

**IMPLEMENTASI KONTROL *PROPORTIONAL* SEBAGAI
KONTROL PERGERAKAN *MOBILE ROBOT ODOMETRY*
DALAM MENUJU KOORDINAT TARGET**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Mohammad Ali Muhsin
NIM: 145150300111026



**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019**

PENGESAHAN

IMPLEMENTASI KONTROL *PROPORTIONAL* SEBAGAI KONTROL PERGERAKAN
MOBILE ROBOT ODOMETRY DALAM MENUJU KOORDINAT TARGET

SKRIPSI

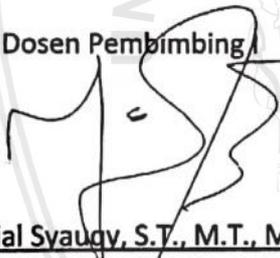
Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Mohammad Ali Muhsin
NIM: 145150300111026

Skripsi ini telah diuji dan dinyatakan lulus pada
18 JULI 2019

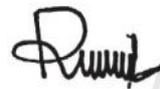
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Dahnial Syaudy, S.T., M.T., M.Sc.
NIK: 201607 870423 1 002

Dosen Pembimbing II



Rizal Maulana, S.T., M.T., M.Sc.
NIK: 201607 891009 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D

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 unsur-unsur 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, 18 Juli 2019



METERAI
TEMPEL
TGL 20
05CB7AEF036653030
6000
ENAM RIBU RUPIAH

Mohammad Ali Muhsin

Mohammad Ali Muhsin

NIM: 145150300111026

KATA PENGANTAR

Dengan menyebut nama Allah Subhanahu Wa Ta'alla yang maha pengasih lagi maha penyayang, puji syukur saya panjatkan atas kehadiran Allah Subhanahu Wa Ta'alla, karena ridha-Nya saya bisa menyelesaikan skripsi ini dan dengan menyelesaikan skripsi ini insya'Allah saya mendapatkan rahmat-Nya. Nabi Muhammad Shalallahu 'Alaihi Wasallam yang telah memberikan sebaik-baiknya contoh dalam meminta pertolongan kepada Allah Subhanahu Wa Ta'alla dan cara melewati segala kesulitan dengan ketenangan. Adapun maksud penyusunan skripsi ini adalah untuk memenuhi salah satu persyaratan dalam memenuhi ujian Sarjan Fakultas Ilmu Komputer dan mengharap rahmat Allah Subhanahu Wa Ta'alla. Judul skripsi yang disusun adalah: "Implementasi Kontrol *Proportional* Sebagai Kontrol Pergerakan *Mobile Robot Odometry* Dalam Menuju Koordinat Target".

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. Bapak Dahnia Syauqy, S.T., M.T., M.Sc. selaku Ketua Program Studi Teknik Komputer dan dosen pembimbing I yang telah banyak memberikan ide dalam penyelesaian skripsi, membimbing dengan sabar dan pengarahan yang sangat bermanfaat dalam penyelesaian skripsi.
2. Bapak Rizal Maulana, S.T., M.T., M.Sc., selaku dosen pembimbing II yang telah banyak memberikan pengarahan dalam menyelesaikan laporan skripsi dan revisi-revisi yang diberikan sangat bermanfaat dalam keilmuan suatu penulisan.
3. Kedua orang tua yang telah meridhai saya, dengan ridhanya maka ridha Allah Subhanahu Wa Ta'alla hadir disetiap proses pengerjaan skripsi saya. Dan juga do'a kedua orang tua yang 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 Informatika Universitas Brawijaya Malang.
7. Seluruh dosen dan karyawan Fakultas Ilmu Komputer Universitas Brawijaya yang telah berperan dalam proses penyelesaian skripsi ini.

8. Fikri Fauzan yang membantu proses skripsi secara berkelompok

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 berguna.

Malang, 18 Juli 2019



Penulis
mohammadalimuhsin@gmail.com

ABSTRAK

Mohammad Ali Muhsin, Implementasi Kontrol *Proportional* Sebagai Kontrol Pergerakan *Mobile Robot Odometry* Dalam Menuju Koordinat Target

Pembimbing: Dahnia Syauqy, S.T., M.T., M.Sc. dan Rizal Maulana, S.T., M.T., M.Sc.

Mobile Robot banyak dikembangkan saat ini dengan salah satu contohnya yaitu *mobile robot odometry* yang dapat merubah posisi dengan hitungan yang rumit. Pada penelitian ini *mobile robot* menggunakan metode odometry dengan kontrol *proportional* memungkinkan robot memperkirakan pergerakan menuju target tujuan. Metode ini sensitiv terhadap kesalahan dan pada penelitian ini digunakan kontrol *Proportional* untuk mengurangi *error* yang ada. *Prototype* ini dibuat menggunakan beberapa komponen yaitu Arduino UNO sebagai pemroses data, sensor encoder dan driver motor L298N. Hasil pengujian dari sensor encoder Fc 03 untuk membaca lubang yang ada pada *disk encoder* diuji sebanyak dua kali yaitu untuk membaca putaran roda maju dan putaran roda mundur. Dari hasil pengujian sensor encoder data yang diperoleh sensor dapat membaca dengan baik *disk encoder* yang diputar. Pada pengujian driver motor terdapat dua data uji dengan masing-masing data diuji sebanyak 5 kali yang didapatkan hasil uji driver motor berhasil 100%. Hasil kontrol *proportional* didapatkan nilai parameter yang sesuai untuk kontrol pergerakan yaitu menggunakan $KP=3.0$. Setelah mendapatkan nilai parameter kontrol *proportional* robot akan bergerak dengan hitungan *odometry* menuju target X dan Y yang telah diberikan melalui *serial monitor IDE Arduino*, jika target sudah benar robot akan berhenti.

Kata kunci: *Mobile Robot, Odometry, Kontrol Proportional*

ABSTRACT

Mohammad Ali Muhsin, *Implementation of Proportional Control As Control of the Movement of Mobile Robot Odometry in Towards Target Coordinates*

Supervisors: Dahniyal Syauqy, S.T., M.T., M.Sc. and Rizal Maulana, S.T., M.T., M.Sc.

Mobile Robot has been developed a lot now with one example, mobile robot odometry which can change positions with complicated calculations. In this study the mobile robot uses the odometry method with proportional control allowing the robot to estimate the movement towards the destination target. This method is sensitive to errors and in this study Proportional controls are used to avoid errors. This prototype is made using several components, namely Arduino UNO as a data processor, encoder sensor and L298N motor driver. The test results from the encoder sensor Fc 03 to read the holes on the disk encoder were tested twice, namely to read the forward wheel rotation and reverse wheel rotation. From the results of testing the encoder sensor the data obtained by the sensor can read well the encoder disk that is playing. In testing the motor driver there are two test data with each data tested 5 times which obtained the test results of the motor driver successfully 100%. Proportional control results obtained the appropriate parameter values for movement control, namely using $KP = 3.0$. After getting the Proportional parameter value the robot will move with a count of odometry towards the X and Y targets that have been given through the Arduino IDE monitor series, if the target is correct the robot will stop.

Keywords: Mobile Robot, Odometry, Proportional Control

DAFTAR ISI

IMPLEMENTASI KONTROL <i>PROPORTIONAL</i> SEBAGAI KONTROL PERGERAKAN <i>MOBILE ROBOT ODOMETRY</i> DALAM MENUJU KOORDINAT TARGET	i
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
<i>ABSTRACT</i>	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	2
1.4 Manfaat	2
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Landasan Kepustakaan	5
2.2 Dasar Teori.....	6
2.2.1 <i>Mobile Robot</i>	6
2.2.2 Arduino Uno	7
2.2.3 Sensor <i>Rotary Encoder</i>	8
2.2.4 <i>Driver Motor DC L298N</i>	9
2.2.5 Motor DC.....	11



2.2.6	Baterai	12
2.2.7	Kontrol <i>Proportional</i>	12
2.2.8	Metode <i>Odometry</i>	13
BAB III METODOLOGI PENELITIAN		14
3.1	Pre Riset.....	14
3.2	Analisis Kebutuhan Sistem	15
3.3	Perancangan Sistem	15
3.4	Implementasi.....	15
3.5	Pengujian dan Analisis.....	16
3.6	Pengambilan Kesimpulan dan Saran.....	16
BAB IV REKAYASA KEBUTUHAN		17
4.1	Gambaran umum sistem	17
4.2	Analisis Kebutuhan Sistem	17
4.3	Kebutuhan fungsional	17
4.4	Kebutuhan non-fungsional	18
4.4.1	Kebutuhan perangkat keras.....	18
4.4.2	Kebutuhan perangkat lunak	18
4.5	Batasan sistem.....	18
BAB V PERANCANGAN DAN IMPLEMENTASI		20
5.1	Perancangan Sistem	20
5.1.1	Gambaran Umum Sistem.....	20
5.1.2	Perancangan <i>Prototype</i> Alat	20
5.1.3	Perancangan Perangkat Keras	22
5.1.4	Perancangan Perangkat Lunak	25
5.2	Implementasi Sistem	30
5.2.1	Implementasi <i>Prototype</i> Alat.....	30
5.2.2	Implementasi Perangkat Keras	31
5.2.3	Implementasi Perangkat Lunak	32
BAB VI PENGUJIAN DAN ANALISIS		37



6.1 Pengujian Sensor Encoder FC 03	37
6.2 Pengujian <i>Driver Motor</i> L298N	39
6.3 Pengujian Kontrol Pada <i>Mobile Robot Odometry</i> Untuk Menuju Koordinat Tujuan	41
6.4 Pengujian Keseluruhan Sistem	49
BAB VII PENUTUP	53
7.1 Kesimpulan	53
7.2 Saran	54
DAFTAR PUSTAKA.....	55



DAFTAR GAMBAR

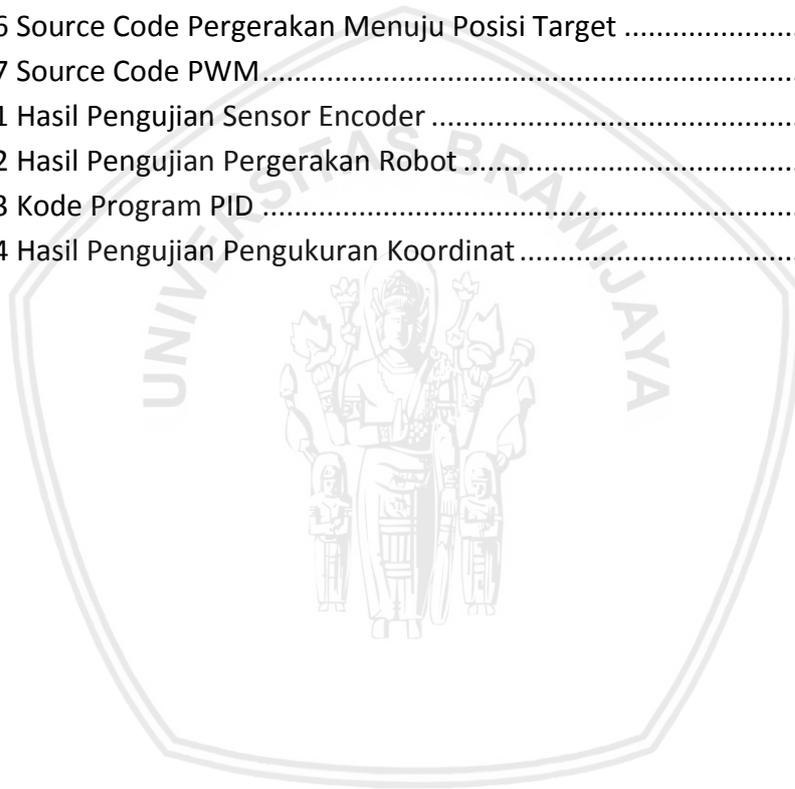
Gambar 2. 1 Mobile Robot.....	6
Gambar 2. 2 Arduino UNO	7
Gambar 2. 3 Skema Rangkaian Arduino Uno	8
Gambar 2. 4 Sensor Rotary Encoder	9
Gambar 2. 5 Driver Motor L298N	9
Gambar 2. 6 Konstruksi Pin IC.....	10
Gambar 2. 7 Simbol dan Bentuk Motor Dc	11
Gambar 2. 8 Baterai	12
Gambar 2. 9 Rumus Dasar Kontrol Proporsional	12
Gambar 2. 10 Blok Diagram Kontrol Proportional	13
Gambar 3. 1 Diagram Alir Metode Penelitian.....	14
Gambar 5. 1 Diagram Blok Sistem.....	20
Gambar 5. 2 Desain Prototype Robot Tampak Atas	21
Gambar 5. 3 Desain Prototype Robot Tampak Bawah	21
Gambar 5. 4 Konfigurasi Driver Motor Dc	22
Gambar 5. 5 Konfigurasi Sensor Encoder.....	23
Gambar 5. 6 Flowchart Akuisisi Data	25
Gambar 5. 7 Flowchart Pembacaan Sensor	26
Gambar 5. 8 Flowchart Kontrol Proportional	27
Gambar 5. 9 Rangkaian Closed Loop Proportional	28
Gambar 5. 10 Flowchart Pergerakan Robot Menuju Target.....	29
Gambar 5. 11 Implementasi Robot Tampak Atas	30
Gambar 5. 12 Implementasi Robot Tampak Bawah	31
Gambar 5. 13 Implementasi Robot Tampak Belakang	31
Gambar 5. 14 Implementasi Robot Tampak Samping Kiri	32
Gambar 6. 1 Bagan Pengujian dan Analisis.....	37
Gambar 6. 2 Pengujian Sensor Encoder.....	38
Gambar 6. 3 Pengujian Pergerakan Robot.....	40
Gambar 6. 4 Pengujian Sistem Kontrol	42
Gambar 6. 5 Pengujian 1 Set Point X Proportional	42
Gambar 6. 6 Pengujian 1 Set Point Y Proportional	43
Gambar 6. 7 Pengujian 2 Set Point X Proportional	43
Gambar 6. 8 Pengujian 2 Set Point Y Proportional	44
Gambar 6. 9 Pengujian 3 Set Point X Proportional	44
Gambar 6. 10 Pengujian 3 Set Point Y Proportional	45
Gambar 6. 11 Pengujian 4 Set Point X PD.....	46

Gambar 6. 12 Pengujian 4 Set Point Y PD	46
Gambar 6. 13 Pengujian 5 Set Point X PI	47
Gambar 6. 14 Pengujian 5 Set Point Y PI.....	47
Gambar 6. 15 Pengujian 6 Set Point X PID	48
Gambar 6. 16 Pengujian 6 Set Point Y PID	48
Gambar 6. 17 Pengujian Pergerakan Robot Menuju Target.....	50



DAFTAR TABEL

Tabel 2. 1 Spesifikasi Arduino Uno.....	7
Tabel 2. 2 Fungsi Pin IC L298	10
Tabel 5. 1 Pin Skematik Driver Motor	23
Tabel 5. 2 Pin Skematik Sensor Encoder	24
Tabel 5. 3 Source Code Akuisisi Data	32
Tabel 5. 4 Source Code Pembacaan Sensor	33
Tabel 5. 5 Source Code Kontrol Pergerakan menggunakan Proportional	34
Tabel 5. 6 Source Code Pergerakan Menuju Posisi Target	35
Tabel 5. 7 Source Code PWM.....	36
Tabel 6. 1 Hasil Pengujian Sensor Encoder	39
Tabel 6. 2 Hasil Pengujian Pergerakan Robot	40
Tabel 6. 3 Kode Program PID	45
Tabel 6. 4 Hasil Pengujian Pengukuran Koordinat	50



BAB I PENDAHULUAN

1.1 Latar Belakang

Mobile robot merupakan salah satu tipe robot yang saat ini banyak dikembangkan oleh para ahli. *Mobile Robot* merupakan sebuah tipe robot yang dapat bergerak menuju tempat satu ke tempat lain (Hartanto, 2005). *Mobile robot* meniru fungsi dari makhluk hidup yang selalu berpindah – pindah tempat, jenis robot seperti ini biasanya dibuat untuk keperluan, seperti: memindahkan barang otomatis, melakukan perjalanan penjelajahan suatu daerah, untuk hiburan (*robotainment*) dan mainan. *Mobile robot* mempunyai beragam kemampuan yang disesuaikan untuk keperluan dan jenis tingkatan pekerjaan tertentu. Dalam bidang industri yang besar robot ini digunakan khusus untuk melakukan pemantauan pada suatu medan yang dianggap berbahaya untuk dijelajahi dan memiliki resiko tinggi bagi keselamatan manusia.

Tantangan yang dihadapi sangat banyak, salah satunya yaitu seberapa jauh *mobile robot* dapat mengatasi gangguan dan diupayakan untuk bisa mencapai sebuah target posisi tertentu. Robot harus mampu mengetahui besarnya gangguan yang dihasilkan dari luar sehingga robot dapat bergerak kembali ke arah yang seharusnya dan berusaha mendekati target posisi yang dituju. Secara teknis, robot tersebut harus bisa mengetahui kemana arah gerak dan jarak tempuhnya untuk menentukan proyeksi lintasan gerak terdekat. Untuk mengetahui arah gerak dan jarak tempuh proyeksi lintasan gerak yang dituju oleh *mobile robot* tersebut biasanya dengan perhitungan metode odometry (Kariyanto, 2011). Data *rotary encoder* digunakan sebagai masukan untuk perhitungan metode odometry. *Rotary encoder* yang terpasang pada roda kanan dan kiri kemudian diproses dengan rumus odometry dan dikonversi ke kinematika *differential drive* sehingga diperoleh data untuk direspon oleh motor kanan dan kiri sehingga *mobile robot* dapat bergerak dan mampu mengetahui posisinya sehingga jarak target saat itu dapat diketahui. Implementasi metode ini sangat tepat dan memberikan banyak keunggulan.

Banyak penelitian yang dilakukan tentang kontrol pergerakan robot secara konvensional. Sedangkan pada kenyataannya suatu robot yang bergerak secara otomatis memiliki banyak tantangan yang dihadapi. Diantaranya adalah sejauh mana robot dapat mengatasi gangguan, dan dituntut dapat mencapai sebuah target koordinat posisi yang telah ditentukan. Banyak *mobile robot* tidak mengetahui dimana posisi robot yang ada sekarang, akibatnya dengan kesalahan yang ada ini kemudian menghasilkan kesalahan pula pada perhitungan jarak tempuhnya (Pasztor, 2010). Pada sistem navigasi robot biasanya dibuat untuk mencapai lokasi tujuan dengan adanya gangguan yang datang dari luar dan robot dapat sampai pada tujuan (Farshad, 2011). Dengan adanya berbagai gangguan yang ada diluar sistem maka diperlukan penambahan kontroler yang mampu menanggulangi gangguan yang ada tersebut. Sistem *proportional* dinilai sangat

cocok untuk sebuah mobile robot karena dengan menggunakan nilai parameter yang ada pada kontrol *proportional* dapat mengurangi *error* yang disebabkan oleh faktor-faktor yang ada diluar sistem. Kontrol *proportional* merupakan sistem yang hanya dapat menghasilkan keluaran apabila terdapat perbedaan data masukan dan data keluaran, artinya sistem ini selalu membutuhkan kesalahan agar dapat memberikan *output*. Pada sistem *mobile robot odometry* ini riskan akan kesalahan yang ada, maka kontrol *proportional* ini memungkinkan dipakai dalam penyelesaian tugas akhir ini. Dengan latar belakang tersebut, diharapkan adanya implementasi sistem *proportional* pada mobile robot menggunakan metode odometri dapat melakukan navigasi untuk mencapai tempat tujuan, dan mampu mengatasi gangguan dari luar yang menyebabkan robot berubah arah agar robot tetap dapat menuju target.

1.2 Rumusan Masalah

Dari hasil permasalahan yang ada pada latar belakang tersebut, terdapat beberapa rumusan masalah untuk untuk penelitian ini yang dituliskan sebagai berikut :

1. Bagaimana cara mengimplementasikan metode odometry yang efektif untuk pada mobile robot?
2. Bagaimana robot dapat melakukan pergerakan agar dapat menuju ketempat target yang sudah ditentukan ?
3. Bagaimana menentukan nilai parameter KP yang sesuai dengan sistem?
4. Bagaimana membuat sistem *Proportional* untuk mengurangi error pada pergerakan mobile robot menuju target?

