhenti dan buzzer berbunyi sebagai notifikasi sudah menemukan garis *finish*.

5.2 Implementasi Sistem

Implementasi merupakan tahap untuk merealisasikan pembuatan sistem berdasarkan perancangan sistem yang telah dilakukan. Pada sub bab ini menjelaskan satu per satu secara rinci terkait implementasi *prototype* alat, implementasi perangkat keras sistem dan implementasi perangkat lunak sistem.

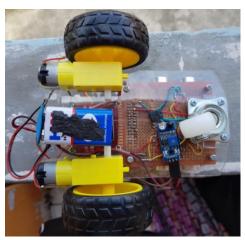
5.2.1 Implementasi Prototype Alat

Dalam mengimplementasikan *prototype* Robot *Maze Wall Follower* ini mengacu pada perancangan di sub bab 5.1.1. Untuk ukuran robotnya yaitu dengan ukuran 16 x 16 x 9 cm³. Bahan yang digunakan untuk membuat tempat tumpuan semua komponen yaitu menggunakan akrilik bening atau transparan dengan ketebalan 0,22cm dengan panjang 14cm dan lebar 9cm. Hasil implementasi *prototype* beserta peletakan komponen elektronik yang digunakan ditunjukkan pada Gambar 5.22 dan Gambar 5.23.



Gambar 5.22 Implementasi Prototype Robot Tampak Atas

Gambar 5.22 menunjukkan *prototype* dari Robot *Maze Wall Follower* tampak atas. Di bagian depan robot terdapat 3 sensor Ultrasonik HC-SR04 yang masing-masing menghadap ke kanan, depan dan kiri menuju dinding labirin. Tepat di belakang sensor Ultrasonik yang menghadap ke depan terdapat satu buah buzzer. Kemudian di bagian tengah robot ada Arduino Nano, dan di belakang Arduino Nano ada Modul L298N *Driver* Motor. Di sebelah kanan Modul L298N *Driver* Motor terdapat 1 Motor DC 9V. Dan di sebelah kiri Modul L298N *Driver* Motor terdapat 1 Motor DC 9V.

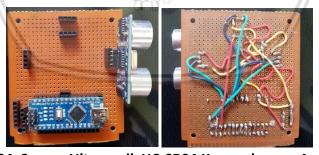


Gambar 5.23 Implementasi Prototype Robot Tampak Bawah

Gambar 5.23 adalah implementasi *prototype* Robot *Maze Wall Follower* tampak bawah. Pada gambar tersebut yang memiliki perbedaan dari tampilan tampak atas adalah paling depan adanya roda bebas yang berfungsi untuk berbelok ke mana saja. Kemudian di belakang roda bebas ada modul sensor Photodioda yang menghadap ke bawah untuk mendeteksi area alas berwarna gelap pada labirin yang digunakan sebagai garis *finish*.

5.2.2 Implementasi Perangkat Keras

Pada sub bab ini menjelaskan implementasi perangkat keras sistem yang mencakup komponen elektronik. Robot *Maze Wall Follower* memiliki beberapa komponen di dalamnya, yaitu Arduino Nano, 3 Sensor Ultrasonik HC-SR04, Modul Sensor Photodioda, Modul L298N *Driver* Motor, 2 Motor DC 9V, Buzzer, dan baterai 9V.



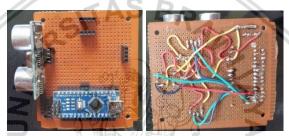
Gambar 5.24 Sensor Ultrasonik HC-SR04 Kanan dengan Arduino Nano

Pada Gambar 5.24 menunjukkan hasil implementasi dari sensor Ultrasonik HC-SR04 bagian sebelah kanan robot yang terhubung dengan Arduino Nano. Untuk menghubungkannya menggunakan kabel *jumper* berwarna biru muda, biru tua, merah dan kuning yang sudah dirangkai pada PCB.



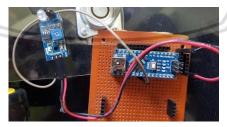
Gambar 5.25 Sensor Ultrasonik HC-SR04 Depan dengan Arduino Nano

Pada Gambar 5.25 menunjukkan hasil implementasi dari sensor Ultrasonik HC-SR04 bagian sebelah depan robot yang terhubung dengan Arduino Nano. Untuk menghubungkannya menggunakan kabel *jumper* berwarna biru muda, biru tua, merah dan kuning yang sudah dirangkai pada PCB.



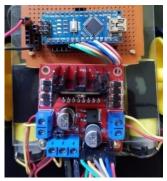
Gambar 5.26 Sensor Ultrasonik HC-SR04 Kiri dengan Arduino Nano

Pada Gambar 5.26 menunjukkan hasil implementasi dari sensor Ultrasonik HC-SR04 bagian sebelah kiri robot yang terhubung dengan Arduino Nano. Untuk menghubungkannya menggunakan kabel *jumper* berwarna biru muda, biru tua, merah dan kuning yang sudah dirangkai pada PCB.



Gambar 5.27 Modul Sensor Photodioda dengan Arduino Nano

Pada Gambar 5.27 menunjukkan hasil implementasi dari Modul Sensor Photodioda yang terhubung dengan Arduino Nano. Untuk menghubungkannya menggunakan kabel *jumper* berwarna abu-abu tua, merah dan hitam. Kemudian Modul sensor tersebut diletakkan di bagian bawah robot di belakang roda bebas berwarna putih.



Gambar 5.28 Modul L298N Driver Motor dengan Arduino Nano

Pada Gambar 5.28 menunjukkan hasil implementasi dari Modul L298N *Driver* Motor yang terhubung dengan Arduino Nano. Untuk menghubungkannya menggunakan kabel *jumper* berwarna hijau, oren, abu-abu, putih, biru dan oren tua. Untuk menghubungkan VCC dan *ground* menggunakan kabel *jumper* berwarna merah dan hitam.



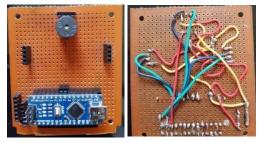
Gambar 5.29 Motor DC Kanan dengan Modul L298N Driver Motor

Pada Gambar 5.29 menunjukkan hasil implementasi dari Motor DC sebelah kanan robot yang terhubung dengan Modul L298N *Driver* Motor. Untuk menghubungkannya menggunakan kabel *jumper* berwarna merah dan kuning.



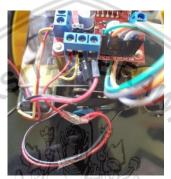
Gambar 5.30 Motor DC Kiri dengan Modul L298N Driver Motor

Pada Gambar 5.30 menunjukkan hasil implementasi dari Motor DC sebelah kiri robot yang terhubung dengan Modul L298N *Driver* Motor. Untuk menghubungkannya menggunakan kabel *jumper* berwarna merah dan kuning.



Gambar 5.31 Buzzer dengan Arduino Nano

Pada Gambar 5.31 menunjukkan hasil implementasi dari Buzzer yang terhubung dengan Arduino Nano. Untuk menghubungkannya menggunakan kabel *jumper* berwarna merah dan kuning yang sudah dirangkai dalam PCB.



Gambar 5.32 Baterai 9V dengan Modul L298N Driver Motor

Pada Gambar 5.32 menunjukkan hasil implementasi dari baterai 9V yang terhubung dengan Modul L298N *Driver* Motor. Untuk menghubungkannya menggunakan kabel *jumper* berwarna merah dan hitam.



Gambar 5.33 Labirin 5x6

Pada Gambar 5.33 menunjukkan hasil implementasi dari pembuatan labirin. Labirin yang dibuat menggunakan kertas karton. Dan implementasi berdasarkan dari perancangan labirin yang dilakukan sebelumnya.

5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak sistem pada sub bab ini menjelaskan proses dalam menjalankan program Robot *Maze Wall Follower* dari perancangan sistem.

Dalam mengimplementasikan perangkat lunak ini menggunakan *software* Arduino IDE 1.6.6. Untuk implementasi dari setiap sensor, buzzer, dan modul L298 N *Driver* Motor DC dibuat 2 poin yaitu belum menggunakan *Free* RTOS dan sudah menggunakan *Free* RTOS. Sedangkan untuk Sistem yang sudah jadi juga dibuat 2 poin yaitu sistem yang tidak menggunakan *Free* RTOS dan sistem yang menggunakan *Free* RTOS.

5.2.3.1 Implementasi Pembacaan Sensor Ultrasonik HC-SR04

Pada implementasi pembacaan sensor HC-SR04 sistem menggunakan Arduino Nano untuk menjalankan program dengan menggunakan bahasa pemrograman C. Digunakan perhitungan dengan pengukuran centi meter (cm) karena pada dasar teori standar penukuran untuk sensor ultrasonik HC-SR04 menggunakan satuan cm. Pada tabel 5.10 adalah *source code* untuk pembacaan sensor ultrasonik HC-SR04 tanpa penerapan *Free* RTOS.

Tabel 5.10 Source Code Sensor Ultrasonik HC-SR04

```
No
                                Source code
1
      int trigPin1 = 12;
      int echoPin1 = 11;
2
3
      int trigPin2 = A3;
4
      int echoPin2 = A2;
5
      int trigPin3 = A1;
6
      int echoPin3 = A0;
7
      void setup() {
8
        Serial.begin (9600);
9
        pinMode(trigPin1, OUTPUT);
10
        pinMode(echoPin1, INPUT);
11
        pinMode(trigPin2, OUTPUT);
12
        pinMode(echoPin2, INPUT);
13
        pinMode(trigPin3, OUTPUT);
14
        pinMode(echoPin3, INPUT);
15
16
      void sensorkiri(){
17
        int kiri1, kiri;
18
        digitalWrite (trigPin1, HIGH);
19
        delayMicroseconds (10);
20
        digitalWrite (trigPin1, LOW);
21
        kiri1 = pulseIn (echoPin1, HIGH);
22
        kiri = (kiri1/2)/29.1;
23
            Serial.print("Sensor: Kiri ");
24
            Serial.print(kiri);
25
            Serial.print("cm
26
27
      void sensordepan() {
28
          int depan1, depan;
29
          digitalWrite (trigPin2, HIGH);
30
          delayMicroseconds (10);
31
          digitalWrite (trigPin2, LOW);
32
          depan1 = pulseIn (echoPin2, HIGH);
33
          depan = (depan1/2)/29.1;
34
            Serial.print("Sensor Depan: ");
35
            Serial.print(depan);
36
            Serial.print("cm
```

```
37
      void sensorkanan(){
38
39
          int kanan1, kanan;
40
          digitalWrite (trigPin3, HIGH);
41
          delayMicroseconds (10);
42
          digitalWrite (trigPin3, LOW);
43
          kanan1 = pulseIn (echoPin3, HIGH);
          kanan = (kanan1/2)/29.1; //konversi ke dalam cm
44
45
            Serial.print("Sensor Kanan: ");
46
            Serial.print(kanan);
            Serial.print("cm");
47
48
49
        void loop() {
50
          Serial.println("\n");
51
          sensorkiri();
52
          sensordepan();
53
          sensorkanan();
54
      delay (500);
55
```

Tabel 5.10 merupakan *source code* yang digunakan pada ketiga Sensor Ultrasonik HC-SR04. Pertama dilakukan inisialisasi pin sensor. Pada program di atas pin trigger digunakan untuk membangkitkan sinyal ultrasonik, trigger akan dibuat dalam kondisi HIGH selama 10 mikro detik(us), kemudian diberikan pulsa LOW pendek untuk memastikan pulsa HIGH telah selesai dilakukan. Kemudian pin echo digunakan untuk sinyal pantulan ultrasonik dari pin trigger. Pada bagian *pulseln* yaitu digunakan untuk menotal waktu tunggu dari echoPin saat HIGH. Jarak dinding dengan sensor diukur dari lama waktu yang dibutuhkan untuk trigger memberikan gelombang yang kemudian hasil pantulan akan diterima oleh echo. Jadi lama waktu dihitung dari gelombang pergi menuju objek atau dinding dan gelombang kembali menuju HC-SR04.

