

OPTIMASI SISTEM NAVIGASI PADA ROBOT PENGANTAR MAKANAN

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

Sony Montela NIM : 1051927

Via Twoifia NIM : 1051928

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2023**

LEMBAR PENGESAHAN

**OPTIMASI SISTEM NAVIGASI PADA ROBOT
PENGANTAR MAKANAN**

Oleh :

Sony Montela/1051927

Via Twoifia/1051928

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

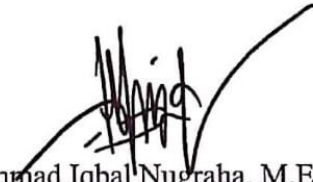
Menyetujui,

Pembimbing 1



(Indra Dwisaputra, M.T.)

Pembimbing 2



(Muhammad Iqbal Nugraha, M.Eng.)

Penguji 1



(Aan Febriansyah, M.T.)

Penguji 2



(Yuchi, M.T.)

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : Sony Montela NIM : 1051927

Nama Mahasiswa 2 : Via Twoifia NIM : 1051928

Dengan Judul : Optimasi Sistem Navigasi pada Robot Pengantar Makanan

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 14 Februari 2023

Nama Mahasiswa

Tanda tangan

1. Sony Montela



.....

2. Via Twoifia



.....

ABSTRAK

Peranan robot semakin hari semakin banyak manfaatnya. Salah satunya adalah pada bidang industri kuliner, yang memanfaatkan teknologi robotika untuk menyelesaikan suatu pekerjaan tertentu, ditambah lagi dengan maraknya kasus Covid-19, maka perlu adanya sebuah robot pengantar makanan yang bertugas untuk mempermudah pekerjaan manusia dan meminimalisir kontak fisik antar manusia. Robot yang dibuat berjalan mengikuti garis menggunakan sensor garis 14 kanal dimultiplekser melalui jalur input ADC 10bit. Pergerakan robot mengikuti garis dikontrol berbasis sistem kontrol PID dengan performa terbaik diperoleh dengan nilai $K_p = 1.0$, $K_i = 0,015$ dan $K_d = 1,0$ setelah dilakukan penalaan secara manual. Sistem perjalanan robot menuju lokasi menerapkan metode navigasi berbasis rotary encoder untuk mengukur jarak tempuh dan arah pergerakan, dan dipantau melalui aplikasi Processing. Berdasarkan hasil percobaan secara menyeluruh, robot dapat mengantarkan makanan ke meja yang diinginkan dengan mengikuti garis yang telah dibuat dengan lamanya perjalanan sangat bergantung kepada beban yang diberikan dan dengan maksimal beban seberat 3kg.

Kata Kunci: Robot, Motor DC, Sensor Garis, PID, Software Processing.

ABSTRACT

The role of robots is getting more and more useful day by day. One of them is in the culinary industry, which utilizes robotics technology to complete a certain job, coupled with the rampant Covid-19 cases, it is necessary to have a food delivery robot tasked with facilitating human work and minimizing physical contact between humans. The created robot walks along the line using a 14-channel line sensor in the multiplexer via a 10bit ADC input line. The movement of the robot following the controlled line based on the PID control system with the best performance was obtained with the values $K_p = 1.0$, $K_i=0.015$ and $K_d = 1.0$ after manual tuning. The robot's journey system to the location applies a rotary encoder-based navigation method to measure mileage and direction of movement, and is monitored through the Processing application. Based on the results of a thorough experiment, the robot can deliver food to the desired table by following a line that has been made with the length of the trip very dependent on the load given and with a maximum load of 3kg.

Keywords: Robot, DC Motor, Line Sensor, PID, Processing Software.

KATA PENGANTAR

Assalamu'alaikum warrahmatullahi wabarakatuh.

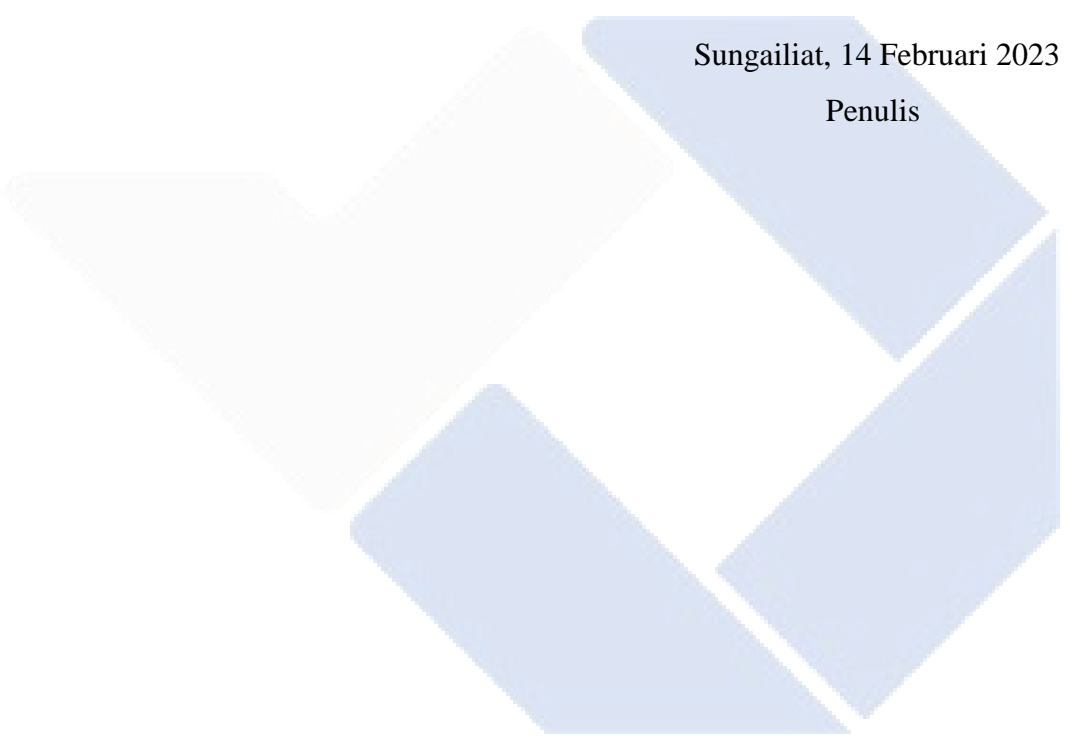
Segala puji dan syukur atas kehadiran Allah SWT., yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Laporan Proyek Akhir yang berjudul “Optimasi Sistem Navigasi pada Robot Pengantar Makanan” tepat pada waktunya. Proyek Akhir ini disusun untuk memenuhi salah satu syarat kelulusan Sarjana Terapan Teknik Elektronika Politeknik Manufaktur Negeri Bangka.

Laporan Proyek Akhir ini dapat diselesaikan berkat adanya usaha dan kerja sama tim yang baik serta adanya bantuan dan dukungan dari berbagai pihak terkait. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua penulis yang telah memberikan dukungan, semangat, dan doa sehingga Laporan Proyek Akhir dapat terselesaikan.
2. Bapak I Made Andik Setiawan, M.Eng, Ph.D. selaku Direktur di Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Indra Dwisaputra, M.T selaku pembimbing 1 yang telah meluangkan waktu, tenaga dan pikiran untuk memberi pengarahan dalam penulisan Laporan Proyek Akhir.
4. Bapak Muhammad Iqbal Nugraha, M.Eng selaku pembimbing 2 yang telah memberikan saran dan solusi selama proses pembuatan Proyek Akhir dan penyusunan laporan.
5. Seluruh staf pengajar di Politeknik Manufaktur Negeri Bangka Belitung.
6. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung.
7. Semua pihak yang telah memberikan bantuan selama proses pembuatan Proyek Akhir yang tidak bisa disebutkan satu persatu.

Dalam pembuatan Laporan Proyek Akhir ini, penulis menyadari bahwa laporan ini masih jauh dari kata sempurna, baik penyusunan maupun penulisannya. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun demi kebaikan penulis kedepannya.

Akhir kata penulis ucapkan terima kasih dan mohon maaf yang sebesar-besarnya. Semoga laporan ini dapat bermanfaat dan berguna sebagaimana yang diharapkan.



Sungailiat, 14 Februari 2023

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERYATAAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN.....	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	3
1.4. Tujuan.....	3
BAB II DASAR TEORI	4
2.1. Robot <i>Line Follower</i>	4
2.2. Mikrokontroler Arduino Mega 2560.....	5
2.3. Motor DC PG36 24VDC.....	6
2.4. <i>Driver</i> Motor IBT-2 BTS7960	8
2.6. Sensor <i>Line Follower 14 Channel</i>	9
2.7. Modul Nirkabel HC-12	10
2.8. USB to TTL <i>Converter</i> CP2102.....	10
2.9. PID (<i>Proportional Integral Derivative</i>)	11
2.10. <i>Software Processing</i>	12
BAB III METODE PELAKSANAAN	13
3.1. Studi Literatur	14
3.2. Tahapan Desain Alat	14
3.2.1. Desain <i>Hardware</i>	15
3.2.2. Desain <i>Software</i>	16

3.3. Pengujian Komponen Robot	17
3.4. Pembuatan Kontruksi Alat	17
3.5. Pembuatan Rangkaian Elektrik	17
3.6. Pembuatan Program	18
3.7. Pengujian Keseluruhan	18
3.8. Pengambilan dan Analisa Data	20
3.9. Pembuatan Laporan Proyek Akhir	20
BAB IV PEMBAHASAN	21
4.1. Keterangan Alat.....	21
4.2. Perencanaan dan Pembuatan Robot	23
4.3. Pengujian Elektrik Robot	26
4.3.1. Konfigurasi Awal Modul HC-12.....	26
4.3.2. Pengujian Koneksi Modul HC-12 dengan USB to TTL CP2102	27
4.3.3. Pengujian Arah Putaran Sensor <i>Encoder</i> CW dan CCW	28
4.3.4. Pengujian Pembacaan Jarak pada Roda	29
4.3.5. Pengujian Sensor <i>Proximity</i>	30
4.3.6. Blok Diagram Sistem	31
4.3.7. Pengujian Persamaan Linier antara PWM dan RPM	32
4.3.8. Pengujian Sensor Garis	32
4.3.8.1. Pengujian Sensor Mengetahui Nilai ADC.....	33
4.3.8.2. Pengujian Hasil <i>Mapping</i> Nilai Sensor	34
4.4. Pengujian Aplikasi	35
4.5. Pengujian Kontrol PID Terhadap Sensor Garis	36
4.6. Pengujian Keseluruhan.....	37
Bab V KESIMPULAN DAN SARAN	40
5.1. Kesimpulan.....	40
5.2. Saran.....	41
DAFTAR PUSTAKA	42
LAMPIRAN	

DAFTAR TABEL

Tabel 4.1 Pin Konfigurasi antara Modul HC-12 dengan Arduino	26
Tabel 4.2 Hasil Pengujian Jarak Roda	30
Tabel 4.3 Hasil Pengujian Jarak Jangkauan Sensor <i>Proximity</i>	31
Tabel 4.4 Hasil Persamaan Linier PWM dan RPM	32
Tabel 4.5 Hasil Nilai ADC (<i>Analog Digital Converter</i>) pada Sensor	33
Tabel 4.6 Hasil <i>Mapping</i> Nilai Sensor	34
Tabel 4.7 Hasil Percobaan Perbandingan PID Terhadap Sensor Garis	36
Tabel 4.8 Hasil Pengujian Menentukan Meja Tanpa Beban	38
Tabel 4.9 Hasil Pengujian Menentukan Meja Menggunakan Beban	38
Tabel 4.10 Hasil Waktu Tempuh Robot Pengantar Makanan.....	39

DAFTAR GAMBAR

Gambar 2.1 Robot <i>Line Follower</i>	4
Gambar 2.2 Arduino Mega 2560	5
Gambar 2.3 Motor DC	6
Gambar 2.4 Aturan Tangan Kiri <i>Fleming</i>	7
Gambar 2.5 <i>Stator</i> Motor DC	7
Gambar 2.6 <i>Rotor</i> atau Jangkar Motor DC	7
Gambar 2.7 <i>Komutator</i> Motor DC	8
Gambar 2.8 Sikat (<i>Brush</i>) Motor DC	8
Gambar 2.9 <i>Driver</i> Motor IBT-2	9
Gambar 2.10 Bentuk Fisik dan Simbol <i>Photodiode</i>	9
Gambar 2.11 Sensor <i>Line Follower</i> 14 <i>Channel</i>	10
Gambar 2.12 Modul Nirkabel HC-12	10
Gambar 2.13 USB to TTL <i>Converter</i> CP2102	11
Gambar 2.14 Blok Diagram Kontrol PID	11
Gambar 3.1 Diagram Alir Metode Pelaksanaan	14
Gambar 3.2 Rancangan Kerangka Robot	15
Gambar 3.3 Blok Diagram	15
Gambar 3.4 Tampilan Sistem Navigasi	16
Gambar 3.5 Rangkaian Elektrik	18
Gambar 3.6 Diagram Alir Pengujian Keseluruhan	20
Gambar 4.1 Jalur Robot <i>Line Follower</i>	22
Gambar 4.2 Pemasangan Roda	23
Gambar 4.3 Pemasangan Sensor Garis	23
Gambar 4.4 Pemasangan Sensor <i>Proximity</i>	24
Gambar 4.5 Pemasangan Modul HC-12	24
Gambar 4.6 Pemasangan LCD Informasi Pengaturan Robot	25
Gambar 4.7 Pemasangan LCD Urutan Meja	25
Gambar 4.8 Pemasangan dan Perakitan Rangkaian Elektrik Komponen	25

Gambar 4.9 Kontruksi Robot	26
Gambar 4.10 Koneksi Modul HC-12 dengan USB to TTL CP2102	27
Gambar 4.11 Arah Putaran CW Bernilai Positif	28
Gambar 4.12 Arah Putaran CCW Bernilai Negatif	28
Gambar 4.13 Blok Diagram Sistem Kontrol Robot	31
Gambar 4.14 Pergerakan Robot Maju Jarak 165 cm	36



DAFTAR LAMPIRAN

Lampiran 1: Daftar Riwayat Hidup

Lampiran 2: Program Arduino Mega 2560

Lampiran 3: Program *Software Processing*



BAB I

PENDAHULUAN

1.1. Latar Belakang

Dengan adanya perkembangan zaman maka ilmu pengetahuan dan teknologi berjalan semakain maju, salah satunya adalah bidang robotika. Peranan robot dalam kehidupan manusia semakin hari semakin banyak manfaatnya. Robot dapat membantu manusia dalam melakukan aktivitas, salah satunya adalah robot dapat digunakan di bidang industri, pertahanan, kesehatan, bahkan industri kuliner yang membutuhkan robot untuk menyelesaikan suatu pekerjaannya. Dalam pemanfaatannya robot tidak sepenuhnya menggantikan manusia, tetapi teknologi robot dapat mengurangi beban kerja, dan memudahkan proses kerja manusia salah satunya yaitu meningkatkan efisiensi waktu, biaya dan meminimalisir saat melakukan kesalahan. Kemudian beralih pada permasalahan yang sedang terjadi sekarang. Maraknya kasus *Covid-19* masih menjadi permasalahan besar untuk Negara Indonesia, sehingga membuat sebagian orang merasa takut jika harus berkontak fisik dengan orang lain. Sulitnya mencegah penyebaran kasus *covid-19* mengharuskan masyarakat Indonesia untuk mengikuti protokol kesehatan yang dihimbaukan oleh pemerintah salah satunya adalah menjaga jarak.

Navigasi adalah proses pengontrolan pergerakan robot dari titik awal hingga akhir[1]. Navigasi robot adalah sistem yang digunakan untuk mengontrol pergerakan robot, dalam navigasi memungkinkan robot untuk bergerak maju, mundur, kiri dan kanan. Sistem navigasi merupakan suatu proses untuk menentukan letak dan arah perjalanan dari keadaan awal[2].

Topik-topik penelitian yang berkaitan dengan robot sudah sering dilakukan oleh peneliti baik yang berada di Negara Indonesia maupun di Negara luar. Janis, et al membuat sebuah robot yang dapat mengantarkan pesanan kedua meja dengan mengikuti jalur berbentuk garis hitam, dengan perancangan robot memiliki 2 arah tempat yaitu lurus untuk meja 1 dan belok kiri untuk meja 2 yang memanfaatkan 2 buah tombol pada masing-masing meja[3]. Saefullah, et al

merancang sebuah robot pengantar pesanan, dengan kendali dan monitor melalui aplikasi web[4]. Aryani, et al merancang sebuah *prototype* alat pengantar makanan yang menggunakan Arduino Mega dan memanfaatkan RFID (Radio *Frequency Identification*), dengan perencanaan apabila *pushbutton* ditekan maka motor *stepper* akan memulai bergerak secara sinkron ke meja *costumer*[5]. Hizburrohman, et al membuat sebuah *prototype* robot pengantar makanan yang dapat mengantarkan makanan ke 2 buah meja melalui monitor dengan menggunakan *App inverter* untuk dapat menjalankan dan mematikan robot[6]. Yanno, et al membuat sebuah robot pengantar makanan dan obat dimasa pandemi yang dapat mengantarkan makanan dan obat-obatan ke 2 ruang dengan mengikuti garis yang telah dibuat secara otomatis, sehingga tidak memerlukan tenaga medis untuk masuk ke dalam ruangan. Penelitian ini menunjukkan Intensitas cahaya sekitar bisa mempengaruhi pembacaan sensor garis yang akan mempengaruhi keberhasilan robot. Hasil pengujian tersebut didapat beberapa kekurangan yaitu robot masih dibuat tanpa sensor jarak yang dapat digunakan untuk mengidentifikasi hambatan, serta masih belum menggunakan sistem kontrol dengan metode *tunning* yang lebih baik[7].