1.3 Tujuan

Setelah menuliskan rumusan masalah yang ada, dapat dihasilkan tujuan dari penelitian dari skripsi sebagai berikut :

1. Dapat mengimplementasikan metode odometry yang efektif pada *mobile robot*.
2. Dapat melakukan pergerakan ke target yang sebelumnya sudah ditentukan.
3. Dapat menentukan nilai parameter KP yang sesuai dengan sistem.
4. Dapat membuat sistem *proportional* yang sesuai untuk mengurangi error pada pergerakan mobile robot agar menuju ke target tujuan yang diinginkan.

1.4 Manfaat

Dengan adanya penelitian yang telah dilakukan diharapkan tercipta manfaat penelitian sebagai berikut :

1. Membantu menemukan target lokasi dengan navigasi yang telah diberikan.
2. Membantu menemukan jalur untuk menuju ke target yang akan dituju.

3. Membantu memberikan navigasi agar dapat bergerak ke target yang diinginkan.

1.5 Batasan Masalah

Mengacu pada hal-hal lain yang tidak ingin dimasukkan dalam pembahasan penelitian yang telah dilakukan dilakukan, dengan membatasi masalah penelitian sesuai yang ada seperti berikut ini :

1. Robot ini dikembangkan menggunakan arduino.
2. Daerah kerja yang dapat dijangkau oleh robot yaitu medan kering dan rata.
3. Robot ini menggunakan sensor *rotary encoder*.

1.6 Sistematika Pembahasan

Untuk mempermudah penulisan skripsi ini digunakan sistematika pembahasan dengan membagi sub bab sebagai berikut :

BAB I : PENDAHULUAN

Pada sub bab ini berisi penjelasan tentang gambaran umum penelitian yang akan dikerjakan pada skripsi ini yang berjudul "Implementasi kontrol *proportional* sebagai kontrol pergerakan mobile robot *odometry* dalam menuju koordinat target".

BAB II : LANDASAN KEPUSTAKAAN

Pada sub bab ini isinya mengenai kajian pustaka yang dipakai dalam skripsi ini tentang "Implementasi kontrol *proportional* pada mobile robot *odometry* untuk menentukan koordinat tujuan". Penjelasan mengenai seluruh komponen yang dipakai dalam pembuatan robot pada skripsi dituliskan pada bab ini.

BAB III : METODE PENELITIAN

Pada sub bab ini menjabarkan tentang urutan pengerjaan yang ada pada penelitian ini. Seluruh urutan pengerjaan seperti literatur, kebutuhan dari sistem, perancangan dan implemetasi sistem, hasil pengujian dan anlisis penelitian dan kesimpulan ditulis semua disini.

BAB IV : REKAYASA KEBUTUHAN

Pada sub bab ini menjabarkan dari kebutuhan sistem yang digunakan dalam melakukan penelitian skripsi dengan judul Implementasi kontrol *proportional* sebagai kontrol pergerakan mobile robot *odometry* dalam menuju koordinat target.

BAB V : PERANCANGAN DAN IMPLEMENTASI

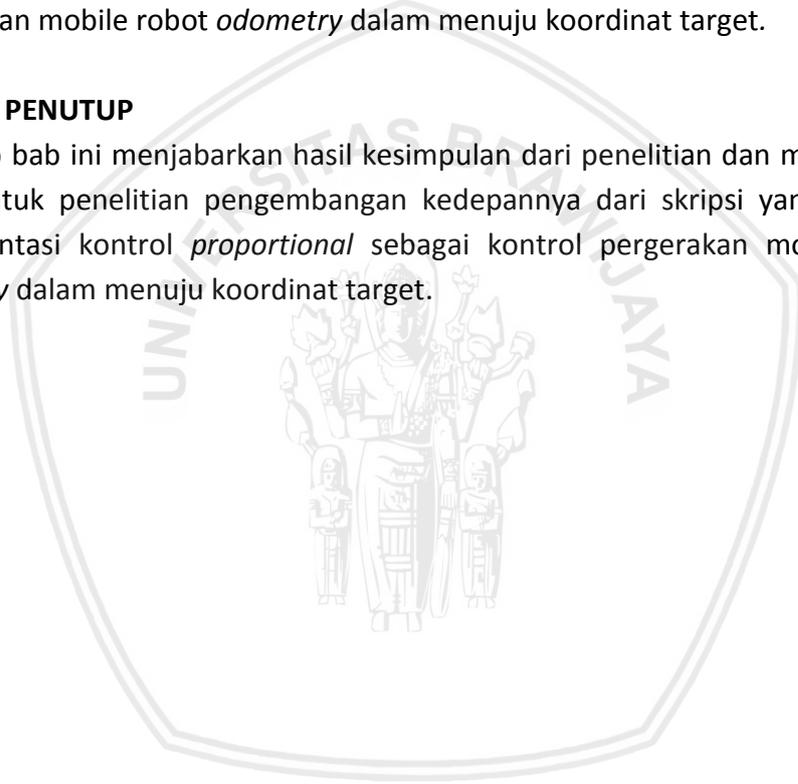
Pada sub bab ini dijabarkan proses perancangan dan implementasi alat mulai dari awal hingga akhir dari penelitian ini yang berjudul Implementasi kontrol *proportional* sebagai kontrol pergerakan mobile robot *odometry* dalam menuju koordinat target.

BAB VI : PENGUJIAN DAN ANALISA

Pada sub bab ini menjabarkan isi tentang hasil pengujian dan analisis dari penelitian yang berjudul Implementasi kontrol *proportional* sebagai kontrol pergerakan mobile robot *odometry* dalam menuju koordinat target.

BAB VII : PENUTUP

Pada sub bab ini menjabarkan hasil kesimpulan dari penelitian dan memberikan saran untuk penelitian pengembangan kedepannya dari skripsi yang berjudul Implementasi kontrol *proportional* sebagai kontrol pergerakan mobile robot *odometry* dalam menuju koordinat target.



BAB II TINJAUAN PUSTAKA

2.1 Landasan Kepustakaan

Untuk dapat memberikan hasil yang diinginkan dalam penelitian ini, dicantumkan beberapa jurnal penelitian yang memiliki kemiripan dari beberapa penelitian terdahulu sebagai sumber referensi dalam melakukan penelitian.

Menurut jurnal yang dikemukakan oleh (Doh, Nakju Lett, Howie Choset dan Wan Kyun Yung, 2006) yang berjudul "*Relative localization using path odometry information*". Semua jenis *mobile robot* dapat mengalami kesalahan lokalisasi. Pendekatan sebelumnya untuk mengakomodasi kesalahan lokalisasi baik menggunakan sensor eksternal seperti laser atau sonar, atau gunakan sensor *internal* seperti *encoders*. Informasi data masukan enkoder terintegrasi untuk mendapatkan posisi robot ini disebut odometri. Kombinasi sensor eksternal dan internal akhirnya digunakan untuk memecahkan masalah kesalahan lokalisasi.

Kemampuan menemukan lokasi dan pembacaan sensor yang akurat penting untuk *mobile robot* otomatis. Secara khusus, *mobile robot* umumnya menggunakan estimasi odometri untuk pemetaan dan lokalisasi. Robot odometri membutuhkan sensor *proprioceptif*, seperti *rotary encoders*, dan parameter kalibrasi sensor sebagai acuan untuk menentukan radius tempuh roda yang efektif. Namun, parameter odometri tersebut memerlukan kalibrasi yang sering karena bergantung pada tekanan ban, keausan, beban robot dan tipe permukaan lantai. Menurut (Pei, Yanming dan Lindsay Kleeman, 2015) yang berjudul "*Robot calibration of sensor poses and region based odometry using offline optimisation of map information*". Kalibrasi pada robot odometri untuk bergerak dan nilai parameter sensor secara signifikan dapat memperbaiki akurasi pemetaan dan pelokalisasian pada robot. Laporan ini menunjukkan kalibrasi yang dilakukan secara terus menerus pada *mobile robot* dan parameter data sensor yang ada pada sistemnya.

Sedangkan menurut (Prakash, D. Vidhya dan A. Elango, 2016) dalam jurnalnya yang berjudul "*Studies on affecting factors of wheel slip and odometry error on the performance of wheeled mobile robots*". *Mobile robot* dapat digunakan untuk melakukan berbagai tugas di berbagai jenis medan. Medan ini adalah bentuk medan datar, tidak dapat diprediksi dan tidak beraturan dari ruang kerja ini yang menuntut tantangan desain dari bentuk robot. Biasanya ada medan yang mungkin terdapat lubang, permukaan medan tidak datar yang menjadi hambatan pada robot. Pada medan yang tidak terprediksi akan ada kesalahan odometri pada slip roda. Robot dapat dirancang dengan mengakomodir prediksi dari lintasan, navigasi, waktu tempuh dan konsumsi energi yang dianggarkan. Masalah teknik ini dapat diatasi dengan desain rotasi roda yang tepat, amplitudo getaran tubuh robot, modulasi waktu perjalanan berbeda dengan kondisi medan (basah, kering, lunak, keras) & efek angin.

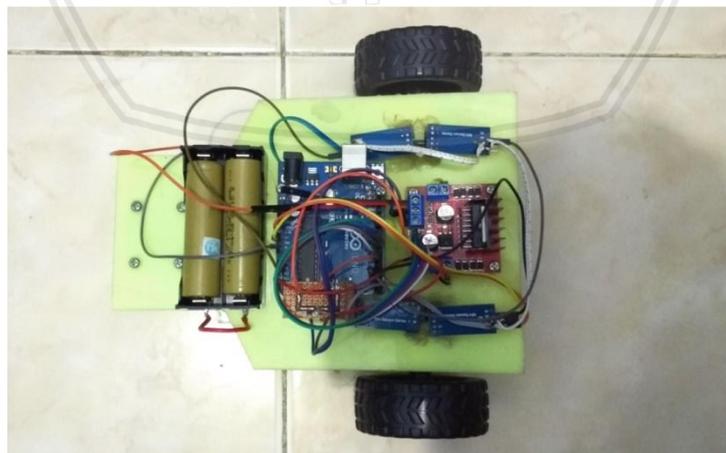
Dengan melihat dari ketiga penelitian yang telah dilakukan di atas kesimpulan yang dihasilkan yaitu kesesuaian metode yang dipakai dalam

penelitian terdahulu, yaitu sama-sama mengimplementasikan metode odometry untuk *Mobile robot*, yang tujuannya adalah membuat pergerakan robot menuju target tujuan dapat sesuai target yang diinginkan. Dari ketiga sistem yang ada tersebut masih ada beberapa hal yang harus ditambahkan. Dengan menambahkan sistem kontrol *proportional* memungkinkan untuk membuat kesalahan yang ada pada sistem sebelumnya dapat dikurangi bahkan dihilangkan. Kelebihan dari sistem *proportional* ini yaitu meminimalisir kesalahan pergerakan robot yang dapat mengakibatkan robot tidak dapat menuju ke target yang telah ditentukan. Agar pada penelitian yang akan dilakukan tidak menghasilkan plagiasi pada penelitian sebelumnya, penulis membuat objek dan judul yang berbeda dengan judul penelitian Implementasi kontrol kontrol *proportional* pada mobile robot *odometry* untuk menentukan koordinat tujuan.

2.2 Dasar Teori

2.2.1 Mobile Robot

Sebuah alat mekanik yang mampu bergerak sendiri dengan pengendali perangkat lunak yang dipengaruhi oleh data yang berasal dari sebuah sensor untuk bergerak atau memindahkan suatu objek disebut robot (Schilling, 2000). Robot dibedakan menjadi beberapa jenis. Dari berbagai jenis robot yang ada salah satunya yaitu *Mobile robot*. Robot ini prinsip kerjanya memakai roda atau kaki untuk dapat bergerak berpindah tempat sesuai tugas yang diberikan pada sistemnya.



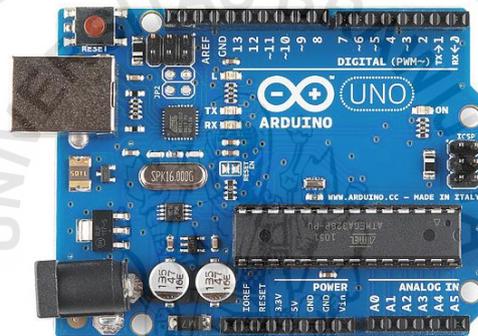
Gambar 2.1 Mobile Robot

Pengelompokan jenis dari medan yang dilalui oleh *mobile robot* dibedakan menjadi empat. Lingkungan tempat bekerja pada *mobile robot* terbagi menjadi tiga macam: robot yang beroperasi di darat contohnya robot beroda, robot yang

beroperasi di udara atau drone dan robot yang beroperasi dibawah air untuk melakukan penyelaman dasar laut. Pada mekanisme pergerakan robot odometry, mekanik yang biasa dipakai yaitu motor Dc dengan masukan data sensor encoder untuk dapat bergerak berpindah lokasi.

2.2.2 Arduino Uno

Arduino Uno adalah sebuah papan mikrokontroler yang bersifat bebas untuk dikembangkan dalam penelitian apapun untuk mengendalikan sebuah *prototype* yang berguna untuk menciptakan dan mempermudah dalam sebuah proyek elektronik dalam berbagai bidang. Prosesor yang digunakan dalam *hardware* ini yaitu Atmel AVR dan untuk bahasa pemrograman pada *softwarena* berupa bahasa pemrograman khusus untuk perangkat ini.



Gambar 2.2 Arduino UNO
(Sumber: robotistan, 2018)

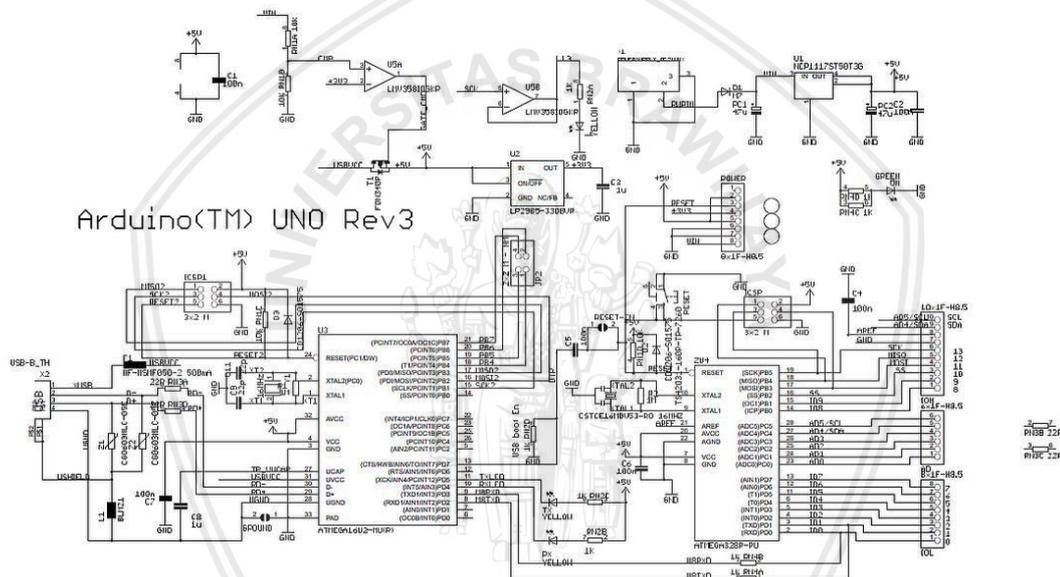
Merk produk Arduino sangat terkenal saat ini. Karena struktur program yang dipakai sangat mudah banyak orang saat ini mempelajari membuat rancangan robot dan rangkaian elektronik moderen dengan produk arduino ini. Untuk mengembangkan sebuah proyek elektronik para profesional juga menggunakan arduino ini dalam mengembangkan projeknya (ecadio.com, 2018).

Tabel 2.1 Spesifikasi Arduino Uno

Chip mikrokontroler	ATmega328P
Tegangan operasi	5V
Tegangan input (yang direkomendasikan, via jack DC)	7V - 12V
Tegangan input (limit, via jack DC)	6V - 20V
Digital I/O pin	14 buah, 6 diantaranya menyediakan PWM
Analog Input pin	6 buah

Arus DC per pin I/O	20 mA
Arus DC pin 3.3V	50 mA
Memori Flash	32 KB, 0.5 KB telah digunakan untuk <i>bootloader</i>
SRAM	2 KB
EEPROM	1 KB
<i>Clock speed</i>	16 Mhz
Dimensi	68.6 mm x 53.4 mm
Berat	25 g

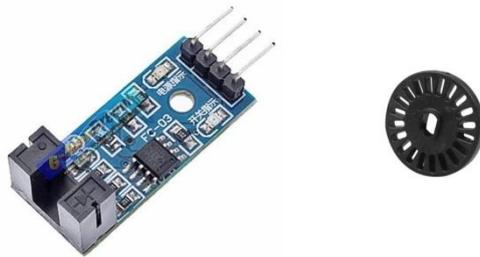
(Sumber: ecadio.com, 2018)



Gambar 2.3 Skema Rangkaian Arduino Uno
(Sumber: pinterest, 2018)

2.2.3 Sensor Rotary Encoder

Untuk dapat memperkirakan gerak dan posisi pergerakan robot pada penelitian ini digunakan sensor *rotary encoder*. Pembacaan posisi ini dilakukan oleh optik yang terpasang pada sensor agar menghasilkan deretan data pulsa yang digunakan untuk menentukan gerakan arah dan posisi pada robot. Nilai sudut dapat diperoleh dari proses pembacaan dari optik tersebut. Informasi tersebut kemudian diproses menjadi nilai digital agar sistem dapat memproses dan diteruskan menjadi menuju kendali mekanik untuk dirubah menjadi gerakan.



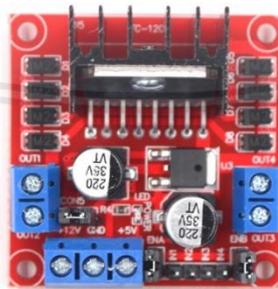
Gambar 2.4 Sensor Rotary Encoder

(Sumber : tokopedia, 2018)

Piringan yang memiliki lubang ini digunakan sebagai masukan data pemrosesan odometri. Prinsip kerjanya setiap cahaya yang dipancarkan optik tidak mengenai penerima optik maka nilai yang dihasilkan = 0. Optik yang disebutkan ini merupakan sebuah photo-transistor yang gunanya sebagai pemancar cahaya diantara lubang encoder. Pengambilan data sensor ini jika dilakukan dengan memutar piringan encoder dengan cepat akan menghasilkan hitungan yang kurang akurat. Perlu dilakukan kalibrasi awal untuk menentukan posisi awal robot agar saat bergerak dapat sesuai dengan menuju target lokasinya.

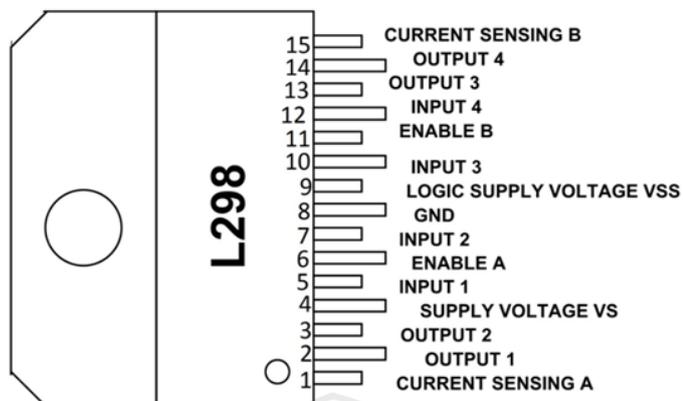
2.2.4 Driver Motor DC L298N

Untuk mengatur dan mengendalikan pergerakan motor DC pada penelitian kali ini digunakan *driver motor DC L298N*. Ruang kerja *driver motor* kali ini digunakan untuk mengendalikan kecepatan motor pada robot saat berpindah posisi. Tegangan yang dipakai saat perangkat ini bekerja yaitu +5V hingga 50V (playground.arduino 2017).