Hasil dari lama waktu pengukuran yang diterima oleh pin echo kemudian di bagi 2 karena gelombang bolak balik secara 2 kali. Untuk mengonversi ke dalam cm (centi meter) maka hasilnya di bagi 29,1. Menggunakan 29,1 dikarenakan kecepatan suara memiliki kecepatan 343 m/detik. Berdasarkan rumus kecepatan suara meter dikonversi ke cm, kemudian 1 detik dikonversi menjadi mikro detik maka nilainya 1.000.000/34.300 = 29,154. Maka untuk mengubah suara menjadi jarak dengan satuan cm yaitu.

Total Jarak =
$$\frac{\text{(lama waktu pengukuran/2)}}{29,1}$$

Data akan ditampilkan dalam serial monitor dengan keterangan sensor kiri, depan dan kanan berapa centimeter dengan jeda 0,5 detik.

5.2.3.2 Implementasi Pembacaan Modul Sensor Photodioda

Pada tabel 5.11 adalah *source code* untuk pembacaan modul sensor photodioda tanpa penerapan *Free* RTOS.

Tabel 5.11 Source Code Modul Sensor Photodioda

```
Source code
No
1
      int FlamePin = A5;
2
      int Flame = HIGH;
3
      void setup() {
4
        pinMode(FlamePin, INPUT);
5
        Serial.begin(9600);
6
7
      void loop() {
8
      Photodioda();
      delay(500);
9
10
11
      void Photodioda () {
12
        Flame = digitalRead(FlamePin);
13
        if (Flame== LOW) {
14
          Serial.println("Belum Menemukan Garis finish"); }
15
          Serial.println("Garis finish !!!");
16
17
        } }
```

Tabel 5.11 merupakan *source code* yang diterapkan pada Modul Sensor Photodioda. Pertama dilakukan inisialisasi pin sensor kemudian sensor akan mendeteksi warna gelap atau terang. Ketika warna terang maka sensor dalam kondisi *LOW* dan bernilai 0 maka pada serial monitor akan ditampilkan bahwa belum menemukan garis *finish*. Dan ketika sensor mendeteksi warna gelap maka dalam kondisi *HIGH* dan bernilai 1 maka pada serial monitor akan ditampilkan keterangan garis *finish*.

5.2.3.3 Implementasi Menentukan Putaran Motor DC pada Modul L298N *Driver* Motor

Pada tabel 5.12 adalah *source code* untuk Menentukan Putaran Motor DC pada Modul L298N *Driver* Motor tanpa penerapan *Free* RTOS.

Tabel 5.12 Source Code Modul L298N Driver Motor

No	Source code
1	<pre>const int motor1_kiri = 3;</pre>
2	<pre>const int motor1_kanan = 9;</pre>
3	<pre>const int motor1_in1 = 4;</pre>
4	<pre>const int motor1_in2 = 5;</pre>
5	<pre>const int motor1_in3 = 7;</pre>
6	<pre>const int motor1_in4 = 6;</pre>
7	<pre>void setup() {</pre>
8	Serial.begin(9600);
9	<pre>pinMode(motor1_kiri , OUTPUT);</pre>
10	<pre>pinMode(motor1_kanan , OUTPUT);</pre>
11	<pre>pinMode(motor1_in1 , OUTPUT);</pre>
12	<pre>pinMode(motor1_in2 , OUTPUT);</pre>
13	<pre>pinMode(motor1_in3 , OUTPUT);</pre>
14	<pre>pinMode(motor1_in4 , OUTPUT);</pre>
15	}
16	<pre>void loop() {</pre>
17	<pre>analogWrite(motor1_kiri, 90);</pre>

```
analogWrite(motor1 kanan,
18
                                     90);
19
          digitalWrite( motor1 in1, 255);
20
          digitalWrite( motor1_in2,
21
          digitalWrite( motor1_in3,
                                        0);
22
          digitalWrite( motor1 in4, 255);
23
      Serial.println("Robot Jalan Lurus");
24
      delay(5000);
25
          analogWrite(motor1 kiri , 90);
25
          analogWrite(motor1 kanan, 90);
26
          digitalWrite( motor1 in1,
27
          digitalWrite (motor1 in2, 255);
          digitalWrite( motor1 in3,
28
29
          digitalWrite (motor1 in4, 255);
30
      Serial.println("Robot Belok Kiri");
31
      delay(5000);
32
          analogWrite(motor1 kiri,
33
          analogWrite(motor1 kanan, 90);
34
          digitalWrite (motor1 in1, 255);
35
          digitalWrite( motor1 in2,
                                        0);
36
          digitalWrite (motor1 in3, 255);
          digitalWrite( motor1 in4,
37
      Serial.println("Robot Belok kanan");
38
39
      delay(5000);
40
          analogWrite(motor1 kiri ,
          analogWrite(motor1 kanan, 90);
41
42
          digitalWrite(motor1_in1, 255);
          digitalWrite( motor1_in2, \land 0);
43
44
          digitalWrite( motor1 in3, 255);
45
          digitalWrite( motor1 in4,
                                     0);
46
          Serial.println("Robot Putar Balik"
47
      delay(5000);
48
          analogWrite ( motor1 kiri ,
          analogWrite ( motor1 kanan ,
49
                                         0);
50
          digitalWrite( motor1_in1,
                                         0);
          digitalWrite( motor1 in2,
51
                                         0);
          digitalWrite( motor1 in3,
52
                                         0);
53
          digitalWrite ( motor1 in4,
                                         0);
54
          Serial.println("Robot Berhenti");
55
      delay(5000);}
```

Tabel 5.12 merupakan *source code* yang digunakan pada Modul L298N *Driver* Motor. Kondisi yang dilakukan yaitu kedua motor DC akan berputar ke arah depan dan nilai PWM Motor DC sebelah kiri dan kanan sama apabila jalan lurus. Nilai PWM motor sebelah kiri dan kanan sama yaitu 90 dan arah putaran motor DC kiri ke belakang dan motor DC kanan ke depan apabila berbelok kiri. Sedangkan apabila akan berbelok kanan maka nilai PWM motor sebelah kanan dan kiri sama yaitu 90 untuk putaran motor DC kiri ke depan dan motor DC kanan ke belakang. Dan ketika akan berputar balik maka PWM motor sebelah kiri dan kanan sama yaitu 90 untuk arah putaran motor DC kiri ke depan dan motor DC kanan ke belakang.

5.2.3.4 Implementasi Notifikasi Buzzer

Pada tabel 5.13 adalah *source code* untuk notifikasi buzzer tanpa penerapan *Free* RTOS.

Tabel 5.13 Source Code Buzzer

```
Source code
No
      int Buzzer = 10;
1
2
      int FlamePin = A5;
3
      int Flame = HIGH;
4
      void setup() {
5
        Serial.begin (9600);
6
        pinMode(Buzzer, OUTPUT);
7
        pinMode(FlamePin, INPUT);
8
9
      void loop() {
10
        Photodioda();
11
12
      void Photodioda () {
13
        Flame = digitalRead(FlamePin)
        if (Flame== LOW) {
14
15
          Serial.println("Warna terang
16
          digitalWrite(Buzzer, HIGH);
17
          delay (500);
18
19
        else{
          Serial.println("Warna Gelap");
20
21
          digitalWrite (Buzzer, LOW);
22
          delay (500);
23
```

Tabel 5.13 merupakan source code dari Buzzer. Pada source code tersebut hampir sama dengan source code pada sensor Photodioda, karena buzzer digunakan sebagai notifikasi saat sensor Photodioda telah menemukan garis finish yaitu objek berwarna gelap. Apabila sensor Photodioda mendeteksi warna terang maka buzzer akan diam dan apabila mendeteksi warna gelap maka buzzer akan berbunyi.

5.2.3.5 Implementasi Sistem Tidak Menggunakan Free RTOS

Pada tabel 5.14 adalah *source code* dari sistem yang sudah berjalan tanpa menggunakan *Free* RTOS.