Berdasarkan uraian diatas maka dibuatlah sebuah *prototype* optimasi sistem navigasi pada robot pengantar makanan. Salah satu tujuan kami membuat *prototype* optimasi sistem navigasi pada robot pengantar makanan ini agar robot dapat mengantarkan makanan ke meja-meja dengan bantuan sistem navigasi agar pergerakan dan posisi robot dapat terpantau melalui monitor. Serta berkaitan dengan maraknya kasus *covid-19*, agar kontak langsung antar manusia bisa diminimalisir. Penelitian ini merupakan pengembangan dari penelitian sebelumnya dari Politeknik Manufaktur Negeri Bangka Belitung.

1.2. Rumusan Masalah

Rumusan masalah pada proyek akhir Optimasi Sistem Navigasi pada Robot Pengantar Makanan ini adalah sebagai berikut:

1. Bagaimana membuat *prototype* robot pengikut garis yang mampu mengantarkan makanan secara otomatis?

2. Bagaimana membuat *prototype* robot agar dapat mengikuti garis/jalur yang telah ditentukan menggunakan sensor garis?
3. Bagaimana membuat sistem navigasi untuk robot pengantar makanan agar sesuai dengan rute/jalur yang telah ditentukan?

1.3. Batasan Masalah

Batasan masalah dari pembuatan Optimasi Sistem Navigasi pada Robot Pengantar Makanan ini adalah:

1. Robot hanya dapat mengikuti garis/jalur yang ditentukan.
2. Robot masih membutuhkan bantuan manusia dalam menginput nomor meja dan meletakkan makanan.
3. Robot hanya dapat digunakan pada lingkungan atau tempat yang intensitas cahaya rendah.
4. Robot hanya bisa mengangkat beban maksimal 3kg.

1.4. Tujuan

Tujuan dari pembuatan Optimasi Sistem Navigasi pada Robot Pengantar Makanan ini adalah:

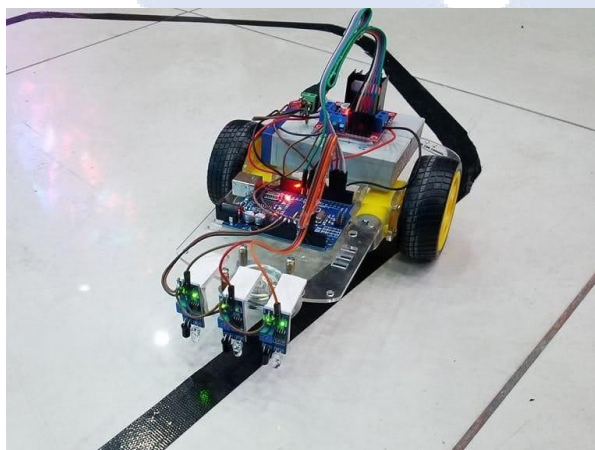
1. Robot dapat mengantar pesanan/makanan secara otomatis ke meja yang ditentukan.
2. Robot dapat mengikuti garis dengan baik sesuai jalur yang telah ditentukan dengan sistem PID.
3. Untuk membuat sistem navigasi pada robot pengantar makanan.

BAB II DASAR TEORI

2.1. Robot *Line Follower*

Robot yang secara otomatis dapat mengikuti jalur garis yang digambar di atas lantai disebut “Robot *Line Follower*”. Fungsi robot *line follower* tergantung pada ide-ide dasar pembacaan sensor dan pengaturan gerakan motor DC[8].

Robot yang mengikuti jalur garis yang didukung oleh rangkaian elektronika dengan dilengkapi roda dan menggunakan motor DC sebagai penggeraknya dikenal sebagai robot *tracer* atau robot *line follower*. Robot *line follower* umumnya dikembangkan dan digunakan dalam bidang penelitian, perlombaan dan industri. Robot pengikut garis umumnya memerlukan 2 jenis sensor yaitu *photodiode* dan LDR. Robot pengikut garis menggunakan motor DC sebagai *Actuator*. Sumber dari robot pengikut garis ini berasal dari baterai/aki. Sensor merupakan salah satu komponen yang paling dibutuhkan pada robot *line follower* dan kemudian digabungkan dengan program agar robot *line follower* dapat bergerak mengikuti jalur/rute[9].



Gambar 2.1 Robot *Line Follower*

2.2. Mikrokontroler Arduino Mega 2560

Mikrokontroler adalah *chip* komputer kecil dalam bentuk IC (*integrated circuit*). Mikrokontroler umumnya digunakan dalam sistem yang ringkas dan terjangkau yang tidak memerlukan komputasi yang begitu rumit, seperti yang ditemukan pada aplikasi *Personal Computer*. *Central Processing Unit*, *Random Access Memory*, *Read Only Memory* dan *port I/O (Input/Output)* adalah bagian-bagian utama mikrokontroler. Adapun *hardware* mikrokontroler, bisa dimanfaatkan untuk transmisi serial, *interupsi*, *enumerasi* dan banyak hal lainnya. *Analog to Digital Converter (ADC)*, *USB controller*, *CAN (Controller Area Network)* adalah contoh mikrokontroler.

Mikrokontroler yang digunakan pada proyek akhir ini adalah Arduino mega 2560. Arduino mega adalah salah satu papan mikrokontroler berdasarkan ATmega 2560. Arduino mega ini mempunyai 54 pin *input/output* digital, pada Arduino mega ini dapat dibagi menjadi 15 pin *output* PWM, 16 pin *input* analog, dan 4 pin *UART*, yang digunakan sebagai *port serial hardware*. *Oscilloscope* Kristal 16 MHz, koneksi USB, colokan listrik, *header ICSP*, dan tombol *reset*. Ini yang dibutuhkan untuk mendukung mikrokontroler[10].



Gambar 2.2 Arduino Mega 2560[10]

2.3. Motor DC PG36 24VDC

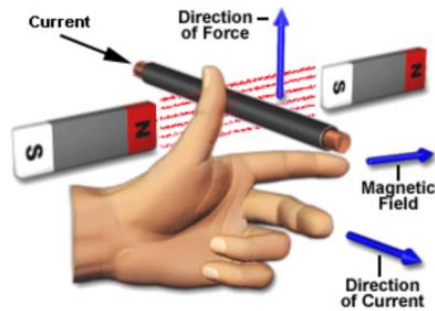
Untuk menghasilkan energi mekanik, Motor DC membutuhkan sumber tegangan searah pada kumparan medan. Pada motor DC terdiri dari 2 kumparan diantaranya yakni kumparan medan yang menciptakan medan magnet dan kumparan jankar yang berfungsi sebagai sumber gaya gerak listrik. Motor akan berputar sebagai akibat dari torsi yang dihasilkan ketika medan magnet dan arus dalam kumparan jankar berinteraksi[11].

Motor DC yang digunakan pada proyek akhir ini adalah motor DC PG36 24 VDC yang dilengkapi dengan *Encoder*, Dengan kapasitas tegangan motor DC adalah 24VDC. *Rotary encoder* adalah Salah satu alat elektromekanis yang dapat digunakan untuk melacak pergerakan dan lokasi. Umumnya *rotary encoder* memanfaatkan sensor optik untuk memperoleh pulsa serial yang bisa diterjemahkan ke dalam gerakan, posisi, dan arah. Sehingga posisi poros benda yang berputar menjadi data yang bisa ditransmisikan ke sirkuit kontrol dalam bentuk kode digital[12].



Gambar 2.3 Motor DC

Cara kerja motor DC yaitu jika arah arus listrik pada kawat terbalik, arah rotasi juga terbalik, ketika medan magnet dan medan listrik saling berinteraksi maka akan menghasilkan gaya mekanik. Simbol tangan atau aturan tangan kiri *Fleming* arah putaran motor ini diciptakan oleh John Ambrose Fleming, sebagai cara sederhana menentukan arah gerakan pada motor listrik. Dinyatakan bahwa jika induk jari, jari tengah dan jari telunjuk tangan kiri anda saling memanjang saling tegak lurus maka jari telunjuk mewakili arah medan magnet, jari tengah menunjukkan arah arus listrik, lalu ibu jari mewakili arah gaya yang dialami oleh poros pada motor dc.

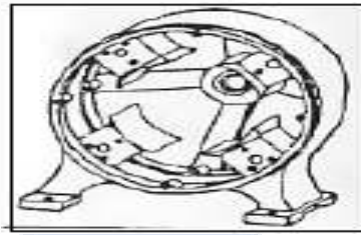


Gambar 2.4 Aturan Tangan Kiri *Fleming*

Adapun kontruksi motor DC secara umum yaitu sebagai berikut:

A. *Stator* Motor DC

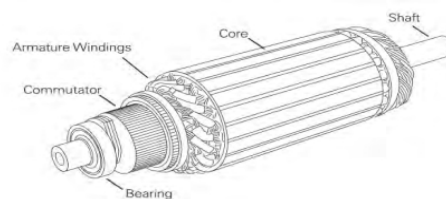
Stator Motor DC merupakan bagian yang berfungsi sebagai sirkuit magnet dengan dua medan kutub, dan medan yang terpasang di bagian dalam *stator*[13]



Gambar 2.5 *Stator* Motor DC[13].

B. *Rotor* atau Jangkar Motor DC

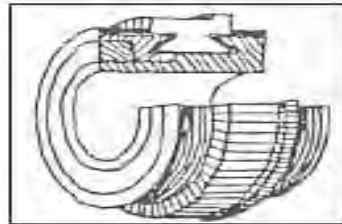
Mengubah energi listrik menjadi energi gerak putar merupakan fungsi dari *rotor* atau jangkar. *Rotor* atau jangkar meliputi serangkaian komponen inti silinder yang dijepit ke poros baja. Pada inti rotor terdapat alur-alur dimana lilitan jangkar dipasang[13].



Gambar 2.6 *Rotor* atau Jangkar Motor DC[13].

C. *Komutator*

Komutator terbuat dari batang tembaga yang ditempa dan dikeraskan yang kemudian dilapisi dengan jenis mika tertentu untuk isolasi. *Komutator* bertugas mengumpulkan arus induksi dari konduktor rotor dan mengubahnya menjadi arus searah melalui *Brush*[13].



Gambar 2.7 *Komutator* Motor DC[13].

D. *Sikat (Brush)*

Brush terbuat dari karbon, grafit, logam grafit atau gabungan karbon dan *graphite* yang dilengkapi dengan pegas pengepresan dan kontak *brush*-nya. *Brush* umumnya dipasangkan dengan menumpangkannya pada sisi *komutator* untuk memberi daya listrik ke motor[13].



Gambar 2.8 *Sikat (Brush)* Motor DC[13].

2.4. *Driver* Motor IBT-2 BTS7960

Driver motor adalah peningkat arus. Motor *driver* berfungsi untuk merubah sinyal kontrol dengan arus rendah menjadi arus lebih tinggi untuk menggerakkan motor. Arus keluaran maksimal, sumber tegangan maksimal, daya keluaran, tegangan beban jenis kemasan dalam jumlah *output* merupakan macam-macam jenis yang dapat dikategorikan dari motor *driver*[14].

Motor *driver* yang akan dipakai pada proyek akhir ini adalah motor *driver* IBT-2 BTS7960. Keunggulan dari *driver* motor BTS7960 karena *driver* ini dapat

bekerja sampai tegangan 27V, dapat mengalirkan arus lebih dari 43A, mudah ditemukan, mudah diinstalasi dan mempunyai kemasan atau bentuk yang kecil sehingga mudah dalam mengatur tempatnya [14].



Gambar 2.9 *Driver Motor IBT-2*[14]

2.5. *Sensor Line Follower 14 Channel*

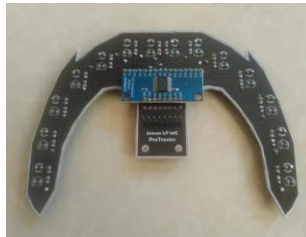
Sensor merupakan salah satu komponen elektronika yang berfungsi merubah suatu besaran khusus menjadi unit analog sehingga bisa dibaca oleh suatu rangkaian elektronik[15]. Sensor berfungsi sebagai pendeteksi atau pengukur sesuatu. Sensor dimanfaatkan untuk mengkonversi suatu mekanis, magnetis, sinar dan lain-lain menjadi tegangan atau arus. Sensor memiliki banyak jenis dan dikelompokkan menjadi beberapa kategori, seperti sensor berdasarkan perubahan lingkungan ada 3 diantaranya sensor fisika, sensor kimia, dan sensor biologi. Dimana tiap-tiap sensor mempunyai kegunaan yang berbeda-beda sehingga pemanfaatannya berdasarkan keperluan yang diinginkan.

Sensor *photodiode* adalah salah satu bentuk sensor yang resistansinya dapat berubah ketika terkena cahaya, termasuk cahaya dari pemancar atau LED. Kecerahan cahaya yang diterima *photodiode* dapat berdampak pada resistansinya. Ketika *photodiode* menerima lebih banyak cahaya atau sinar, nilai resistansinya menurun, dan sebaliknya, ketika *photodiode* menerima intensitas cahaya yang lebih sedikit, nilai resistansinya meningkat[16].



Gambar 2.10 Bentuk Fisik dan Simbol *Photodiode*[16].

Sensor yang digunakan pada proyek akhir ini menggunakan sensor *line follower 14 channel*. Sensor *line follower 14 channel* merupakan penggabungan dari beberapa sensor *photodiode*. Sensor yang digunakan berbahan PCB *Fiber High Quality* Menggunakan *optical* sensor LED dan *photodiode* yang Menggunakan teknik *multiplexer 16 bit* untuk mengakses ke 14 sensor, sehingga penggunaan pin Arduino Mega 2560 lebih hemat.



Gambar 2.11 Sensor *Line Follower 14 Channel*

2.6. Modul Nirkabel HC-12

Modul Nirkabel HC-12 RF UART nirkabel HC-12 merupakan modul transmisi data nirkabel tertanam multi saluran. Frekuensi radio 433, 473.0 MHz, saluran komunikasi dapat diatur. Langkahnya adalah 400KHz, total 100 saluran. Maksimal daya pancar modul adalah 100mW (20dbM) dan udara sensitivitas penerima -116dBm dari *baudrate* 5000bps, jarak komunikasi sekitar 500 meter. Modul HC-12 terdiri dari 9 pin dan blok antena RF ANT. Pin modul HC-12 terdiri dari VCC, GND, RXD, TXD, SET, ANT, GND, GND, NC, ANT dan ANT, dimana setiap pin mempunyai kegunaan yang berbeda-beda.



Gambar 2.12 Modul Nirkabel HC-12[17]

2.7. USB to TTL Converter CP2102

CP2102 adalah jembatan pengontrol USB ke UART yang sangat terintegrasi yang menyediakan solusi sederhana untuk memperbarui desain RS-232 ke USB menggunakan komponen minimal dan ruang PCB. Pengontrol fungsi USB di CP2102 adalah perangkat kecepatan penuh yang sesuai dengan USB 2.0 dengan

transceiver dan pencocokan *on chip* dan *resistor pull-up*. Pengontrolan fungsi USB mengelola semua *transfer* data antara USB to UART serta permintaan perintah yang dihasilkan oleh pengontrol *host* USB dan perintah untuk mengontrol fungsi UART. Antarmuka UART CP2102 terdiri dari sinyal data TX (mengirim) dan RX (menerima)[18].



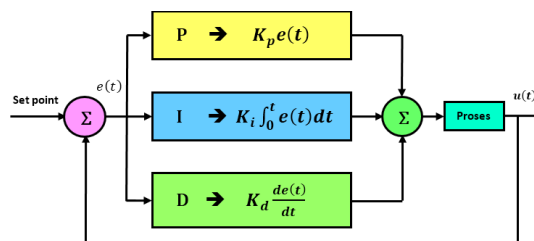
Gambar 2.13 USB to TTL Converter CP2102[18].

2.8. PID (*Proportional Integral Derivative*)

Proportional Integral Derivative atau yang sering disingkat PID merupakan *controller* gabungan dari ketiga *controller* yaitu *controller proportional*, *controller integral* dan *controller derivative*[19]. PID adalah bentuk kontrol berbasis umpan balik (*feedback*) yang memungkinkan manajemen sistem yang presisi. Sistem kontrol pada PID digunakan untuk mendeteksi kesalahan (*error*) dari pengukuran *variable* masukan (sensor) agar keluaran sistem dapat sesuai dengan nilai *setpoint* untuk menghasilkan *error* seminimal mungkin[20].

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}$$

K_p merupakan kontrol proporsional yang bisa berguna untuk memperbaiki *respon transien* spesifiknya *rise time* dan *settling time*. K_i merupakan kontrol integral yang mempunyai sifat yang agak lambat dari kontrol proporsional. Sedangkan K_d merupakan kontrol *derivative* yang hanya akan beralih ketika ada perubahan *error*, sehingga kontrol *derivative* harus digunakan dengan kontrol lainnya[21].



Gambar 2.14 Blok Diagram Kontrol PID [21].

