

BAB III

ANALISIS DAN DESAIN

Dalam skripsi ini akan dirancang sebuah prototype kecil dari robot yang dapat membantu kerja di perindustrian. Pada penelitian ini prototype akan menggunakan *infrared sensor* dan *Load Cell*. Prototype akan dibangun berbasis arduino yang kemudian akan menggunakan sensor *load cell* untuk mendeteksi berat. Jika berat sudah sesuai maka robot akan bergerak mengikuti garis.

3.1 Analisis Permasalahan

Pada bagian ini penulis akan menjabarkan analisis permasalahan yang ada.

3.1.1 Pokok Permasalahan

Di Indonesia terdapat banyak perindustrian seperti industri tekstil, industri makanan, dan industri otomotif. Tidak hanya memberikan dampak positif bagi Indonesia berupa pemasukan pajak, perindustrian juga memberikan dampak negatif bagi para pekerja nya berupa masalah kesehatan dan keamanan pada saat bekerja. Tidak hanya kesehatan dan keamanan pada saat bekerja, keefektifan pada saat bekerja juga merupakan masalah pada perindustrian. Ketidakefektifan ini bisa berupa kendaraan transportasi barang yang sudah melakukan perjalanan meskipun muatan pada kendaraan belum penuh. Kendaraan yang dikendarai oleh manusia juga sering kali membuat kendaraan sering menabrak barang-barang pada perindustrian. Biaya tenaga kerja yang mahal juga menjadi kendala dalam perindustrian ketika ingin memakai tenaga dari seorang pekerja.

3.1.2 Solusi

Berdasarkan pokok permasalahan diatas, peneliti menawarkan solusi. Solusi yang ditawarkan adalah sebuah robot yang dapat melakukan navigasi secara otomatis dan akan mulai bergerak ketika muatan sudah terisi penuh. Hal ini memungkinkan agar kerja dari industri menjadi lebih optimal.

Penggunaan robot dipilih dikarenakan penggunaan robot tidak akan memerlukan biaya tenaga manusia sehingga bisa menghemat biaya. Penggunaan robot juga dipilih dikarenakan penggunaan robot dapat menentukan apakah jumlah muatan pada kendaraan sudah sesuai dengan kapasitas angkut dari kendaraan. Penggunaan robot juga bisa membuat navigasi pada industri menjadi lebih baik.

3.1.3 Analisis Kebutuhan Sistem

Berdasarkan solusi yang ditawarkan oleh peneliti, peneliti akan merancang robot yang mampu melakukan navigasi secara otomatis. Oleh karena itu peneliti memerlukan komponen yang mampu mengatur navigasi dan mampu menggerakkan robot. Peneliti juga memerlukan komponen yang mampu mendeteksi jalan pada perindustrian.

Selain dapat melakukan navigasi, robot juga mampu menghitung jumlah muatan pada kendaraan sudah sesuai dengan kapasitas angkut dari kendaraan. Oleh karena itu peneliti memerlukan komponen yang dapat mengukur berat dari muatan.

3.1.3.1 Pemilihan Alat Sebagai Pusat Kontrol

Peneliti telah membandingkan 2 pilihan alat pusat kontrol untuk dipakai dalam penelitian ini. Perbandingan tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1 Perbandingan Arduino Nano dengan Raspberry Pi

Perbandingan	Arduino Nano	Raspberry Pi
Harga (Berdasarkan di Tokopedia per Maret 2019)	Rp. 38.900,00	Rp. 520.000,00
Daya	7V - 12V	5V
Berat(g)	37	45
Ukuran(mm)	43 x 18	85.6 x 53.98 x 17
RAM	8 KB	1 GB
<i>Processor</i>	ATmega328P	ARM BCM 2837
Digital I/O Pins	14	14
Analog Inputs	8	0
USB Ports	1	1-2
Ethernet	-	10/100 Mbps

Perbandingan	Arduino Nano	Raspberry Pi
OS	-	Raspbian, Ubuntu MATE, Snappy Ubuntu Core, Windows 10,RISC OS, Debian, Arch Linux ARM
Bahasa Program	Arduino	C, C++, Java, Phyton

Pada penelitian ini, peneliti akan menggunakan arduino dikarenakan harga yang lebih murah. Ukuran dari Arduino nano lebih kecil sehingga cocok digunakan sebagai *prototype*. Penggunaan Arduino Nano sendiri sudah memenuhi kebutuhan dalam penelitian ini.