Tabel 5.14 Source Code Sistem tanpa Free RTOS

No	Source code
1	int Buzzer = 10;
2	<pre>int FlamePin = A5;</pre>
3	int motor1 kiri = 3;
4	<pre>int motor1_in1 = 4;</pre>
5	<pre>int motor1_in2 = 5;</pre>
6	<pre>int motor1_kanan = 9;</pre>
7	<pre>int motor1_in3 = 7;</pre>
8	<pre>int motor1_in4 = 6;</pre>
9	int trigger_kiri = 12;

```
10
            int echo kiri = 11;
11
            int trigger depan = A3;
12
            int echo depan = A2;
13
            int trigger kanan = A1;
14
            int echo kanan = A0;
15
      void setup()
16
17
             Serial.begin(9600);
18
              pinMode(trigger depan, OUTPUT);
19
              pinMode(echo depan, INPUT);
2.0
              pinMode(trigger kiri, OUTPUT);
21
              pinMode(echo kiri, INPUT);
22
              pinMode(trigger kanan, OUTPUT);
23
              pinMode(echo kanan, INPUT);
24
              pinMode( motor1 in1 , OUTPUT);
25
              pinMode( motor1 in2 , OUTPUT);
26
              pinMode( motor1_in3 , OUTPUT);
27
              pinMode( motor1 in4 , OUTPUT);
28
              pinMode(motor1 kiri, OUTPUT);
29
              pinMode(motor1 kanan, OUTPUT);
30
              pinMode (Buzzer, OUTPUT);
              pinMode(FlamePin, INPUT);
31
32
33
      void loop() {
       int hasil_depan, hasil_kiri, hasil_kanan, jarak_kanan,
34
35
      jarak_kiri, jarak_depan;
36
              digitalWrite (trigger_kiri, LOW);
37
              delayMicroseconds (2);
38
              digitalWrite (trigger kiri, HIGH);
39
              delayMicroseconds(5);
              jarak_kiri = pulseIn (echo_kiri, HIGH);
hasil_kiri = (jarak_kiri/2/)29.1;
40
41
              Serial.print("Sensor kiri : ");
42
43
              Serial.print(hasil kiri);
              Serial.println(" cm");
44
              digitalWrite (trigger depan, LOW);
45
46
              delayMicroseconds (2);
47
              digitalWrite (trigger depan, HIGH);
48
              delayMicroseconds(5);
49
              jarak depan = pulseIn (echo depan, HIGH);
50
              hasil depan = (jarak depan/2)/29.1;
51
              Serial.print("Sensor depan : ");
              Serial.print(hasil_depan);
52
53
              Serial.println(" cm");
54
              digitalWrite (trigger kanan, LOW);
55
              delayMicroseconds (2);
56
              digitalWrite (trigger kanan, HIGH);
              delayMicroseconds(5);
57
58
              jarak kanan = pulseIn (echo kanan, HIGH);
59
              hasil_kanan = (jarak_kanan/2)/29.1;
60
              Serial.print("Sensor kanan : ");
61
              Serial.print(hasil kanan);
62
              Serial.println(" cm");
63
      int Flame = HIGH;
64
      Flame = digitalRead(FlamePin);
65
      if (Flame== LOW) {
66
          if(hasil depan >8)
67
68
              if(hasil kiri >7 && hasil kiri< 13)
```

```
69
70
                                                 90);
                     analogWrite (motor1 kiri,
71
                     analogWrite (motor1 kanan, 90);
72
                     digitalWrite( motor1 in1,
                                                 255);
73
                     digitalWrite( motor1 in2,
                                                  0);
74
                     digitalWrite( motor1_in3,
                                                   0);
75
                     digitalWrite( motor1 in4,
                                                 255);
76
                     Serial.println("Robot jalan lurus");
77
                         digitalWrite(Buzzer, HIGH);
78
79
              if(hasil kiri >=13)
80
81
                     analogWrite(motor1 kiri,
                                                70);
82
                     analogWrite(motor1 kanan, 130);
83
                    digitalWrite (motor1 in1, 255);
84
                    digitalWrite (motor1 in2,
85
                     digitalWrite ( motor1 in3,
86
                     digitalWrite (motor1 in4, 255);
87
                     Serial.println("Robot belok kiri");
                     digitalWrite(Buzzer, HIGH);
88
89
90
              if(hasil_kiri <=7)</pre>
91
                     analogWrite(motor1 kiri,
92
                    analogWrite(motor1_kanan, 70);
93
                    digitalWrite( motor1_in1, 255);
94
95
                    digitalWrite( motor1 in2, 0 );
                    digitalWrite( motor1 in3, 0);
96
                    digitalWrite( motor1 in4, 255);
97
98
             Serial.println("Robot belok kanan sedikit");
99
                     digitalWrite(Buzzer, HIGH);
100
101
102
         if(hasil kiri
                         <=20 && hasil kanan >20 && hasil_depan
103
      <=8) {
104
            analogWrite ( motor1 kiri ,
                                           90);
            analogWrite ( motor1 kanan ,
105
                                           90);
106
            digitalWrite( motor1 in1,
                                           255);
            digitalWrite( motor1 in2,
107
                                             0);
            digitalWrite( motor1 in3,
108
                                           255);
            digitalWrite ( motor1 in4,
109
                                             0);
110
            Serial.println("Robot belok kanan");
111
            digitalWrite(Buzzer, HIGH);
112
            delay(800);
113
114
         if(hasil kiri
                          >20 && hasil kanan >20 && hasil depan
115
      <=8) {
116
            analogWrite( motor1 kiri ,
                                          90);
            analogWrite( motor1_kanan ,
117
118
            digitalWrite( motor1_in1,
                                            0);
            digitalWrite( motor1_in2,
119
                                          255);
            digitalWrite( motor1_in3,
120
                                            0);
            digitalWrite( motor1_in4,
                                          255);
121
122
            Serial.println("Robot belok kiri ");
123
            digitalWrite(Buzzer, HIGH);
124
            delay(800);
125
126
         if(hasil kiri
                          >20 && hasil kanan <=20 && hasil depan
127
      <=8) {
```

```
90);
128
            analogWrite( motor1 kiri ,
129
            analogWrite( motor1 kanan , 90);
130
            digitalWrite( motor1 in1,
                                            0);
131
            digitalWrite( motor1_in2,
                                          255);
132
            digitalWrite( motor1 in3,
                                            0);
133
            digitalWrite( motor1 in4,
                                          255);
            Serial.println("Robot belok kiri ");
134
135
            digitalWrite(Buzzer, HIGH);
136
            delay(800);
137
         if(hasil kiri <=20 && hasil kanan <=20 && hasil depan
138
139
      <=8)
140
141
            analogWrite( motor1 kiri ,
                                          90);
142
            analogWrite( motor1 kanan ,
143
            digitalWrite( motor1 in1,
144
            digitalWrite( motor1 in2,
                                            0);
            digitalWrite( motor1 in3,
                                          255);
145
            digitalWrite ( motor1 in4,
146
            Serial.println("Robot putar balik");
147
148
            digitalWrite(Buzzer, HIGH);
149
            delay(1200);
90
151
152
      else
153
            analogWrite ( motor1 kiri ,
154
            analogWrite ( motor1 kanan ,
155
            digitalWrite( motor1 in1,
                                           0);
156
            digitalWrite( motor1 in2,
                                           0);
157
            digitalWrite( motor1 in3,
                                           0);
            digitalWrite( motor1 in4,
158
                                           0);
159
            Serial.println("Robot Berhenti");
            Serial.println("Menemukan Garis finish");
160
161
            digitalWrite (Buzzer, LOW);
162
163
```

Pada tabel 5.14 adalah source code sistem yang sudah jadi tanpa menggunakan Free RTOS. Yang pertama dilakukan adalah inisialisasi tiap pin yang digunakan oleh sensor dan modul L298N Driver motor. Kemudian di bagian void setup() menentukan di bagian pin mana yang menjadi input dan output nya. Dan untuk perhitungan dari sensor ultrasonik, mendeteksi garis berwarna terang dan gelap oleh sensor photodioda, output dari sistem yang berupa pemberian nilai PWM serta arah gerakan dari motor DC, dan buzzer sebagai notifikasi di letakkan pada void loop().

Yang pertama dilakukan pada *void loop()* yaitu inisialisasi variabel yang digunakan untuk perhitungan dengan rumus pada program sensor ultrasonik. Kemudian di bawahnya dilakukan program perhitungan untuk pengukuran jarak pada sensor ultrasonik kiri, depan, dan kanan. Setelah itu dilakukan inisialisasi untuk sensor photodioda dengan keterangan *HIGH* adalah sat mendeteksi garis berwarna gelap. Setelah itu jika kondisi sensor photodioda LOW atau mendeteksi garis berwarna terang maka akan menuju pengecekan apakah jarak di depan robot

lebih dari 8 cm. Jika kondisi tersebut terpenuhi maka di dalamnya dilakukan pengecekan jika jarak kiri lebih dari 7 cm dan kurang dari 20 cm robot jalan lurus, bila jarak dinding kiri lebih dari atau sama dengan 20 cm maka robot belok kiri, bila jarak kiri kurang dari atau sama dengan 7 cm maka robot belok sedikit ke arah kanan supaya robot tetap berada di tengah labirin. Kondisi kedua yaitu apabila jarak dinding kiri kurang dari atau sama dengan 20 cm, jarak dinding kanan lebih dari 20 cm, dan jarak dinding depan kurang dari atau sama dengan 8 maka robot belok kanan sebesar 90 derajat dengan cara PWM Motor DC kiri bernilai 90 ke arah depan dan nilai PWM Motor DC kanan bernilai 90 ke arah belakang dengan jeda 0,8 detik. Kondisi ketiga yaitu apabila jarak dinding kiri lebih dari 20 cm, jarak dinding kanan lebih dari 20 cm, dan jarak dinding depan kurang atau sama dengan 8 cm maka robot belok kiri sebesar 90 derajat dengan cara PWM Motor DC kiri bernilai 90 ke arah belakang dan nilai PWM Motor DC kanan 90 ke arah depan dengan jeda 0,8 detik. Kondisi ke empat adalah apabila jarak dinding kiri lebih dari 20 cm, jarak dinding kanan kurang dari atau sama dengan 20 cm, dan jarak dinding depan kurang dari atau sama dengan 8 cm maka robot belok kiri sebesar 90 derajat dengan cara PWM Motor DC kiri bernilai 90 ke arah belakang dan nilai PWM Motor DC kanan 90 ke arah depan dengan jeda 0,8 detik. Kondisi ke lima adalah apabila jarak dinding kiri kurang dari 20 cm, jarak dinding depan kurang dari atau sama dengan 8 cm, jarak dinding kanan kurang dari 20 cm maka robot putar balik haluan kanan sebesar 180 derajat dengan cara nilai PWM Motor DC kiri 90 ke arah depan dan jeda PWM Motor DC kanan bernilai 90 ke arah depan dengan jeda 1,2 detik. Jika kondisi sensor photodioda HIGH atau mendeteksi garis berwarna gelap robot langsung berhenti dan buzzer berbunyi.

5.2.3.6 Implementasi Sistem Menggunakan Free RTOS

Pada sub bab ini akan dibahas implementasi RTOS pada setiap *task void* yang digunakan pada sistem. Pada *task* sensor Ultrasonik dibuat secara terpisah yaitu TaskUltrasonikKiri, TaskUltrasonikDepan, dan TaskUltrasonikKanan. Kemudian pada *task* sensor photodioda dan buzzer dijadikan dalam satu *task* karena buzzer bekerja dalam satu *void* dengan modul sensor photodioda dan diberi nama TaskPhoto. Dan *task* yang terakhir yaitu pada TaskMotor yang merupakan *output* dari sistem untuk menentukan nilai PWM yang akan diberikan ke motor DC yang dilakukan putaran roda untuk kondisi tertentu.

1. Implementasi Pembuatan Task Pada Real Time Operating System (RTOS)

Pada *source code* yang digunakan dalam sistem dengan RTOS dibuat menjadi potongan-potongan supaya bisa dijelaskan tiap poin-poin yang penting.

Tabel 5.15 Potongan Program Inisialisasi Pin

```
RTOS Robot.ino
     1
     #include <Arduino FreeRTOS.h>
2
      int Buzzer=10;
3
      int FlamePin = A5;
4
      int flame;
5
      int trigPin[3] = {12, A3, A1};
6
      int echoPin[3] = \{11, A2, A0\};
7
      int hasil[3];
8
      int jarak[3];
9
      int motor1 kiri = 3;
10
      int motor1 in1 = 4;
11
      int motor1 in2 = 5;
12
      int motor1 kanan = 9;
13
      int motor1 in3 = 7;
14
      int motor1 in4 = 6;
15
      uint16 t sensorsekarang = 0;
```

Pada tabel 5.15 hal yang paling atas atau pertama dilakukan adalah memasukkan *library* dari *Free* RTOS. Kemudian dilakukan inisialisasi tiap pin dari Buzzer, Sensor Ultrasonik HC-SR04, dan Modul L298N *Driver* Motor. Untuk inisialisasi pin buzzer dan sensor photodioda menggunakan int yaitu bilangan bulat, untuk pin trigger, echo, hasil, dan jarak menggunakan int *array*, untuk pin modul motor menggunakan konstanta bilangan bulat supaya tidak bernilai negatif, dan untuk sensorsekarang yang merupakan inisialisasi yang digunakan pada *task* tiap sensor ultrasonik menggunakan bilangan yang tidak bulat supaya memori yang digunakan lebih sedikit. Pada buzzer menggunakan pin digital 10, pada modul sensor photodioda menggunakan pin Analog 5, pada sensor ultrasonik bagian kiri menggunakan pin Digital, sedangkan untuk sensor ultrasonik depan dan kanan menggunakan pin Analog, dan untuk pin Modul L298N *Driver* Motor menggunakan pin Digital. Deklarasi *task* pada RTOS ditampilkan pada Tabel 5.16.