Gambar 2.5 Driver Motor L298N

(Sumber: makerfabs.com, 2018)



Gambar 2.6 Konstruksi Pin IC
(Sumber : components101.com, 2018)

a. Datasheet Pin IC L298

Tabel 2.2 Fungsi Pin IC L298

Nama Pin	Fungsi Kegunaan
Pin Tegangan	
Pin 8, GND	Tersambung pada ground
Pin 4, Tegangan VS	Tersambung pada tegangan +5V
Pin 9, Tegangan <i>logic</i>	Tegangan untuk kontrol logikal
H-Bridge A	
Pin 1, Tegangan Arus <i>H-Bridge A</i>	Tersambung pada ground dan resistor untuk mengontrol keluaran arus
Pin 2, Keluaran 1 Pin 3, Keluaran 2	Keluaran dari <i>H-Bridge A</i>
Pin 5, Masukan 1 Pin 7, Masukan 2	TTL kontrol input <i>H-Bridge A</i>
Pin 6, <i>Enable A</i>	TTL <i>enable input</i>
H-Bridge B	
Pin 15, Tegangan Arus <i>H-Bridge B</i>	Tersambung pada ground dan resistor untuk mengontrol keluaran arus
Pin 13, Keluaran 3 Pin 14, Keluaran 4	Keluaran dari <i>H-Bridge B</i>
Pin 10, Masukan 3 Pin 12, Masukan 4	TTL kontrol input <i>H-Bridge B</i>
Pin 11, <i>Enable B</i>	TTL <i>enable input</i>



b. Fitur *Driver Motor DC L298N*

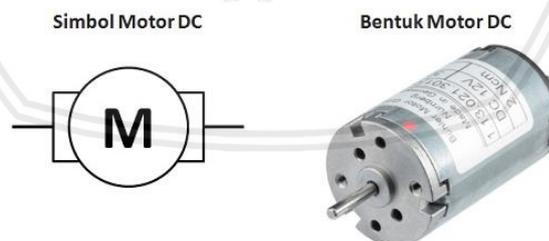
Karakteristik driver motor ini sudah dapat digunakan pada sistem yang akan dibuat pada prototype robot odometri. karakteristik yang dimiliki *driver motor DC IC L298N* sesuai dengan *datasheet* adalah sebagai berikut:

1. Tegangan operasi antara +5V sampai +46V.
2. Tegangan maksimum yang boleh dipakai 50V.
3. Tegangan maksimum *input* dan tegangan *enable* sebesar +7V.
4. TTL kontrol *inputs*.
5. Total disipasi tegangan sebesar 25W.
6. Temperatur pengoperasian antara -23°C sampai 130°C.
7. Temperatur penyimpanan antara -40°C sampai 150°C.

2.2.5 Motor DC

Untuk dapat melakukan pergerakan menuju suatu target tujuan robot membutuhkan sebuah komponen berupa motor DC. Tegangan searah digunakan untuk memfungsikan Motor DC. Penggunaan Motor DC ini pada peralatan elektronik biasanya digunakan untuk kipas DC, bor pcb DC, penggerak robot, dan lainnya.

Saat berputar, putaran per-menit yang dihasilkan oleh motor DC biasa disebut RPM (*Revolutions per minute*) dan arah putaran dari motor DC dapat diatur dengan mengubah polaritas pada tegangan yang diberikan. Spesifikasi dari motor DC yang ada sangat beragam sesuai dengan fungsi yang akan digunakan dalam sistem.



Gambar 2.7 Simbol dan Bentuk Motor DC

(Sumber: Teknik Elektronika, 2018)

Saat melakukan putaran tanpa dipasangi beban, motor DC hanya menggunakan sedikit arus yang dipakai. Jika melakukan putaran dan dipasangi beban maka jumlah arus yang dipakai bisa meningkat sampai ratusan persen tergantung dari nilai beban yang dipasang (Teknik Elektronika, 2018).

2.2.6 Baterai

Baterai merupakan sumber tegangan yang digunakan sebuah perangkat elektronik untuk dapat bekerja sebagaimana fungsinya. Pada tugas akhir kali ini, sumber tegangan yang dipakai adalah baterai 4,2V untuk dapat mengaktifkan komponen elektronik yang dipakai dalam penelitian ini.



Gambar 2.8 Baterai
(Sumber: shopee, 2018)

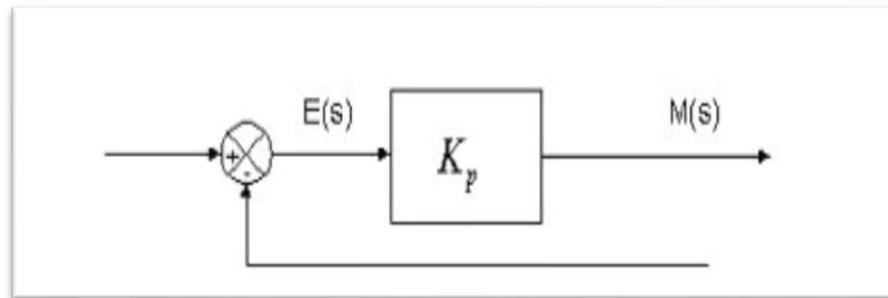
Peralatan elektronik yang menggunakan baterai tidak perlu mendapatkan sumber tegangan langsung dari listrik menggunakan kabel dan dapat dibawa pergi kemana-mana dengan sangat mudah. Jenis baterai ada dua macam, baterai sekali pakai dan baterai isi ulang (Manika. Anki, 2016).

2.2.7 Kontrol *Proportional*

Pengontrol proporsional sebuah kontrol yang memiliki nilai keluaran yang seimbang (proporsional) dengan besarnya nilai sinyal *error* yang ada. Pengontrol P dikatakan secara sederhana merupakan hasil perkalian dari nilai parameter K_p dengan nilai data masukan. Sinyal masukan yang berubah akan mengakibatkan secara langsung sistem memberikan keluaran sinyal sebesar pengalinya.

$$P \rightarrow K_p e(t)$$

Gambar 2.9 Rumus Dasar Kontrol Proporsional
(Sumber: robotics-university, 2015)



Gambar 2.10 Blok Diagram Kontrol *Proportional*

(Sumber: infoelektro, 2008)

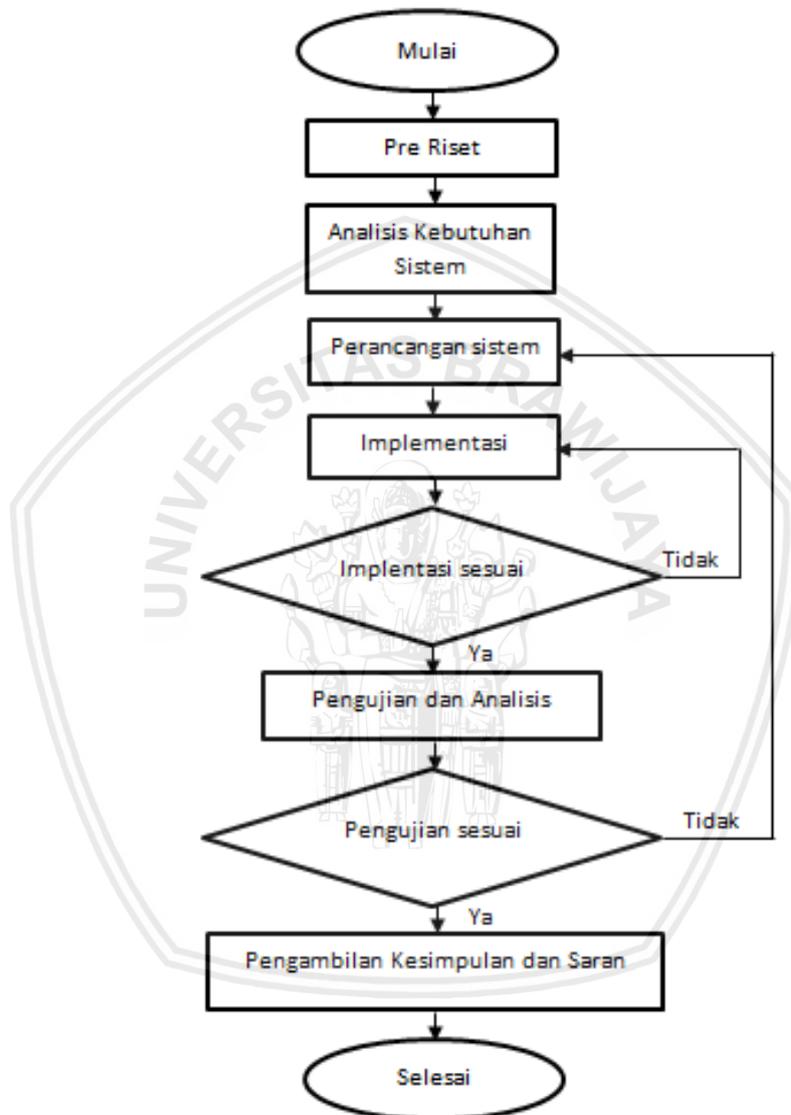
Pada gambar 2.10 yang menunjukkan rangkaian blok dari kontrol proporsional yang memperlihatkan hubungan antara nilai besaran seting, nilai besaran aktual dengan nilai keluaran pengontrol P. Selisih yang ditimbulkan antara besaran seting dan besaran aktual disebut sinyal kesalahan. Hasil perbedaan ini nantinya akan mempengaruhi sistem kontrol sehingga mampu memberikan sinyal positif agar mempercepat mencapai nilai seting atau sinyal negatif untuk memperlambat mencapai sinyal seting. Terdapat 2 parameter pada pengontrol proporsional, pita proporsional dan konstanta proporsional. Pita proporsional mencerminkan daerah kerja kontroler aktif sedangkan nilai faktor penguatan sinyal pada sinyal kesalahan K_p .

2.2.8 Metode *Odometry*

Metode *Odometry* adalah sebuah metode dengan mengambil penggunaan data dari sensor gerak yang dapat memperkirakan perubahan posisi suatu alat dari waktu ke waktu. Dalam dunia robotik, odometri digunakan sebuah robot beroda untuk memprediksi posisi awal terhadap target tujuan yang kan dituju. Dikarenakan saat melakukan pengukuran kecepatan dari waktu ke waktu dalam memperkirakan suatu target posisi yang diinginkan robot, metode ini riskan akan kesalahan perhitungan. Untuk mendapatkan sistem pengukuran yang efektif pada metode odometri dilakukan pengambilan data yang cepat dan akurat, kalibrasi sensor dan pemrosesan yang baik (Bayu. Sandi Marta, Fernando. Ardilla, A.R. Anom Besari, 2011).

BAB III METODOLOGI PENELITIAN

Pada bab metodologi penelitian dituliskan langkah dalam menjelaskan metode dan alur penulisan dalam skripsi secara bertahap. Alur dari penulisan yang terdapat pada metodologi penelitian dapat di lihat pada diagram alir dibawah ini.



Gambar 3. 1 Diagram Alir Metodologi Penelitian

3.1 Pre Riset

Langkah pertama kali yang dilakukan dalam sebuah penelitian yaitu melakukan pra-riset. Rujukan penelitian yang sudah ada dikumpulkan berguna untuk mendapatkan sebuah rumusan masalah. Kegiatan pra-riset dilakukan

dengan mengamati masalah yang akan dipecahkan hingga dapat dituliskan pada skripsi ini.

Memperkuat akar permasalahan yang diamati sebagai dasar teori pada penelitian yang akan dilakukan menjadi dasar untuk membuat kontrol *proportional* pada *mobile robot odometry*.

3.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem yang akan dibuat diperlukan untuk mempersiapkan beberapa kebutuhan yang akan dipakai. Ada dua kebutuhan yang digunakan pada penelitian kali ini. kebutuhan fungsional yang harus ada pada implementasi dan kebutuhan non fungsional untuk membuat dan merealisasikan dari kebutuhan fungsional. Dalam penelitian kali ini yang dihasilkan dalam kebutuhan fungsional yaitu nilai jarak tempuh, input parameter proporsional, *tracking* jarak dan dapat menuju target. Dari kebutuhan non fungsional yang dipakai pada penelitian kali ini berupa Arduino uno, *driver motor* L298N, sensor *rotary encoder* dan IDE Arduino. Jika kebutuhan ini bisa terpenuhi akan dapat mengimplentasikan kontrol pergerakan pada mobile robot odometry dengan baik agar dapat bergerak menuju target tujuan.

3.3 Perancangan Sistem

Setelah melakukan analisis kebutuhan yang ada pada sistem. sekarang sudah saatnya dilakukan proses perancangan sistem. pada proses ini dibuat semua rancangan desain dari robot yang akan dibuat beserta konfigurasi rangkaian yang dipakai. Dalam perancangan dimulai dari merancang arsitektural sebuah rangkaian dari komponen yang ada pada *mobile robot*. Pembuatan block diagram berfungsi untuk melihat dari mana alur kerja dari *prototype* itu berjalan. Proses perancangan alat dimulai dari merancang berbagai komponen yang ada menjadi sebuah *prototype* robot. Dilanjutkan dengan perancangan sebuah program yang akan ditulis dan diproses melalui IDE Arduino. Kemudian setelah semua selesai akan dilanjutkan dengan pengujian alat.

3.4 Implementasi

Tahap berikutnya yaitu implementasi dari perncangan yang telah dibuat sebelumnya. Tahapan ini untuk mengetahui kemampuan sistem yang telah dibuat telah sesuai yang diharapkan. Dengan melihat tahapan ini dapat dilihat kesesuaian alat untuk pengembangan yang akan dilakukan selanjutnya serat batasan dari implementasi terhadap sistem dan program yang dibuat. Spesifikasi lingkungan pengembangan yang digunakan yaitu lingkungan dengan keadaan rata dan kering. Batasan yang ada dalam implementasi digunakan untuk

membatasi komponen dalam sistem yang akan digunakan. Dilanjutkan dengan mengimplementasikan sebuah program untuk dimasukan didalam sistem.

3.5 Pengujian dan Analisis

Dalam alur ini dilakukan untuk menguji bahwa robot yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan sistem. untuk melihat seberapa mampu sistem dalam melakukan sebuah tugas yang ada dilakukan proses uji untuk mendapatkan hasil respon dari sistem. dalam mengetahui respon sistem yang ada dibuat pengujian per blok, seperti pengujian keakuratan sensor rotary encoder, pengujian kendali driver motor dc, pengujian sistem kontrol dan pengujian keseluruhan sistem yang dibuat. Hasil pengujian yang dilakukan akan dianalisis untuk mengetahui kelemahan sistem yang telah dibuat sebelumnya. Hal ini berguna untuk melihat semua fungsi dalam sistem yang akan berjalan untuk mengambil dan mengolah data.

3.6 Pengambilan Kesimpulan dan Saran

Proses terakhir yang dilakukan pada alur metode penelitian yaitu pengambilan kesimpulan dan memberikan saran masukan. Hasil dari kesimpulan diadapat dari pengujian yang telah dilakukan guna menjawab rumasn masalah yang dibuat. Pengembangan selanjutnya dalam penelitian ini diberikan saran untuk memudahkan penelitian selanjutnya tentang implementasi kontrol *proportional* pada *mobile robot odometry* dalam menuju koordinat target.

BAB IV REKAYASA KEBUTUHAN

4.1 Gambaran umum sistem

Implementasi kontrol *proportional* sebagai kontrol pergerakan mobile robot *odometry* dalam menuju koordinat target. Metode *Odometry* ini menghitung nilai dari putaran roda yang dibaca oleh sensor encoder yang diproses menggunakan arduino. Kebutuhan pengguna dapat terpenuhi dengan melihat perhitungan jarak melalui serial monitor arduino. Dengan sensor encoder pengguna dapat melihat seberapa akurat jarak yang diberikan kepada robot. Kontrol *proportional* ini digunakan untuk mengontrol pergerakan robot agar dapat bergerak menuju target tujuan.

4.2 Analisis Kebutuhan Sistem

Pengguna yang ingin menggunakan akan mengirim sebuah koordinat tertentu kepada robot, kemudian robot akan menerima instruksi tersebut dan menyelesaikannya. Pengguna nantinya akan dikirim kembali data *tracking* yang sudah diambil oleh robot. Sehingga pengguna dapat mengetahui seberapa akurat data yang akan diambil oleh robot dengan membandingkannya melalui pengukuran *tracking* aslinya.

4.3 Kebutuhan fungsional

Kebutuhan fungsional ini wajib dapat dipenuhi oleh sistem yang ada pada robot odometry ini. Kebutuhan yang harus dipenuhi oleh sistem ialah:

1. Sistem mendapatkan nilai jarak tempuh data uji

Kegunaan dari fungsi ini dapat menampilkan nilai jarak tempuh dari data uji menggunakan Odometri. Nilai jarak yang ditempuh yang didapat digunakan untuk perbandingan jarak asli dengan jarak yang ditempuh robot.

2. Sistem dapat melakukan perhitungan nilai parameter *proportional*

Fungsi ini digunakan untuk menghitung nilai parameter *proportional* yang dimasukkan. Kemudian data tersebut diolah menggunakan metode *odometry* dan dikontrol menggunakan kontrol *proportional* agar data sesuai dengan yang diinginkan.

3. Sistem dapat melakukan tracking area

Fungsi ini berguna untuk memberikan data area yang sudah dilewati oleh mobile robot. Untuk melakukan tracking digunakan metode odometry pada sistem.

4. Sistem dapat melakukan pergerakan menuju target tujuan

Fungsi ini berguna untuk melakukan pergerakan menuju target yang telah ditentukan. Robot bergerak menuju target yang sudah ditentukan tanpa ada kesalahan.

4.4 Kebutuhan non-fungsional

Dalam memenuhi kebutuhan non-fungsional yang ada pada sistem, terdapat kebutuhan untuk perangkat keras dan kebutuhan untuk perangkat lunak.

4.4.1 Kebutuhan perangkat keras

Ini adalah beberapa komponen yang digunakan pada kebutuhan perangkat keras dalam sistem:

1. Arduino UNO
merupakan mikrokontroler yang berperan sebagai inti pemrosesan dari *prototype*. Kegunaan Pin I/O yang ada pada Arduino UNO sebagai tempat menyambungkan sensor dan juga driver motor untuk membangun sebuah *prototype*.
2. Driver Motor digunakan untuk mengatur arah dan kecepatan motor DC.
3. Sensor Encoder digunakan untuk menghitung putaran dari roda motor DC.
4. Motor DC digunakan untuk penggerak dari *mobile robot*.

4.4.2 Kebutuhan perangkat lunak

Kebutuhan perangkat lunak untuk menunjang kebutuhan dalam sistem ini terdiri dari:

1. Arduino IDE digunakan untuk menulis program dan mengirim program pada sistem robot yang dibuat.

4.5 Batasan sistem

Implementasi kontrol *proportional* sebagai kontrol pergerakan mobile robot *odometry* dalam menuju koordinat target memerlukan batasan implementasi sistem agar tidak keluar dari perancangan yang telah dibuat.

1. Implementasi kontrol *proportional* sebagai kontrol pergerakan mobile robot *odometry* ini memakai Arduino UNO sebagai bagian mikrokontroler pemroses data yang ada didalam sistem.
2. Penelitian yang berjudul Implementasi kontrol *proportional* sebagai kontrol pergerakan mobile robot dibuat dalam bentuk *prototype*.

3. Untuk mendapatkan data masukan, sensor encoder digunakan untuk menghitung putaran roda sebagai input data.
4. Sistem menggunakan metode *Odometry*.



BAB V PERANCANGAN DAN IMPLEMENTASI

Pada bab 5 ini akan membahas tentang perancangan dari implementasi kontrol *proportional* sebagai kontrol pergerakan *mobile robot Odometry*.

5.1 Perancangan Sistem

Pada sub-bab ini menjelaskan tentang perancangan sistem pada bagian perangkat keras dan perangkat lunak.

5.1.1 Gambaran Umum Sistem

Gambar 5.1 merupakan diagram blok yang menunjukkan bagian dari keseluruhan sistem.