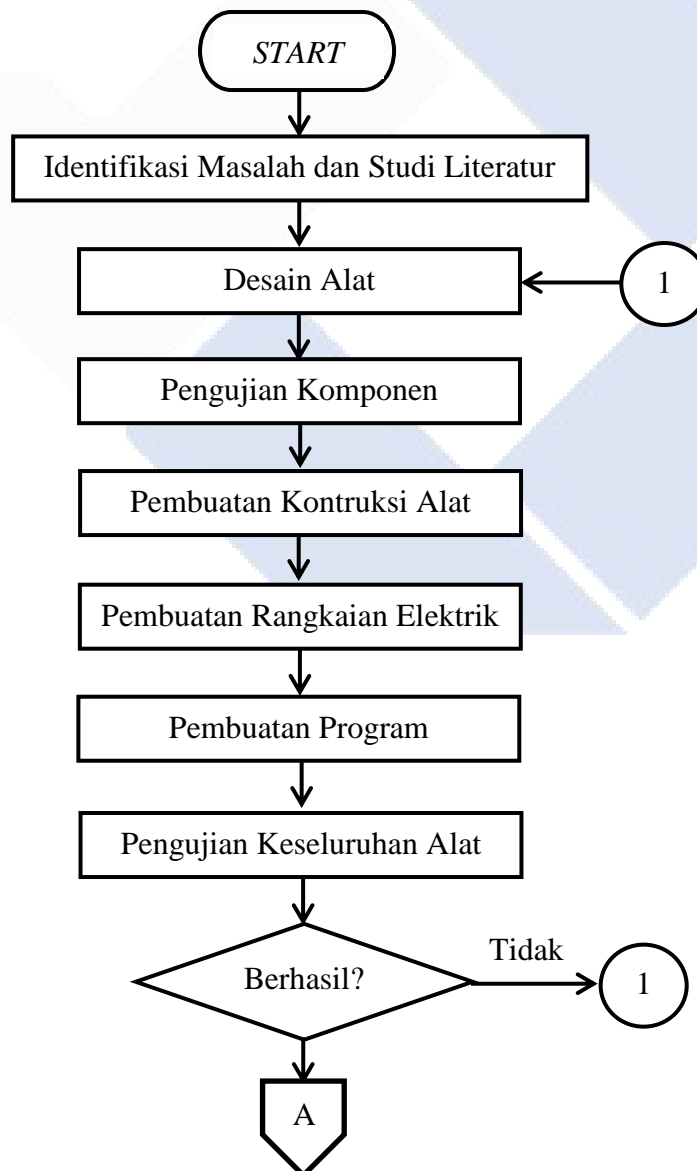
2.9. *Software Processing*

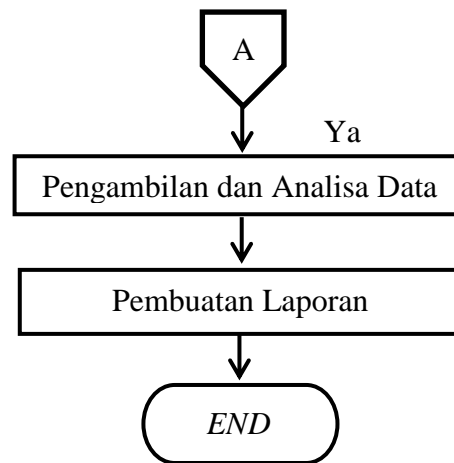
Processing merupakan *sketchbook software* bahasa pemrograman untuk mempelajari cara membuat kode program. *Processing* telah menawarkan literasi *software* dalam seni visual dan literasi visual dalam teknologi.

Processing dikembangkan sebagai *sketchbook software*, alat produksi, dan cara untuk mengajarkan dasar-dasar pemrograman komputer dalam pengaturan visual. Perangkat lunak *Processing* ditulis dalam bahasa pemrograman berbasis teks yang sangat cocok untuk membuat dan memodifikasi grafik. *Processing* bertujuan untuk mencapai kesederhanaan antara kejelasan dan fitur lanjutan. Banyak grafik komputer dan teknik antarmuka, seperti gambar vektor/raster, pemrosesan *image*, pemodelan warna, operasi *mouse* dan *keyboard*, komunikasi jaringan, dan pemrograman berorientasi objek, dibuat lebih gampang untuk memfasilitasi pengajaran oleh sistem *processing*. Dengan bantuan *libraries*-nya, kemampuan pemrosesan dapat dengan cepat diperluas untuk memperoleh suara, mengirim dan menerima data dalam berbagai format, serta mengimpor dan mengekspor format file 2D dan 3D[22].

BAB III METODE PELAKSANAAN

Untuk mempermudah dalam proses pengerjaan proyek akhir dengan topik “Optimasi Sistem Navigasi pada Robot Pengantar Makanan”, maka dibuatlah beberapa langkah-langkah ataupun tahapan proses dalam pengerjaannya. Metode pelaksanaan dari pembuatan proyek akhir ini dapat diperhatikan pada *flowchart* dibawah ini:





Gambar 3.1 Diagram Alir Metode Pelaksanaan

3.1. Studi Literatur

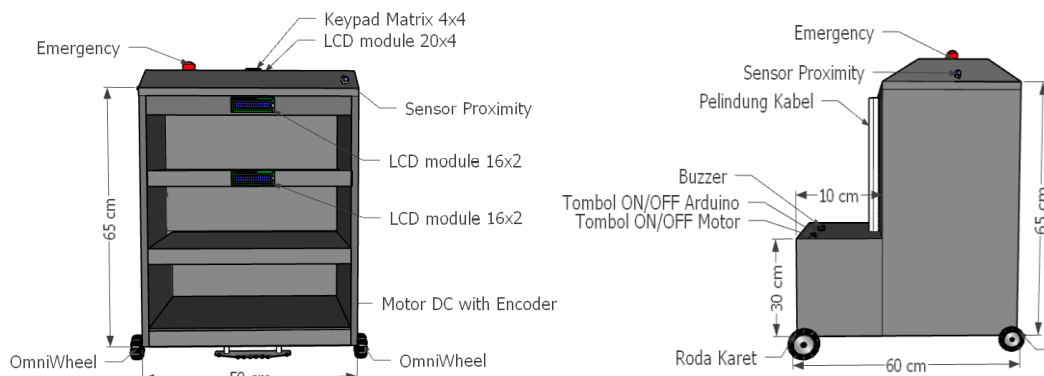
Studi literatur pada Proyek Akhir ini bertujuan untuk mendapatkan informasi tentang data-data referensi yang diperlukan sebagai bahan acuan dalam pengerjaan Proyek Akhir. Metode pengumpulan informasi serta data-data melalui buku, jurnal, dan skripsi yang berkaitan dengan Proyek Akhir dengan *topic* “Optimasi Sistem Navigasi pada Robot Pengantar Makanan”. Studi literatur merupakan mempelajari kumpulan-kumpulan informasi yang bersumber dari internet maupun buku. Kumpulan-kumpulan informasi bertujuan untuk mengetahui cara kerja alat, *hardware* elektrik dan prosedur kerja yang nantinya akan dipakai. Setelah Kumpulan-kumpulan informasi digabungkan selanjutnya data akan diolah menjadi referensi sebagai dasar acuan untuk prosedur pembuatan proyek akhir.

3.2. Tahapan Desain Alat

Tahap desain alat pada proyek akhir ini dibagi menjadi 2 bagian diantaranya, yaitu desain perangkat keras (*Hardware*) dan desain perangkat lunak (*software*). Gambaran umum tentang alat yang akan dibuat adalah tujuan dari tahap ini.

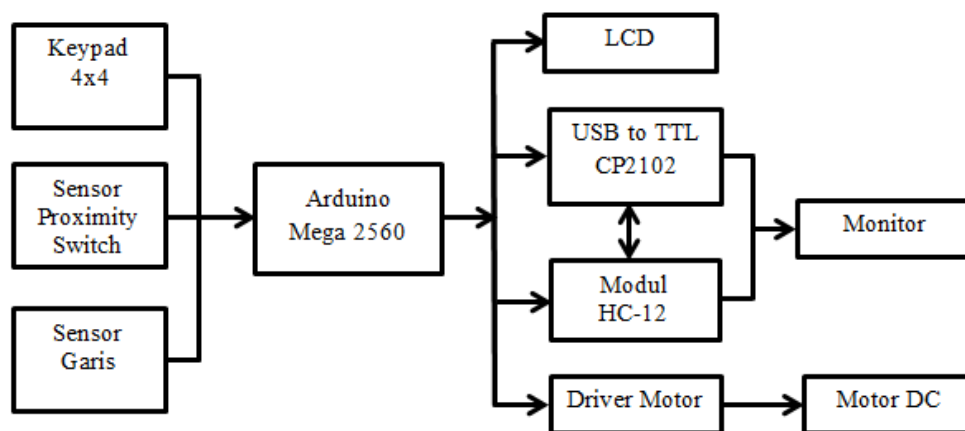
3.2.1. Desain *Hardware*

Desain konstruksi dapat diawali dengan membuat desain kerangka robot, menetapkan komponen dan membuat diagram blok sistem yang akan digunakan dalam pembuatan Proyek Akhir.



Gambar 3.2 Rancangan Kerangka Robot

Setelah desain kerangka robot dan menetapkan komponen, langkah berikutnya yaitu membuat blok diagram. Diagram blok mempunyai beberapa tujuan termasuk membuatnya lebih mudah memahami bagaimana sistem beroperasi, memeriksa bagaimana jaringan beroperasi, dan mengidentifikasi kekurangan dalam sistem yang dibangun. Diagram blok yang dibuat adalah diagram blok dari kontrol robot. Cara kerja robot akan lebih jelas setelah melihat diagram blok dibawah ini.



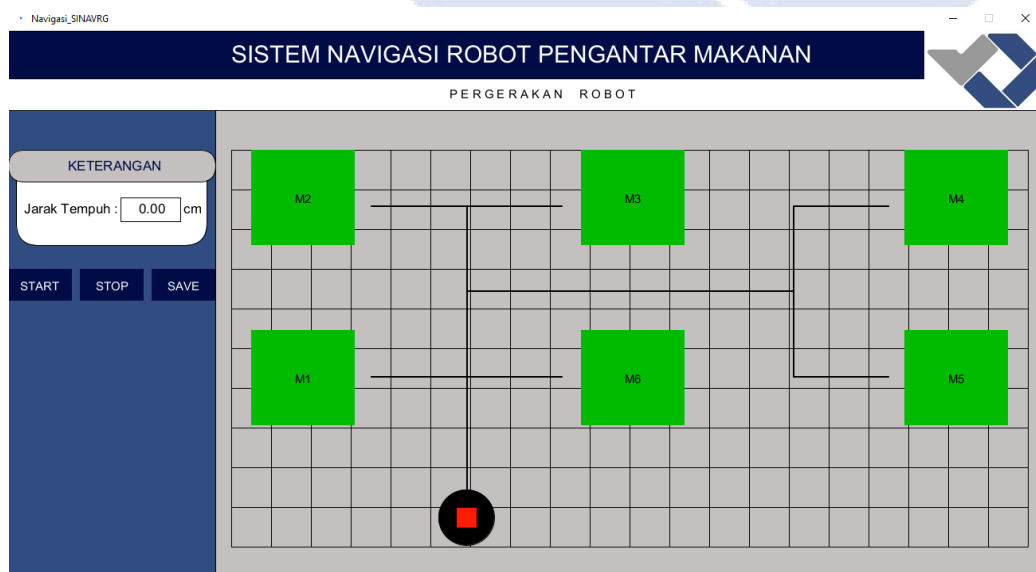
Gambar 3.3 Blok Diagram

Berikut ini adalah beberapa fungsi tiap-tiap diagram blok:

1. Blok Masukan : Blok Masukan ini terdiri dari beberapa sensor yang bertujuan memberikan sinyal masukan kepada blok proses/pemrosesan yaitu Arduino Mega 2560.
2. Blok Proses : Blok proses/pemrosesan terdiri dari Arduino Mega 2560, merupakan mikrokontroler yang telah dikonfigurasi untuk memproses semua aktivitas masukan maupun keluaran komponen yang terhubung ke Arduino.
3. Blok keluaran : Blok keluaran terdiri dari Modul HC-12 dan USB to TTL CP2102 yang berfungsi sebagai pengirim data ke monitor (PC) secara *wireless*, LCD berfungsi untuk menampilkan tulisan dan yang terakhir motor *driver* berfungsi sebagai penggerak/pengatur kecepatan dan pengubah arah motor.

3.2.2. Desain Software

Tahap ini dapat diawali dengan membuat desain alat dan membuat rute (jalur) untuk navigasi pada aplikasi. Desain perangkat lunak (*software*) merupakan tahap untuk merancang aplikasi sistem yang mempunyai tujuan diantaranya yaitu, menjalankan, menghentikan, serta mengetahui jarak tempuh dari pergerakan robot.



Gambar 3.4 Tampilan Sistem Navigasi

3.3. Pengujian Komponen Robot

Tahapan selanjutnya adalah tahap uji coba *hardware* elektrik robot yang bermaksud untuk melihat apakah *hardware* elektrik yang digunakan bisa beroperasi secara baik sesuai kegunaannya.

Adapun uji coba yang akan dilakukan diantaranya yaitu:

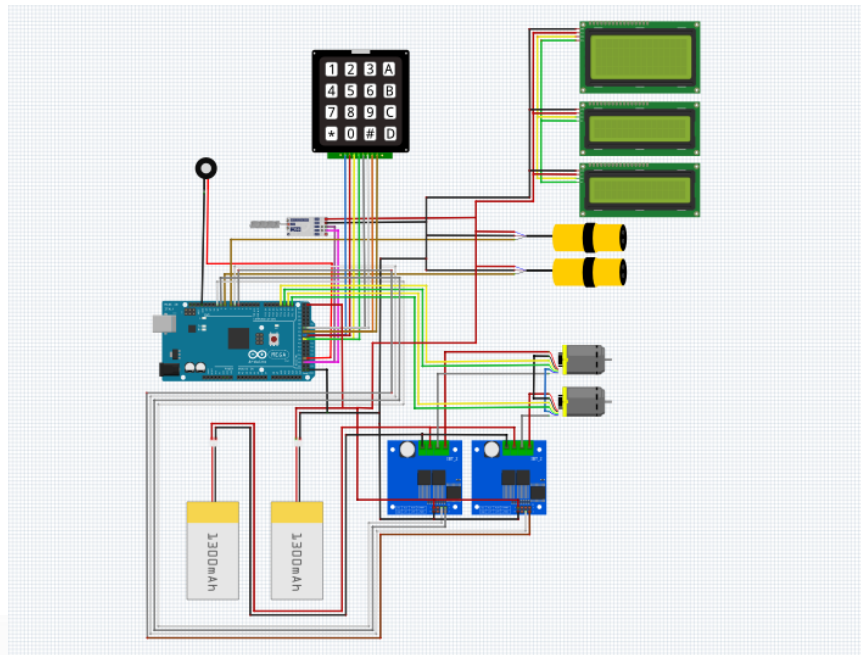
1. Uji coba Motor DC
2. Uji coba *Driver* motor
3. Uji coba Sensor garis
4. Uji coba modul nirkabel HC-12
5. Uji coba USB to TTL CP12102

3.4. Pembuatan Kontruksi Alat

Tahapan selanjutnya yaitu pengerjaan kontruksi alat dengan mengikuti referensi desain kerangka robot yang sudah dirancang. Dikarenakan proyek akhir yang akan kami buat melanjutkan proyek akhir penelitian sebelumnya maka tahapan kontruksi hanya menambah beberapa bagian kontruksi yang diinginkan.

3.5. Pembuatan Rangkaian Elektrik

Tahap ini hampir sama dengan tahapan pembuatan kontruksi alat. Dimana pada tahap pembuatan rangkaian elektrik ini mengikuti acuan dari desain sketsa yang telah buat. Tahap pembuatan rangkaian elektrik dapat dikerjakan dengan cara pemasangan kabel *jumper* ke tiap-tiap komponen, dimana tiap-tiap komponen akan disambungkan ke Arduino Mega 2560.