Tabel 5.16 Potongan Program Pendefinisian *Task* RTOS

Tabel 5.16 adalah semua *task* yang digunakan dalam sistem. Dalam penerapan *Free* RTOS langkah pertama yang harus dilakukan adalah mendefinisikan *task* yang akan digunakan. Sesuai dengan banyak proses yang digunakan dalam sistem, pada sistem ini ada 5 proses maka didefinisikan 5 *task* yaitu *TaskUltrasonikKiri()*, *TaskUltrasonikDepan()*, *TaskUltrasonikKanan()*, *TaskPhoto()*, dan *TaskMotor()*.

Untuk void setup yang digunakan pada RTOS yang digunakan ditampilkan pada Tabel 5.17.

Tabel 5.17 Potongan Program Void Setup()

```
RTOS Robot.ino
1
     Serial.begin (9600);
2
      for (int i = 0; i < 3; i + +) {
3
       pinMode(trigPin[i], OUTPUT);
4
5
      for (int i = 0; i < 3; i + +) {
6
       pinMode(echoPin[i], INPUT);
7
8
       pinMode( motor1 in1 , OUTPUT);
9
       pinMode( motor1 in2 , OUTPUT);
       pinMode( motor1_in3 , OUTPUT);
10
11
       pinMode( motor1 in4 , OUTPUT);
12
       pinMode(motor1 kiri, OUTPUT);
       pinMode(motor1 kanan, OUTPUT);
13
       pinMode(Buzzer, OUTPUT);
14
15
```

Pada tabel 5.17 di dalam *void setup()* pertama kali dilakukan adalah menentukan *baudrate* dari program. Kemudian untuk pin trigger dan pin echo pada sensor ultrasonik dibuat *looping* untuk mendefinisikan pin trigger dan echo merupakan *output* nya, dan hasil dari pengukuran dari sensor bisa dipanggil dalam *task* yang lainnya. Kemudian menentukan *input* atau *output* dari pin yang digunakan dengan pinMode. Setelah itu di dalam *void setup()* juga disertakan pembuatan tiap *task* yang digunakan. Untuk program pembuatan *task* ditampilkan pada tabel 5.18.

Tabel 5.18 Potongan Program Pembuatan Task RTOS

```
RTOS Robot.ino
     xTaskCreate(
2
       TaskKiri
       , (const portCHAR *) "Ultrasonik1"
3
4
         128
       , NULL
5
6
       , 1
7
       , NULL);
8
     xTaskCreate(
9
       TaskDepan
10
       , (const portCHAR *) "Ultrasonik2"
11
         128
       , NULL
12
       , 1
13
       , NULL);
14
15
     xTaskCreate(
16
       TaskKanan
       , (const portCHAR *) "Ultrasonik3"
17
        128
18
       , NULL
19
20
         1
```

```
, NULL);
21
22
      xTaskCreate(
23
        TaskPhoto
        , (const portCHAR *) "Photodioda"
24
         128
25
        , NULL
26
27
        , 1
28
        , NULL);
29
       xTaskCreate(
30
        TaskMotor
        , (const portCHAR *) "Motor"
31
        , 128
32
        , NULL
33
        , 1
34
35
        , NULL);
36
```

Pada potongan program di tabel 5.18 pada task 3 sensor ultrasonik, 1 sensor Photodioda, dan task motor memiliki nilai prioritas 1 dengan tujuan agar sistem ini berjalan secara paralel. Untuk penjadwalan dengan task diatur oleh RTOS. Jalan dari program dimulai dari task yang memiliki posisi paling atas dan memiliki waktu eksekusi paling cepat. Jika task tersebut mendeteksi jarak yang sama semua dan mengakibatkan waktu eksekusi dari tiap task hampir sama, maka task yang akan di eksekusi terlebih dahulu adalah task yang memiliki posisi paling atas atau TasUltrasonik1 yaitu sensor Ultrasonik yang terletak di bagian kiri. Pada bagian Stack Depth di bagian task sensor Ultrasonik dan Photodioda bernilai 128 yang bertujuan untuk mengalokasikan ukuran stack pada task tersebut sebesar 128. Dan Stack Depth di bagian task motor bernilai 128 yang bertujuan untuk mengalokasikan ukuran stack pada task tersebut sebesar 128. Semakin banyak komputasi yang dilakukan task maka membutuhkan semakin besar Stack Depth. Penentuan nilai 128 didapat dari Stack Depth default dari Free RTOS. Untuk potongan program proses penghitungan yang dilakukan oleh sensor ultrasonik ditampilkan pada tabel 5.19.

Tabel 5.19 Potongan Program Proses Penghitungan Nilai Sensor Ultrasonik

```
RTOS Robot.ino
1
      int ultrasonik(int a) {
2
          digitalWrite (trigPin[a], LOW);
3
          delayMicroseconds (2);
4
          digitalWrite (trigPin[a], HIGH);
5
          delayMicroseconds(5);
6
          jarak[a] = pulseIn (echoPin[a], HIGH);
7
          hasil[a] = ((jarak[a])/2)/29.1;
8
          Serial.print("Sensor ");
9
          Serial.print(a);
          Serial.print(": ");
10
11
          Serial.print(hasil[a]);
12
          Serial.println(" cm");
13
       }
```

Pada tabel 5.19 dilakukan inisialisasi terhadap ultrasonik bahwa merupakan bilangan bulat. *Hasil* [a] merupakan hasil dari perhitungan nilai jarak antara sensor dengan objek, nama a merupakan variabel yang kemudian bisa dipanggil dengan *hasil* [0] yang merupakan hasil dari sensor bagian kiri, *hasil* [1] yang merupakan hasil dari sensor bagian kenan.

2. Implementasi Pembuatan Task Tiap Sensor HC-SR04

Setelah *task* dibuat dan diatur kemudian dilakukan pengisian *task* tersebut dengan kode program yang sudah ditentukan untuk sensor Ultrasonik HC-SR04, Modul sensor Photodioda, buzzer, dan Modul L298N *Driver* Motor. Pada sensor Ultrasonik dan Photodioda menampilkan nilai hasil akuisisi data *sensing*. Potongan program *TaskUltrasonikKiri()* ditampilkan pada tabel 5.20.

Tabel 5.20 Potongan Program Task UltrasonikKiri

```
RTOS Robot.ino
     .....//...../
     void TaskKiri(void *pvParameters) {
2
        (void) pvParameters;
3
       Serial.begin (9600);
4
       for (;;)
5
       sensorsekarang = 0;
6
7
       ultrasonik (sensorsekarang);
8
     vTaskDelay(3);
9
```

Pada tabel 5.20 di atas adalah *task* yang digunakan untuk sensor Ultrasonik HC-SR04 sebelah kiri. Sensor kiri di inisialisasikan dengan nama 0 dari hasil *looping* yang telah di lakukan pada potongan program pada tabel 5.17. Pada potongan program di atas dibuat parameter dari perhitungan yang dilakukan pada tabel 5.19. Hasil dari pembacaan sensor akan ditampilkan pada serial monitor dengan *baudrate* 9600 dan diberi keterangan sensor kiri berapa cm. Kemudian Potongan program untuk *TaskUltrasonikDepan()* ditampilkan pada tabel 5.21.

Tabel 5.21 Potongan Program Task Ultrasonik Depan

```
RTOS Robot.ino
1
      void TaskDepan(void *pvParameters) {
2
        (void) pvParameters;
3
          Serial.begin (9600);
4
        for (;;)
5
6
        sensorsekarang = 1;
7
        ultrasonik(sensorsekarang);
8
      vTaskDelay(3);
9
        } }
```

Pada tabel 5.21 di atas adalah *task* yang digunakan untuk sensor Ultrasonik HC-SR04 sebelah depan. Sensor depan di inisialisasikan dengan nama 1 dari hasil *looping* yang telah di lakukan pada potongan program pada tabel 5.17. Pada potongan program di atas dibuat parameter dari perhitungan yang dilakukan pada tabel 5.19. Hasil dari pembacaan sensor akan ditampilkan pada serial monitor dengan *baudrate* 9600 dan diberi keterangan sensor kiri berapa cm. Potongan program untuk *TaskUltrasonikKanan()* ditampilkan pada tabel 5.22.

Tabel 5.22 Potongan Program Task UltrasonikKanan

Pada tabel 5.22 di atas adalah *task* yang digunakan untuk sensor Ultrasonik HC-SR04 sebelah kanan. Sensor depan di inisialisasikan dengan nama 2 dari hasil *looping* yang telah di lakukan pada potongan program pada tabel 5.17. Pada potongan program di atas dibuat parameter dari perhitungan yang dilakukan pada tabel 5.19. Hasil dari pembacaan sensor akan ditampilkan pada serial monitor dengan *baudrate* 9600 dan diberi keterangan sensor kiri berapa cm.

3. Implementasi Pembuatan Task Photodioda

Pada *TaskPhoto()* di dalamnya ada program modul sensor photodioda untuk mendeteksi warna gelap dan terang. Potongan program untuk *TaskPhoto()* ditampilkan pada tabel 5.23.

Tabel 5.23 Potongan Program Task Photodioda dan Buzzer

```
RTOS_Robot.ino

void TaskPhoto(void *photo) {
   (void) photo;
   for (;;)
   {
    flame = digitalRead(FlamePin);
    vTaskDelay(3);
   }
}
```

Pada tabel 5.23 di atas variabel flame sebagai hasil pembacaan dari sensor photodioda. Di dalam *TaskPhoto* tidak dilakukan perhitungan karena hanya membaca kondisi HIGH atau LOW.

4. Implementasi Pembuatan Task Motor

Pada *TaskMotor()* mempunyai prioritas 2 pada program yaitu setelah sensor. Dan merupakan *output* dari program. Potongan program untuk *TaskMotor()* ditampilkan pada tabel 5.24.