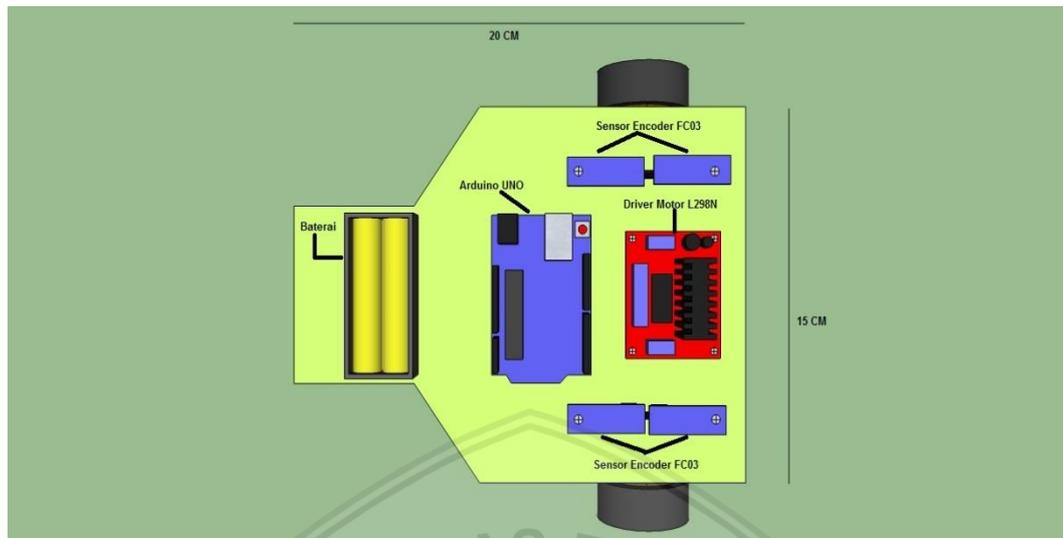


Gambar 5.1 Diagram Blok Sistem

Pada Gambar umum diagram blok keseluruhan sistem yang terdiri dari beberapa komponen dibuat untuk merancang robot *odometry*. Perangkat keras yang dibutuhkan terdiri dari komponen meliputi Arduino UNO yang disambungkan dengan driver motor DC dan sensor *encoder*. Komponen tersebut saling terhubung sesuai dengan Gambar 5.1.

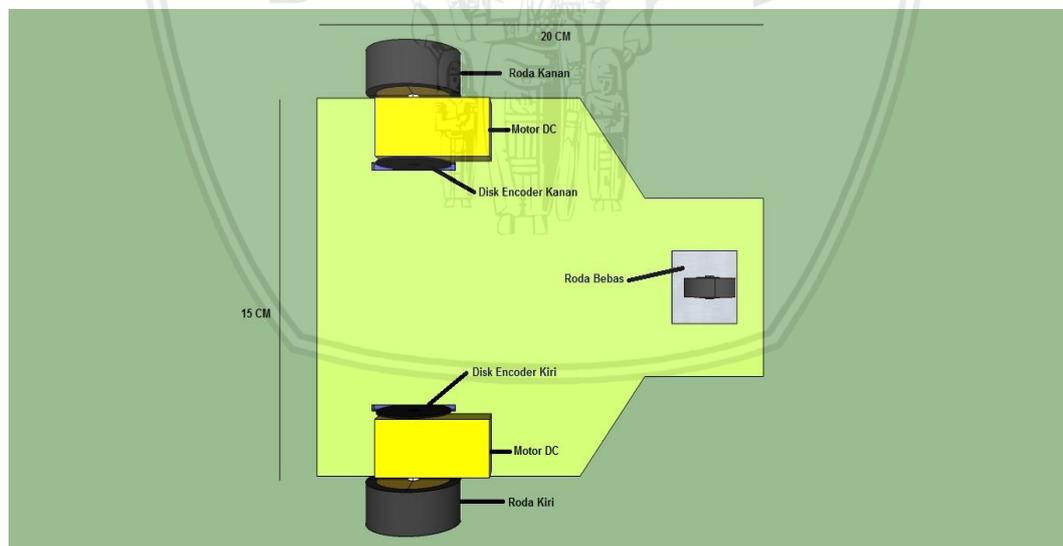
5.1.2 Perancangan *Prototype* Alat

Letak Setiap komponen harus diperhatikan agar dalam tahap implementasi prototype dapat ditetapkan sesuai dengan yang di rencanakan pada Gambar 5.2 dan 5.3



Gambar 5.2 Desain *Prototype* Robot Tampak Atas

Pada Gambar 5.2 adalah rancangan dari *mobile robot* tampak atas. Pada bagian depan terdapat baterai. Di sebelah tengah terdapat Arduino UNO dan Driver Motor DC terpasang pada bagian tengah belakang robot. Sensor Encoder terpasang pada bagian kanan dan kiri.



Gambar 5.3 Desain *Prototype* Robot Tampak Bawah

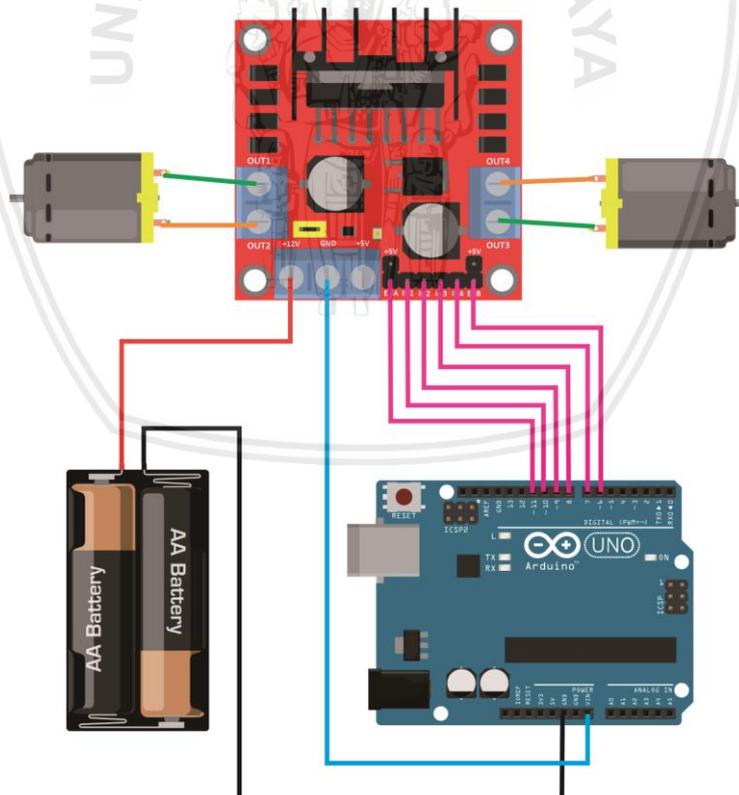
Pada Gambar 5.3 adalah robot tampak bagian bawah. Terlihat pada bagian depan bawah terpasang roda bebas untuk bergerak 360 derajat. Di bagian belakang terpasang 2 motor DC yang saling terpasang roda disamping luarnya. Di bagian samping dalam motor DC terdapat *rotary* untuk input pembacaan sensor encoder.

5.1.3 Perancangan Perangkat Keras

Pada blok diagram Gambar 5.1 tentang gambaran umum sistem. Pemrosesan data pada sistem yaitu menggunakan Arduino UNO. Pada bagian sensor menggunakan 4 sensor encoder. Terdapat juga *driver motor DC* untuk mengendalikan kedua motor DC yang digunakan untuk menggerakkan roda belakang dan juga *rotary*.

5.1.3.1 Konfigurasi Driver Motor DC

Pada Gambar 5.4 merupakan skematik dari perancangan driver motor DC yang terhubung ke Arduino UNO. 2 Kapasitor 22uF yang berfungsi menyimpan dan melepas muatan listrik. Fungsi dari *driver motor* ini berguna untuk mengendalikan kecepatan dari *Motor DC* agar dapat bergerak sebagaimana fungsinya dan juga untuk mengatur PWM yang digunakan untuk menggerakkan roda robot. Semua pin yang ada pada *driver motor* ini akan dihubungkan semua ke Arduino.



Gambar 5.4 Konfigurasi Driver Motor DC

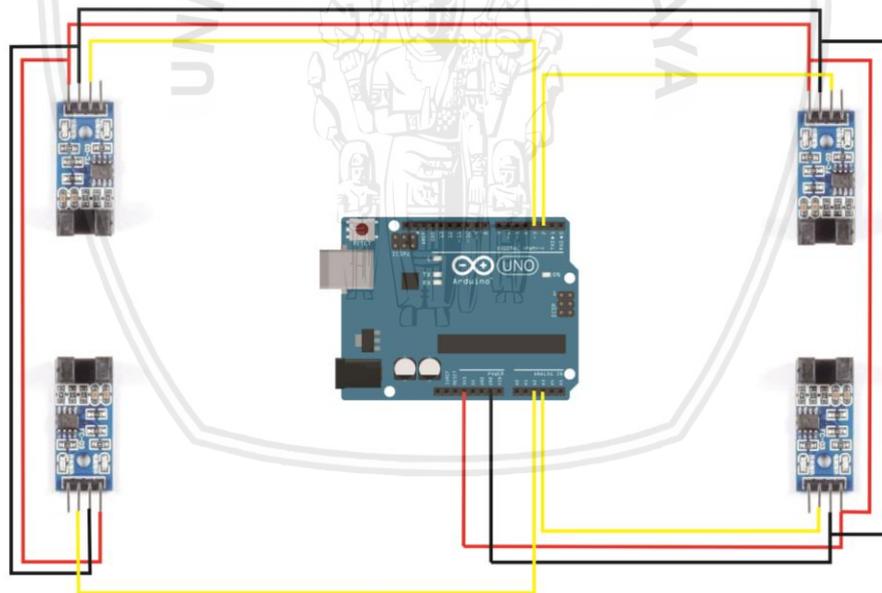
Tabel 5.1 merupakan keterangan dari konfigurasi pin yang ada pada Gambar 5.4. Pada tabel ini menunjukkan letak pemasangan pin dari *driver motor*.

Tabel 5.1 Pin Skematik *Driver Motor*

NO	Arduino Uno	<i>Driver Motor DC</i>
1.	PIN 6	EN B <i>Driver motor</i>
2.	PIN 7	IN 4 <i>Driver motor</i>
3.	PIN 8	IN 3 <i>Driver motor</i>
4.	PIN 9	IN 2 <i>Driver motor</i>
5.	PIN 10	IN 1 <i>Driver motor</i>
6.	PIN 11	EN A <i>Driver motor</i>
7.	PIN Vin	GND <i>Driver motor</i>
8.	PIN Gnd	(-) <i>Batteray</i>

5.1.3.2 Konfigurasi Sensor Encoder

Pada Gambar 5.5 merupakan konfigurasi pin dari keempat sensor encoder. Pada perancangan sensor encoder terdapat 4 sensor yang digunakan. Untuk penjelasan dari konfigurasi pin terdapat pada Tabel 5.2



Gambar 5.5 Konfigurasi Sensor Encoder

Pada Gambar 5.5 merupakan skematik perangkat keras untuk merancang sensor *encoder* yang mana pin dari VCC (warna merah) terhubung dengan VCC dan pin GND (warna hitam) terhubung dengan GND, Pin D0 pada Sensor *Encoder* kiri terhubung dengan Arduino UNO pada Pin 3 *digital* dan Pin D0 pada Sensor *Encoder* kanan terhubung dengan Arduino UNO pada Pin 2 *digital*. Untuk sensor *encoder* kiri1 terhubung pada Arduino pin 3 *analog*. Sedangkan untuk sensor



encoder kanan1 terhubung pada Arduino pin 2 *analog*. Untuk keberadaan sensor encoder maupun sensor encoder1 memiliki fungsi yang berbeda. Sensor encoder kiri-kanan berfungsi untuk melakukan perhitungan maju /positif. Sedangkan untuk sensor encoder1 kiri-kanan berfungsi untuk menghitung mundur/negatif.

Tabel 5.2 merupakan keterangan yang menunjukkan konfigurasi dari pin Arduino UNO yang tersambung pada *sensor encoder*.

Tabel 5.2 Pin Skematik Sensor Encoder

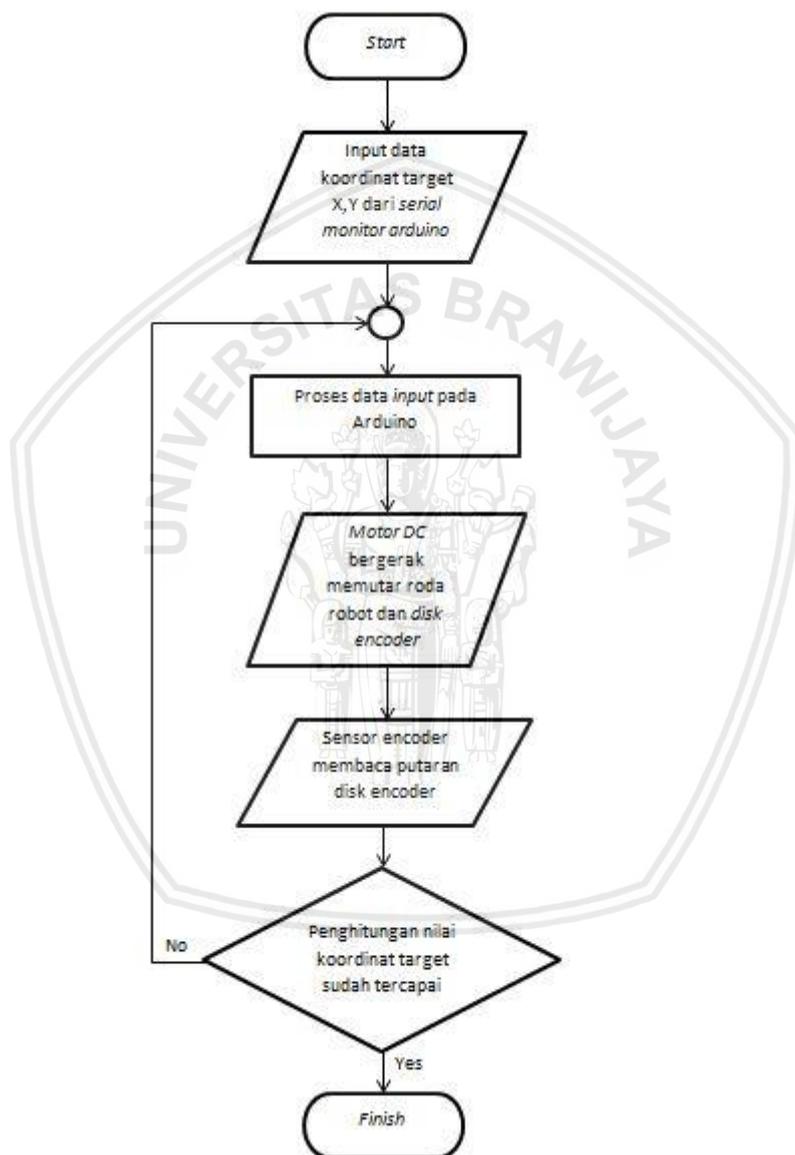
NO	Arduino Uno	Sensor Encoder
1.	PIN 2	D0 Sensor <i>Encoder</i> Kanan
2.	PIN 3	D0 Sensor <i>Encoder</i> Kiri
3.	PIN A2	D0 Sensor <i>Encoder</i> Kanan1
4.	PIN A3	D0 Sensor <i>Encoder</i> Kiri1



5.1.4 Perancangan Perangkat Lunak

Pada sub bab ini menjelaskan perancangan dari perangkat lunak. Mulai dari perancangan akuisisi data, *odometry*, PID, dan pergerakan robot.

5.1.4.1 Perancangan Proses Akuisisi Data

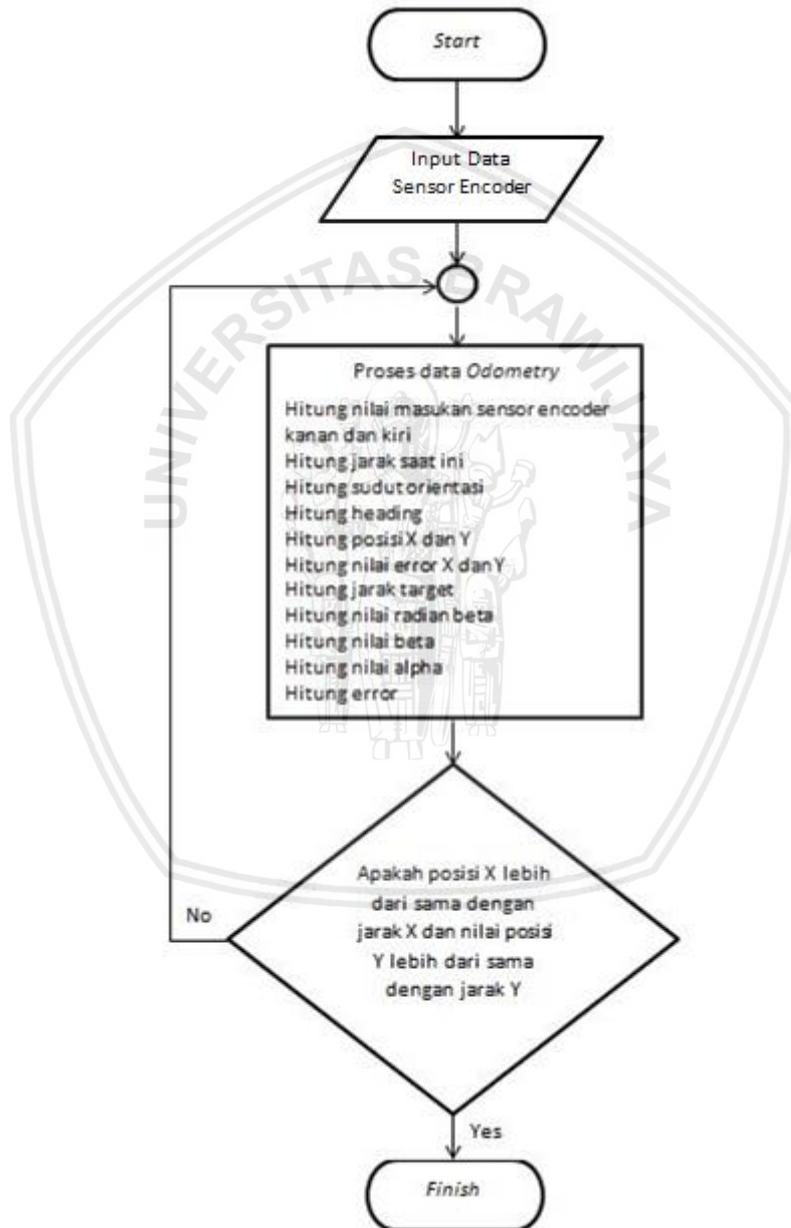


Gambar 5.6 Flowchart Akuisisi Data

Pada Gambar 5.6 merupakan flowchart dari proses akuisisi data yang akan dijalankan. Proses pertama yang dilakukan dengan menginput koordinat X,Y dari serial monitor IDE Arduino. Inputan kemudian akan diproses oleh arduino yang kemudian akan menggerakkan *motor DC* dan *disk encoder*. *Disk encoder* yang

berputar akan dibaca oleh sensor *encoder* dimana proses pembacaan sensor tersebut akan menghasilkan data yang akan digunakan robot menuju target tujuan robot.