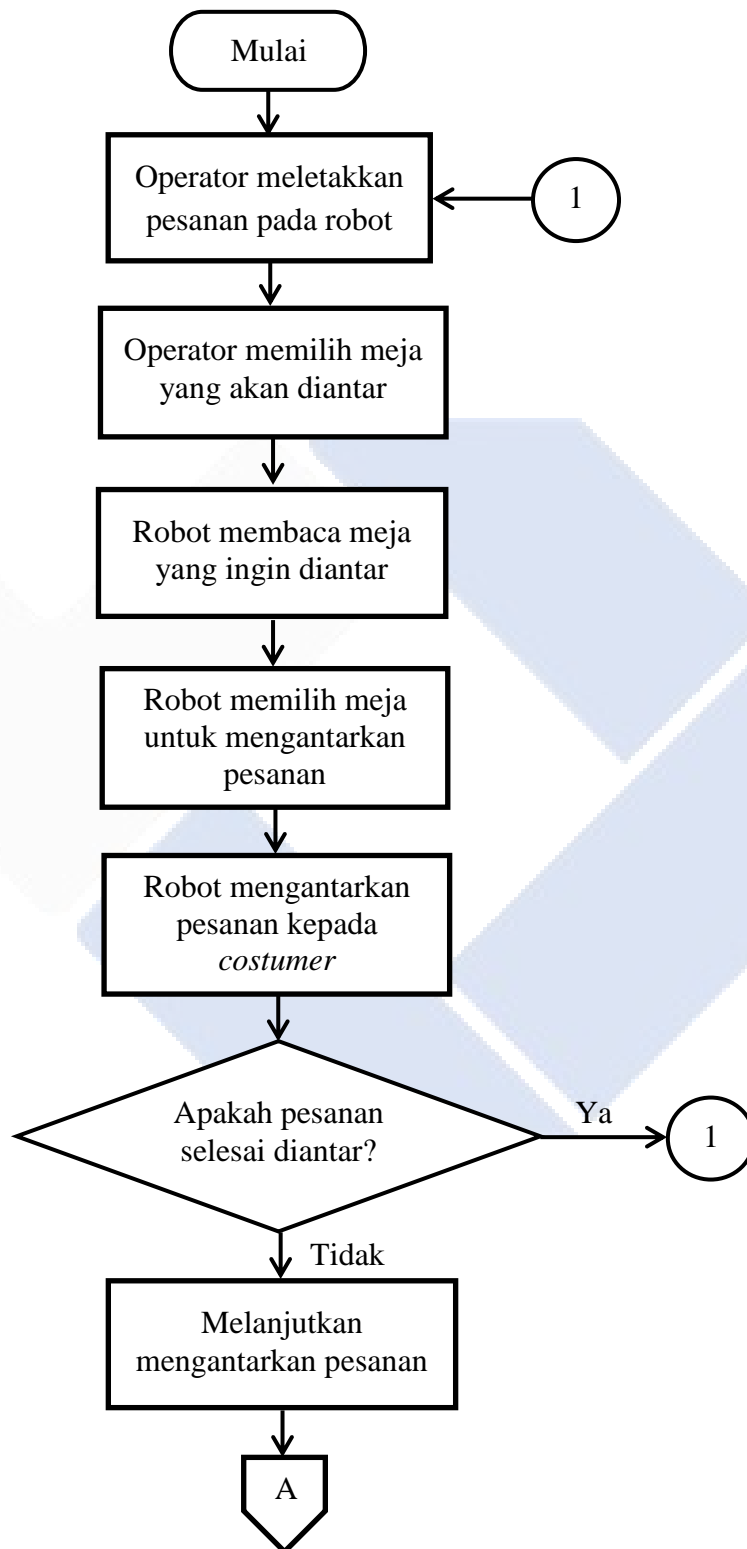
Gambar 3.5 Rangkaian Elektrik

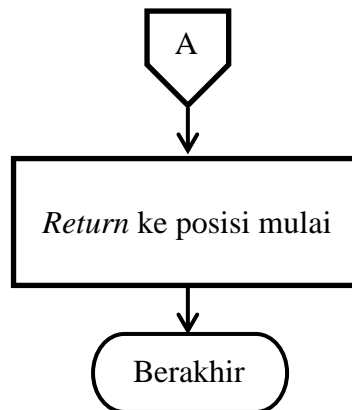
3.6. Pembuatan Program

Tahap selanjutnya yaitu tahap pembuatan program bagi beberapa komponen yang digunakan. Setelah selesai pembuatan program, kemudian program-program tersebut akan digabungkan agar robot dapat beroperasi selaras yang diinginkan. Dengan menggunakan sistem kontrol PID dan sistem navigasi, maka robot akan mengikuti garis dan mengantarkan makanan sesuai dengan intruksi yang telah diberikan.

3.7. Pengujian Keseluruhan

Pada tahap ini dilakukan pengujian keseluruhan untuk memahami apakah alat yang telah dikerjakan dapat beroperasi dan berfungsi sesuai keinginan. Apabila percobaan berhasil, informasi dan data-data yang dikumpulkan akan digunakan untuk menarik kesimpulan dan memberikan rekomendasi untuk studi selanjutnya.





Gambar 3.6 Diagram Alir Pengujian Keseluruhan

3.8. Pengambilan dan Analisa Data

Tahapan ini adalah tahapan dimana informasi dan data yang dihasilkannya dari pengujian alat akan dikumpulkan dan dianalisa. Tahap ini bermaksud untuk mengetahui kelebihan dan kekurangan dari alat yang dikerjakan, baik dari konstruksi, sistem pengontrolan maupun pengaplikasiannya.

3.9. Pembuatan Laporan Proyek Akhir

Tahapan ini adalah tahapan akhir dari pengerjaan proyek akhir. Tahapan ini bermaksud untuk menyimpulkan keseluruhan dari proses pengerjaan proyek akhir, dimana tahapan ini akan memberikan berbagai informasi yang didapatkan dari proses pembuatan alat yang dikerjakan.

BAB IV

PEMBAHASAN

Pada Bab 4 ini akan menguraikan proses pengerjaan proyek akhir berdasarkan metode yang digunakan dalam pembuatan serta pengujian optimasi sistem navigasi pada robot pengantar makanan. Berikut uraian dari bab pembahasan ini:

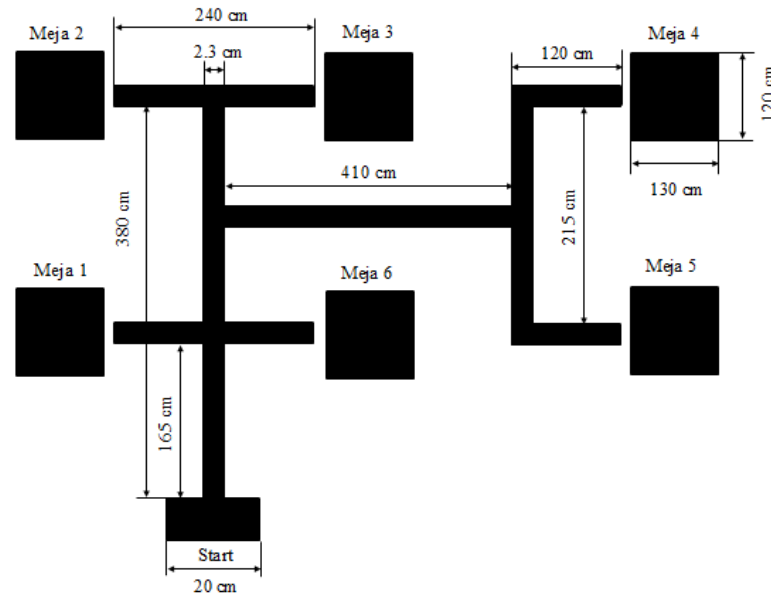
1. Keterangan alat
2. Perencanaan dan pembuatan robot
3. Pengujian elektrik robot
4. Pengujian sensor
5. Pengujian aplikasi
6. Pengujian kontrol PID terhadap sensor garis
7. Pengujian keseluruhan

4.1. Keterangan Alat

Robot pengantar makanan dirancang untuk dapat bergerak mengikuti jalur yang sudah ditentukan. Robot akan bergerak stabil mengikuti garis yang ditentukan dikarenakan adanya sistem kontrol PID yang digunakan. Robot pengantar makanan akan berhenti apabila robot telah sampai ke meja tempat tujuan. Apabila robot telah sampai ke meja tempat tujuan *costumer* dapat mengambil makanan di laci. Setelah *costumer* selesai mengambil makanan di laci kemudian *costumer* dapat meletakkan tangan pada sensor *proximity*.

Mekanis robot pengantar makanan yang digunakan adalah holo dan triplek ukuran 3 mm yang dipasang sesuai dengan bentuk dari konstruksi untuk menutupi holo tersebut. *Actuator* yang digunakan pada robot pengantar makanan yaitu 2 buah motor DC serta 2 buah roda karet yang terpasang di bagian belakang pada masing-masing motor DC dan 2 roda *free wheel* yang terpasang pada bagian depan robot. Robot pengantar makanan menggunakan sensor LF dengan 14 *channel* yang dipasang pada bagian bawah robot. Pada setiap laci robot akan

dipasang LCD untuk menampilkan nomor meja *costumer*. Sumber yang digunakan pada robot pengantar makanan adalah 2 baterai/aki. Sumber kontrol menggunakan *buck converter*. Berikut merupakan jalur untuk robot *line follower*.



Gambar 4.1 Jalur Robot *Line Follower*

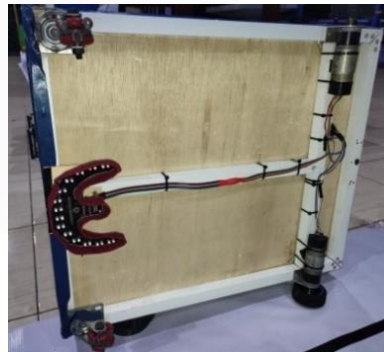
Cara kerja dari Robot pengantar makanan ini adalah pertama-tama operator akan menghidupkan robot dengan cara saklar yang berada dibelakang robot di ON terlebih dahulu. Selanjutnya operator akan memilih meja yang akan diantarkan makanan. Sebelum robot mengantarkan makanan, operator harus secara teratur mengkalibrasi sensor garis pada robot untuk memperhitungkan *error* terkait intensitas cahaya dalam pembacaan sensor. Kalibrasi sensor dilakukan secara bertahap, pertama kalibrasi sensor putih dan kemudian sensor hitam. Setelah kalibrasi selesai, robot akan secara otomatis mengikuti jalur yang ditentukan. Seperti yang terlihat pada gambar di atas, robot akan mulai di tempat awalnya sebelum memilih meja atau lokasi tujuan yang diinginkan. Adanya sistem navigasi pada robot bertujuan agar pergerakan dan posisi robot dapat terpantau melalui monitor.

4.2. Perencanaan dan Pembuatan Robot

Pada perancangan dan pembuatan robot pengantar makanan ini dilakukan proses pembuatan konstruksi serta pemasangan komponen dan alat yang digunakan pada robot. Adapun pembuatan konstruksi dan pemasangan komponen terbagi menjadi beberapa tahapan. Berikut tahapan yang dikerjakan dalam proses pembuatan robot pengantar makanan:

a. Pemasangan Roda

Pada tahap ini yaitu pemasangan roda pada pangkalan robot. Jumlah roda yang dipasang ada 4 buah roda yang terdiri dari 2 buah roda karet yang dipasang pada masing-masing motor DC di posisi bagian belakang dan 2 buah roda *free wheel* yang dipasang di bagian depan. Posisi pemasangan roda perlu diperhatikan untuk mengantisipasi adanya kendala pada pergerakan robot. Gambar 4.2 menunjukkan bagaimana roda dipasang.



Gambar 4.2 Pemasangan Roda

b. Pemasangan Sensor Garis

Pemasangan sensor garis yang dipasang di bawah bagian depan robot yang bertujuan agar sensor dapat mendeteksi garis/jalur yang ditentukan. Gerakan dari motor DC diinput menggunakan sensor garis ini. Gambar 4.3 menunjukkan bagaimana sensor garis dipasang.



Gambar 4.3 Pemasangan Sensor Garis

c. Pemasangan Sensor *Proximity*

Pemasangan sensor ini terletak pada bagian atas robot dan bagian depan robot. Dimana sensor ini akan berfungsi untuk memberikan sinyal *input* ke Arduino. Pemasangan sensor *proximity* di bagian atas berfungsi untuk memberikan perintah untuk bergerak/berjalan apabila robot telah sampai ke meja tujuan sedangkan pemasangan sensor *proximity* di bagian depan samping robot berfungsi untuk mendeteksi penghalang yang ada di depan robot. Pemasangan sensor *Proximity* dapat dilihat pada gambar 4.4.



Gambar 4.4 Pemasangan Sensor *Proximity*

d. Pemasangan Modul HC-12

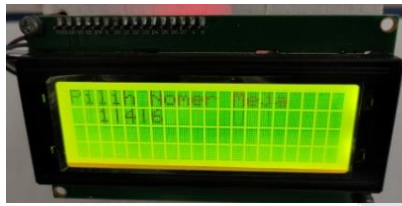
Pemasangan Modul HC-12 diposisikan dibelakang body robot dan di *port* USB laptop/PC yang dikombinasikan dengan USB *converter* to TTL. Pada proyek akhir ini modul HC-12 yang digunakan berjumlah 2 buah. Modul HC-12 berfungsi untuk komunikasi serial jarak jauh antara beberapa mikrokontroler. sehingga modul HC-12 berguna sebagai pengirim dan penerima data ke monitor (PC).



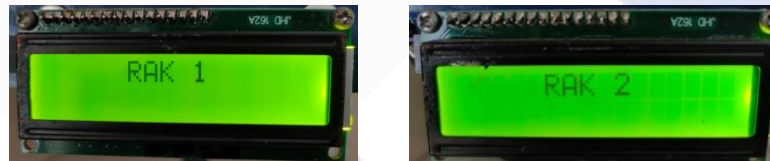
Gambar 4.5 Pemasangan Modul HC-12

e. Pemasangan LCD

Pemasangan LCD diposisikan dibagian depan setiap laci makanan dan dibagian atas robot. Pemasangan LCD dibagian depan laci berfungsi untuk menginformasikan urutan meja apabila *costumer* ingin mengambil pesannya sedangkan pemasangan LCD di bagian atas robot berfungsi untuk menginput nomor meja yang diinginkan serta melihat tampilan pengaturan robot seperti *setting* kalibrasi.



Gambar 4.6 Pemasangan LCD Informasi Pengaturan Robot



Gambar 4.7 Pemasangan LCD Urutan Meja

f. Pemasangan dan Perakitan Rangkaian Elektrik Komponen

Pemasangan beberapa komponen ditempatkan pada bagian khusus yang telah disiapkan dibagian belakang robot. Dimana komponen yang dipasang adalah Arduino, baterai, *buck converter* dan *driver* motor DC. Pemasangan dan Perakitan Rangkaian Elektrik Komponen dapat dilihat pada gambar 4.8.



Gambar 4.8 Pemasangan dan Perakitan Rangkaian Elektrik Komponen

g. Hasil Pemasangan Komponen dan Kontruksi Robot

Setelah pemasangan keseluruhan kontruksi dan komponen paada robot selesai, maka hasil dari robot pengantar makanan dapat dilihat pada gambar 4.9.



Gambar 4.9 Kontruksi Robot

4.3. Pengujian Elektrik Robot

Pada pengujian ini bertujuan untuk melihat apakah komponen-komponen yang digunakann bisa beroperasi dengan baik sesuai dengan kegunaannya. Pengujian dilakukan dengan menggunakan beberapa program pada Arduino.

4.3.1. Konfigurasi Awal Modul HC-12

Konfigurasi pertama dalam pengujian modul HC-12 adalah men-*setting* terlebih dahulu modul HC-12 dengan menggunakan perintah *AT command* pada *serial monitor*. cara untuk men-*setting*nya yaitu hubungkan pin “*Set*” dari modul ke pin *ground*. Berikut pin konfigurasi modul HC-12 dengan pin arduino.

Tabel 4.1 Pin konfigurasi antara Modul HC-12 dengan Arduino

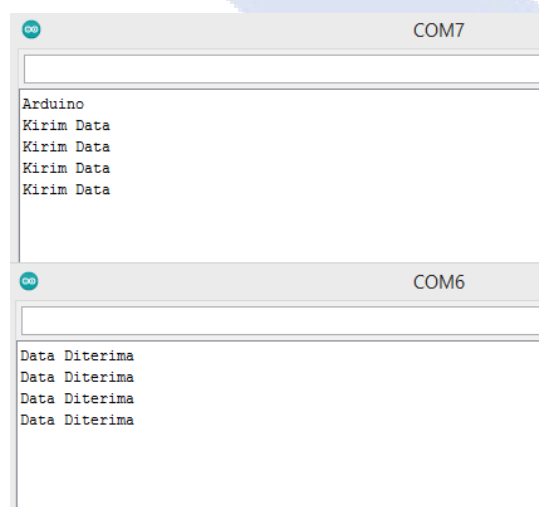
Pin Modul HC-12	Pin Arduino
VCC	5V
GND	GND
RX	51
TX	50
SET	GND

Langkah selanjutnya adalah masuk ke *serial monitor* pada arduino, perintah yang ditulis yaitu:

- AT digunakan untuk menguji modul HC-12 apakah berfungsi atau tidak.
- AT+V digunakan untuk mengetahui informasi versi dari modul HC-12
- AT+B115200 digunakan untuk mengatur *baudrate* yang digunakan, kemudian modul akan mengembalikan “OK+B115200”
- AT+C100 digunakan untuk mengatur saluran komunikasi nirkabel. kemudian modul akan mengembalikan “OK+C100”
- AT+P8 digunakan untuk mengatur tingkat daya pancar modul HC-12. kemudian modul akan mengembalikan “OK+CP8”, daya pancar modul adalah +20dbm.

4.3.2. Pengujian Koneksi Modul HC-12 dengan USB to TTL CP2102

Pengujian ini dilakukan untuk mengetahui apakah Modul HC-12 dan USB to TTL CP2102 sudah terkoneksi atau belum. Pengujian ini menggunakan 2 buah modul HC-12 dan 1 buah USB to TTL CP2102. Pengujian dilakukan dengan cara menghubungkan USB to TTL CP2102 dan modul HC-12 ke laptop sedangkan modul HC-12 yang satunya lagi dihubungkan ke arduino. Pengujian menggunakan program pada Arduino. Gambar 4.10 menunjukkan hasil pengujian koneksi.