Tabel 5.24 Potongan Program Task Motor

```
RTOS Robot.ino
      void TaskMotor(void *motor)
1
2
3
         (void) motor;
4
         for (;;)
5
6
       if(flame==LOW) {
7
      if(hasil[1] >8){
8
         if(hasil[0] > 7 \&\& hasil[0] < 20)
9
10
           analogWrite(motor1_kiri, 90);
           analogWrite(motor1_kanan, 90);
11
12
           digitalWrite ( motor1 in1, 255);
13
           digitalWrite( motor1 in2, 0 );
14
           digitalWrite( motor1_in3, 0);
15
           digitalWrite( motor1 in4, 255 )
           digitalWrite(Buzzer, HIGH);
16
17
                   }
         if(hasil[0] >= 20)
18
19
                 {
20
           analogWrite(motor1 kiri, 70);
21
           analogWrite(motor1 kanan, 130);
22
           digitalWrite( motor1 in1, 255);
23
           digitalWrite( motor1 in2, 0 );
24
           digitalWrite( motor1 in3, 0);
25
           digitalWrite( motor1 in4, 255 );
26
           digitalWrite(Buzzer, HIGH);
27
28
         if(hasil[0] < 7)
29
30
           analogWrite(motor1 kiri, 130);
31
           analogWrite(motor1 kanan, 70);
           digitalWrite( motor1 in1, 255);
32
33
           digitalWrite( motor1 in2, 0 );
34
           digitalWrite( motor1 in3, 0);
35
           digitalWrite( motor1 in4, 255 );
36
           digitalWrite(Buzzer, HIGH);
37
       }else {
         if(hasil[0] <=20 && hasil[2] >20)
38
           analogWrite( motor1_kiri , 90);
analogWrite( motor1_kanan , 90);
39
           digitalWrite( motor1_in1, 255);
digitalWrite( motor1_in2, 0 );
digitalWrite( motor1_in3, 255);
digitalWrite( motor1_in4, 0 );
40
41
42
43
           Serial.println("Robot belok kanan");
44
           Serial.println("Belum Menemukan Garis finish");
```

```
45
          digitalWrite(Buzzer, HIGH);
46
          delay(800);
47
48
        if(hasil[0] >20 && hasil[2] >20)
49
50
          analogWrite( motor1_kiri , 90);
51
          analogWrite( motor1 kanan , 90);
52
          digitalWrite( motor1 in1, 0);
53
          digitalWrite( motor1 in2, 255 );
54
          digitalWrite( motor1 in3, 0);
55
          digitalWrite( motor1 in4, 255 );
          Serial.println("Robot belok kiri ");
56
57
          Serial.println("Belum Menemukan Garis finish");
58
          digitalWrite(Buzzer, HIGH);
59
          delay(800);
60
61
        if(hasil[2]
                     <=20 && hasil[0] >20)
62
63
          analogWrite( motor1_kiri , 90);
          analogWrite( motor1 kanan , 90);
64
65
          digitalWrite( motor1 in1, 0);
66
          digitalWrite(motor1 in2, 255);
          digitalWrite( motor1 in3, 0);
67
          digitalWrite( motor1_in4, 255 );
68
          Serial.println("Robot belok kiri ");
69
          Serial.println("Belum Menemukan Garis finish");
70
71
          digitalWrite(Buzzer, HIGH);
          delay(800);
72
73
74
        if(hasil[0] <20
                           && hasil[2] <=20)
75
          analogWrite( motor1_kiri , 90);
76
          analogWrite( motor1_kanan , 90);
77
          digitalWrite( motor1_in1, 255);
digitalWrite( motor1_in2, 0 );
78
79
          digitalWrite( motor1 in3, 255);
80
          digitalWrite( motor1 in4, 0 );
81
82
          Serial.println("Robot putar balik");
83
          Serial.println("Belum Menemukan Garis finish");
84
          digitalWrite(Buzzer, HIGH);
85
          delay(1200);
86
87
88
      }else if(flame==HIGH){
89
          analogWrite ( motor1_kiri ,
90
          analogWrite ( motorl_kanan ,
                                         0);
          digitalWrite( motorl_in1,
91
                                         0);
          digitalWrite( motor1_in2,
92
                                         0);
          digitalWrite( motor1_in3,
93
                                         0);
94
          digitalWrite( motor1_in4,
                                         0);
95
          Serial.println("Robot Berhenti");
96
          Serial.println("Menemukan Garis finish");
97
          digitalWrite(Buzzer, LOW);
98
99
          vTaskDelay(3);
100
101
```

Pada tabel 5.24 yaitu TaskMotor(), yang dilakukan pertama kali adalah apakah kondisi dari sensor photodioda LOW atau HIGH. Jika kondisi sensor photodioda LOW atau mendeteksi garis berwarna terang maka akan menuju pengecekan apakah jarak di depan robot lebih dari 8 cm. Jika kondisi tersebut terpenuhi maka di dalamnya dilakukan pengecekan jika jarak kiri lebih dari 7 cm dan kurang dari 20 cm robot jalan lurus, bila jarak dinding kiri lebih dari atau sama dengan 20 cm maka robot belok kiri, bila jarak kiri kurang dari atau sama dengan 7 cm maka robot belok sedikit ke arah kanan supaya robot tetap berada di tengah labirin. Kondisi kedua yaitu apabila jarak dinding kiri kurang dari atau sama dengan 20 cm, jarak dinding kanan lebih dari 20 cm, dan jarak dinding depan kurang dari atau sama dengan 8 maka robot belok kanan sebesar 90 derajat dengan cara PWM Motor DC kiri bernilai 90 ke arah depan dan nilai PWM Motor DC kanan bernilai 90 ke arah belakang dengan jeda 0,8 detik. Kondisi ketiga yaitu apabila jarak dinding kiri lebih dari 20 cm, jarak dinding kanan lebih dari 20 cm, dan jarak dinding depan kurang atau sama dengan 8 cm maka robot belok kiri sebesar 90 derajat dengan cara PWM Motor DC kiri bernilai 90 ke arah belakang dan nilai PWM Motor DC kanan 90 ke arah depan dengan jeda 0,8 detik. Kondisi ke empat adalah apabila jarak dinding kiri lebih dari 20 cm, jarak dinding kanan kurang dari atau sama dengan 20 cm, dan jarak dinding depan kurang dari atau sama dengan 8 cm maka robot belok kiri sebesar 90 derajat dengan cara PWM Motor DC kiri bernilai 90 ke arah belakang dan nilai PWM Motor DC kanan 90 ke arah depan dengan jeda 0,8 detik. Kondisi ke lima adalah apabila jarak dinding kiri kurang dari 20 cm, jarak dinding depan kurang dari atau sama dengan 8 cm, jarak dinding kanan kurang dari 20 cm maka robot putar balik haluan kanan sebesar 180 derajat dengan cara nilai PWM Motor DC kiri 90 ke arah depan dan nilai PWM Motor DC kanan bernilai 90 ke arah depan dengan jeda 1,2 detik. Jika kondisi sensor photodioda HIGH atau mendeteksi garis berwarna gelap robot langsung berhenti dan buzzer berbunyi.

Untuk penggabungan *source code* sistem yang menggunakan *Free* RTOS ditampilkan pada tabel 5.25.

Tabel 5.25 Source Code Sistem dengan Free RTOS

```
RTOS Robot.ino
      #include <Arduino FreeRTOS.h>
1
      int Buzzer=10;
3
      int FlamePin = A5;
4
      int flame;
5
      int trigPin[3] = {12, A3, A1};
      int echoPin[3] = {11, A2, A0};
6
7
      int hasil[3];
8
      int jarak[3];
9
       int motor1 kiri = 3;
       int motor1 in1 = 4;
10
       int motor1 in2 = 5;
11
       int motor1 kanan = 9;
12
13
       int motor1 in3 = 7;
       int motor1 in4 = 6;
14
```

```
15
      uint16 t sensorsekarang = 0;
16
      void TaskKiri(void *pvParameters);
17
      void TaskDepan(void *pvParameters);
18
      void TaskKanan(void *pvParameters);
19
      void TaskPhoto(void *pvParameters);
20
      void TaskMotor(void *pvParameters);
21
      void setup() {
22
        Serial.begin(115200);
23
       for (int i = 0; i < 3; i + +) {
24
        pinMode(trigPin[i], OUTPUT);
25
26
       for (int i = 0; i < 3; i + +) {
27
        pinMode(echoPin[i], INPUT);
28
29
        pinMode( motor1 in1 , OUTPUT);
30
        pinMode( motor1 in2 , OUTPUT);
31
        pinMode( motor1_in3 , OUTPUT);
32
        pinMode( motor1 in4 , OUTPUT);
33
        pinMode(motor1 kiri, OUTPUT);
34
        pinMode(motor1 kanan, OUTPUT);
35
        pinMode(Buzzer, OUTPUT);
36
      xTaskCreate(
37
        TaskKiri
38
        , (const portCHAR
39
          128
40
         NULL
        , 1
41
        , NULL);
42
43
      xTaskCreate(
44
        TaskDepan
        , (const portCHAR *)
45
                               "Ultrasonik2"
46
         128
        , NULL
47
        , 1
48
49
        , NULL);
50
      xTaskCreate(
51
        TaskKanan
        , (const portCHAR *) "Ultrasonik3"
52
        , 128
53
54
        , NULL
55
        , 1
56
        , NULL);
57
      xTaskCreate(
58
        TaskPhoto
59
        , (const portCHAR *) "Photodioda"
60
         128
        , NULL
61
        , 1
62
        , NULL);
63
64
       xTaskCreate(
65
        TaskMotor
        , (const portCHAR *) "Motor"
66
67
         128
        , NULL
68
        , 1
69
70
        ,NULL);
71
72
      int ultrasonik(int a){
73
          digitalWrite (trigPin[a], LOW);
```

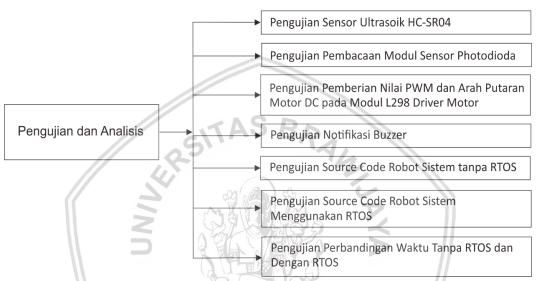
```
delayMicroseconds (2);
74
75
          digitalWrite (trigPin[a], HIGH);
76
          delayMicroseconds(5);
77
          jarak[a] = pulseIn (echoPin[a], HIGH);
78
          hasil[a] = (jarak[a])/29/2;
79
          Serial.print("Sensor ");
80
          Serial.print(a);
81
          Serial.print(": ");
82
          Serial.print(hasil[a]);
83
          Serial.println(" cm");
84
85
      void TaskKiri(void *kiri) {
86
        (void) kiri;
87
        Serial.begin (115200);
88
        for (;;)
89
90
        sensorsekarang = 0;
        ultrasonik(sensorsekarang);
91
92
      vTaskDelay(3);
93
        } }
      void TaskDepan(void *depan) {
94
95
        (void) depan;
96
          Serial.begin (115200);
97
        for (;;)
98
        {
        sensorsekarang = 1;
99
100
        ultrasonik(sensorsekarang);
101
        vTaskDelay(3);
102
        } }
103
      void TaskKanan(void *kanan) {
         (void) kanan;
104
105
          Serial.begin (115200);
106
        for (;;)
107
108
        sensorsekarang = 2;
109
        ultrasonik (sensorsekarang);
110
        vTaskDelay(3);
111
112
      void TaskPhoto(void *photo) {
113
        (void) photo;
114
        for (;;)
115
116
        flame = digitalRead(FlamePin);
117
        vTaskDelay(3);
118
        } }
119
      void TaskMotor(void *motor) // This is a task.
120
121
        (void) motor;
122
        for (;;)
123
        {
124
      if(flame==LOW) {
125
      if(hasil[1] >8){
126
        if (hasil[0] > 7 \&\& hasil[0] < 20)
127
128
          analogWrite(motor1 kiri, 90);
          analogWrite(motor1_kanan, 90);
129
130
          digitalWrite( motor1 in1, 255);
131
          digitalWrite( motor1_in2, 0 );
132
          digitalWrite( motor1_in3, 0);
```

```
digitalWrite ( motor1_in4, 255 );
133
134
          digitalWrite(Buzzer, HIGH);
135
        if(hasil[0] >=20)
136
137
                    {
138
          analogWrite(motor1_kiri, 70);
139
          analogWrite(motor1 kanan, 120);
140
          digitalWrite( motor1 in1, 255);
141
          digitalWrite( motor1 in2, 0 );
142
          digitalWrite( motor1 in3, 0);
143
          digitalWrite( motor1 in4, 255 );
144
          digitalWrite(Buzzer, HIGH);
145
146
        if(hasil[0] < 7)
147
                    {
148
          analogWrite(motor1 kiri, 120);
149
          analogWrite(motor1 kanan, 70);
150
          digitalWrite (motor1 in1, 255);
          digitalWrite( motor1 in2, 0 );
151
          digitalWrite( motor1 in3, 0);
152
153
          digitalWrite( motor1 in4, 255 );
          digitalWrite(Buzzer, HIGH);
154
155
          } }
156
      else
157
158
        if(hasil[0]
                      <=20 && hasil[2] >20)
159
                    {
                        MI
          analogWrite( motor1 kiri , 90);
160
          analogWrite( motor1 kanan , 90);
161
162
          digitalWrite( motor1 in1, 255);
          digitalWrite( motor1 in2, 0 );
163
164
          digitalWrite( motor1 in3, 255);
          digitalWrite( motor1_in4, 0 );
165
          Serial.println("Robot belok kanan ");
166
          Serial.println("Belum Menemukan Garis finish");
167
168
          digitalWrite(Buzzer, HIGH);
169
          delay(700);
170
171
                     <=20 && hasil[2] <=20)
        if(hasil[0]
172
173
          analogWrite( motor1 kiri , 90);
174
          analogWrite( motor1_kanan , 90);
175
          digitalWrite( motor1_in1, 255);
176
          digitalWrite( motor1_in2, 0 );
177
          digitalWrite( motor1_in3, 255);
178
          digitalWrite( motor1_in4, 0 );
179
          Serial.println("Robot putar balik");
180
          Serial.println("Belum Menemukan Garis finish");
181
          digitalWrite(Buzzer, HIGH);
182
          delay(1200);
183
          } } }
184
      else if(flame==HIGH){
185
          analogWrite ( motor1_kiri ,
                                         0);
186
          analogWrite ( motor1 kanan ,
                                         0);
          digitalWrite( motor1_in1,
                                         0);
187
188
          digitalWrite( motor1 in2,
                                         0);
189
          digitalWrite( motor1 in3,
                                         0);
190
                                         0);
          digitalWrite( motor1 in4,
191
          Serial.println("Robot Berhenti");
```



BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai pengujian dan analisis dari hasil pengujian yang telah dilakukan berdasarkan sistem yang sudah dibuat. Pengujian dan analisis bertujuan untuk mengetahui apakah sistem yang sudah dibuat sesuai dengan analisis kebutuhan yang diinginkan. Pengujian terhadap sistem dilakukan dalam beberapa tahapan dan lebih dari 3 kali pengujian supaya mendapatkan hasil yang akurat. Sedangkan analisis dilakukan untuk membuat kesimpulan dari pengujian yang sudah dilakukan.



Gambar 6.1 Grafik Pengujian dan Analisis

6.1 Pengujian Pembacaan Sensor Ultrasonik HC-SR04

Di sistem ini, sensor Ultrasonik HC-SR04 digunakan untuk mendeteksi dan menghitung jarak antara dinding dengan sensor. Sehingga pada pengujian Sensor HC-SR04 ini diperlukan adanya dinding atau objek untuk melihat dari hasil pembacaan sensor HC-SR04.

6.1.1 Tujuan Pengujian

Untuk mengetahui tingkat keakuratan dari ketiga sensor HC-SR04 yang digunakan, yaitu sensor bagian kiri, depan dan kanan apakah berfungsi dengan baik atau tidak. Hasil dari pengukuran sensor Ultrasonik HC-SR04 apakah sama dengan pengukuran menggunakan penggaris yang digunakan dalam kehidupan nyata dalam satuan centimeter. Menggunakan satuan centimeter karena di dalam source code pada penghitungan sensor ultrasonik di konversikan ke cm.

6.1.2 Prosedur Pengujian

Untuk menguji tingkat keakuratan dari ketiga sensor Ultrasonik HC-SR04 yang digunakan maka sensor akan dilakukan percobaan untuk mengukur jarak yang

dilakukan secara terpisah untuk masing-masing sensor Ultrasonik HC-SR04. Pengujian menggunakan program khusus untuk HC-SR04 yang digunakan untuk mengukur jarak dan menampilkannya pada Serial monitor. Prosedur pengujian sensor HC-SR04 sebagai berikut:

- 1. Menghubungkan prototipe alat dengan komputer atau laptop menggunakan kabel USB
- 2. Membuka *source code* yang dibuat untuk mengukur jarak menggunakan sensor HC-SR04. Setelah itu melakukan *compile* dan mengunggah *source code* program ke Arduino Nano.
- Memberi suatu objek padat yang digunakan untuk memantulkan sinyal Ultrasonik dari sensor dan mengamati hasil yang ditampilkan pada serial monitor.
- 4. Melakukan perpindahan pada objek padat tersebut tiap dua centimeter, menyesuaikan perpindahan objek tersebut seperti pengukuran pada penggaris.
- 5. Mengumpulkan data hasil pengukuran tersebut dan menghitung persentase *error* dari setiap sensor dengan persamaan :

6.1.3 Hasil dan Analisis Pengujian

Hasil dari pengujian akurasi dan persentase *error* dari setiap sensor yang digunakan dalam sistem *multitasking* ditampilkan pada tabel 6.1.

Tabel 6.1 Hasil Pengujian Sensor Ultrasonik HC-SR04

Pengujian Ke-	Pengukuran Penggaris	Sensor Yang Diukur (cm)		Error (%)	
Ke-	(cm)	Kiri	Depan	Kanan	
1	4	4	4	4	0
2	6	6	6	6	0
3	8	8	8	8	0
4	10	10	10	10	0
5	12	12	12	12	0
6	14	14	14	14	0
7	16	16	16	16	0
Rata-rata <i>Error</i>					0

Berdasarkan analisis Tabel 6.1, masing-masing sensor mengukur jarak yang sudah ditentukan mulai dari 4 cm hingga 16 cm. Jarak yang diukur dengan penggaris dan pengukuran oleh sensor memiliki hasil yang sama.

Berdasarkan analisis Tabel 6.1, masing-masing sensor memiliki rata-rata *error* 0% yang di mulai dari jarak 4 cm sampai 16 cm. Percobaan dilakukan sebanyak 7 kali yaitu dimulai dari jarak 4 cm kemudian setiap pengujian selanjutnya dilakukan

penambahan 2 cm, maka pengukuran sampai dengan 16 cm. Dengan percobaan sebanyak 7 kali dapat disimpulkan sensor yang dipakai untuk sistem multi-sensor bekerja dengan sangat baik untuk pemrosesan sistem. Jika sensor memiliki ratarata persentase *error* 0% maka kinerja dari sistem di bagian *output* sistem bisa dikatakan bagus dan akurat.

6.2 Pengujian Pembacaan Modul Sensor Photodioda

Di sistem ini, sensor photodioda digunakan untuk mendeteksi alas atau dasar labirin yang berwarna gelap atau terang. Pada pengujian sensor photodioda ini diperlukan adanya objek yang berwarna terang dan gelap untuk melihat dari hasil pembacaan sensor photodioda.

6.2.1 Tujuan Pengujian

Untuk mengetahui tingkat keakuratan dari modul sensor Ultrasonik dalam mendeteksi warna yang berwarna gelap dan terang.

6.2.2 Prosedur Pengujian

Untuk menguji tingkat keakuratan dari modul sensor photodioda akan dilakukan percobaan untuk mendeteksi objek yang berwarna merah, putih, hitam, coklat, biru dan hijau. Pengujian menggunakan program khusus untuk modul sensor photodioda yang digunakan untuk mendeteksi warna gelap dan terang dan menampilkannya pada Serial monitor. Prosedur pengujian sensor photodioda sebagai berikut.

- 1. Menghubungkan prototipe alat dengan PC menggunakan kabel USB.
- 2. Membuka *source code* yang dibuat untuk mendeteksi warna menggunakan sensor photodioda. Setelah itu melakukan *compile* dan mengunggah *source code* program ke Arduino Nano.
- Memberi suatu objek berwarna yang berbeda-beda dan mengamati hasil yang ditampilkan pada serial monitor.
- 4. Membandingkan apakah keterangan yang ditampilkan pada serial monitor Arduino IDE dan kenyataan warna objek sama.

6.2.3 Hasil dan Analisis Pengujian

Hasil dari pengujian tingkat akurasi keberhasilan dari sensor photodioda yang digunakan dalam sistem ditampilkan pada tabel 6.2.

Tabel 6.2 Hasil Pengujian Sensor Photodioda

Pengujian Ke-	Warna Objek	Nilai <i>Output</i> pembacaan sensor	Pembacaan Sensor Photodioda	Error (%)
1	Merah	150	Warna Terang	0
2	Putih	145	Warna Terang	0
3	Hitam	1023	Warna Gelap	0
4	Coklat	152	Warna Terang	0
5	Abu-abu	148	Warna Terang	0
6	Hijau	151	Warna Terang	0
	0			

Berdasarkan analisis Tabel 6.2 Modul sensor photodioda digunakan untuk membaca warna yang berbeda yaitu sebanyak 6 warna. Dan dalam hasil pembacaan dibuat dengan dua kategori yaitu warna terang dan gelap. Dari hasil pembacaan tersebut dapat dilihat jika warna gelap mempunyai nilai *output* sensor di atas 900 dan kategori terang di bawah 900. Dari pengujian yang sudah dilakukan sensor photodioda memiliki rata-rata *error* 0%. Maka sensor photodioda yang digunakan dalam sistem memiliki tingkat keakuratan yang sangat bagus.

6.3 Pengujian Pemberian Nilai PWM dan arah Putaran Motor DC pada Modul L298N *Driver* Motor

Di sistem ini, Modul L298N *Driver* Motor digunakan untuk menentukan kecepatan dan arah putaran dari kedua Motor DC. Pada pengujian Modul L298N *Driver* Motor dilakukan uji coba untuk robot berbelok kiri, jalan lurus, belok kanan, putar balik, dan berhenti.

6.3.1 Tujuan Pengujian

Untuk mengetahui tingkat keakuratan dari Modul L298N *Driver* Motor saat melakukan belok kiri, jalan lurus, belok kanan, putar balik dan berhenti. Apakah salah satu dari kedua Motor DC kecepatannya berkurang, bertambah, atau tetap saat kondisi tertentu.

6.3.2 Prosedur Pengujian

Untuk menguji tingkat keakuratan dari Modul L298N *Driver* Motor akan dilakukan percobaan untuk belok kiri, kanan, jalan lurus, putar balik, dan berhenti. Pengujian menggunakan program khusus untuk Modul L298N *Driver* Motor yang digunakan untuk melakukan kondisi yang ditentukan menampilkannya pada Serial monitor. Prosedur pengujian Modul L298N *Driver* Motor sebagai berikut.

1. Menghubungkan prototipe alat dengan komputer atau laptop menggunakan kabel USB.

- 2. Membuka *source code* yang dibuat untuk Modul L298N *Driver* Motor. Setelah itu melakukan *compile* dan mengunggah *source code* program ke Arduino Nano.
- 3. Membuat kondisi Modul L298N *Driver* Motor untuk berbelok kiri, jalan lurus, belok kanan, putar balik ,dan berhenti.
- 4. Mengamati pergerakan dari Motor DC kiri dan kanan.

6.3.3 Hasil dan Analisis Pengujian

Hasil dari pengujian tingkat akurasi keberhasilan dari Modul L298N *Driver* Motor yang digunakan dalam sistem ditampilkan pada tabel 6.3.