5.1.4.2 Perancangan Pembacaan Sensor Encoder Dengan Metode Odometry

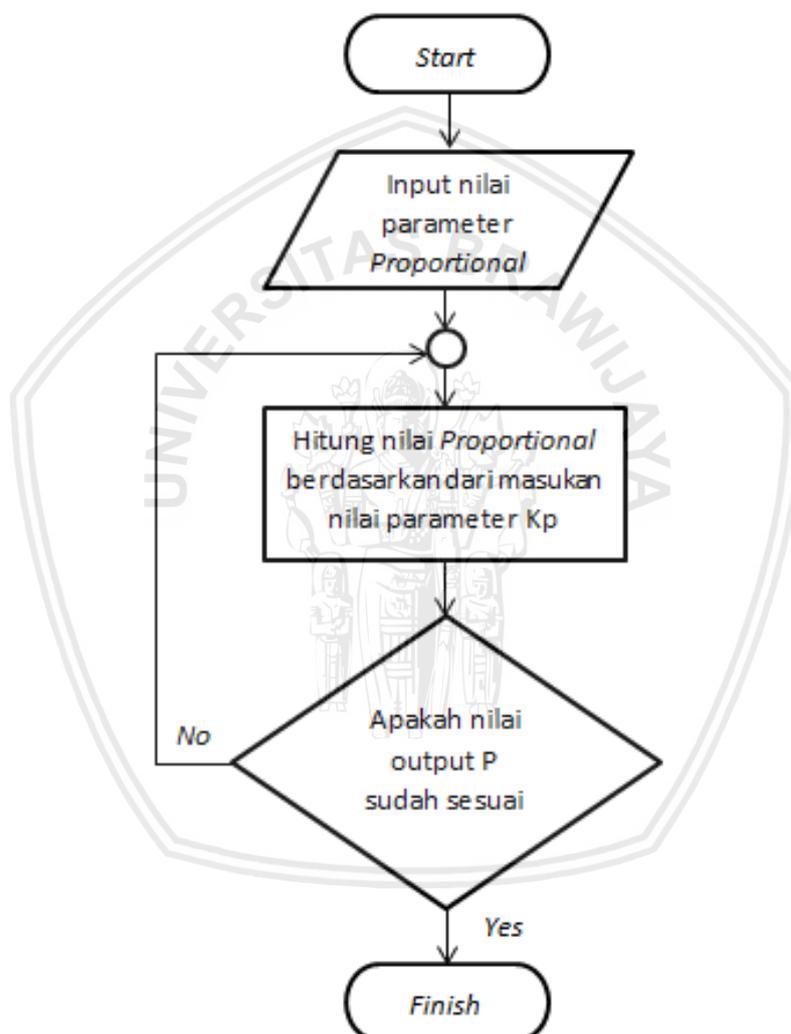


Gambar 5.7 Flowchart Pembacaan Sensor

Pada proses pembacaan sensor dapat dilihat pada flowchart Gambar 5.7. input data dimasukan untuk mendapatkan koordinat target yang akan dituju.

Selanjutnya data inputan tadi akan diproses pada metode odometry. Pada proses odometry ini sensor encoder akan membaca nilai input data yang telah dimasukan. Data sensor encoder akan dihitung dengan menggunakan proses odometry. Jika nilai jarak dan posisi sudah sesuai robot telah berhasil mencapai target tujuan yang telah ditentukan.

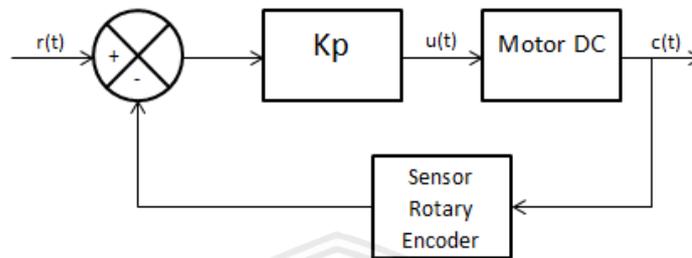
5.1.4.3 Perancangan Kontrol *Proportional*



Gambar 5.8 Flowchart Kontrol *Proportional*

Proses kontrol *proportional* dapat dilihat pada Gambar 5.8. Proses pertama yaitu menginput nilai parameter KP. Pada penelitian kali ini nilai KP dimasukan secara manual dan dilihat respon sitem yang paling bagus. Dari respon sistem yang paling bagus nantinya yang akan dipakai pada tahap pengujian. Nilai parameter tersebut akan dihitung sehingga menghasilkan nilai output

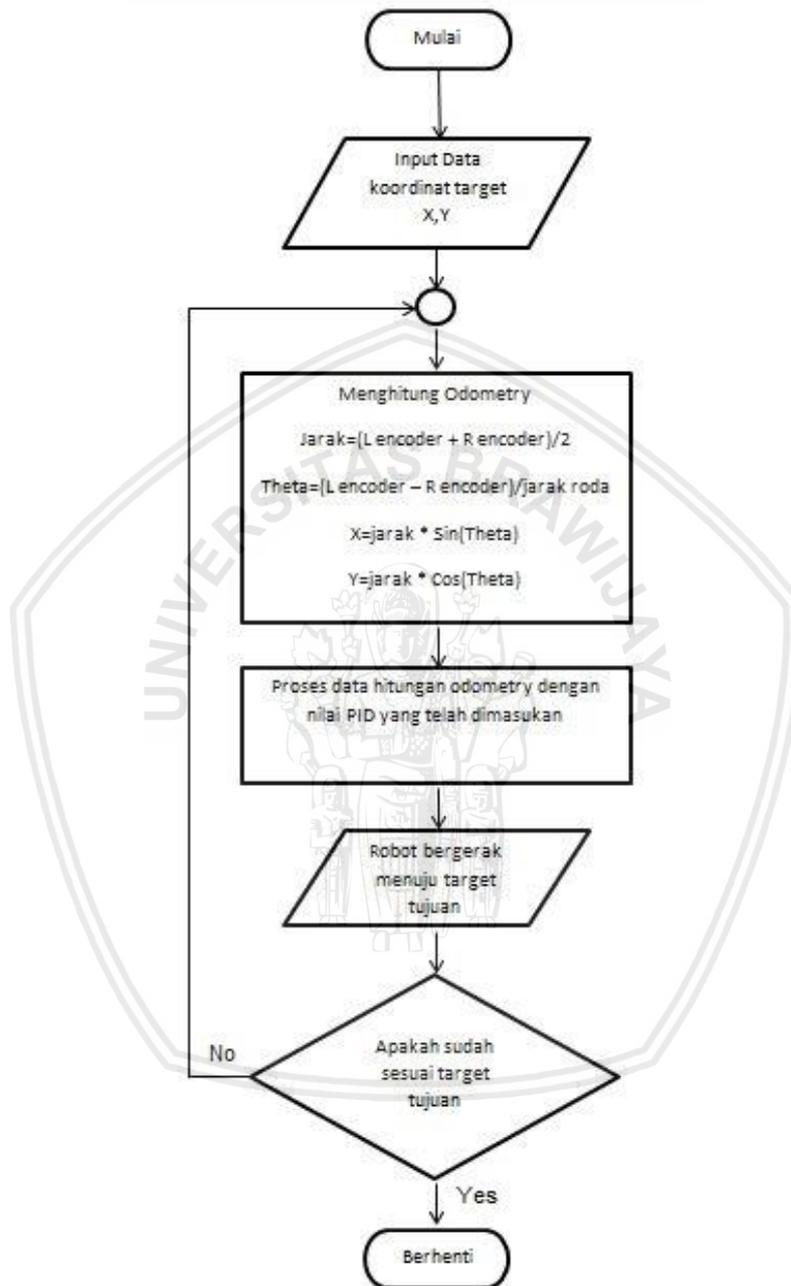
proportional yang digunakan untuk mengontrol pergerakan robot untuk menuju koordinat target tujuan.



Gambar 5.9 Rangkaian Closed Loop Proportional

Gambar 5.9 merupakan rangkaian closed loop dari sistem kontrol *proportional* yang telah dibuat. Sistem closed loop ini menunjukkan hubungan masukan dan keluaran dari sistem loop tertutup ini. *Output* dari sistem tertutup ini mempunyai pengaruh langsung pada pengontrolan. Sistem ini juga merupakan sistem berumpan balik.

5.1.4.4 Perancangan Proses Pergerakan Robot Menuju Target



Gambar 5.10 Flowchart Pergerakan Robot Menuju Target

Pada Gambar 5.10 merupakan *flowchart* perancangan pergerakan robot menuju target tujuan. Proses pertama yaitu menginputkan koordinat target tujuan X, Y. Sistem ini mendapatkan nilai koordinat X,Y, dan theta awal=0. Kemudian jika robot bergerak akan mulai menghitung koordinat baru dengan rumus odometry. Dari perhitungan rumus tersebut diperoleh koordinat baru

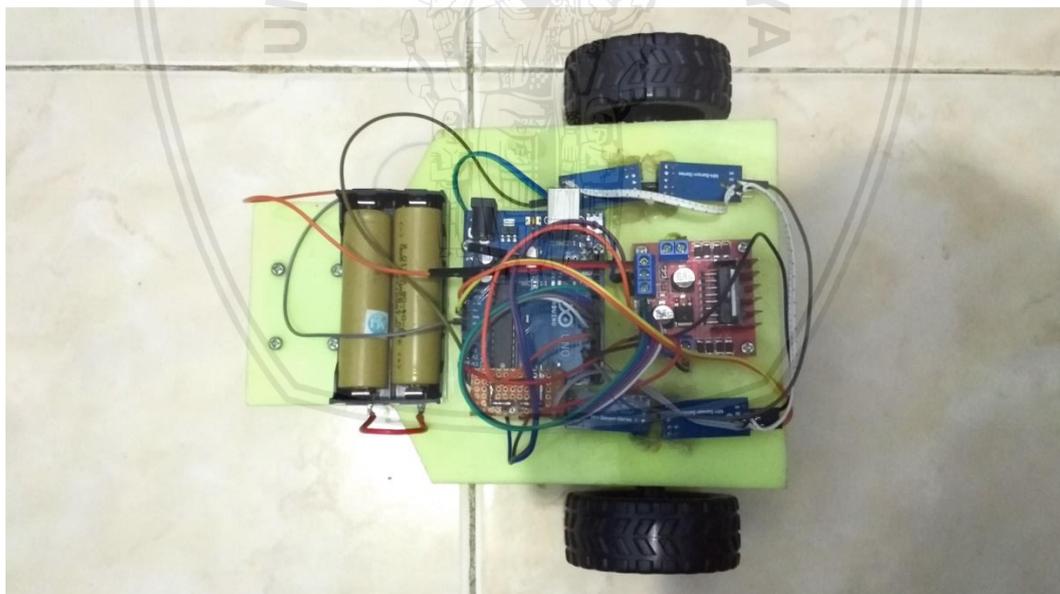
yang akan didapat oleh robot. Proses ini akan dikontrol oleh sistem *proportional* agar robot dapat bergerak ke arah target. Jika koordinat yang dituju tidak sesuai maka robot akan terus menghitung nilai X,Y, theta agar mendapatkan koordinat yang sesuai target dari input data.

5.2 Implementasi Sistem

Pada sub bab ini akan membahas mengenai implementasi dari sistem yang telah dirancang. Hasil perancangan yang akan diimplementasikan agar robot dapat direalisasikan.

5.2.1 Implementasi Prototype Alat

Pada Gambar 5.11 merupakan implementasi dari prototype robot. Untuk bahan penunjangnya menggunakan akrilik sebagai tempat ditaruhnya semua komponen. Pada bagian tengah robot ditempatkan arduino UNO. Dibagian sisi depan terdapat baterai dan disisi belakang terdapat *driver motor*. Peletakan sensor diletakan pada bagian sisi kiri dan kanan robot.



Gambar 5.11 Implementasi Robot Tampak Atas

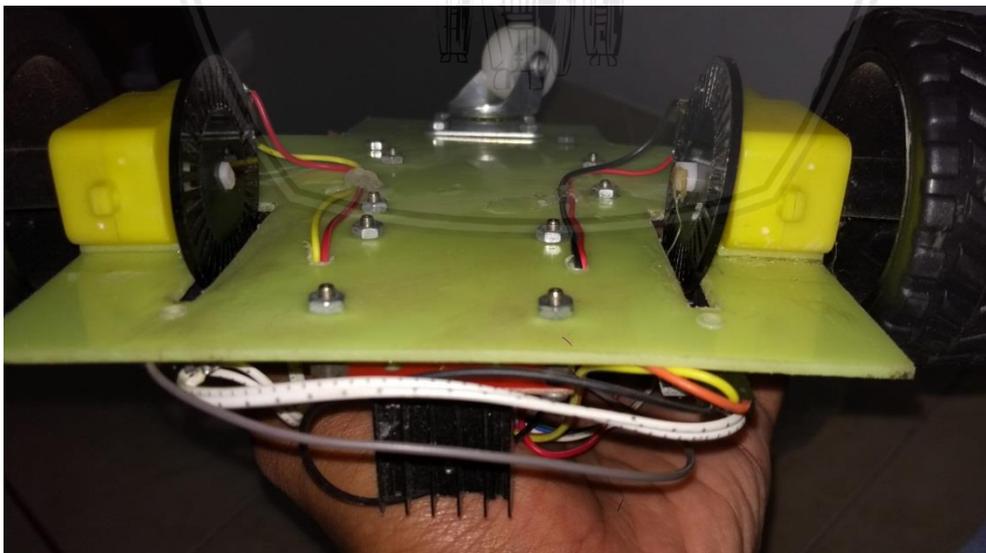
Pada Gambar 5.12 merupakan bagian bawah dari robot. Dibagian sisi depan robot terdapat roda bebas agar robot dapat berbelok. Dibagian belakang sisi kanan dan kiri terpasang motor DC untuk menggerakkan roda robot dan juga disk encoder. Peletakan *disk encoder* yaitu pada bagian sisi dalam motor DC. *Disk encoder* ini yang akan berfungsi sebagai input data dari sensor encoder.



Gambar 5.12 Implementasi Robot Tampak Bawah

5.2.2 Implementasi Perangkat Keras

Implementasi perangkat keras dapat dilihat pada Gambar 5.13 merupakan robot bagian belakang. Terdapat komponen sensor encoder dan juga *disk encoder* yang terpasang pada motor DC yang dikendalikan oleh *driver motor* L298N.



Gambar 5.13 Implementasi Robot Tampak Belakang

lubang yang ada pada disk encoder nantinya akan digunakan sebagai pembacaan sensor. Masing – masing sensor terpasang pada bagian atas kiri dan kanan robot.



Gambar 5.14 Implementasi Robot Tampak Samping Kiri

5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak menjelaskan program yang akan ditulis pada arduino. Perancangan perangkat lunak dilakukan agar dapat memfungsikan semua komponen dengan baik dan merealisasikan metode dengan baik.

5.2.3.1 Implementasi Proses Akuisisi Data

Di bawah ini merupakan implementasi perangkat lunak dari proses akuisisi data. Pada program di bawah ini bermaksud untuk melakukan proses dari akuisisi data pada sistem.

Tabel 5.3 *Source Code* Akuisisi Data

No.	Source Code
1	<code>void setup()</code>
2	<code>{</code>
3	<code> Serial.begin(9600); //baud rate serial</code>
4	<code> pinMode(A3, INPUT); //input untuk data 2 rotary encoder</code>
5	<code> pinMode(2, INPUT); //input untuk interrupt 1 rotary</code>
6	<code> encoder</code>
7	<code> pinMode(A2, INPUT);</code>

```

8   pinMode(3, INPUT);
9   for(int i=7; i<=10; i++){
10    pinMode(i, OUTPUT);    // pin motor output
11  }
12  attachInterrupt(digitalPinToInterrupt(interruptPin3),
13  countRight, CHANGE);
14  attachInterrupt(digitalPinToInterrupt(interruptPin2),
15  countLeft, CHANGE);
16  }
17  void loop() {
18    while(!starting)
19    {
20      if(Serial.available())
21      {
22        char c = Serial.read();
23        if( c== '\n' )
24        {
25          parseData(dataSerial, " ");
26          dataSerial = "";
27          xDistance = valSplit[0].toFloat();
28          yDistance = valSplit[1].toFloat();
29          Serial.print(xDistance);    Serial.print("    ");
30          Serial.println(yDistance);
31          delay(2000);
32          starting = true;
33        }
34        else
35        {
36          dataSerial += String(c);
37        }
38      }
39    }

```

Pada Tabel 5.3 baris 1 sampai 10 merupakan setup variable yang digunakan untuk inputan data yang akan dimasukan dari *serial monitor* arduino. Data koordiant yang akan di input oleh pengguna nantinya digunakan robot untuk bergerak ke target tujuan sesuai yang di input oleh pengguna. Kemudian pada bais 17 sampai 39 data koordinat *input* yang telah dimasukan oleh pengguna melalui *serial monitor* arduino akan diproses sehingga robot dapat bergerak.

5.2.3.2 Implementasi Pembacaan Sensor Dengan Metode Odometry

Tabel 5.4 Source Code Pembacaan Sensor

No.	Source Code
1	while (starting)
2	{
3	double leftRotaryMm = 2 * 3.14 * rRoda * (double(rotaryLeft) / resolusiEncLeft);
4	double rightRotaryMm = 2 * 3.14 * rRoda * (double(rotaryRight) / resolusiEncRight);
5	currentDistance = (leftRotaryMm + rightRotaryMm) / 2;
6	sudutOrientasi = (leftRotaryMm - rightRotaryMm) /



	<pre> wheelBase; 7 heading = sudutOrientasi*180/3.14; 8 xPos = currentDistance * sin(sudutOrientasi); 9 yPos = currentDistance * cos(sudutOrientasi); 10 errorX = xDistance - xPos; 11 errorY = yDistance - yPos; 12 targetDistance = sqrt((pow(errorX,2)+0.5) + (pow(errorY,2)+0.5)); 13 double radianBeta = asin(errorX/targetDistance); 14 beta = radianBeta * 180/3.14; 15 alpha = beta - heading; 16 error = alpha; 17 if(xPos >= xDistance && yPos >= yDistance) 18 { 19 while(1) 20 { 21 backward(0,0); 22 } 23 } </pre>
--	---

Pada Tabel 5.4 merupakan *Source Code Odometry* yang berfungsi sebagai program utama untuk menghitung nilai dari *odometry* yang didapat dari sensor encoder. Sensor encoder akan membaca inputan dari lubang yang ada pada disk encoder yang berputar. Jarak yang dihasilkan robot saat roda berputar selama satu putaran yaitu 22 cm. Program ini adalah program utama bagi keseluruhan fungsi program yang ada.

5.2.3.3 Implementasi Kontrol *Proportional*

Tabel 5.5 *Source Code* kontrol pergerakan menggunakan *proportional*

No.	Source Code
1	<pre>P = Kp * error;</pre>
2	<pre>MV = P; //output Proporsional</pre>
3	<pre>if(MV>=-Max_MV && MV <= Max_MV)</pre>
4	<pre>{</pre>
5	<pre> p_l = max_pwm_ref + MV;</pre>
6	<pre> p_r = max_pwm_ref - MV;</pre>
7	<pre> if (p_l < 0) p_l = 0;</pre>
8	<pre> if (p_l > 200) p_l = 200;</pre>
9	<pre> if (p_r < 0) p_r = 0;</pre>
10	<pre> if (p_r > 200) p_r = 200;</pre>
11	<pre> forward(p_l,p_r);</pre>
12	<pre>}</pre>
13	<pre>else</pre>
14	<pre>{</pre>
15	<pre> p_l = 75; p_r = 75;</pre>
16	<pre> forward(p_l,p_r);</pre>
17	<pre>}</pre>

Pada Tabel 5.5 merupakan *Source Code* untuk menghitung nilai parameter *proportional* yang digunakan sebagai kontrol pergerakan robot agar robot dapat bergerak menuju target tujuan. Pada baris 1 sampai 11 merupakan proses untuk menghitung nilai *output* dari parameter *proportional*.