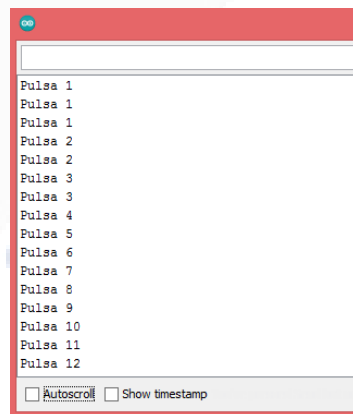


Gambar 4.10 Koneksi Modul HC-12 dengan USB to TTL CP2102

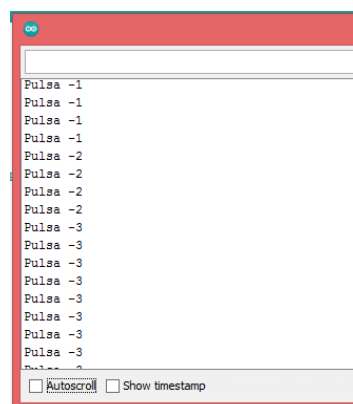
Berdasarkan pengujian pada gambar 4.12, dapat dilihat bahwa pada *serial monitor* COM 7 yaitu *serial monitor* arduino mengirimkan data kemudian data yang dikirim akan diterima oleh *serial monitor* COM 6 yaitu *serial monitor* yang terhubung dengan Laptop, setelah Modul HC-12 terkoneksi maka pada *serial monitor* COM 6 akan menampilkan *text* “Data Diterima”. Dari pengujian ini dapat disimpulkan bahwa kedua Modul HC-12 dapat terkoneksi dengan baik untuk dapat saling mengirim dan menerima data.

4.3.3. Pengujian Arah Putaran Sensor Encoder CW dan CCW

Pada Pengujian sensor ini dilakukan untuk mengetahui arah putaran sensor pada motor DC. Pengujian ini dilakukan dengan memeriksa nilai pada nilai digital I/O ketika interupsi terjadi. Pengujian dilakukan dengan menggunakan program pada Arduino untuk mendeteksi arah putaran sensor.



Gambar 4.11 Arah Putaran CW Bernilai Positif



Gambar 4.12 Arah Putaran CCW Bernilai Negatif

4.3.4. Pengujian Pembacaan Jarak pada Roda

Sebelum pengujian tahap pertama dalam pembacaan jarak pada roda adalah mengkonversi nilai pulsa dari *output rotary encoder* menjadi informasi untuk kecepatan putaran roda. Berikut rumus perhitungan kecepatan putaran roda.

- Jumlah pulsa dalam 1 putaran penuh adalah 135 ppr (*pulses per revolution*).

- Jumlah putaran :

$$\text{Putaran} = \text{pulsa/ppr}$$

$$\text{Putaran} = \text{pulsa}/135$$

- Nilai RPM :

$$\text{RPM} = (60/\text{sampling time}) \times \text{putaran}$$

- Nilai kecepatan sudut :

$$\omega = 2\pi/60 \times \text{RPM}$$

Setelah mendapatkan nilai kecepatan sudut maka langkah selanjutnya yaitu mengubah kecepatan sudut menjadi kecepatan linier untuk mendapatkan nilai jarak. dalam kondisi nyata kecepatan putaran roda dapat berubah-ubah yang dapat diakibatkan dari beberapa faktor seperti, kondisi gesekan permukaan dengan roda, kendala roda apabila terjadi *slip* dan faktor lainnya sehingga perhitungan dilakukan dengan teknik *sampling time*. *Sampling time* yang digunakan pada proyek akhir ini adalah 200 ms, maka dapat diartikan perhitungan jarak dilakukan setiap 200 ms sekali. Berikut rumus perhitungan dibawah ini.

- Kecepatan linier :

$$V = r/1000 \times \omega$$

$$V = 33,5/1000 \times \omega$$

- Jarak Roda :

$$S = \text{sampling time} \times V$$

$$S = 0.2 \times V$$

- Jarak total roda :

$$St += s$$

Keterangan:

ω = Kecepatan sudut (rad/s)

V = Kecepatan Linier (m/s)

S = Jarak (m)

S_t = Jarak total (m)

Langkah selanjutnya yaitu pengujian dengan menggerakkan roda sesuai dengan jarak yang ditentukan. Hasil pengujian dapat dilihat pada tabel 4.2.

Tabel 4.2 Hasil Pengujian Jarak Roda

Percobaan		
Jarak (cm)	Hasil (cm)	Error (cm)
50	49,89	0,11
100	98,52	1,48
150	149,44	0,56
200	197,62	2,40
250	249,02	0,98
300	299,44	0,56
Rata-rata error		1.015

Pengujian ini dilakukan untuk mengetahui nilai jarak roda dari pembacaan *encoder*. Pengujian dilakukan secara manual dengan menggerakkan robot sesuai jarak yang ditentukan. Dari data pada tabel 4.2 dapat disimpulkan bahwa nilai percobaan jarak yang ditentukan dengan hasil pengujian jarak roda nilainya mendekati sama dengan nilai *error* yang kecil. Maka dapat disimpulkan pengujian jarak roda yang didapat dari nilai *encoder* dapat digunakan untuk proyek akhir.

4.3.5. Pengujian Sensor *Proximity*

Percobaan sensor *proximity* ini dilakukan untuk mengetahui jarak maksimal dari sensor. Pengujian ini menggunakan 2 buah sensor *proximity*, dimana *proximity* 1 berada dibagian depan sebelah kiri robot sedangkan *proximity* 2 berada dibagian atas sebelah kanan robot. Pengujian dilakukan secara manual dengan mengukur jarak maksimalnya.

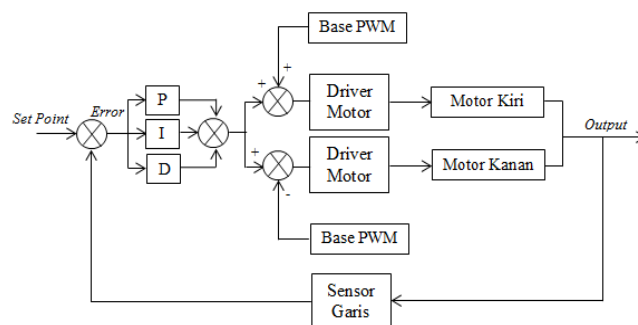
Tabel 4.3 Hasil Pengujian Jarak Jangkauan Sensor *Proximity*

<i>Proximity 1</i>	Jarak jangkauan	<i>Proximity 2</i>	Jarak jangkauan
4cm	Ya	4cm	Ya
6cm	Ya	6cm	Ya
8cm	Ya	8cm	Ya
10 cm	Ya	10 cm	Ya
12 cm	Ya	12 cm	Ya
14 cm	Ya	14 cm	Ya
16 cm	Ya	16 cm	Tidak
20 cm	Ya	20 cm	Tidak
23.5 cm	Ya	23.5 cm	Tidak

Dari tabel 4.3 ini dapat disimpulkan bahwa penggunaan sensor *proximity 1* dapat mendeteksi penghalang/keberadaan objek dengan jarak maksimal 23,5 cm, sedangkan penggunaan sensor *proximity 2* dapat mendeteksi penghalang atau keberadaan objek dengan jarak maksimal 14 cm. Apabila penghalang/keberadaan objek diatas batas maksimal dari sensor *proximity 1* dan sensor *proximity 2* maka sensor tidak dapat mendeteksinya.

4.3.6. Blok Diagram Sistem

Blok diagram sistem bertujuan untuk mempermudah mengidentifikasi langkah-langkah dari suatu sistem kontrol pada robot. Gambar 4.13 menunjukkan diagram blok sistem kontrol robot.



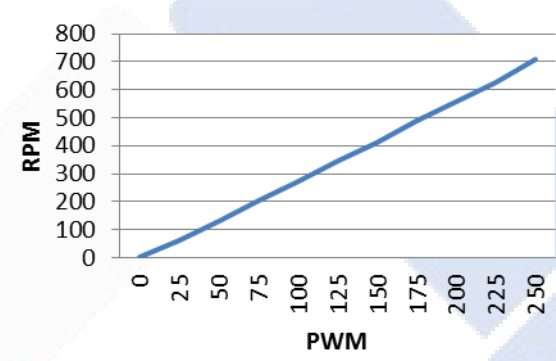
Gambar 4.13 Blok Diagram Sistem Kontrol Robot

4.3.7. Pengujian Persamaan Linier antara PWM dan RPM

Pengujian ini dilakukan untuk melihat berapa nilai RPM yang dihasilkan berdasarkan PWM yang diberikan, pengujian ini bertujuan agar dapat membantu dalam pembuatan program.

Tabel 4.4 Hasil Persamaan Linier PWM dan RPM

PWM(x)	RPM(y)	Garis Persamaan Linier PWM dan RPM
0	0	
25	60	
50	132	
75	202	
100	272	
125	348	
150	412	
175	489	
200	559	
225	628	
250	711	



Berdasarkan hasil data pengujian dari tabel 4.4, dapat dilihat bahwa nilai PWM yang diberikan akan mempengaruhi nilai RPM, semakin besar nilai PWM maka akan semakin besar pula RPM yang dihasilkan dari Motor DC karena kecepatan putarannya akan semakin cepat. Pengujian ini dilakukan secara manual dengan cara *mensetting* nilai PWM. Nilai RPM di dapat dari pengukuran menggunakan alat ukur *tachometer*.

4.3.8. Pengujian Sensor Garis

Pada pengujian sensor garis ini dilakukan 2 pengujian yaitu pengujian sensor untuk mengetahui nilai ADC dan pengujian sensor untuk mengetahui nilai *error*-nya.

4.3.8.1. Pengujian Sensor Mengetahui Nilai ADC

Pengujian sensor ini dilakukan untuk mengetahui perbandingan nilai ADC (*Analog Digital Converter*) pada garis hitam dan garis putih.

Tabel 4.5 Hasil Nilai ADC (*Analog Digital Converter*) pada Sensor

No	Warna Putih	Warna Hitam	Warna Abu-abu Muda	Warna Abu-abu Tua
1	228	913	466	677
2	158	899	458	572
3	138	921	424	562
4	130	912	407	542
5	134	920	433	596
6	127	914	488	591
7	196	940	584	709
8	135	906	431	593
9	139	895	420	576
10	254	846	542	669
11	150	923	470	580
12	148	905	464	596
13	137	880	485	625
14	413	922	528	754

Berdasarkan data diatas, dapat dilihat bahwa nilai ADC (*Analog Digital Converter*) setiap warna memiliki nilai yang berbeda-beda. Dikarenakan nilai analog *input* 0-1023 dibagi 2 untuk warna hitam dan warna putih, maka *setpoint* warna hitam ≥ 600 sedangkan *setpoint* warna putih ≤ 500 . Dari tabel diatas nilai warna putih dibawah 500 dan nilai warna hitam diatas 600. Sedangkan warna abu-abu muda datanya lebih mendekati nilai warna garis putih dan warna abu-abu tua datanya lebih mendekati warna garis hitam. Nilai ADC dipengaruhi oleh banyaknya intensitas cahaya yang diterima dari sinar LED, apabila sensor *photodiode* menerima intensitas cahaya lebih banyak maka resistansinya menurun, dan sebaliknya, ketika sensor *photodiode* menerima intensitas cahaya yang lebih

sedikit, nilai resistansinya meningkat. Maka dari data diatas dapat disimpulkan nilai ADC sesuai yang diinginkan. Pengujian ini dilakukan dengan cara menghubungkan sensor garis dan arduino dengan menggunakan program.

4.3.8.2. Pengujian Hasil *Mapping* Nilai Sensor

Pengujian sensor ini dilakukan untuk mengetahui nilai *error* pada sensor garis dan membuktikan bahwa sensor dapat bekerja dengan baik. Hasil *mapping* dari nilai *Process Variabel* ditunjukkan pada tabel di bawah ini.

Tabel 4.6 Hasil *Mapping* Nilai Sensor

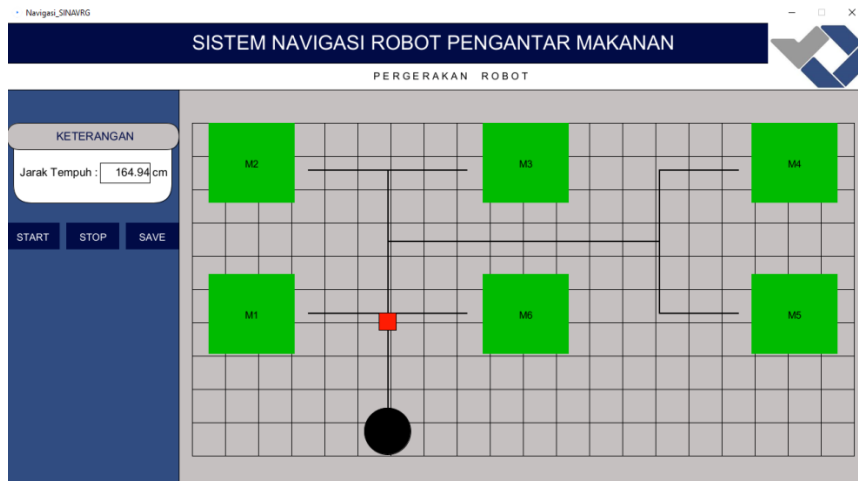
Posisi Sensor	Nilai PV (<i>error</i>)
0b00000000000001	6
0b000000000000010	5
0b0000000000000100	4
0b00000000000001000	3
0b000000000000010000	2
0b0000000000000100000	1
0b0000000000000011	6
0b00000000000000110	5
0b000000000000001100	4
0b0000000000000011000	3
0b00000000000000110000	2
0b000000000000001100000	1
0b0000000000000011000000	0
0b00000000000000110000000	-1
0b000000000000001100000000	-2
0b0000000000000011000000000	-3
0b00000000000000110000000000	-4
0b000000000000001100000000000	-5
0b0000000000000011000000000000	-6
0b000000000000000100000000	-1
0b0000000000000001000000000	-2

0b0001000000000000	-3
0b0010000000000000	-4
0b0100000000000000	-5
0b1000000000000000	-6

Dari tabel 4.6 ini dapat dilihat bahwa jumlah sensor yang digunakan yaitu 14 buah sensor *photodiode*. Dari data tersebut diasumsikan sebagai *Process Variable* (PV) dari kondisi beberapa sensor, nilai inilah yang dijadikan nilai error dari pembacaan posisi sensor. Nilai bilangan biner 1 pada posisi sensor mendefinisikan garis hitam dan nilai bilangan biner 0 posisi sensor mendefinisikan garis putih. Apabila kondisi sensor mendeteksi garis hitam sesuai yang diasumsikan dengan posisi sensor pada tabel 4.3 nilai *error*-nya akan diketahui. Apabila nilai *error*-nya adalah 0 maka posisi sensor berada di tengah garis hitam, posisi ini merupakan posisi yang ideal dari pembacaan sensor garis sehingga pergerakan robot maju kedepan akan semakin baik karena kecepatan putaran motor DC kanan dan putaran motor DC kiri sama. Jika nilai *error*-ya besar maka ini akan mempengaruhi putaran motor pada robot sehingga pergerakan robot kurang baik. Berdasarkan hasil dari tabel 4.6 dapat disimpulkan bahwa nilai yang diasumsikan sesuai dengan hasil pengujian yang dilakukan.

4.4. Pengujian Aplikasi

Pengujian dilakukan menggunakan laptop dengan cara menghubungkan USB to TTL *Converter* CP2102 dengan Modul HC-12 ke laptop dan menghubungkan Modul HC-12 yang satunya lagi ke arduino dengan pengaturan Modul HC-12 (*bluetooth*) yang sama. Kemudian buka aplikasi *processing* yang berjudul “Navigasi_SINAVRG”, lalu klik “*run*” untuk menjalankan aplikasi. Tombol “*START*” pada aplikasi berfungsi untuk menjalankan aplikasi atau pergerakan robot, Tombol “*STOP*” berfungsi untuk memberhentikan pergerakan robot, sedangkan tombol “*SAVE*” berfungsi untuk menyimpan data jarak tempuh dari pergerakan robot kemudian aplikasi akan tertutup secara otomatis.



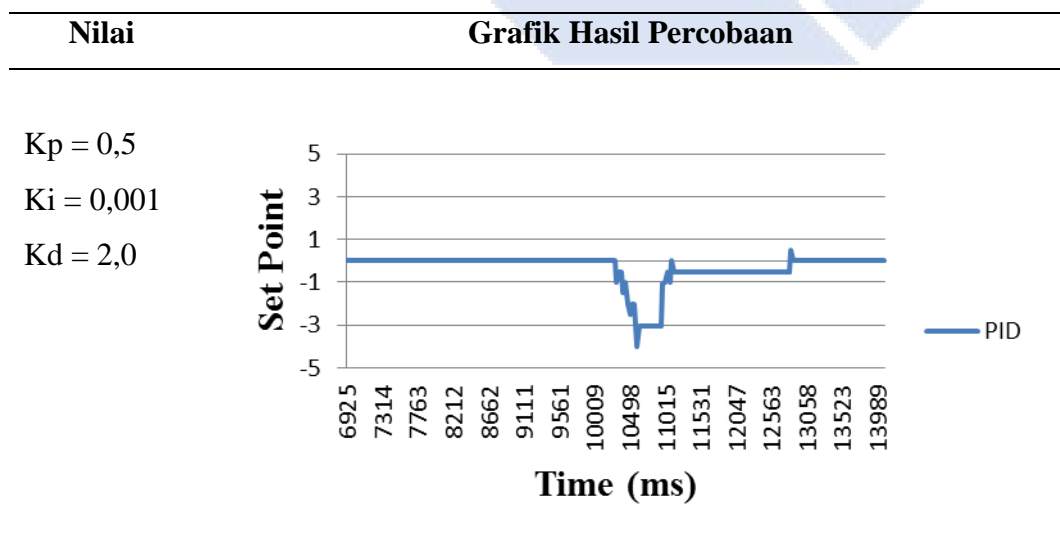
Gambar 4.14 Pergerakan Robot Maju Jarak 165 cm

Dapat dilihat dari gambar 4.14, bahwa tampilan dari sistem navigasi bisa mengetahui posisi pergerakan maju dari robot. *Error* pada pergerakan maju sumbu $y = 0,06$ cm. Pengujian dilakukan dengan menekan tombol *start* pada laptop.