Tabel 6.3 Hasil Pengujian Modul L298N Driver Motor

Pengujian Ke-	Kondisi	Keterangan Putaran Motor DC		Keterangan arah Putaran Motor DC	
ile.		Kiri	Kanan	kiri	kanan
1	Belok Kiri	Berputar	Berputar	Ke belakang	Ke depan
2	Jalan Lurus	Berputar	Berputar	Ke depan	Ke depan
3	Belok Kanan	Berputar	Berputar	Ke depan	Ke belakang
4	Putar Balik	Berputar	Berputar	Ke depan	Ke belakang
5	Berhenti	Diam	Diam	Diam	Diam

Berdasarkan analisis Tabel 6.3 Modul L298N *Driver* Motor memberikan nilai PWM untuk Motor DC kiri dan kanan sama namun arah putaran dari motor DC berbeda sesuai kondisi yang sedang berlangsung. Dari tabel dilihat hasil yang sesuai dengan program yang dijalankan, maka Modul L298N *Driver* Motor bisa dikatakan akurat dalam menggerakkan kedua Motor DC. Hasil tersebut didapatkan apabila sumber tegangan menuju Modul bernilai 9V. Digunakan PWM maksimal 90 karena supaya robot dapat berjalan dengan kecepatan yang pas dan tidak terlalu cepat sehingga saat dilakukan perubahan nilai PWM maka berputarnya robot akurat dan tidak menabrak dinding labirin.

6.4 Pengujian Notifikasi Buzzer

Di sistem ini, buzzer digunakan sebagai notifikasi jika sensor photodioda telah mendeteksi warna gelap. Apabila sensor photodioda mendeteksi warna terang maka buzzer diam dan apabila sensor photodioda mendeteksi warna gelap maka buzzer akan berbunyi terus.

6.4.1 Tujuan Pengujian

Untuk mengetahui tingkat keakuratan dari buzzer dalam berbunyi saat sensor Photodioda mendeteksi warna yang berwarna gelap. Dan diam saat photodioda mendeteksi warna yang berwarna terang.

6.4.2 Prosedur Pengujian

Untuk menguji tingkat keakuratan dari buzzer akan dilakukan percobaan untuk mendeteksi objek yang berwarna hitam dan putih. Pengujian menggunakan program khusus untuk modul sensor photodioda dan buzzer yang digunakan untuk membuat buzzer menyala atau mati dan menampilkannya pada Serial monitor. Prosedur pengujian buzzer sebagai berikut.

- 1. Menghubungkan prototipe alat dengan komputer atau laptop menggunakan kabel USB.
- 2. Membuka *source code* yang dibuat untuk mendeteksi warna menggunakan sensor photodioda dan untuk menyalakan buzzer. Setelah itu melakukan *compile* dan mengunggah *source code* program ke Arduino Nano.
- 3. Memberi suatu objek berwarna gelap dan terang dan mengamati yang terjadi pada buzzer menyala atau mati.
- 4. Membandingkan apakah kondisi buzzer sesuai dengan warna objek.

6.4.3 Hasil dan Analisis Pengujian

Hasil dari pengujian tingkat akurasi keberhasilan dari buzzer yang digunakan dalam sistem ditampilkan pada tabel 6.4.

Tabel 6.4 Hasil Pengujian Buzzer

Pengujian Ke-	Warna Objek	Kondisi Buzzer
1	Putih	Diam
2	Hitam	Berbunyi
3	Merah —	Diam
4	Cokelat	Diam
5	Abu-abu	Diam
6	Hijau	Diam

Berdasarkan analisis Tabel 6.4 buzzer akan berbunyi apabila sensor Photodioda mendeteksi warna terang. Dibuat 6 sampel warna yang kelihatan tingkat perbedaan warna gelap dan terangnya yaitu putih dan hitam . dari hasil percobaan tersebut sangat akurat, saat Photodioda mendeteksi warna terang maka dalam kondisi diam, dan saat mendeteksi warna gelap buzzer berbunyi.

6.5 Pengujian Arah Gerak Robot pada Sistem Robot *Maze Wall Follower* tanpa *Free* RTOS dan menenggunakan *Free* RTOS

Dari semua source code yang digunakan pada komponen Robot Maze Wall Follower kemudian digabungkan menjadi sebuah sistem jadi. Pada sistem yang sudah jadi ini belum dilakukan pengujian untuk sistem tanpa RTOS dan dengan RTOS yang merupakan untuk mempercepat multitasking sistem. Pada pengujian

ini akan dilakukan untuk mengetahui pergerakan robot saat menemui jalan lurus, pertigaan belok kiri, pertigaan robot belok kanan, perempatan, jalan buntu, dan menemukan garis *finish*. Untuk masing-masing pengujian dilakukan sebanyak tiga kali supaya mendapatkan hasil yang akurat, kecuali saat pengujian untuk menemukan garis *finish* dilakukan sekali saja sudah cukup karena tempat dari garis *finish* dipindah-pindah.

6.5.1 Tujuan Pengujian

Untuk mengetahui tingkat keakuratan dari robot saat jalan lurus, belok kanan, belok kiri, putar balik, dan berhenti. Dan untuk mengetahui ketika robot sudah menemukan *goal* apakah berhenti dan buzzer berbunyi.

6.5.2 Prosedur Pengujian

Prosedur pengujian yang dilakukan untuk mengamati tingkat keakuratan robot saat melakukan tindakan ketika menemukan pertigaan, perempatan, jalan buntu, belum menemukan garis *finish* dan menemukan garis *finish* adalah sebagai berikut.

- 1. Menghubungkan prototipe alat dengan komputer atau laptop menggunakan kabel USB.
- 2. Membuka *source code* sistem yang digunakan tanpa *Free* RTOS. Setelah itu melakukan *compile* dan mengunggah *source code* programnya ke Arduino Nano.
- 3. Mencabut kabel USB dari PC menuju Arduino Nano dan menghubungkan menuju *power* bank. Serta menghubungkan modul L298N *Driver* motor dengan baterai 9 V.
- 4. Melakukan pengujian sebanyak 3 kali untuk setiap kondisi yang ada dan pengujian satu kali untuk garis *finish*.
- 5. Mengamati pergerakan dari robot.

Pada saat akan melakukan pengujian terhadap sistem yang menggunakan RTOS, maka poin 2 dilakukan perubahan mengunggah source code sistem yang menggunakan RTOS. Untuk tahap sebelum dan sesudahnya sama.

6.5.3 Hasil dan Analisis Pengujian

Hasil dari pengujian tingkat akurasi keberhasilan sistem yang tidak menggunakan *Free* RTOS ditampilkan pada tabel 6.6.

Tabel 6.5 Hasil Pengujian Sistem tanpa Free RTOS dan dengan Free RTOS

Penguji an Ke-	Kondisi Labirin	Pengujian Setiap Poin	Arah Pergerakan robot	Catatan
	Jalan Lurus	1	Lurus	-
1	(Depan >8 cm, Kiri <=20	2	Lurus	-
	cm , Kanan <=20 cm)	3	Lurus	-
	Jalan Perempatan	1	Belok Kiri	-
2	(depan >8 cm, Kiri >20	2	Belok Kiri	-
	cm , Kanan >20 cm)	3	Belok kiri	-
	Jalan Pertigaan Kiri	1	Belok Kiri	-
3	(depan <=8 cm, Kiri >20	2	Belok Kiri	-
cm , Kanar	cm , Kanan <=20 cm)	3	Belok Kiri	-
	Jalan Pertigaan Kanan	1	Belok Kanan	-
4	(depan <=8 cm, Kiri <=20	2	Belok Kanan	-
	cm, Kanan >20 cm)	3	Belok Kanan	-
	Jalan Buntu	1 C ¹ D	Putar Balik	-
5	(depan <=8 cm, Kiri <=20	A 2 5 0	Putar Balik	-
	cm , Kanan <=20 cm)	3	Putar Balik	-
	Garis finish	Di jalan lurus	Berhenti	Garis finish diletakkan di tengah jalan lurus
		Di pertigaan Belok Kiri	Berhenti	Garis finish diletakkan di tengah pertigaan
6		Di pertigaan Belok Kanan	Berhenti	Garis <i>finish</i> diletakkan di tengah pertigaan
		Di perempatan	Berhenti	Garis finish diletakkan di tengah perempatan
		Di jalan buntu	Berhenti	Garis finish diletakkan di tengah jalan buntu

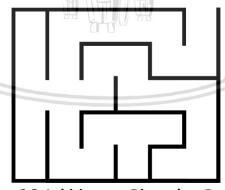
Pada tabel 6.5 adalah hasil dari pengujian untuk sistem yang tidak menggunakan Free RTOS dan menggunakan RTOS, robot melakukan tindakan atau output ketika menemukan kondisi labirin dengan jalan lurus, jalan perempatan, jalan pertigaan belok kiri, jalan pertigaan belok kanan, dan jalan buntu. Untuk setiap percobaan dapat dilihat pada tabel 6.7 yang dilakukan sebanyak 3 kali dan hasilnya robot dapat melakukan tindakan yang sesuai dengan yang diinginkan. Sedangkan untuk pengujian terhadap garis finish dilakukan sebanyak 4 kondisi dan setiap kondisi dilakukan 1 kali pengujian. Untuk garis finish yang diletakkan di jalan lurus yaitu saat kiri dan kanan robot ada dinding dan di bagian depan robot tidak ada dinding, hasilnya robot dapat berhenti saat mendeteksi garis berwarna hitam

dan buzzer berbunyi. untuk pertigaan yang belok kiri yaitu saat kiri tidak ada dinding, kanan ada dinding, dan depan ada dinding, hasilnya robot dapat berhenti saat mendeteksi garis berwarna hitam dan buzzer berbunyi. untuk pertigaan yang belok kanan yaitu saat kiri ada dinding, kanan tidak ada dinding, dan depan ada dinding, hasilnya robot dapat berhenti saat mendeteksi garis berwarna hitam dan buzzer berbunyi. untuk di perempatan yaitu saat kiri tidak ada dinding, kanan tidak ada dinding, dan depan tidak ada dinding, hasilnya robot dapat berhenti saat mendeteksi garis berwarna hitam dan buzzer berbunyi. dan yang terakhir yaitu untuk di jalan buntu yaitu saat kiri ada dinding, kanan ada dinding, dan depan ada dinding, hasilnya robot dapat berhenti saat mendeteksi garis berwarna hitam dan buzzer berbunyi.

Hasil dari pengujian sistem yang menggunakan RTOS dan tidak menggunakan RTOS sama, karena dalam pengujian tersebut hanya untuk menguji apakah robot melakukan tindakan yang sesuai dengan yang diinginkan pada penelitian yang dilakukan.

6.6 Pengujian Perbandingan Waktu *Tanpa* Free RTOS dan Dengan *Free* RTOS

Membandingkan kecepatan Robot *Maze Wall Follower* dalam menyelesaikan labirin dalam menemukan *goal* atau garis *finish*. Dilakukan perpindahan start dan garis *finish* sebanyak 3 kali dengan jarak antara start dan *finish* yang melewati minimal 20 tikungan supaya mendapatkan hasil yang akurat. Penghitungan kecepatan menggunakan *stopwatch* dengan kondisi labirin 5x6 yang telah ditentukan. Untuk bentuk dari labirin dapat dilihat pada gambar 6.2.