5.2.3.4 Implementasi Proses Pergerakan Robot Menuju Target

Tabel 5.6 *Source Code* pergerakan menuju posisi target

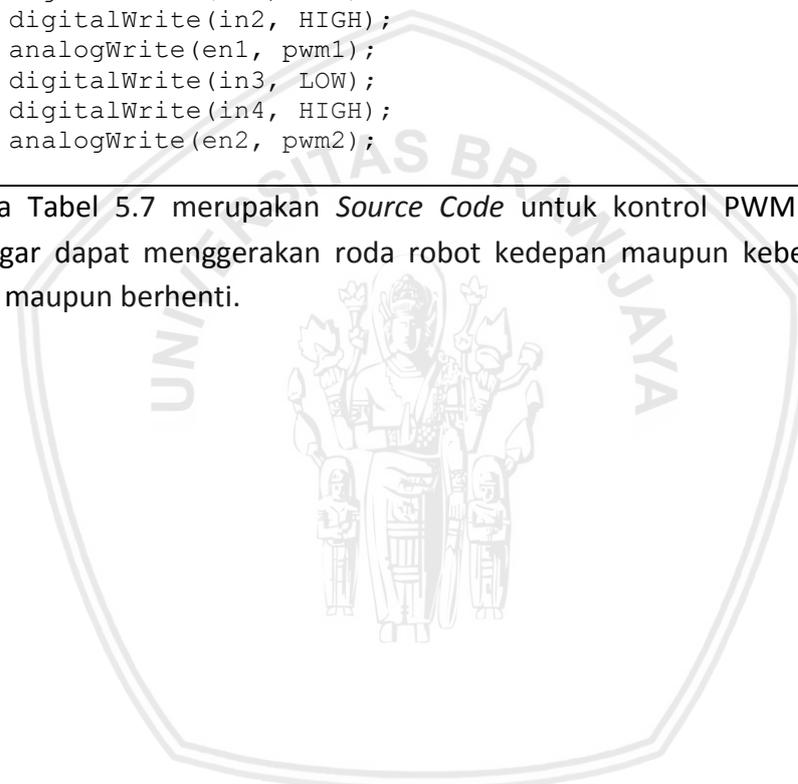
No.	Source Code
1	void countLeft() {
2	if (digitalRead(2) == HIGH) {
3	if (digitalRead(A3) == LOW) {
4	rotaryLeft++; //CW
5	}
6	else {
7	rotaryLeft--; //CCW
8	}
9	}
10	else {
11	if (digitalRead(A3) == HIGH) {
12	rotaryLeft++; // CW
13	}
14	else {
15	rotaryLeft--; // CCW
16	}
17	}
18	}
19	void countRight()
20	{
21	if (digitalRead(3) == HIGH) {
22	if (digitalRead(A2) == LOW) {
23	rotaryRight++; // CW
24	}
25	else
26	{
27	rotaryRight--; // CCW
28	}
29	}
30	else {
31	if (digitalRead(A2) == HIGH) {
32	rotaryRight++; // CW
33	}
34	else {
35	rotaryRight--; // CCW
36	}
37	}
38	}

Pada Tabel 5.6 merupakan *Source Code* untuk bergerak menuju koordinat target setelah menghitung nilai odometry dan juga konstanta *proportional*. Program akan membaca nilai dari sensor dengan cara menghitung putaran dari *disk encoder*. Jika putaran *disk encoder* bergerak sesuai arah jarum jam maka program akan menghitung maju dan nilai data akan ditambahkan jika *disk encoder* bergerak berlawanan arah jarum jam maka program akan menghitung mundur dan nilai akan dikurangi.

Tabel 5.7 Source Code PWM

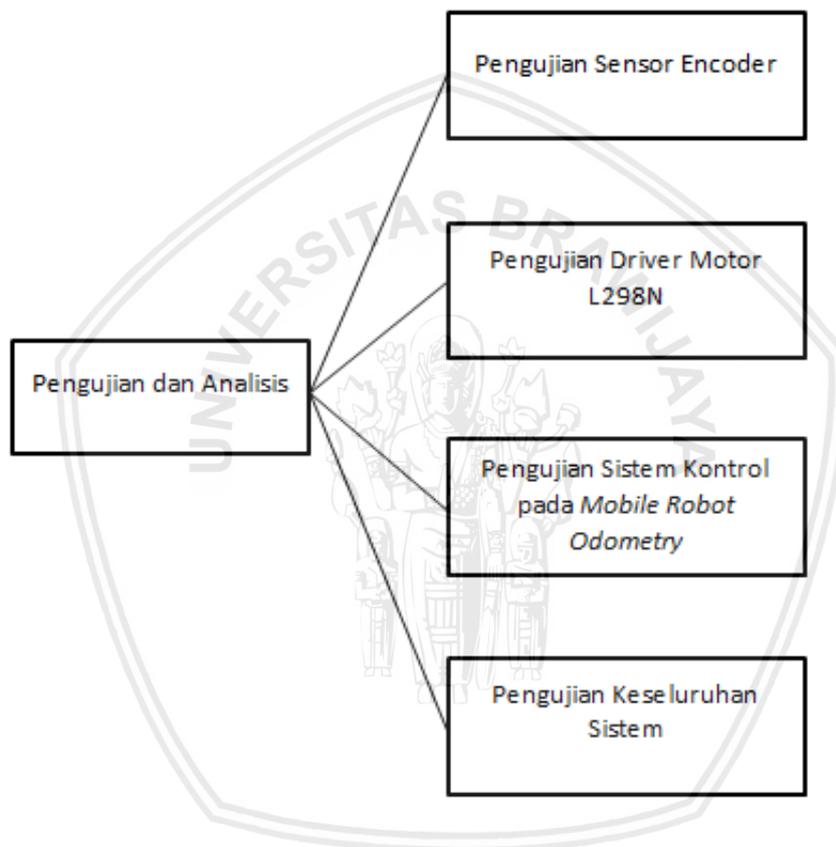
No.	Source Code
1	<code>void forward(int pwm1, int pwm2){</code>
2	<code>digitalWrite(in1, HIGH);</code>
3	<code>digitalWrite(in2, LOW);</code>
4	<code>analogWrite(en1, pwm1);</code>
5	<code>digitalWrite(in3, HIGH);</code>
6	<code>digitalWrite(in4, LOW);</code>
7	<code>analogWrite(en2, pwm2);</code>
8	<code>}</code>
9	<code>void backward(int pwm1, int pwm2){</code>
10	<code>digitalWrite(in1, LOW);</code>
11	<code>digitalWrite(in2, HIGH);</code>
12	<code>analogWrite(en1, pwm1);</code>
13	<code>digitalWrite(in3, LOW);</code>
14	<code>digitalWrite(in4, HIGH);</code>
15	<code>analogWrite(en2, pwm2);</code>
16	<code>}</code>

Pada Tabel 5.7 merupakan *Source Code* untuk kontrol PWM dari *driver motor* agar dapat menggerakkan roda robot kedepan maupun kebelakang dan berjalan maupun berhenti.



BAB VI PENGUJIAN DAN ANALISIS

Pada bab ini pembahasan pengujian dan analisis dari pengujian sensor encoder, pengujian driver motor dan pengujian kontrol pada *mobile robot odometry*.



Gambar 6.1 Bagan Pengujian dan Analisis

6.1 Pengujian Sensor Encoder FC 03

Pada pengujian sensor encoder dengan melakukan pengujian respon pembacaan lubang dari disk encoder yang dijadikan bahan pengujian pada sensor.

6.1.1 Tujuan Pengujian

Untuk mengetahui keakuratan sensor encoder dalam membaca lubang pada *disk encoder* dan mengetahui tampilan presentasi keberhasilannya.

6.1.2 Prosedur Pengujian

Pada Gambar 6.2 prosedur pengujian sensor encoder dengan membaca lubang yang ada pada *disk encoder*. Pengujian menguji keakuratan dalam menghitung lubang yang telah melewati sensor.



Gambar 6.2 Pengujian Sensor Encoder

1. Menghubungkan robot dengan laptop menggunakan kabel USB.
2. Buka source code yang disiapkan untuk pengujian sensor encoder. *Compile source code* tersebut ke dalam mikrokontroler Arduino UNO.
3. Putar Roda agar *disk encoder* dapat berputar untuk melihat respon sensor encoder.
4. Lakukan putaran maju untuk melihat respon nilai sensor encoder pada *serial monitor*.
5. Lakukan putaran mundur untuk melihat respon nilai sensor encoder pada *serial monitor*.
6. Mengumpulkan hasil data pengujian sensor encoder.

6.1.3 Hasil dan Analisis Pengujian

Hasil dari pengujian sensor encoder dalam membaca lubang *disk encoder* yang digunakan dalam sistem ditampilkan pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian Sensor Encoder

Putaran Maju		Putaran Mundur	
Sensor Kanan	Sensor kiri	Sensor Kanan	Sensor kiri
1	1	-1	-1
2	2	-2	-2
3	3	-3	-3
4	4	-4	-4
5	5	-5	-5
6	6	-6	-6
7	7	-7	-7
8	8	-8	-8
9	9	-9	-9
10	10	-10	-10
11	11	-11	-11
12	12	-12	-12
13	13	-13	-13
14	14	-14	-14
15	15	-15	-15

Sensor Encoder bekerja dengan membaca lubang yang ada pada *disk encoder*. Dari data percobaan diatas dapat dilihat bahwa sensor encoder dapat membaca lubang pada disk encoder. Tetapi kelemahan sensor encoder ini jika perputaran *disk encoder* semakin cepat pembacaan sensor juga semakin tidak akurat.

6.2 Pengujian Driver Motor L298N

Pada pengujian ini, *driver motor* L298N digunakan untuk menentukan gerakan putaran motor DC. Pengujian *driver motor* L298N dilakukan untuk mengetahui gerakan maju dan mundur robot.

6.2.1 Tujuan Pengujian

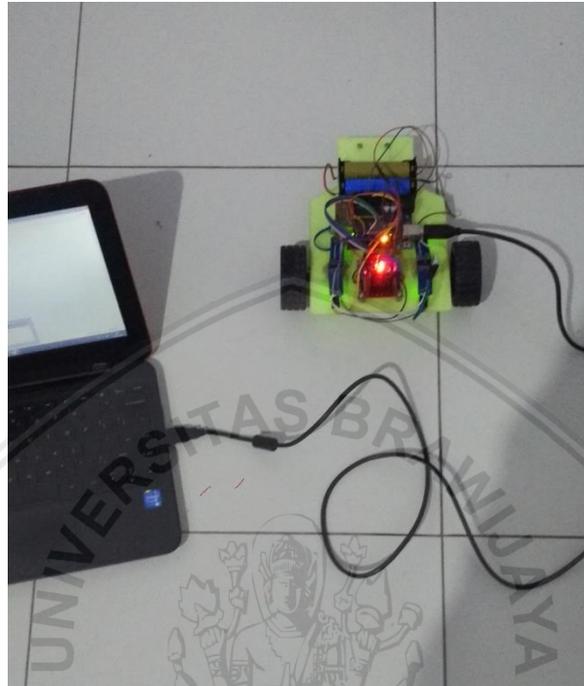
Untuk mengetahui tingkat keberhasilan dari driver motor L298N saat melakukan pergerakan maju dan mundur.

6.2.2 Prosedur Pengujian

1. Menghubungkan robot dengan laptop menggunakan kabel USB.
2. Membuka *source code* yang dibuat untuk *driver motor* L298N. Kemudian *compile source code* tersebut ke mikrokontroler Arduino UNO.

3. Buat kondisi *driver motor* L298N untuk bergerak maju maupun mundur.
4. Amati pergerakan dari motor DC

Berikut Gambar 6.3 robot melakukan pengujian pergerakan.



Gambar 6.3 Pengujian Pergerakan Robot

6.2.3 Hasil dan Analisis Pengujian

Hasil dari pengujian tingkat akurasi keberhasilan dari driver motor L298n yang digunakan dalam sistem ditampilkan pada Tabel 6.2.

Tabel 6.2 Hasil Pengujian Pergerakan Robot

Pergerakan	Percobaan	Keterangan	% Keberhasilan
Maju	1	Berhasil	100%
	2	Berhasil	
	3	Berhasil	
	4	Berhasil	
	5	Berhasil	
Mundur	1	Berhasil	100%
	2	Berhasil	
	3	Berhasil	
	4	Berhasil	
	5	Berhasil	

Hasil dari pengujian pergerakan driver motor yang telah diuji dapat dilihat pada Tabel 6.2. Pada pengujian driver motor ini sistem yang ada telah sesuai dan bekerja dengan baik. Rata – rata hasil dari pengujian driver motor bernilai 100%.

Pada pengujian ini masing – masing diuji sebanyak 5 kali. Pengujian dilakukan menguji pergerakan maju dan mundur robot. Hasil pengujian kedua parameter tersebut memperlihatkan bahwa robot dapat bergerak maju dan mundur.

6.3 Pengujian Kontrol Pada *Mobile Robot Odometry* Untuk Menuju Koordinat Tujuan

Dalam pengujian kontrol ini, robot dicoba untuk melakukan pergerakan menuju target tujuan yang telah diberikan oleh *user* dan juga di berikan nilai parameter kontrol. Pengujian ini berguna untuk melihat sistem kontrol yang sesuai untuk dapat mengontrol pergerakan *mobile robot* dengan metode *odometry* untuk menuju koordinat target tujuan.

6.3.1 Tujuan Pengujian

Untuk mengetahui apakah sistem kontrol yang sesuai untuk dapat mengontrol pergerakan *mobile robot* dengan metode *odometry* menuju koordinat target tujuan.

6.3.2 Prosedur Pengujian

1. Menghubungkan robot dengan laptop menggunakan kabel USB.
2. Membuka *source code* yang digunakan dalam sistem kontrol untuk pergerakan mobile robot menggunakan metode *odometry* menuju koordinat target. Lakukan *compile* program ke mikrokontroler Arduino UNO.
3. Berikan nilai koordinat target X,Y pada *serial monitor* IDE Arduino.
4. Amati nilai *output set point* posisi koordinat X,Y pada *serial monitor*.
5. Bandingkan hasil nilai parameter setiap sistem kontrol untuk melihat respon *set point* yang paling bagus.

Berikut Gambar 6.4 percobaan robot melakukan pengujian sistem kontrol.



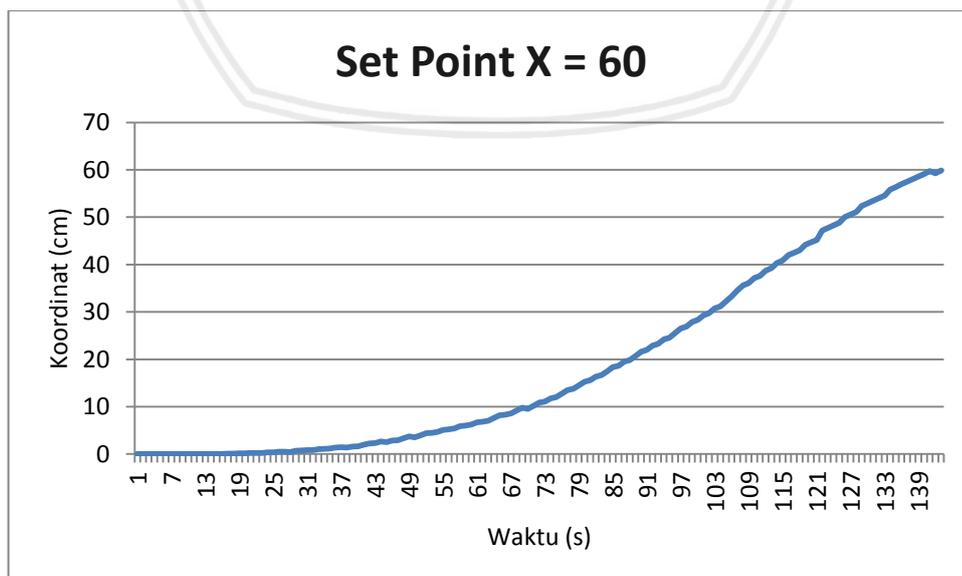
Gambar 6.4 Pengujian Sistem Kontrol

6.3.3 Hasil dan Analisis Pengujian

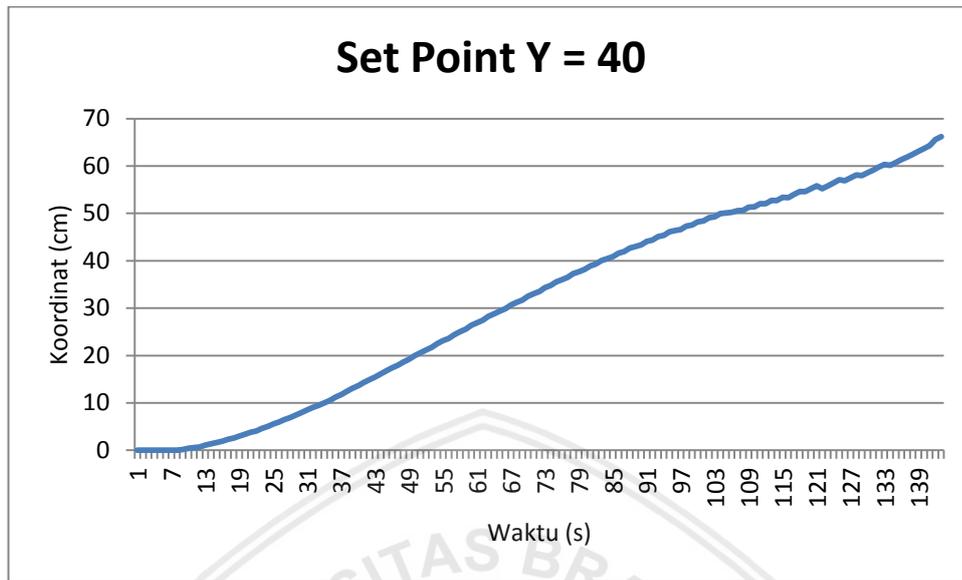
Hasil dari pengujian sistem kontrol yang dipakai dalam pergerakan *mobile robot* dengan metode *odometry* untuk menuju koordinat target tujuan.