4.5. Pengujian Kontrol PID Terhadap Sensor Garis

Pengujian dilakukan dengan menggunakan *setpoint* 0 untuk memeriksa seberapa besar pengaruh nilai PID terhadap pembacaan sensor garis. Hasil pengujian dapat dilihat pada grafik dibawah ini.

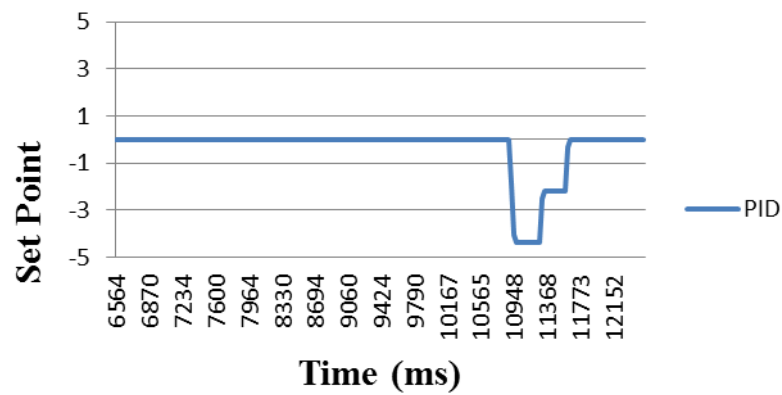
Tabel 4.7 Hasil Percobaan Perbandingan PID Terhadap Sensor Garis



$K_p = 1,0$

$K_i = 0,015$

$K_d = 1,0$



Berdasarkan percobaan pada tabel 4.7 hasil dari kedua Percobaan ini menggunakan *base* PWM yang sama yaitu 45 PWM untuk motor kiri dan 28 PWM untuk motor kanan. Dari percobaan ini dapat dilihat bahwa pemberian nilai K_p , K_i , dan K_d berpengaruh terhadap pembacaan sensor garis. Pada kondisi awal pergerakan robot berjalan cukup stabil, pada grafik ke-1 terjadi *error* pada detik 10,334 dan kembali stabil pada detik 12,833. Sedangkan pada grafik ke-2 terjadi *error* pada detik 10,915 kemudian kembali stabil pada detik 11,626. Maka dari hasil data tabel 4.7 dapat disimpulkan bahwa hasil pengujian penggunaan PID pada grafik ke-2 lebih baik dikarenakan dengan sistem kontrol PID yang digunakan pada grafik 2 perbaikan pembacaan *error* pada sensor garis lebih cepat untuk mencapai nilai *setpoint* yaitu 0.

4.6. Pengujian Keseluruhan

Pada tahap ini dilakukan pengujian keseluruhan untuk memahami apakah alat yang telah dikerjakan dapat beroperasi dan berfungsi sesuai keinginan. Hasil pengujian keseluruhan robot pengantar makanan dapat dilihat pada tabel 4.8.

Tabel 4.8 Hasil Pengujian Menentukan Meja Tanpa Beban

Percobaan (kali)	Beban (kg)	Nomor Meja	Keberhasilan	Persentase (%)
7	-	Meja 1	5 kali	70
7	-	Meja 2	6 kali	75
7	-	Meja 3	3 kali	40

Tabel 4.9 Hasil Pengujian Menentukan Meja Menggunakan Beban

Percobaan (kali)	Beban (kg)	Nomor Meja	Keberhasilan	Persentase (%)
3	1 kg	Meja 1	2 kali	80
3	0,8 kg	Meja 1	2 kali	70
3	1 kg	Meja 2	2 kali	75
3	3 kg	Meja 2	2 kali	75
2	0,8 kg	Meja 3	1 kali	40

Berdasarkan hasil pengujian pada tabel 4.8 dan tabel 4.9 yang dilakukan dengan beberapa kali percobaan, dapat dilihat bahwa keberhasilan robot dalam mengantarkan makanan dengan diberikannya beban, hal ini dikarenakan pada saat robot diberikan beban maka perputaran motor DC akan berpengaruh sehingga pergerakan robot akan semakin *smooth* maka untuk pembacaan sensor garis akan lebih stabil baik itu bergerak maju atau saat mendeteksi persimpangan. Pada saat pengujian robot menuju meja ke-2 dengan beban 3kg maka pergerakan robot kurang baik dikarenakan beban yang diberikan terlalu berat sehingga mempengaruhi putaran dari motor DC yang menyebabkan motor DC berputar lambat. Dari data hasil pengujian diatas dapat disimpulkan bahwa pertama, keberhasilan robot mengantarkan makanan pada saat diberikan beban. Kedua, apabila robot diberikan beban terlalu berat maka pergerakan robot akan semakin lambat, maka pemberian beban pada robot harus disesuaikan. Informasi tentang data waktu tempuh dapat dilihat pada tabel dibawah ini.

Tabel 4.10 Hasil Waktu Tempuh Robot Pengantar Makanan

Nomor Meja	Beban (kg)	Waktu (s)
Meja 1	-	15.13
Meja 2	-	18.56
Meja 3	-	18.87
Meja 1	1 kg	21.65
Meja 1	0.8 kg	20.15
Meja 2	1 kg	24.48
Meja 2	3 kg	29.44
Meja 3	1 kg	25.67

Berdasarkan hasil data pada tabel diatas pemberian beban pada robot mempengaruhi waktu tempuh. Dilihat pada tabel 4.10 pada meja 1 apabila robot mengantarkan makanan tanpa beban waktu tempuh yang dihasilkan 15.13 detik sedangkan apabila menggunakan beban 1kg dan 0.8kg waktu tempuh yang dihasilkan 21.65 detik dan 20,15 detik. Maka dari data hasil pengujian dapat disimpulkan apabila robot tidak diberikan beban maka waktu tempuh yang dihasilkan lebih cepat dan sebaliknya semakin berat beban yang diberikan pada robot maka waktu tempuhnya akan semakin lama.

BAB V

KESIMPULAN DAN SARAN

5.1. KESIMPULAN

Berdasarkan dari data hasil pengujian di atas maka dapat ditarik kesimpulan sebagai berikut:

1. Berdasarkan hasil percobaan secara menyeluruh, robot dapat mengantarkan makanan ke meja yang diinginkan dengan mengikuti garis yang telah dibuat, dengan lamanya perjalanan sangat bergantung kepada beban yang diberikan dan dengan maksimal beban seberat 3kg. Salah satu contoh yaitu robot dapat bergerak ke meja 1 dengan waktu tempuh 21,65 detik dengan diberikan beban 1kg. Sedangkan jika tanpa beban robot dapat bergerak meja 1 dengan waktu tempuh 15.13 detik.
2. Penentuan nilai K_p , K_i , dan K_d menggunakan metode *trial* dan *error*. Hasil pengujian pergerakan robot terbaik dengan pengaturan nilai $K_p = 1,0$, $K_i=0,015$ dan $K_d = 1,0$. Pembacaan *error* pada sensor garis yaitu -4 dengan waktu *error* 10,915 detik sampai 11,626 detik.
3. Sistem navigasi yang digunakan pada penelitian ini dapat menjalankan dan memberhentikan robot serta menampilkan hasil jarak tempuh dari pergerakan robot.
4. Robot yang dibuat berjalan mengikuti garis menggunakan sensor garis 14 kanal dimultiplekser melalui jalur *input* ADC 10bit. Sensor photodiode menghasilkan nilai ADC (0-1023). *Setpoint* nilai ADC pada warna putih ≤ 500 , dan *setpoint* nilai ADC pada warna hitam ≥ 600 .

5.2. SARAN

Dari hasil penelitian yang sudah dilakukan, masih terdapat beberapa kekurangan, untuk itu penulis mengharapkan adanya pengembangan yang dilakukan. Berikut beberapa saran yang ingin disampaikan oleh penulis:

1. Pembuatan robot pengantar makanan sebaiknya menggunakan metode sistem *tunning* yang lebih baik lagi.
2. Perlu ditambahkan orientasi untuk menentukan arah hadap dari pergerakan robot pada *software processing*.



DAFTAR PUSTAKA

- [1] L. D. Kawuri, I. Setiawan, and Darjat, "Sistem Navigasi Personal Sensor MEMS (Micro Electro-Mechanical System)," *Makal. Tugas Akhir*, 2011.
- [2] E. A. Budianto, Jamir, and D. Kisbianty, "Perancangan Navigasi Pada Robot Sepak Bola Beroda Menggunakan Metode PID (Proposional Integer Delivtif)," *J. Ilm. Mhs. Inform.*, pp. 1–10, 2019.
- [3] D. A. N. Janis, D. Pang, and Wuwung, "Rancang Bangun Robot Pengantar Makanan Line follower," *J. Tek. Elektro dan Komput.*, pp. 1–10, 2014.
- [4] A. Saefullah, E. Sunandar, and M. N. Rifai, "Prototipe Robot Pengantar Makanan Berbasis Arduino Mega Dengan Interface Web Browser," *Raharja Open J. Syst.*, vol. 10, no. 2, pp. 269–279, 2017.
- [5] D. Aryani, I. J. Dewanto, and Alfiantoro, "Prototype Alat Pengantar Makanan Berbasis Arduino Mega," *J. Pengkaj. dan Penerapan Tek. Inform.*, vol. 12, no. 2, pp. 242–250, 2019.
- [6] W. Hizburrohman, I. Al Khoiri, and J. I. Santoso, "Pengembangan Prototype Robot Pengantar Makanan," *Tugas Akhir Politek. Harapan Bersama*, 2019.
- [7] G. A. Yanno and M. I. Zuhdi, "Rancangan Bangun Prototype Robot Pengantar Makanan dan Obat Dimasa Pandemi," 2022.
- [8] D. Y. Krisna and S. Satrio, "Perancangan Robot Line Follower Pemadam Api," *J. Algoritm. Log. dan Komputasi*, no. 2, pp. 176–189, 2019.
- [9] Siswanto and H. T. Sigit, "Pelatihan Pembuatan Robot Line Follower untuk Meningkatkan Pengetahuan Robotika pada Siswa SMK Negeri 1 Kramatwatu," *J. Pemberdaya. Masy.*, vol. 1, pp. 230–240, 2019.
- [10] H. A. Dharmawan, "*Mikrokontroler Konsep Dasar dan Praktis*," Cetakan Pe. Malang: UBMedia Universitas Brawijaya Malang, 2017. [Sarean]. Available at: <https://www.google.co.id/books/edition/Mikrokontroler/GQJODwAAQBAJ?hl=id&gbpv=1&dq=Mikrokontroler&pg=PA21&printsec=frontcover>

- [11] N. Nugroho and S. Agustina, "Analisa Motor DC (Direct Current) Sebagai Penggerak Mobil Listrik," *Mikrotiga*, vol. 2, no. 1, pp. 28–34, 2015.
- [12] A. E. Wahyudi, "Perancangan dan Pembuatan Alat Ukur Jarak Digital Berbasis Arduino Menggunakan Sensor Rotary Encoder," *Karya Ilm. Univ. Muhammadiyah Surakarta*, 2015.
- [13] M. N. Yuski, W. Hadi, and A. Saleh, "Rancang Bangun Jangkar Motor DC," *Berk. Sainstek*, 2017.
- [14] D. H. Yapriyono and J. Dewanto, "Perancangan Spion Elektrik Tipe Tanduk Pada Bus Pariwisata Berukuran Besar," *J. Tek. Mesin*, vol. 16, no. 1, pp. 9–16, 2016, doi: 10.9744/jtm.16.1.9-16.
- [15] M. B. U. Kaleka, "Thermistor Sebagai Sensor Suhu," *J. Ilm. Din. Sains*, pp. 8–11, 2017.
- [16] A. Adella, M. Kamal, and A. Finawan, "Rancang Bangun Robot Mobile Line Follower Pemindah Minuman Kaleng Berbasis Arduino," *J. TEKTRONIKA*, vol. 2, no. 2, 2018.
- [17] Hc-12 Wireless Rf Uart Communication Module V2.4 User Manual, or. 10, 2018, [Sarean]. Available at: www.hc01.com
- [18] G. S. Kavitate and K. S. Gorde, "NFC Ticketing Mobile Service," *J. Electr. Electron. Eng.*, vol. 9, no. 2, 2019, doi: 10.9790/1676-09232833.
- [19] E. B. Prinandika, "Sistem Pengaturan Kecepatan Motor Pada Robot Line Follower Berbeban Menggunakan Kontroler PID," *J. Mhs. TEUB*, pp. 1–8, 2014.
- [20] H. Miftahul, Firdaus, and Derisma, "Pengontrolan Kecepatan Mobile Robot Line Follower Dengan Sistem Kendali PID," *J. TELKA*, vol. 2, no. 2, 2016.
- [21] K. Joni, M. Ulum, and Z. Abidin, "Robot Line Follower Berbasis Kendali Proportional- Integral-Derivative (PID) Untuk Lintasan Dengan Sudut Ekstrim," *J. Infotel*, vol. 8, no. 2, pp. 138–142, 2016.
- [22] F. Ben Reas Casey, "*Processing A Programming Handbook for Visual Designers and artists.*" London, Inggris, 2007. [Sarean]. Available at: <https://books.google.co.id/books?hl=id&lr=&id=tqW75bfJkxIC&oi=fnd&pg=PR19&dq=+software+sketchbook+and+a+language+for+learning+how+>

to+code&ots=So-RWJOK.

7&sig=YF9jROwjuZYuhTOcZUx6nRyE1Qg&redir_esc=y#v=onepage&q

=software sketchbook and a language for learnin





LAMPIRAN 1
DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Sony Montela
Tempat Tanggal Lahir : Pangkalpinang, 09 April 2002
Alamat Rumah : Jl Kalamaya 1 Bacang RT03/RW01
Kec. Bukit Intan
Kota Pangkalpinang
Kab. Bangka
Prov. Kep. Bangka Belitung
No.Telp/ HP : 082129087710
Email : sonymontela009@gmail.com
Jenis Kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 20 Pangkalpinang (2007-2013)
SMP Negeri 10 Pangkalpinang (2013-2016)
SMA Negeri 2 Pangkalpinang (2016-2019)

3. Pendidikan Non Formal

-

Sungailiat, 14 Februari 2023

(Sony Montela)

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Via Twoifia
Tempat Tanggal Lahir : Belinyu, 12 November 2001
Alamat Rumah : Komp. KD Panji RT 013
Desa Gunung Muda
Kec. Belinyu
Kab. Bangka
Prov. Kep. Bangka Belitung
No.Telp/ HP : 083183485003
Email : viajaya285@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 23 Parit 4 Belinyu (2007-2013)
SMP Negeri 2 Belinyu (2013-2016)
SMK YPN Belinyu (2016-2019)

3. Pendidikan Non Formal

-

Sungailiat, 14 Februari 2023

(Via Twoifia)



LAMPIRAN 2
PROGRAM ARDUINO MEGA 2560

PROGRAM LINE FOLLOWER

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
LiquidCrystal_I2C lcd2(0x26, 16, 2);
LiquidCrystal_I2C lcd3(0x23, 16, 2);
#include <EEPROM.h>
#include <Keypad.h>
#include <SoftwareSerial.h>
#define RX 50 //Connect to the TX pin of the HC-12
#define TX 51 //Connect to the RX pin of the HC-12
SoftwareSerial portSerial(50, 51);
//PIN MOTOR DRIVER
const int dir_ka=9;
const int ena_ka=10;
const int dir_ki=5;
const int ena_ki=6;
//PIN SENSOR
const int en=13;
const int S0=A0;
const int S1=A1;
const int S2=A2;
const int S3=A3;
const int sig=A6;
//PIN PROXYMITY
const int objek1=7;
const int objek2=8;
//PIN KEYPAD
const byte ROWS = 4; //baris
const byte COLS = 4; //kolom
//PIN BUZZER
const int buzzer=48;
//PIN ENCODER
const int CH_A1=2; // Kanan
const int CH_B1=40;
const int ch_A2=3; //Kiri
const int ch_B2=41;
//PIN EMERGENCY
const int emg=52;
//PIN SUARA
const int suara=53;

//VARIABEL GLOBAL
uint8_t loopForMenu=1, menuUtama=0, menu_Settings=0, menuSensors_Calibration=0,
menuPID_Settings=0, step=0;
bool Key1_state=1, Key2_state=1, Key3_state=1, Key4_state=1, Key5_state=1, Key6_state=1,
Key7_state=1, KeyA_state=1, KeyB_state=1, KeyC_state=1, KeyD_state=1, KeyE_state=1,
KeyF_state=1, KeyG_state=1;
bool lastPosition;
int dataRaw[14], dataGarisPutih[14], dataGarisHitam[14], treshold[14], Vut[14], Vlt[14],
sensorsValueHysterisis[14];
unsigned int currentSensorsState;
uint16_t_Const14bitValue;
int8_t Error,_SP,_PV, lastError;
```