Gambar 6.2 Labirin yang Digunakan Pengujian

Pada gambar 6.2 adalah labirin yang digunakan untuk pengujian Robot *Maze Wall Following Left*. Untuk panjang 1 kotak adalah 33 cm, jadi total keliling dari *maze* yang digunakan adalah 2(5x33)+2(6x33) = 728 cm.

6.6.1 Tujuan Pengujian

Untuk mengetahui perbandingan waktu dalam menemukan *goal* dari Robot *Maze Wall Follower* yang menggunakan *Free* RTOS dan tidak menggunakan *Free* RTOS. Dilakukan pengujian penghitungan waktu untuk menyelesaikan labirin dalam menemukan *goal* karena untuk melihat sistem jika menggunakan *Free* RTOS apakah bisa lebih cepat dalam *multitasking* sensor dan *output*.

6.6.2 Prosedur Pengujian

Prosedur pengujian yang dilakukan 2 kali yaitu pertama robot *maze* tanpa *Free* RTOS dan robot *maze* menggunakan *Free* RTOS. Dan untuk setiap kondisi dilakukan sebanyak 3x supaya hasil akurat. Untuk prosedur pengujian Robot *Maze Wall Following Left* tanpa *Free* RTOS adalah sebagai berikut.

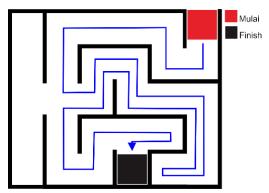
- 1. Menghubungkan prototipe alat dengan komputer atau laptop menggunakan kabel USB
- 2. Membuka *source code* sistem tanpa *Free* RTOS. Setelah itu melakukan compile dan mengunggah *source code* programnya ke Arduino Nano.
- Mencabut kabel USB dari PC menuju Arduino Nano dan menghubungkan menuju power bank. Serta menghubungkan Modul L298N Driver Motor DC dengan baterai 9 V.
- 4. Melakukan pengujian sebanyak 7 kali.
- 5. Menghitung waktu yang dibutuhkan robot dalam menemukan goal.

Untuk prosedur pengujian *Robot Maze Wall Following left* menggunakan *Free* RTOS adalah sebagai berikut.

- Menghubungkan prototipe alat dengan komputer atau laptop menggunakan kabel USB
- Membuka source code sistem yang telah menggunakan Free RTOS. Setelah itu melakukan compile dan mengunggah source code programnya ke Arduino Nano
- Mencabut kabel USB dari PC menuju Arduino Nano dan menghubungkan menuju power bank. Serta menghubungkan Modul L298N Driver Motor DC dengan baterai 9 V.
- 4. Melakukan pengujian sebanyak 7 kali.
- 5. Menghitung waktu yang dibutuhkan robot dalam menemukan *goal*.

6.6.3 Hasil dan Analisis Pengujian

Pengujian pertama dilakukan pada titik *finish* yang diletakkan pada jalan buntu dan robot akan menemukannya setelah melewati 23 tikungan dan 2 jalan buntu.



Gambar 6.3 Labirin yang Digunakan Pengujian Menemukan Goal

Pada gambar 6.3 robot berjalan dari titik berwarna merah menuju *goal* sesuai dengan arah garis panah. Dan ketika mencapai *goal* robot berhenti dan buzzer berbunyi. Jarak antara titik mulai dan *finish* adalah 959 cm. Pengukuran dengan cara jalur yang dilewati oleh robot diberi benang mulai dari start menuju finish, kemudian dilakukan pengukuran dengan penggaris terhadap benang. Untuk hasil pengujian terhadap Robot *Maze Wall Follower* pada sistem yang tidak menggunakan RTOS dan tanpa RTOS ditampilkan pada tabel 6.8.

Tabel 6.6 Hasil Pengujian Menemukan Goal

1 4 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6			
NO.	Waktu yang dibutuhkan (detik)		
Pengujian ke-	Robot Tidak	Robot	
	Menggunakan	Menggunakan	
	Free RTOS	Free RTOS	
1	69,95	65,59	
2	70,23	67,89	
3	79,34	67,75	
4	80,21	66,4	
5	81,21	72,1	
6	70,32	70,3	
7	78,11	68,5	
Rata-rata	75,62429	68,36143	

Tabel 6.6 adalah hasil pengujian terhadap Robot *Maze Wall Follower* sebanyak 7 kali percobaan untuk menemukan *goal* pada setiap sistem yang tidak menggunakan RTOS dan tanpa RTOS. Rata-rata dari robot *maze* tanpa RTOS yaitu 75,62429 detik dan robot *maze* yang menggunakan RTOS yaitu 68,36143 detik. Dari rata-rata tersebut dapat dicari selisihnya, 75,62429-68,36143= 7,262857 detik.

BAB 7 PENUTUP

Bab ini berisi kesimpulan berdasarkan dari pengujian dan analisis yang sudah dilakukan selama proses penelitian dan saran yang berisi hal-hal yang diperlukan dalam melakukan pengembangan sistem untuk topik skripsi selanjutnya.

7.1 Kesimpulan

Berdasarkan pengujian dan Analisis yang dilakukan terhadap tugas akhir ini maka didapatkan kesimpulan sebagai berikut.

- 1. Robot mendeteksi dari keberadaan dinding dengan menggunakan sensor ultrasonik HC-SR04 sebanyak 3 unit yang dipasang pada bagian depan robot pada bagian kiri, depan, dan kanan robot. Untuk setiap sensor diletakkan berdiri dan menghadap ke dinding. Dari hasil pengujian yang sudah dilakukan setiap sensor ultrasonik sebanyak 7 kali dengan jarak yang berbeda-beda memiliki *error 0%*. Sedangkan untuk mendeteksi keberadaan garis *finish* yang berada di alas/ dasar dari labirin menggunakan modul sensor photodioda yang dilakukan pengujian sebanyak 6 kali dengan warna yang berbeda-beda memiliki *error* 0%.
- 2. Metode Simple Maze Wall Follower diimplementasikan dengan cara robot mengikuti dinding bagian kiri pada labirin. Robot akan mengecek dinding kiri ada dinding atau tidak, dan ketika kiri tidak ada dinding maka robot akan langsung belok kiri tanpa memedulikan di depan dan kanan ada dinding atau tidak. Sedangkan untuk menemukan garis finish robot berhenti saat mendeteksi garis berwarna gelap entah itu di jalan lurus, perempatan, pertigaan, dan jalan buntu.
- 3. Implementasi *Free* RTOS ke dalam sistem *multitasking* yaitu dengan cara membuat *task* untuk masing-masing sensor dan modul L298N. Pada sistem digunakan 5 *task* yang memiliki prioritas 1 bertujuan agar sistem memproses setiap *task* dengan bersamaan dan sistem penjadwalan diatur sepenuhnya oleh RTOS. Untuk sensor digunakan sebagai *input* kepada sistem dan Modul L298N *Driver* Motor DC motor digunakan sebagai *output* dari sistem. Dengan menggunakan *library* yang terdapat pada program Arduino IDE.
- 4. Hasil rata-rata waktu untuk menemukan goal pada labirin dengan jarak tempuh 959 cm dan melewati 23 tikungan dan 2 jalan buntu. Robot Maze Wall Follower tanpa RTOS adalah 75,62429 detik dan dengan RTOS adalah 68,36143 detik dengan 7 kali pengulangan. Dan selisih dari hasil rata-rata tersebut adalah 7,262857 detik. Dengan begitu maka RTOS dapat menghemat waktu 7,262857 detik ketika Robot Maze Wall Follower dalam menemukan goal pada labirin pada jarak 959 cm.

7.2 Saran

Berdasarkan pengujian dan Analisis yang sudah dilakukan terhadap tugas akhir ini maka ada beberapa saran yang dapat digunakan untuk pengembangan sistem ini atau implementasi dengan sistem yang lain sebagai berikut:

- 1. Untuk pengembangan sistem selanjutnya diharapkan robot dapat menemukan jalur terdekat dari start menuju *finish* saat kembali ke titik awal mulai robot setelah menemukan *goal*.
- 2. Diharapkan untuk penelitian selanjutnya, sensor photodioda diganti dengan sensor warna, sehingga bisa diimplementasikan saat mendeteksi warna tertentu maka robot melakukan tindakan tertentu.



DAFTAR PUSTAKA

- Anjaswati, I. T. (2013). Sensor Photodioda. Surabaya: UNAIR.
- Arduino. (2006, 04 13). *User Manual*. Arduino Nano (V2.3).
- Arduino. (2018). PWM. Amerika: Arduino.
- Aribowo, T., & Herwanto, H. W. (2011). Kontrol Motor DC Penggerak Roda Pada Robot Cerdas. *Tekno*, 41-45.
- Ariwibowo, T., & Herwanto, H. W. (2011). Kontrol Motor DC Penggerak Roda Pada Robot Cerdas. *Tekno*, 45.
- Boxall, J. (2014). *Tutorial L298N Dual Motor Controller Module 2A and Arduino*. Australia: Tronixlabs Pty Ltd.
- BoxElectronica. (2018). *Proximity Sensor / Obstacles.* Travessa, Portugal: BoxElectronica.
- Chandane, M. P. (2016). RTOS (Real Time Operating System). IJEEE, 361.
- Jones, M. P. (2018). HC-SR04 Ultrasonic Ranging for Arduino. Florida: Assoc.Inc.
- Kingstate. (2011). *Piezo Buzzer*. Tamshui, Taiwan: Kingstate Electronics Corporation.
- Multicomp. (2012). Miniature Motor Model MM10. Indiana: Newark element14.
- Ozdisan. (2018). HC-SR04 Ultrasonik Modul. Intanbul: Ozdisan Elektronik A.S.
- Pratama, A. (2007). Analisis Penerapan Algoritma Backtracking Pada Pencarian Jalan Keluar di Dalam Labirin . *Makalah IF2251 Strategi Algoritmik*, 1-2.
- Saman, & Abdramane. (2013). simple mapping.
- Saputro, F. C., & Haryanto. (2015). Robot Maze Solving Untuk Menyelesaikan Tiga Persimpangan. *Jurnal Ilmiah Go Infotech*, 32-33.
- Shofa, F., & Andrasto, T. (2015). Penerapan Metode Simple Maze Pada Robot Wall Follower Untuk Menyelesaikan Jalur Dalam Menelusuri Sebuah Labirin. Edu Elektrika Journal, 36-37.
- Sulistyowati, R., & Febriantoro, D. D. (2012). Perancangan Prototype Sistem Kontrol Dan Monitoring Pembatas Daya Listrik Berbasis Mikrokontroler. *Jurnal IPTEK Vol 16 No.1*, 27.
- Sutaya, I. W. (2015). Peningkatan Kinerja Perangkat Elektronik Berbasis Mikrokontroler AVR 8 Bit Dengan Menggunakan RTOS (Real Time Operating System). *Peningkatan Kinerja Perangkat Elektronik*, 12.
- Sutrisna, R. Y. (2014). Algoritma Runut-Balik pada Robot Pemadam Api. *Makalah IF2211*, 1-2.
- Udani, A., Marhaeni, & Jampel, N. (2011). Implementasi Teknik Maze Untuk Mengembangkan Kreativitas Dan Kemampuan Kognitif Anak. *Universitas Pendidikan Ganesha Singaraja, Indonesia*, 3-4.

Yultrisna, & Sofiyan, A. (2013). Rancang Bangun Robot Solving Maze Dengan Algoritma Depth First Search. *Momentum*, 87.