Percobaan 1

Parameter $P = 0.8$

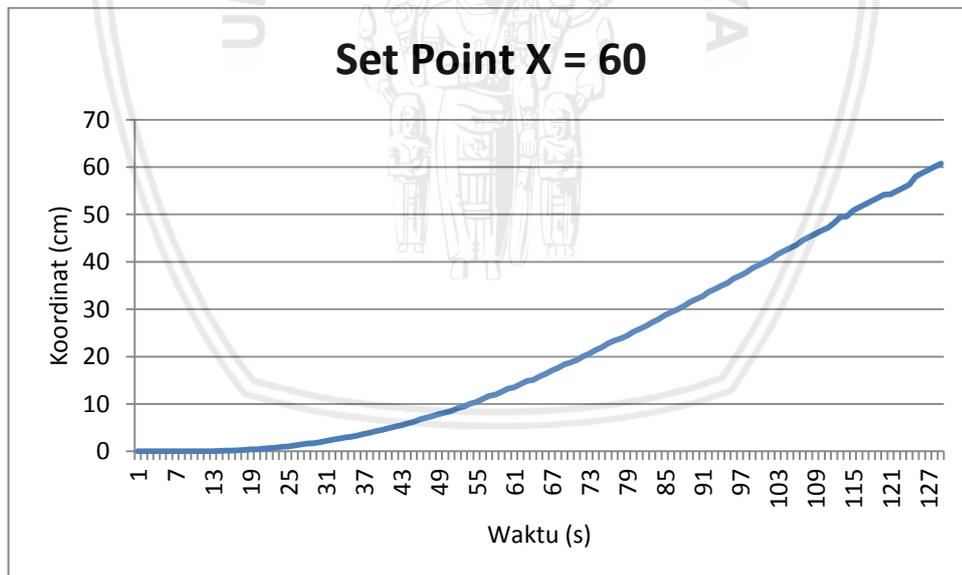


Gambar 6.5 Pengujian 1 Set point X *Proportional*

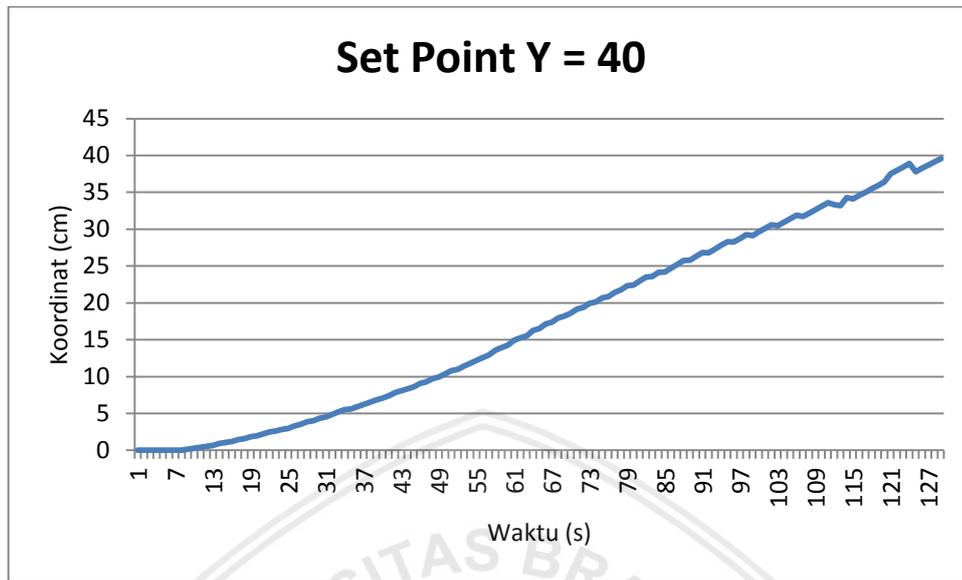


Gambar 6.6 Pengujian 1 Set point Y *Proportional*

Percobaan 2
Parameter P = 3.0

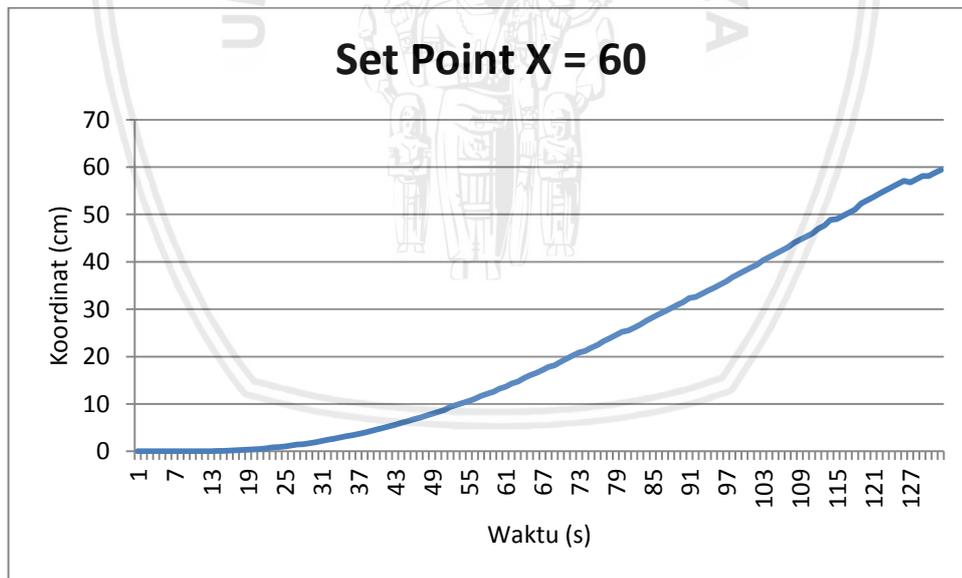


Gambar 6.7 Pengujian 2 Set point X *Proportional*

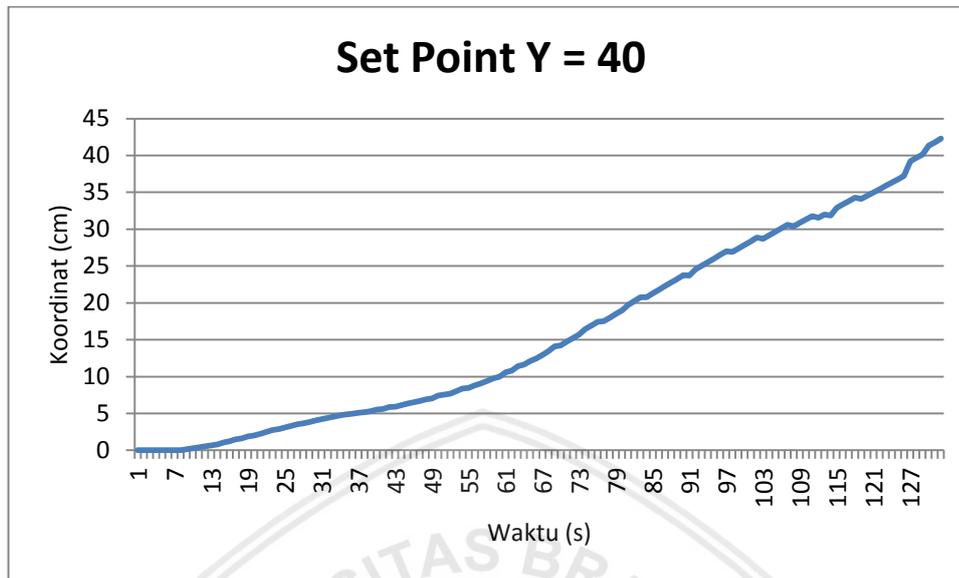


Gambar 6.8 Pengujian 2 Set point Y *Proportional*

Percobaan 3
Parameter P = 4.8



Gambar 6.9 Pengujian 3 Set point X *Proportional*



Gambar 6.10 Pengujian 3 Set point Y *Proportional*

Percobaan 4

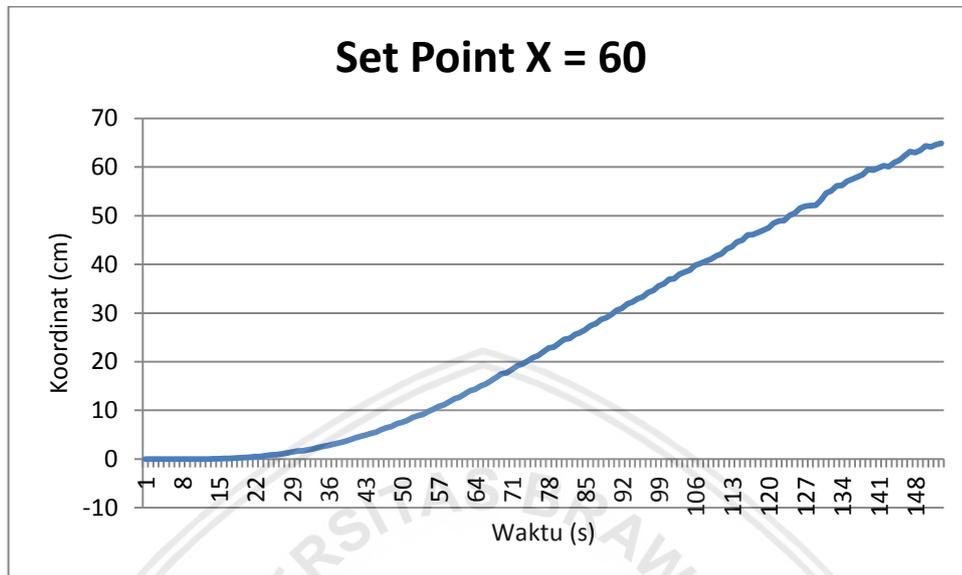
Pada percobaan ini menggunakan kode program yang ada parameter PID.

Tabel 6.3 Kode Program PID

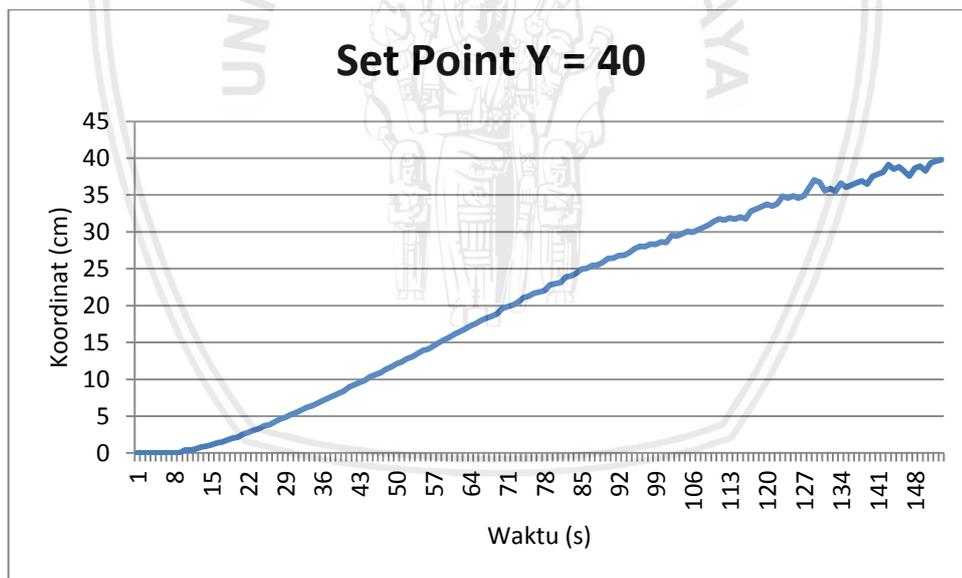
No.	Source Code
1	P = Kp * error;
2	D1 = Kd*8;
3	D2 = D1 / Ts;
4	D3 = error - last_error;
5	D = D2 * D3;
6	I1 = Ki/8;
7	I2 = error + last_error;
8	I3 = I1 * I2;
9	I = I3 * Ts;
10	last_error = error;
11	Pd = P + D;
12	MV = Pd + I; //output Proporsional
13	if (MV>=Max_MV && MV <= Max_MV)
14	{
15	pid_l = max_pwm_ref + MV;
16	pid_r = max_pwm_ref - MV;
17	if (pid_l < 0) pid_l = 0;
18	if (pid_l > 200) pid_l = 200;
19	if (pid_r < 0) pid_r = 0;
20	if (pid_r > 200) pid_r = 200;
21	forward(pid_l,pid_r);
22	}
23	else
24	{
25	pid_l = 75; pid_r = 75;
26	forward(pid_l,pid_r);
27	}



Parameter P = 4.0 , D = 1.5



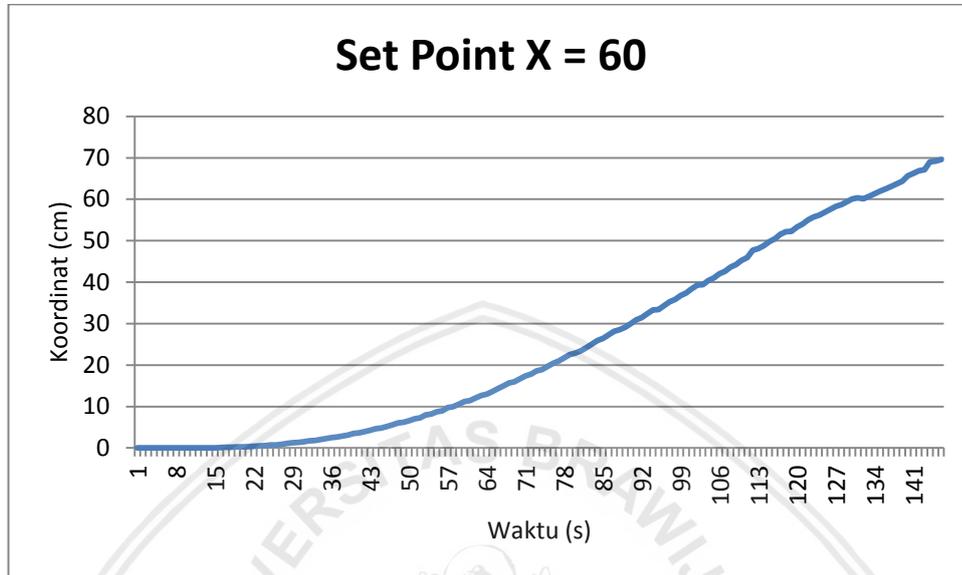
Gambar 6.11 Pengujian 4 Set point X PD



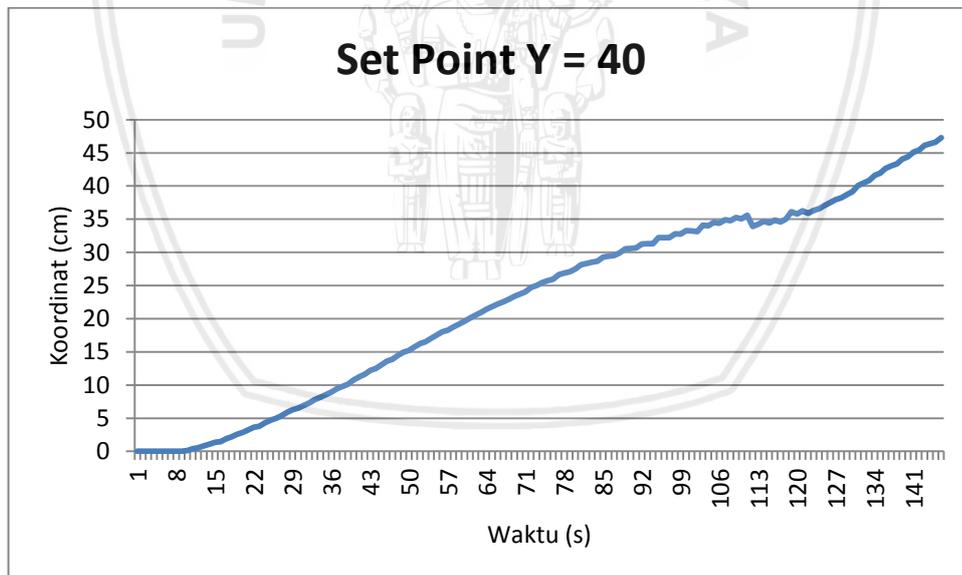
Gambar 6.12 Pengujian 4 Set point Y PD

Percobaan 5

Parameter $P = 1.4$, $I = 0.5$



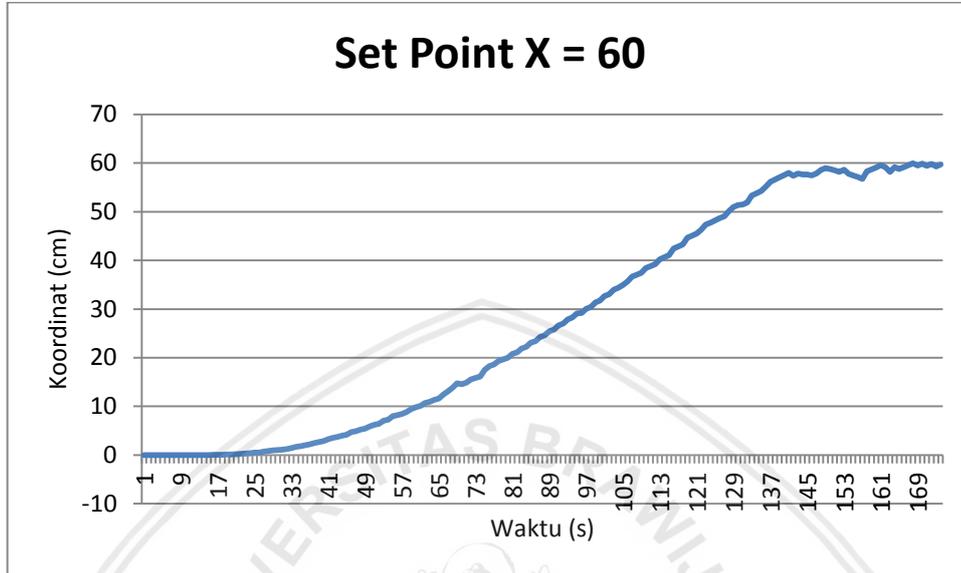
Gambar 6.13 Pengujian 5 Set point X PI



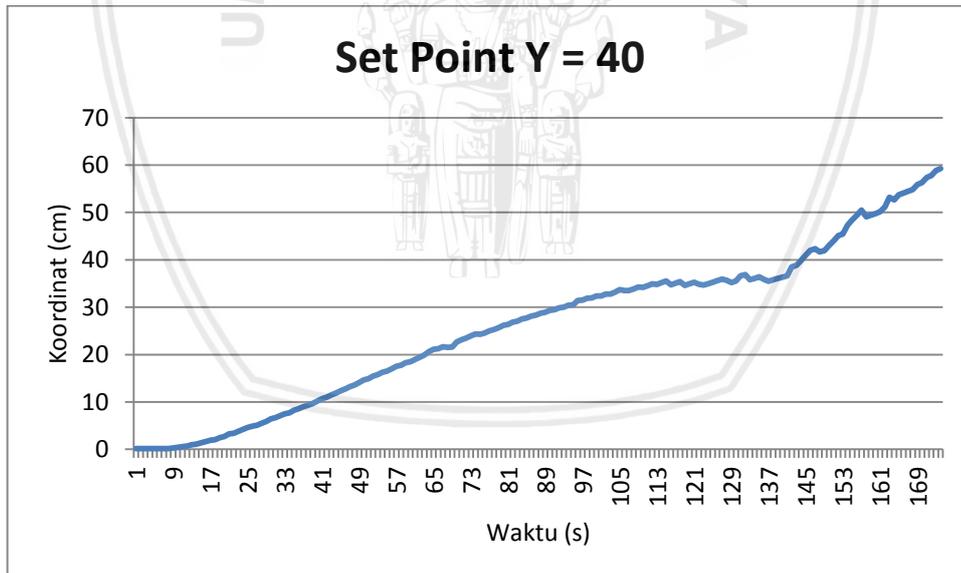
Gambar 6.14 Pengujian 5 Set point Y PI

Percobaan 6

Parameter $P = 2.4$, $I = 0.5$, $D = 1.6$



Gambar 6.15 Pengujian 6 Set point X PID



Gambar 6.16 Pengujian 6 Set point Y PID

Dari semua pengujian parameter kontrol yang telah diuji tersebut. Parameter yang menunjukkan nilai terbaik yaitu pada percobaan kedua dengan nilai parameter kontrol, $P = 3,0$. Nilai parameter tersebut dipilih karena dari nilai set point yang ditentukan telah terpenuhi. Nilai parameter Proportional dipakai pada sistem ini berdasarkan dari tercapainya set point dan juga waktu yang dihasilkan lebih cepat dari pada nilai parameter uji lainnya. Sedangkan untuk

parameter uji yang ada nilai parameter yang ada melebihi nilai set point yang telah ditentukan.

6.4 Pengujian Keseluruhan Sistem

Dalam pengujian keseluruhan sistem ini, robot dicoba untuk melakukan pergerakan menuju target tujuan yang telah diberikan oleh *user* dan juga di berikan nilai parameter *proportional*. Dalam pengujian keseluruhan sistem ini berguna untuk melihat respon pergerakan robot menuju koordinat target tujuan.

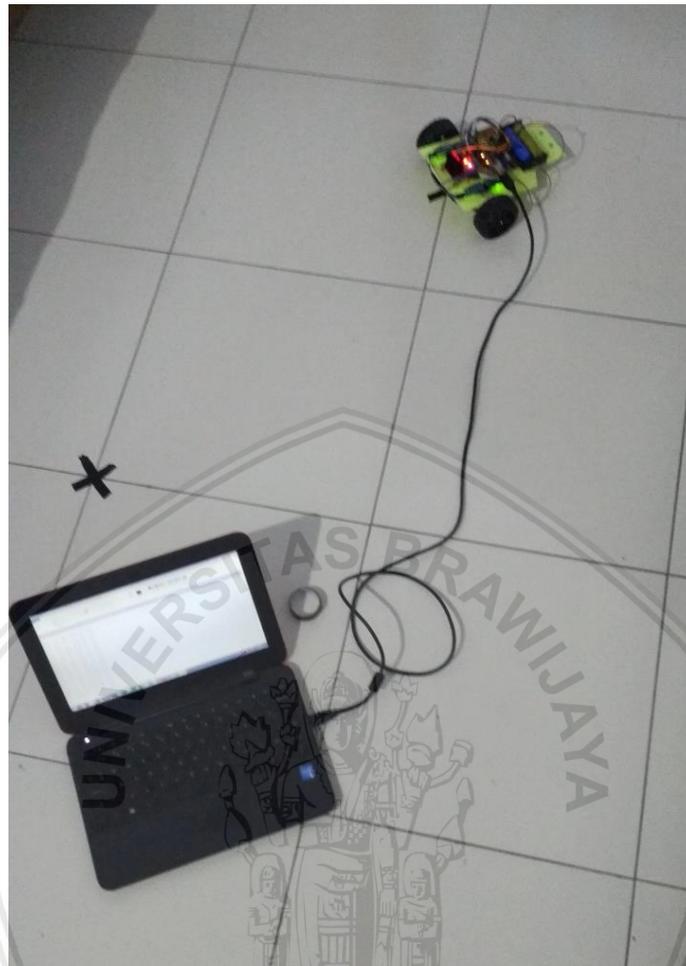
6.4.1 Tujuan Pengujian

Untuk mengetahui apakah semua sistem sudah dapat berfungsi dengan baik sehingga robot dapat bergerak menuju koordinat target.