```

float Kp, Ki, Kd;
long* ptrKp, * ptrKi, * ptrKd;
const uint8_t Ts=4;
float Pro, igl, der, PID;
int16_t leftMotorPWM=0, rightMotorPWM=0, SP_PWM=0;
long savedMillisValue;
int z0, z1, z2, z3, z4, z5, z6, z7, z8, z9, z10, z11, z12, z13, xsensor, proxym1, proxym2,
emergency1, meja, mejab, mejac, mejad, mejae, mejaf, mejag, oktet = 0;
char Key;
int kanan, kiri;
//Encoder
int encoder1,encoder2,rpm1,rpm2,stepp,val;
float putaran1,putaran2,v1,v2,w1,w2,s1,s2,st1,st2,stot,stotx,stoty,koorX,koorY,sy;
unsigned long waktu;
uint8_t h; //int 0-25
char Keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
//PIN KEYPAD = 32 33 34 35 36 37 38 39
byte rowPins[ROWS] = {36, 37, 38, 39} ;
byte colPins[COLS] = {32, 33, 34, 35};
Keypad pad4x4 = Keypad( makeKeymap(Keys), rowPins, colPins, ROWS, COLS );
///int logic, x;
ISR(TIMER2_OVF_vect)
{
  Sensors_Value_Using_Hysterisis();
  PV_Define();
  PID_Control("ENABLE");
}
void setup()
{Serial.begin(115200);portSerial.begin(115200);lcd.begin();lcd.backlight();char key =
pad4x4.getKey();char
val;motor_begin();sensor_begin();proxy();Emergency();Read_EEPROM_Data();Starting_Menu();
lcd2_begin();lcd3_begin();encoder();bacaencoder1();bacaencoder2();buzzer1(); Suaraku();}
void loop()
{nilaiencoder();Keykey();read_proxy1();read_proxy2();Gency();Stop(); }

```

PROGRAM ENCODER

```

voidencoder(){
pinMode(CH_A1,INPUT_PULLUP);pinMode(CH_B1,INPUT_PULLUP);attachInterrupt
(digitalPinToInterrupt(2),bacaencoder1,RISING);
pinMode(ch_A2,INPUT_PULLUP);pinMode(ch_B2,INPUT_PULLUP);attachInterrupt
(digitalPinToInterrupt(3),bacaencoder2,RISING);waktu=millis();}
void bacaencoder1(){if(digitalRead(CH_B1) != digitalRead (CH_A1)){encoder1--;}
else {encoder1++;}}
void bacaencoder2(){if(digitalRead(ch_B2) != digitalRead (ch_A2)){encoder2--;}
else {encoder2++;}}
void nilaiencoder() {
unsigned long waktusekarang = millis();
if (waktusekarang-waktu>=200) {
waktu=waktusekarang;
}
}

```

```

putaran1=(float)encoder1/135;
putaran2=(float)encoder2/135;
rpm1=(60/0.20)*putaran1; //Membaca nilai RPM
rpm2=(60/0.20)*putaran2; //Membaca nilai RPM
w1=10.466667*rpm1; // Membaca kecepatan sudut
w2=10.466667*rpm2;
v1=0.0335*w1; //Membaca kecepatan linier
v2=0.0335*w2;
s1=0.20*v1; //Membaca jarak per 200ms
s2=0.20*v2;
st1=st1+s1; //Membaca jarak total
st2=st2+s2;
stot=(st1+st2)/2;
sy = stot;
koorX = koorX;
koorY = -sy;
encoder1=0; //Untuk mereset encoder
encoder2=0;
portSerial.print(sy); portSerial.print("");
portSerial.print(koorX); portSerial.print("");
portSerial.print(koorY); portSerial.println("");
}}
void Start() {
  if (portSerial.available()){
    char val = portSerial.read();
    if(val=='D'){Key5_state = 0;}
    else{Key5_state = 1;}
    if(val=='S'){Key7_state = 0;}
    else{Key7_state = 1;}}
void Stop(){Read_Key_State();bacaNilaiKeypad();Start();while (Key7_is_Pressed()){ berhenti();
lcd.clear();}}

```

PROGRAM ALGORITMA

```

void _2_Sensors_Algorithm_Lurus()
{
  Sensors_Value_Using_Hysterisis();
  PV_Define();
  //Hitam 1, Putih 0
  if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 11
  {
    Forward(56, 38);
  }
  if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&

```



```

sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 10 Robot dikanan
{
    Turn_Left(0, 38);
    lastPosition = 0;
    delay(500);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==1 && sensorsValueHysterisis[13]==1) // 01 Robot dikiri
{
    Turn_Right(38, 0);
    lastPosition = 1;
    delay(500);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==1 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0)
{
    if(lastPosition==0){Turn_Left(0, 38); }
    else if(lastPosition==1){Turn_Right (38, 0); }
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==1 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Left(0, 38);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==1 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Left(0, 38);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==1 && sensorsValueHysterisis[5]==1 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&

```

```

sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Left(0, 38);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Right(56, 0);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Right(56, 0);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Right(56, 0);
}
}
//SIMPANG 4 KIRI
void _2_Sensors_Algorithm_Simpang_4_Kiri()
{
    Sensors_Value_Using_Hysterisis();
    PV_Define();
    //Hitam 1, Putih 0
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 11
    {
        Forward(56, 38);
    }
}
// SIMPANG 4 BELOK KIRI

```



```

    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
    {
        Turn_Right(56, 0);
    }
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
    {
        Turn_Right(56, 0);
    }
}
//SIMPANG 4 KANAN
void _2_Sensors_Algorithm_Simpang_4_Kanan0()
{
    Sensors_Value_Using_Hysterisis();
    PV_Define();
    //Hitam 1, Putih 0
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 11
    {
        Forward(56, 38);
    }
    // SIMPANG 4 BELOK KANAN
    if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==1 && sensorsValueHysterisis[13]==1) // 01 Robot simpang
    {
        Turn_Right(38, 0);
        lastPosition = 1;
        delay(500);
    }
    if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==1 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&

```



```

    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
    {
        Turn_Right(56, 0);
    }
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
    {
        Turn_Right(56, 0);
    }
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
    {
        Turn_Right(56, 0);
    }
}
//SIMPANG 3 KANAN
void _2_Sensors_Algorithm_Simpang_3_Kanan0()
{
    Sensors_Value_Using_Hysterisis();
    PV_Define();
    //Hitam 1, Putih 0
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 11
    {
        Forward(56, 38);
    }
    // SIMPANG 3 BELOK KANAN
    if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==1 && sensorsValueHysterisis[13]==1) // 01 Robot simpang

```

```

    {
      Turn_Right(38, 0);
      lastPosition = 1;
      delay(500);
    }
    if(sensorsValueHysteresis[0]==0 && sensorsValueHysteresis[1]==1 &&
sensorsValueHysteresis[2]==0 && sensorsValueHysteresis[3]==0 &&
sensorsValueHysteresis[4]==0 && sensorsValueHysteresis[5]==0 &&
sensorsValueHysteresis[6]==0 && sensorsValueHysteresis[7]==0 &&
sensorsValueHysteresis[8]==0 && sensorsValueHysteresis[9]==0 &&
sensorsValueHysteresis[10]==0 && sensorsValueHysteresis[11]==0 &&
sensorsValueHysteresis[12]==1 && sensorsValueHysteresis[13]==0) // 01 Robot simpang
    {
      Turn_Right(38, 0);
      lastPosition = 1;
      delay(500);
    }
    if(sensorsValueHysteresis[0]==1 && sensorsValueHysteresis[1]==0 &&
sensorsValueHysteresis[2]==0 && sensorsValueHysteresis[3]==0 &&
sensorsValueHysteresis[4]==0 && sensorsValueHysteresis[5]==0 &&
sensorsValueHysteresis[6]==0 && sensorsValueHysteresis[7]==0 &&
sensorsValueHysteresis[8]==0 && sensorsValueHysteresis[9]==0 &&
sensorsValueHysteresis[10]==0 && sensorsValueHysteresis[11]==0 &&
sensorsValueHysteresis[12]==0 && sensorsValueHysteresis[13]==1) // 01 Robot simpang
    {
      Turn_Right(38, 0);
      lastPosition = 1;
      delay(500);
    }
    //////////////////////////////////////
    if(sensorsValueHysteresis[0]==0 && sensorsValueHysteresis[1]==0 &&
sensorsValueHysteresis[2]==0 && sensorsValueHysteresis[3]==0 &&
sensorsValueHysteresis[4]==0 && sensorsValueHysteresis[5]==0 &&
sensorsValueHysteresis[6]==0 && sensorsValueHysteresis[7]==0 &&
sensorsValueHysteresis[8]==0 && sensorsValueHysteresis[9]==1 &&
sensorsValueHysteresis[10]==1 && sensorsValueHysteresis[11]==0 &&
sensorsValueHysteresis[12]==0 && sensorsValueHysteresis[13]==0)
    {
      if(lastPosition==0){Turn_Left(0, 38); }
      else if(lastPosition==1){Turn_Right (38, 0); }
    }
    if(sensorsValueHysteresis[0]==0 && sensorsValueHysteresis[1]==0 &&
sensorsValueHysteresis[2]==0 && sensorsValueHysteresis[3]==0 &&
sensorsValueHysteresis[4]==0 && sensorsValueHysteresis[5]==1 &&
sensorsValueHysteresis[6]==1 && sensorsValueHysteresis[7]==0 &&
sensorsValueHysteresis[8]==0 && sensorsValueHysteresis[9]==0 &&
sensorsValueHysteresis[10]==0 && sensorsValueHysteresis[11]==0 &&
sensorsValueHysteresis[12]==0 && sensorsValueHysteresis[13]==0) // 00
    {
      Turn_Left(0, 38);
    }
    if(sensorsValueHysteresis[0]==0 && sensorsValueHysteresis[1]==0 &&
sensorsValueHysteresis[2]==0 && sensorsValueHysteresis[3]==0 &&
sensorsValueHysteresis[4]==0 && sensorsValueHysteresis[5]==1 &&
sensorsValueHysteresis[6]==1 && sensorsValueHysteresis[7]==1 &&

```

```

sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Left(0, 38);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==1 && sensorsValueHysterisis[5]==1 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Left(0, 38);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Right(56, 0);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Right(56, 0);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Right(56, 0);
}
}
//SIMPANG 3 KIRI
void _2_Sensors_Algorithm_Simpang_3_Kiri()
{
    Sensors_Value_Using_Hysterisis();
    PV_Define();
    //Hitam 1, Putih 0
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&

```



```

sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 11
{
    Forward(56, 38);
}
// SIMPANG 3 BELOK KIRI
if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==1 && sensorsValueHysterisis[13]==1) // 01 Robot simpang
{
    Turn_Left(0, 38);
    lastPosition = 0;
    delay(500);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==1 && sensorsValueHysterisis[13]==0) // 01 Robot simpang
{
    Turn_Left(0, 38);
    lastPosition = 0;
    delay(500);
}
if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==1) // 01 Robot simpang
{
    Turn_Left(0, 38);
    lastPosition = 0;
    delay(500);
}
////////////////////
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==1 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0)
{
    if(lastPosition==0){Turn_Left(0, 38); }
}

```



```

sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Right(56, 0);
}
}
//SIMPANG Y KANAN
void _2_Sensors_Algorithm_Simpang_Y_Kanan0()
{
    Sensors_Value_Using_Hysterisis();
    PV_Define();
    //Hitam 1, Putih 0
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 11
    {
        Forward(56, 38);
    }
    // SIMPANG Y BELOK KANAN
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==1 && sensorsValueHysterisis[13]==1) // 01 Robot simpang
    {
        Turn_Right(38, 0);
        lastPosition = 1;
        delay(500);
    }
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==1 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==1 && sensorsValueHysterisis[13]==1) // 01 Robot simpang
    {
        Turn_Right(38, 0);
        lastPosition = 1;
        delay(500);
    }
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==0 &&

```



```

    Turn_Right(56, 0);
}
if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==0 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Right(56, 0);
}
}
//SIMPANG Y KIRI
void _2_Sensors_Algorithm_Simpang_Y_Kiri0()
{
    Sensors_Value_Using_Hysterisis();
    PV_Define();
    //Hitam 1, Putih 0
    if(sensorsValueHysterisis[0]==0 && sensorsValueHysterisis[1]==0 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 11
    {
        Forward(56, 38);
    }
    // SIMPANG Y BELOK KIRI
    if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 01 Robot simpang
    {
        Turn_Left(0, 38);
        lastPosition = 0;
    }
    if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==1 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==0 &&
sensorsValueHysterisis[8]==0 && sensorsValueHysterisis[9]==0 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 01 Robot simpang
    {
        Turn_Left(0, 38);
        lastPosition = 0;
    }
    if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==0 && sensorsValueHysterisis[3]==0 &&
sensorsValueHysterisis[4]==0 && sensorsValueHysterisis[5]==0 &&

```



```

sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==0 && sensorsValueHysterisis[11]==0 &&
sensorsValueHysterisis[12]==0 && sensorsValueHysterisis[13]==0) // 00
{
    Turn_Right(56, 0);
}
}
//HITAM
void _2_Sensors_Algorithm_Dead0()
{
    Sensors_Value_Using_Hysterisis();
    PV_Define();
    //Hitam 1, Putih 0
    if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==1 && sensorsValueHysterisis[3]==1 &&
sensorsValueHysterisis[4]==1 && sensorsValueHysterisis[5]==1 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==1 && sensorsValueHysterisis[11]==1 &&
sensorsValueHysterisis[12]==1 && sensorsValueHysterisis[13]==1)
    {
        berhenti();
    }
}
void _2_Sensors_Algorithm_Dead()
{
    Sensors_Value_Using_Hysterisis();
    PV_Define();
    //Hitam 1, Putih 0
    if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==1 && sensorsValueHysterisis[3]==1 &&
sensorsValueHysterisis[4]==1 && sensorsValueHysterisis[5]==1 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==1 && sensorsValueHysterisis[11]==1 &&
sensorsValueHysterisis[12]==1 && sensorsValueHysterisis[13]==1)
    {
        berhenti();
        lcd.setCursor(6,2); lcd.print("RAK == 1");
    }
}
void _2_Sensors_Algorithm_Dead2()
{
    Sensors_Value_Using_Hysterisis();
    PV_Define();
    //Hitam 1, Putih 0
    if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==1 && sensorsValueHysterisis[3]==1 &&
sensorsValueHysterisis[4]==1 && sensorsValueHysterisis[5]==1 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==1 && sensorsValueHysterisis[11]==1 &&
sensorsValueHysterisis[12]==1 && sensorsValueHysterisis[13]==1)
    {
        berhenti();
    }
}

```

```

        lcd.setCursor(6,2); lcd.print("RAK == 2");
    }
}
void _2_Sensors_Algorithm_Dead3()
{
    Sensors_Value_Using_Hysterisis();
    PV_Define();
    //Hitam 1, Putih 0
    if(sensorsValueHysterisis[0]==1 && sensorsValueHysterisis[1]==1 &&
sensorsValueHysterisis[2]==1 && sensorsValueHysterisis[3]==1 &&
sensorsValueHysterisis[4]==1 && sensorsValueHysterisis[5]==1 &&
sensorsValueHysterisis[6]==1 && sensorsValueHysterisis[7]==1 &&
sensorsValueHysterisis[8]==1 && sensorsValueHysterisis[9]==1 &&
sensorsValueHysterisis[10]==1 && sensorsValueHysterisis[11]==1 &&
sensorsValueHysterisis[12]==1 && sensorsValueHysterisis[13]==1)
    {
        berhenti();
        lcd.setCursor(6,2); lcd.print("RAK == 3");
    }
}

////////////////////////////////////STEP////////////////////////////////////
void _2_Sensors_Algorithm_Meja_1()
{
    if (step==0){
        _2_Sensors_Algorithm_Simpang_4_Kiri0();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
    if (step==1){
        _2_Sensors_Algorithm_Simpang_4_Kanan0();
        _2_Sensors_Algorithm_Dead0();
    }
}

void _2_Sensors_Algorithm_Meja_2()
{
    if (step==0){
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
    if (step==1){
        _2_Sensors_Algorithm_Simpang_Y_Kanan0();
        _2_Sensors_Algorithm_Dead0();
    }
}

void _2_Sensors_Algorithm_Meja_3()
{
    if (step==0){
        _2_Sensors_Algorithm_Simpang_3_Kanan0();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
}

```

```

if (step==1){
    _2_Sensors_Algorithm_Simpang_Y_Kiri2();
}
if (step==2){
    _2_Sensors_Algorithm_Lurus();
    _2_Sensors_Algorithm_Dead0();
}
}

void _2_Sensors_Algorithm_Meja_4()
{
    if (step==0){
//    _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Simpang_Y_Kanan0();
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
    if (step==1){
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Simpang_Y_Kanan0();
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Dead0();
    }
}

void _2_Sensors_Algorithm_Meja_5()
{
    if (step==0){
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Simpang_Y_Kanan0();
        _2_Sensors_Algorithm_Simpang_3_Kanan0();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
    if (step==1){
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Simpang_Y_Kiri0();
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Dead0();
    }
}

void _2_Sensors_Algorithm_Meja_6()
{
    if (step==0){
        _2_Sensors_Algorithm_Simpang_4_Kanan0();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
    if (step==1){
        _2_Sensors_Algorithm_Simpang_4_Kiri0();
        _2_Sensors_Algorithm_Dead0();
    }
}

```