6.4.2 Prosedur Pengujian

1. Menghubungkan robot dengan laptop menggunakan kabel USB.
2. Membuka *source code* yang dibuat kontrol *proportional* untuk pergerakan mobile robot menggunakan metode *odometry* menuju koordinat target. Lakukan *compile* program ke mikrokontroler Arduino UNO.
3. Berikan nilai koordinat target X,Y pada *serial monitor* IDE Arduino.
4. Amati nilai *output set point* posisi koordinat X,Y pada *serial monitor*.
5. Bandingkan hasil *output* dari pengukuran robot yang ada pada *serial monitor* dengan ukuran asli pergerakan robot.

Berikut Gambar 6.17 robot melakukan pergerakan menuju target tujuan X=60, Y=40.



Gambar 6.17 Pengujian Pergerakan Robot Menuju Target

6.4.3 Hasil dan Analisis Pengujian

Hasil dari pengujian keseluruhan sistem pada pergerakan *mobile robot* dengan metode *odometry* untuk menuju koordinat target tujuan dapat dilihat pada tabel 6.3.

Tabel 6.4 Hasil Pengujian Pengukuran Koordinat

Koordinat X, Y	Parameter PID	Percobaan	Hasil Pengukuran Robot		Hasil pengukuran <i>Real</i>		Selisih	
			X (cm)	Y (cm)	X (cm)	Y (cm)	X (cm)	Y (cm)
X = 60 Y = 40	P = 3.0	1	59.65	40.04	58	42	1.65	-1.96
		2	59.44	41.08	58	43	1.44	-1.92
		3	59.89	41.39	60	42	-0.11	-0.61

X = 80 Y = 60	P = 3.0	1	79.78	60.09	76	60	3.78	0.09
		2	78.92	61.92	80	63	-1.08	-1.08
		3	79.49	61.58	77	62	2.49	-0.42
X = 0 Y = 70	P = 3.0	1	-0.95	69.80	0	75	-0.95	-5.2
		2	0.00	69.67	0	77	0	-7.33
		3	0.00	69.67	0	75	0	-5.33
X = 70 Y = 60	P = 3.0	1	69.85	60.47	67	63	2.85	-2.53
		2	69.44	60.12	68	61	1.44	-0.88
		3	69.44	60.12	67	63	2.44	-2.88
X = 70 Y = 50	P = 3.0	1	69.85	49.69	68	49	1.85	0.69
		2	69.63	49.53	67	51	2.63	-1.47
		3	69.72	51.04	70	47	-0.28	4.04
X = 60 Y = 50	P = 3.0	1	59.77	51.74	60	49	-0.23	2.74
		2	59.74	50.31	57	51	2.74	-0.69
		3	59.56	51.57	57	52	2.56	-0.43
X = 80 Y = 40	P = 3.0	1	80.42	39.35	79	38	1.42	1.35
		2	79.56	41.65	78	38	1.56	3.65
		3	80.00	40.50	80	37	0	3.5
X = 80 Y = 50	P = 3.0	1	79.58	50.33	80	46	-0.42	4.33
		2	79.35	50.18	80	48	-0.65	2.18
		3	80.14	49.17	80	50	0.14	-0.83
X = 90 Y = 50	P = 3.0	1	89.70	51.73	89	48	0.7	3.73
		2	89.70	51.73	92	46	-2.3	5.73
		3	89,70	51.73	92	46	-2.3	5.73
X = 100 Y = 80	P = 3.0	1	99.26	81.31	103	78	-3.74	3.31
		2	99.68	81.66	102	78	-2.32	3.66
		3	99.47	81.49	100	80	-0.53	1.49
X = 100 Y = 50	P = 3.0	1	99.67	50.46	100	48	-0.33	2.46
		2	100.95	49.40	100	50	0.95	-0.6
		3	100.95	49.40	100	50	0.95	-0.6
X = 85 Y = 50	P = 3.0	1	85.93	49.55	86	48	-0.07	1.55
		2	86.16	49.69	85	46	1.16	3.69
		3	84.55	51.87	85	50	-0.45	1.87
X = 120 Y = 60	P = 3.0	1	118.62	62.10	122	60	-3.38	2.1
		2	121.24	59.32	124	60	-2.76	-0.68
		3	121.49	59.44	120	60	1.49	-0.56
X = 130 Y = 60	P = 3.0	1	129.54	61.21	130	57	-0.46	4.21
		2	130.98	59.73	131	60	-0.02	-0.27
		3	129.30	61.10	129	65	0.3	-3.9
X = 110 Y = 70	P = 3.0	1	109.78	71.54	108	72	1.78	-0.46
		2	110.17	69.67	108	70	2.17	-0.33
		3	110.40	69.82	105	73	5.4	-3.18

X = 95 Y = 55	P = 3.0	1	94.76	59.93	97	54	-2.24	5.93
		2	94.48	56.21	98	52	-3.52	4.21
		3	94.25	56.07	94	55	0.25	1.07
X = 130 Y = 50	P = 3.0	1	129.73	50.83	133	45	-3.27	5.83
		2	129.55	54.87	132	47	-2.45	7.87
		3	129.91	52.95	130	50	-0.09	2.95
X = 130 Y = 80	P = 3.0	1	129.72	82.04	132	80	-2.28	2.04
		2	129.78	79.62	130	84	-0.22	-4.38
		3	129.72	82.04	130	84	-0.28	-1.96
X = 90 Y = 60	P = 3.0	1	89.21	61.65	94	60	-4.79	1.65
		2	91.77	59.80	90	62	1.77	-2.2
		3	89.71	60.21	95	62	-5.29	-1.79
X = 80 Y = 70	P = 3.0	1	79.69	70.91	82	71	-2.31	-0.09
		2	80.34	69.56	84	69	-3.66	0.56
		3	80.34	69.56	81	75	-0.66	-5.44
Rata – rata selisih X dan Y							-0.12	0.57

Dari Tabel 6.3 merupakan hasil dari pengujian keseluruhan sistem yang ada. Hasil pengukuran koordinat target oleh robot kemudian dibandingkan dengan hasil ukuran asli pada pergerakan robot. Rata – rata hasil selisih perbandingan perhitungan robot dengan ukuran asli yaitu X = -0.12 cm dan Y = 0.57 cm. Dengan nilai rata – rata tersebut selisih perbandingannya tidak terlalu besar dan seluruh sistem telah bekerja dengan baik.

BAB VII PENUTUP

Bab 7 ini berisi mengenai kesimpulan dari hasil analisis pengujian dan penelitian yang telah dilakukan pada bab sebelumnya dan saran yang dapat digunakan untuk mengembangkan penelitian selanjutnya.

7.1 Kesimpulan

Berdasarkan dari pengujian dan analisis yang telah dilakukan selama penelitian pada tugas akhir ini maka diperoleh kesimpulan sebagai berikut :

1. Untuk mengimplementasikan metode odometry pada mobile robot yang diterapkan dalam penelitian ini yaitu dengan menggunakan sensor rotary encoder. Sensor rotary encoder ini bekerja dengan cara menghitung putaran disk encoder yang terdapat lubang – lubang sebagai masukan data untuk sensor encoder. Dari pembacaan sensor ini sistem dapat memperoleh data dan akan diproses dengan metode odometry. Apabila pembacaan sensor kurang akurat maka implementasi sistem metode odometry tidak akan efektif. Dari hasil pengujian dan analisis dapat dilihat bahwa metode odometry dapat diterapkan secara efektif pada sistem dengan melihat nilai rata – rata perbandingan selisih perhitungan koordinat target yang dihitung oleh robot dan pengukuran asli tidak memiliki selisih nilai yang jauh.
2. Robot dapat bergerak menuju target tujuan dengan menerapkan metode odometry yang telah diuji. Data hasil pembacaan sensor encoder diproses dengan metode odometry dengan menghitung jarak. Jarak ini diperoleh dari hasil pembacaan sensor encoder. Dari hasil menghitung jarak didapatkan nilai theta yang akan digunakan untuk menghitung posisi koordinat target X dan Y. Hasil pengujian keseluruhan sistem pergerakan robot, robot dapat bergerak sesuai dengan target yang telah diberikan.
3. Nilai parameter kontrol *proportional* yang digunakan diperoleh dengan cara manual yaitu menguji satu persatu parameter nilai KP yang akan diuji. pengujian kontrol ini membandingkan kontrol proportional dengan kontrol lainnya (PI, PD, dan PID). Pengujian ini untuk melihat respon sistem dapat sesuai dengan yang diinginkan. Dengan memasukan nilai parameter KP saja dapat dilihat pada pengujian sistem kontrol, sistem sudah sesuai dengan yang diinginkan. Nilai tersebut sudah sesuai dengan set point parameter yang telah ditentukan. Melihat respon tersebut sistem ini hanya menggunakan nilai parameter $KP = 3.0$.

4. Untuk membuat sistem kontrol *proportional* yang sesuai untuk mengurangi error yaitu dengan memastikan nilai parameter KP yang telah diuji sesuai dengan set point yang telah diberikan. Nilai parameter yang diberikan pada robot saat pengujian dengan menguji satu persatu nilai parameter KP yang dimasukkan. Dengan melihat hasil pengujian yang telah dilakukan nilai parameter $KP = 3.0$ sesuai dengan sistem untuk menggerakkan robot menuju koordinat target tujuan. Hasil pengujian menunjukkan selisih dari pengukuran robot dan pengukuran asli tidak berbeda jauh dan menunjukkan *error* yang dihasilkan sedikit.

7.2 Saran

Berdasarkan dari pengujian dan analisis yang telah dilakukan selama peneliatian pada tugas akhir ini maka ada beberapa saran untuk pengembangan alat kedepannya :

1. Untuk kedepannya pada pengiriman koordinat target tujuan dapat menggunakan aplikasi dan komunikasi tanpa menggunakan kabel.
2. Pada pengembangan selanjutnya sensor rotary encoder dapat diperbarahui dengan sensor optical encoder yang memiliki akurasi lebih tinggi.

DAFTAR PUSTAKA

- Doh, Nakju Lett, Howie Choset dan Wan Kyun Yung, 2006 yang berjudul "Relative localization using path odometry information"
- Pei, Yanming dan Lindsay Kleeman, 2015 yang berjudul "Robot calibration of sensor poses and region based odometry using offline optimisation of mapinformation"
- Prakash, D. Vidhya dan A. Elango 2016 dalam jurnalnya yang berjudul "Studies on affecting factors of wheel slip and odometry error on the performance of wheeled mobile robots"
- Pengertian Fungsi dan Kegunaan Arduino, <https://ariefeeiiggeennblog.wordpress.com/2014/02/07/pengertian-fungsi-dan-kegunaan-arduino/>, dikunjungi februari 2018
- Teori Motor DC Dan Jenis-Jenis Motor DC, <http://elektronika-dasar.web.id/teori-motor-dc-dan-jenis-jenis-motor-dc/>, dikunjungi februari 2018
- Mesriah, Ria. Jenis-jenis atau Macam-macam Robot, <http://www.matadunia.id/2016/04/jenis-jenis-atau-macam-macam-robot.html>, dikunjungi februari 2018
- Kho, Dickson. Pengertian Baterai dan Jenis-jenisnya <http://teknikelektronika.com/pengertian-baterai-jenis-jenis-baterai/>, dikunjungi februari 2018
- Teknik Elektronika. (2017). *Pengertian Motor DC dan Prinsip Kerjanya*. Retrieved from Teknik Elektronika: <http://teknikelektronika.com/pengertian-motor-dc-prinsip-kerja-dc-motor/>
- Manika, A. (2016, November 25). *Kenapa Baterai pada Akhirnya Mati?* Diambil kembali dari blajar: <http://blajar.org/pertanyaan/10814/>
- Bayu. Sandi Marta, Fernando. Ardilla, A.R. Anom Besari, (2011). Path Tracking Pada Mobile Robot Dengan Umpan Balik Odometry

LAMPIRAN

Source Code

No.	Kode Program
1	<code>#include <math.h> //add library untuk sin cos dan matematika</code>
2	<code>#define en1 6 //pwm pin motor kiri</code>
3	<code>#define in1 7 //in 1 pin motor kiri</code>
4	<code>#define in2 8 //in 2 pin m motor kiri</code>
5	<code>#define en2 11 //pwm pin motor kanan</code>
6	<code>#define in3 9</code>
7	<code>#define in4 10</code>
8	<code>const byte interruptPin3 = 3; //pin interrupt kiri</code>
9	<code>const byte interruptPin2 = 2; //pin interrupt kanan</code>
10	<code>int rotaryRight, rotaryLeft;</code>
11	<code>double rRoda = 3.25;</code>
12	<code>double resolusiEncLeft = 75;</code>
13	<code>double resolusiEncRight = 75;</code>
14	<code>double wheelBase = 20.0;</code>
15	<code>double currentDistance, sudutOrientasi, heading, xPos, yPos;</code>
16	<code>double errorX, errorY, targetDistance, beta, alpha, lamda;</code>
17	<code>double xDistance = 0; //samping</code>
18	<code>double yDistance = 0; //ke atas</code>
19	<code>double error, last_error, MV, D, D1, D2, D3, I, I1, I2, I3, P, Pd, pid_l, pid_r;</code>
20	<code>double Max MV = 1000;</code>
21	<code>double max_pwm_ref = 180;</code>
22	<code>double Kp=3.0;</code>
23	<code>double Kd=0.0 ;</code>
24	<code>double Ki=0.0;</code>
25	<code>double Ts=0.1; //waktu menuju zero point</code>
26	<code>boolean starting = false;</code>
27	<code>String valSplit[3];</code>
28	<code>String dataSerial;</code>
29	<code>void setup()</code>
30	<code>{</code>
31	<code> Serial.begin(9600); //baud rate serial</code>
32	<code> pinMode(A3, INPUT); //input untuk data 2 rotary encoder</code>
33	<code> pinMode(2, INPUT); //input untuk interrupt 1 rotary encoder</code>
34	<code> pinMode(A2, INPUT);</code>
35	<code> pinMode(3, INPUT);</code>

36	for(int i=7; i<=10; i++)
37	{
38	pinMode(i, OUTPUT); // pin motor output
39	}
40	attachInterrupt(digitalPinToInterrupt(interruptPin3), countRight, CHANGE);
41	attachInterrupt(digitalPinToInterrupt(interruptPin2), countLeft, CHANGE);
42	}
43	void loop()
44	{
45	while(!starting)
46	{
47	if(Serial.available())
48	{
49	char c = Serial.read();
50	if(c== '\n')
51	{
52	parseData(dataSerial, " ");
53	dataSerial = "";
54	xDistance = valSplit[0].toFloat();
55	yDistance = valSplit[1].toFloat();
56	Serial.print(xDistance); Serial.print(" ");
57	Serial.println(yDistance);
58	delay(2000);
59	starting = true;
60	}
61	Else
62	{
63	dataSerial += String(c);
64	}
65	}
66	while(starting)
67	{
68	double leftRotaryMm = 2 * 3.14 * rRoda * (double(rotaryLeft) / resolusiEncLeft);
69	double rightRotaryMm = 2 * 3.14 * rRoda * (double(rotaryRight) / resolusiEncRight);
70	currentDistance = (leftRotaryMm + rightRotaryMm) / 2;
71	sudutOrientasi = (leftRotaryMm - rightRotaryMm) / wheelBase;
72	heading = sudutOrientasi*180/3.14;
73	xPos = currentDistance * sin(sudutOrientasi);
74	yPos = currentDistance * cos(sudutOrientasi);
75	errorX = xDistance - xPos;

76	<code>errorY = yDistance - yPos;</code>
77	<code>targetDistance = sqrt((pow(errorX,2)+0.5) + (pow(errorY,2)+0.5));</code>
78	<code>double radianBeta = asin(errorX/targetDistance);</code>
79	<code>beta = radianBeta * 180/3.14;</code>
80	<code>alpha = beta - heading;</code>
81	<code>error = alpha;</code>
82	<code>if(xPos >= xDistance && yPos >= yDistance)</code>
83	<code>{</code>
84	<code>while(1)</code>
85	<code>{</code>
86	<code>backward(0,0);</code>
87	<code>}</code>
88	<code>}</code>
89	<code>P = Kp * error; //output nilai Proportional</code>
90	<code>D1 = Kd*8; // konstanta derivative dijadikan 8 bit, dikarenakan mikrokontroller hanya 8 bit</code>
91	<code>D2 = D1 / Ts; //KD/Time sampling</code>
92	<code>D3 = error - last_error; //pengurangan nilai error sekarang dengan nilai error sebelumnya</code>
93	<code>D = D2 * D3; //output nilai Derivative</code>
94	<code>I1 = Ki/8; //Ki di ubah ke 8 bit</code>
95	<code>I2 = error + last_error; //summing dari nilai error sekarang dan nilai error sebelumnya</code>
96	<code>I3 = I1 * I2;</code>
97	<code>I = I3 * Ts; // output nilai Integral</code>
98	<code>last_error = error; //error sekarang dijadikan error sebelumnya, kemudian akan di loop ke atas lagi</code>
99	<code>Pd = P + D;</code>
100	<code>MV = Pd + I; //output PID</code>
101	<code>//=====//</code>
102	<code>if(MV>=-Max MV && MV <= Max MV)</code>
103	<code>{</code>
104	<code>pid_l = max_pwm_ref + MV;</code>
105	<code>pid_r = max_pwm_ref - MV;</code>
106	<code>if (pid_l < 0) pid_l = 0;</code>
107	<code>if (pid_l > 200) pid_l = 200;</code>
108	<code>if (pid_r < 0) pid_r = 0;</code>
109	<code>if (pid_r > 200) pid_r = 200;</code>
110	<code>forward(pid_l,pid_r);</code>
111	<code>}</code>
112	<code>Else</code>
113	<code>{</code>
114	<code>pid_l = 75; pid_r = 75;</code>
115	<code>forward(pid_l,pid_r);</code>

116	}
117	//Serial.print(error); Serial.print("\t");
118	//Serial.print(MV); Serial.print("\t");
119	//Serial.print(pid_l); Serial.print("\t");
120	//Serial.print(pid_r); Serial.print("\t");
121	Serial.print(xPos); Serial.print("\t");
122	Serial.print(yPos); Serial.print("\t");
123	Serial.println();
124	}
125	}
126	void countLeft()
127	{
128	if (digitalRead(2) == HIGH)
129	{
130	if (digitalRead(A3) == LOW)
131	{
132	rotaryLeft++; // CW
133	}
134	else
135	{
136	rotaryLeft--; // CCW
137	}
138	}
139	else // must be a high-to-low edge on channel A
140	{
141	// check channel B to see which way encoder is turning
142	if (digitalRead(A3) == HIGH)
143	{
144	rotaryLeft++; // CW
145	}
146	else
147	{
148	rotaryLeft--; // CCW
149	}
150	}
151	}
152	void countRight()
153	{
154	if (digitalRead(3) == HIGH)
155	{
156	if (digitalRead(A2) == LOW)

157	{
158	rotaryRight++; // CW
159	}
160	else
161	{
162	rotaryRight--; // CCW
163	}
164	}
165	else // must be a high-to-low edge on channel A
166	{
167	// check channel B to see which way encoder is turning
168	if (digitalRead(A2) == HIGH)
169	{
170	rotaryRight++; // CW
171	}
172	else
173	{
174	rotaryRight--; // CCW
175	}
176	}
177	}
178	void forward(int pwm1, int pwm2)
179	{
180	digitalWrite(in1, HIGH);
181	digitalWrite(in2, LOW);
182	analogWrite(en1, pwm1);
183	digitalWrite(in3, HIGH);
184	digitalWrite(in4, LOW);
185	analogWrite(en2, pwm2);
186	}
187	void backward(int pwm1, int pwm2)
188	{
189	digitalWrite(in1, LOW);
190	digitalWrite(in2, HIGH);
191	analogWrite(en1, pwm1);
192	digitalWrite(in3, LOW);
193	digitalWrite(in4, HIGH);
194	analogWrite(en2, pwm2);
195	}
196	void parseData(String text, String key)
197	{

198	int countSplitSecond=0;
199	int lastIndexSecond=0;
200	text += key;
201	//Serial.println(text);
202	for(int j = 0; j < text.length(); j++)
203	{
204	if(text.substring(j, j+1) == key)
205	{
206	valSplit[countSplitSecond] = text.substring(lastIndexSecond,j);
207	lastIndexSecond = j + 1;
208	//Serial.print(countSplitSecond);
209	//Serial.print(":");
210	//Serial.println(valSplit[countSplitSecond]);
211	countSplitSecond++;
212	}
213	}
214	}