```

void _2_Sensors_Algorithm_Meja_1_2()
{
  if (step==0){
    _2_Sensors_Algorithm_Simpang_4_Kiri0();
    _2_Sensors_Algorithm_Dead();
    read_proxy1();
  }
  if (step==1){
    _2_Sensors_Algorithm_Simpang_4_Kiri0();
    _2_Sensors_Algorithm_Simpang_3_Kiri0();
    _2_Sensors_Algorithm_Dead2();
    read_proxy1();
  }
  if (step==2){
    _2_Sensors_Algorithm_Simpang_Y_Kanan0();
    _2_Sensors_Algorithm_Dead0();
  }
}

```

```

void _2_Sensors_Algorithm_Meja_1_3()
{
  if (step==0){
    _2_Sensors_Algorithm_Simpang_4_Kiri0();
    _2_Sensors_Algorithm_Dead();
    read_proxy1();
  }
  if (step==1){
    _2_Sensors_Algorithm_Simpang_4_Kiri0();
    _2_Sensors_Algorithm_Simpang_3_Kanan0();
    _2_Sensors_Algorithm_Dead2();
    read_proxy1();
  }
  if (step==2){
    _2_Sensors_Algorithm_Simpang_Y_Kiri0();
    _2_Sensors_Algorithm_Dead0();
  }
}

```

```

void _2_Sensors_Algorithm_Meja_1_4()
{
  if (step==0){
    _2_Sensors_Algorithm_Simpang_4_Kiri0();
    _2_Sensors_Algorithm_Dead();
    read_proxy1();
  }
  if (step==1){
    _2_Sensors_Algorithm_Simpang_4_Kiri0();
    _2_Sensors_Algorithm_Simpang_Y_Kanan0();
    _2_Sensors_Algorithm_Simpang_3_Kiri0();
    _2_Sensors_Algorithm_Lurus();
    _2_Sensors_Algorithm_Dead2();
    read_proxy1();
  }
  if (step==2){
    _2_Sensors_Algorithm_Lurus();
  }
}

```

```
    _2_Sensors_Algorithm_Simpang_Y_Kanan0();  
    _2_Sensors_Algorithm_Simpang_3_Kiri0();  
    _2_Sensors_Algorithm_Dead0();  
  }  
}
```

```
void _2_Sensors_Algorithm_Meja_1_5()  
{  
  if (step==0){  
    _2_Sensors_Algorithm_Simpang_4_Kiri0();  
    _2_Sensors_Algorithm_Dead();  
    read_proxy1();  
  }  
  if (step==1){  
    _2_Sensors_Algorithm_Simpang_4_Kiri0();  
    _2_Sensors_Algorithm_Simpang_Y_Kanan0();  
    _2_Sensors_Algorithm_Simpang_3_Kiri0();  
    _2_Sensors_Algorithm_Lurus();  
    _2_Sensors_Algorithm_Dead2();  
    read_proxy1();  
  }  
  if (step==2){  
    _2_Sensors_Algorithm_Simpang_Y_Kanan0();  
    _2_Sensors_Algorithm_Simpang_3_Kiri0();  
    _2_Sensors_Algorithm_Lurus();  
    _2_Sensors_Algorithm_Dead0();  
  }  
}
```

```
void _2_Sensors_Algorithm_Meja_1_6()  
{  
  if (step==0){  
    _2_Sensors_Algorithm_Simpang_4_Kiri0();  
    _2_Sensors_Algorithm_Dead();  
    read_proxy1();  
  }  
  if (step==1){  
    _2_Sensors_Algorithm_Lurus();  
    _2_Sensors_Algorithm_Dead2();  
    read_proxy1();  
  }  
  if (step==2){  
    _2_Sensors_Algorithm_Simpang_4_Kiri0();  
    _2_Sensors_Algorithm_Dead0();  
  }  
}
```

```
void _2_Sensors_Algorithm_Meja_2_3()  
{  
  if (step==0){  
    _2_Sensors_Algorithm_Simpang_3_Kiri0();  
    _2_Sensors_Algorithm_Dead();  
    read_proxy1();  
  }  
  if (step==1){
```

```

    _2_Sensors_Algorithm_Lurus();
    _2_Sensors_Algorithm_Dead2();
    read_proxy1();
}
if (step==2){
    _2_Sensors_Algorithm_Simpang_Y_Kiri0();
}
if (step==3){
    _2_Sensors_Algorithm_Lurus();
    _2_Sensors_Algorithm_Dead0();
}
}

void _2_Sensors_Algorithm_Meja_2_4()
{
    if (step==0){
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
    if (step==1){
        _2_Sensors_Algorithm_Simpang_Y_Kanan0();
        _2_Sensors_Algorithm_Simpang_Y_Kiri0();
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Dead2();
        read_proxy1();
    }
    if (step==2){
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Simpang_Y_Kanan0();
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Dead0();
    }
}

void _2_Sensors_Algorithm_Meja_2_5()
{
    if (step==0){
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
    if (step==1){
        _2_Sensors_Algorithm_Simpang_Y_Kanan0();
        _2_Sensors_Algorithm_Simpang_Y_Kiri0();
        _2_Sensors_Algorithm_Simpang_3_Kanan0();
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Dead2();
        read_proxy1();
    }
    if (step==2){
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Simpang_Y_Kiri0();
        _2_Sensors_Algorithm_Simpang_3_Kiri0();

```



```

    _2_Sensors_Algorithm_Dead0();
  }
}

void _2_Sensors_Algorithm_Meja_2_6()
{
  if (step==0){
    _2_Sensors_Algorithm_Simpang_4_Kanan0();
    _2_Sensors_Algorithm_Dead2();
    read_proxy1();
  }
  if (step==1){
    _2_Sensors_Algorithm_Simpang_4_Kanan0();
    _2_Sensors_Algorithm_Simpang_3_Kiri0();
    _2_Sensors_Algorithm_Dead();
    read_proxy1();
  }
  if (step==2){
    _2_Sensors_Algorithm_Simpang_Y_Kanan0();
    _2_Sensors_Algorithm_Dead0();
  }
}

void _2_Sensors_Algorithm_Meja_3_4()
{
  if (step==0){
    _2_Sensors_Algorithm_Simpang_3_Kanan0();
    _2_Sensors_Algorithm_Dead();
    read_proxy1();
  }
  if (step==1){
    _2_Sensors_Algorithm_Simpang_Y_Kiri0();
    _2_Sensors_Algorithm_Simpang_3_Kiri0();
    _2_Sensors_Algorithm_Lurus();
    _2_Sensors_Algorithm_Dead2();
    read_proxy1();
  }
  if (step==2){
    _2_Sensors_Algorithm_Lurus();
    _2_Sensors_Algorithm_Simpang_Y_Kanan0();
    _2_Sensors_Algorithm_Simpang_3_Kiri0();
    _2_Sensors_Algorithm_Dead0();
  }
}

void _2_Sensors_Algorithm_Meja_3_5()
{
  if (step==0){
    _2_Sensors_Algorithm_Simpang_3_Kanan0();
    _2_Sensors_Algorithm_Dead();
    read_proxy1();
  }
  if (step==1){
    _2_Sensors_Algorithm_Simpang_Y_Kiri0();
    _2_Sensors_Algorithm_Simpang_3_Kanan0();
  }
}

```

```

    _2_Sensors_Algorithm_Lurus();
    _2_Sensors_Algorithm_Dead2();
    read_proxy1();
}
if (step==2){
    _2_Sensors_Algorithm_Lurus();
    _2_Sensors_Algorithm_Simpang_Y_Kiri0();
    _2_Sensors_Algorithm_Simpang_3_Kiri0();
    _2_Sensors_Algorithm_Dead0();
}
}

```

```

void _2_Sensors_Algorithm_Meja_3_6()
{
    if (step==0){
        _2_Sensors_Algorithm_Simpang_4_Kanan0();
        _2_Sensors_Algorithm_Dead2();
        read_proxy1();
    }
    if (step==1){
        _2_Sensors_Algorithm_Simpang_4_Kanan0();
        _2_Sensors_Algorithm_Simpang_3_Kanan0();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
    if (step==2){
        _2_Sensors_Algorithm_Simpang_Y_Kiri3();
    }
    if (step==3){
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Dead0();
    }
}

```

```

void _2_Sensors_Algorithm_Meja_4_5()
{
    if (step==0){
        _2_Sensors_Algorithm_Simpang_Y_Kanan0();
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
    if (step==1){
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Dead2();
        read_proxy1();
    }
    if (step==2){
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Simpang_Y_Kiri0();
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Dead0();
    }
}
}

```

```

void _2_Sensors_Algorithm_Meja_4_6()
{
    if (step==0){
        _2_Sensors_Algorithm_Simpang_4_Kanan0();
        _2_Sensors_Algorithm_Dead2();
        read_proxy1();
    }
    if (step==1){
        _2_Sensors_Algorithm_Simpang_4_Kanan0();
        _2_Sensors_Algorithm_Simpang_Y_Kanan0();
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
    if (step==2){
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Simpang_Y_Kanan0();
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Dead0();
    }
}

```

```

void _2_Sensors_Algorithm_Meja_5_6()
{
    if (step==0){
        _2_Sensors_Algorithm_Simpang_4_Kanan0();
        _2_Sensors_Algorithm_Dead2();
        read_proxy1();
    }
    if (step==1){
        _2_Sensors_Algorithm_Simpang_4_Kanan0();
        _2_Sensors_Algorithm_Simpang_Y_Kanan0();
        _2_Sensors_Algorithm_Simpang_3_Kanan0();
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Dead();
        read_proxy1();
    }
    if (step==2){
        _2_Sensors_Algorithm_Lurus();
        _2_Sensors_Algorithm_Simpang_Y_Kiri0();
        _2_Sensors_Algorithm_Simpang_3_Kiri0();
        _2_Sensors_Algorithm_Dead0();
    }
}

```



LAMPIRAN 3
PROGRAM *PROCESSING*

PROGRAM SOFTWARE PROCESSING

```
import processing.serial.*;
import controlP5.*;
Serial myport;
ControlP5 cp5;
PFont Font1, Font2, Font3;
PImage photo;
Table table;
String val, namafile;
float koorX, koorY, st, Jarak;
color c1 = #010B46; // Memberi Warna
color c2 = #ffffff;
color c3 = #304C81;
color c4 = #C5C0C0;
color c5 = #00BB00;
int x1 = 562;
int y1 = 600;

void setup ()
{
  background(200);
  size(1300,700);
  photo = loadImage("Polman.png");
  myport = new Serial(this, "COM6", 115200);
  cp5 = new ControlP5(this);
  Font1 = createFont("Century.vlw", 15);
  Font2 = createFont("TimesNewRoman.vlw", 30);
  Font3 = createFont("TimesNewRoman.vlw", 17);

  // Menampilkan tombol START, STOP dan SAVE
  cp5.addButton("START")
    .setPosition(0,300)
    .setSize(80,40)
    .setFont(Font1)
    .setColorBackground(color(c1))
    .setColorForeground(color(0,255,0))
    ;

  cp5.addButton("STOP")
    .setPosition(90,300)
    .setSize(80,40)
    .setFont(Font1)
    .setColorBackground(color(c1))
    .setColorForeground(color(0,255,0))
    ;

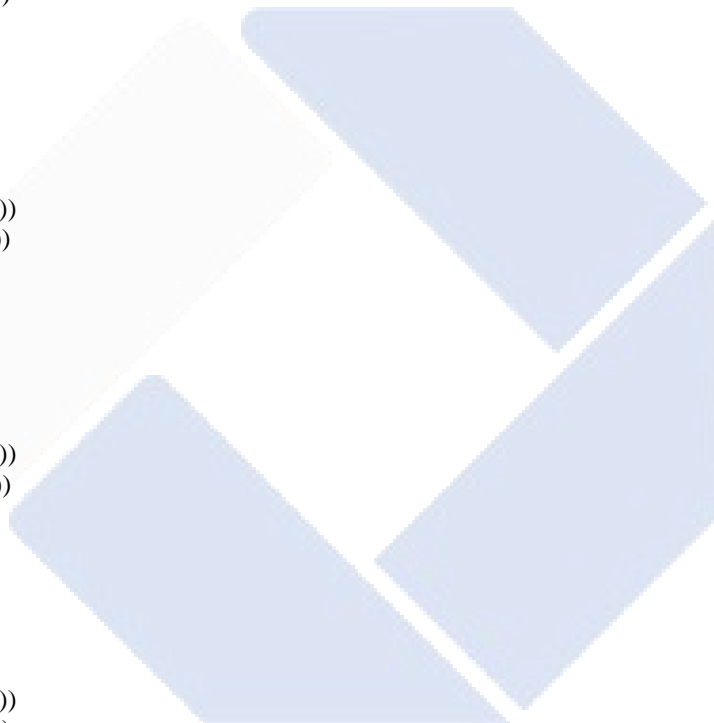
  cp5.addButton("SAVE")
    .setPosition(180,300)
    .setSize(80,40)
    .setFont(Font1)
    .setColorBackground(color(c1))
    .setColorForeground(color(0,255,0))
    ;
}
```

```

// Menampilkan tombol Meja-Meja
cp5.addButton("M1")
.setPosition(305,377)
.setSize(130,120)
.setFont(Font1)
.setColorBackground(color(c5))
.setColorForeground(color(c1))
.setColorLabel(color(0,0,0))
;
cp5.addButton("M2")
.setPosition(305,150)
.setSize(130,120)
.setFont(Font1)
.setColorBackground(color(c5))
.setColorForeground(color(c1))
.setColorLabel(color(0,0,0))
;
cp5.addButton("M3")
.setPosition(719,150)
.setSize(130,120)
.setFont(Font1)
.setColorBackground(color(c5))
.setColorForeground(color(c1))
.setColorLabel(color(0,0,0))
;
cp5.addButton("M4")
.setPosition(1125,150)
.setSize(130,120)
.setFont(Font1)
.setColorBackground(color(c5))
.setColorForeground(color(c1))
.setColorLabel(color(0,0,0))
;
cp5.addButton("M5")
.setPosition(1125,377)
.setSize(130,120)
.setFont(Font1)
.setColorBackground(color(c5))
.setColorForeground(color(c1))
.setColorLabel(color(0,0,0))
;
cp5.addButton("M6")
.setPosition(719,377)
.setSize(130,120)
.setFont(Font1)
.setColorBackground(color(c5))
.setColorForeground(color(c1))
.setColorLabel(color(0,0,0))
;

table = new Table();
table.addColumn("Jarak_Tempuh");
}

```



```

void draw()
{
// Membuat kotak tampilan awal
fill(c1);
rect(0,0,1300,60);
fill(c2);
rect(0,60,1300,40);
fill(c3);
rect(0,100,260,600);
fill(c4);
rect(260,100,1300,600);
// Menampilkan Foto
image(photo, 1150, 0);

// Membuat Grid
fill(0);
for (int i = 280; i < 1290; i += 50){
  line (i,150,i,650);
}
for (int i = 150; i < 680; i += 50){
  line (280,i,1280,i);
}

// Membuat garis sumbu Y
line(575,220,575,600); //Tengah
line(576,220,576,600);
line(985,220,985,435); //M4 & M5
line(986,220,986,435);

// Membuat garis sumbu X
line(455,435,695,435); //M1
line(455,436,695,436);
line(455,220,695,220); //M2
line(455,221,695,221);
line(576,327,986,327); //Tengah
line(576,328,986,328);
line(985,220,1105,220); // M4
line(985,221,1105,221);
line(985,435,1105,435); // M5
line(985,436,1105,436);

// Membuat Kotak Home
ellipse(575,613,70,70); // Titik awal Robot/Home

// Membuat simulasi robot
smooth();
fill(#FF1C03);
square(x1+koorX, y1+koorY, 26);

// Menampilkan text
fill(255);
textFont(Font2);
text("SISTEM NAVIGASI ROBOT PENGANTAR MAKANAN",280,40);
fill(0);
textFont(Font1);

```



```

text(" P E R G E R A K A N   R O B O T ",550,85);

// Kotak Keterangan
fill(255);
rect(10, 150, 239, 120, 28);
fill(c4);
rect(0, 150, 260, 40, 28);
fill(255);
rect(141,210,75,30);
fill(c1);
textFont(Font3);
text("KETERANGAN",75,176);
fill(0);
text("Jarak Tempuh : ",20,230);
text(nf(st,1,2),164,230);
text("cm",220,230);

// Serial Komunikasi Arduino ke Processing
if(myport.available(>0)
{
  val = myport.readStringUntil('\n');
  if (val != null){
    String [] list = split(val, ',');
    st = float(list[0]);
    koorX = float(list[1]);
    koorY = float(list[2]);
  }

  int h = hour();
  int min = minute();
  int sec = second();
  TableRow newRow = table.addRow();
  newRow.setFloat("Jarak_Tempuh", st);
}}
void START()
{
  myport.write('D');
}
void STOP()
{
  myport.write('S');
}

void SAVE(){
  myport.write('A');
  int dy = day();
  int m = month();
  int y = year();
  int h = hour();
  int min = minute();
  int sec = second();
  namafile = ("Berusaha/" + "Berusaha-" + "[" + str(dy) + str(m) + str(y) + "]" +
  str(h) + str(min) + str(sec) + ".csv");
  saveTable(table, namafile);
  exit();}

```