3.1.3.2 Pemilihan Alat untuk Mendeteksi Garis

Peneliti telah membandingkan 2 pilihan alat pendeteksi garis untuk dipakai dalam penelitian ini. Perbandingan tersebut dapat dilihat pada Tabel 3.2.

Tabel 3.2 Perbandingan alat pendeteksi garis

Perbandingan	TCRT5000	Sensor inframerah
Harga (Berdasarkan di Tokopedia per Maret 2019)	Rp. 5.900,00	Rp. 6.500,00
Daya	3.3V – 5V	3.3V – 5V
Chipset	LM393	LM393
Jarak Efektif	0,2mm - 15mm	2cm – 30cm
Ukuran	3,2mm x 1,4mm	3,1mm x 1,5mm
Hasil output	Digital (0 dan 1) & Analog	Digital (0 dan 1)

Pada penelitian ini peneliti akan menggunakan sensor inframerah TCRT5000 sebagai alat untuk mendeteksi garis. Pada TRCT5000 hasil keluaran yang dihasilkan berupa nilai digital dan analog dibandingkan dengan sensor inframerah yang hanya memberikan hasil keluaran digital. Hasil keluaran analog diperlukan oleh peneliti untuk mendapatkan nilai

yang akurat. Oleh karena itu pada penelitian ini peneliti akan menggunakan sensor TCRT5000.

3.1.3.3 Pemilihan Alat untuk Mendeteksi Berat

Pada penelitian ini, peneliti akan menggunakan *load cell* sebagai alat untuk mengukur berat dari muatan pada robot. Penggunaan *load cell* juga akan menggunakan modul HX711 untuk mengkonversikan data dari *load cell* agar bisa dibaca oleh mikrokontroller. *Load cell* pada penelitian ini akan menggunakan *load cell 1kg*, penggunaan ini dipilih karena sudah mencukupi kebutuhan dari penelitian. Spesifikasi modul HX711 dapat dilihat pada Tabel 3.3.

Tabel 3.3 Spesifikasi modul HX711

Keterangan	HX711
Harga (Berdasarkan di Tokopedia per Maret 2019)	Rp. 50.000,00
Daya	2 V – 5,5 V
Output Rate	10SPS / 80SPS
Noise Rejection	50Hz / 60Hz
Temperatur kerja	-40 sampai dari +85 celcius
Ukuran	40mm x 20mm

3.1.3.4 Pemilihan Alat untuk Menggerakkan Robot

Penelitian ini DC Motor 1:48 dipilih sebagai penggerak dari robot. Pemilihan DC Motor ini dikarenakan ukurannya yang kecil dan Motor ini sudah dilengkapi dengan Gearbox yang memudahkan perakitan robot. Spesifikasi dari DC Motor dapat dilihat pada Tabel 3.4.

Tabel 3.4 Spesifikasi DC Motor

Keterangan	DC Motor
Harga (Berdasarkan di Tokopedia per Maret 2019)	Rp. 25.900,00
Daya	3V – 12V
Rasio	1: 48
Kecepatan roda (Meter/Menit)	3V (20), 5V (39), 6V (48)
Arus	70mA - 250mA
Berat	50gram
Torsi	Hingga 0,8kg
Diameter Roda	66mm
Lebar Roda	27mm

3.1.3.5 Pemilihan Alat untuk Mengontrol Motor

Peneliti telah membandingkan 2 pilihan alat yang dapat mengontrol motor untuk dipakai dalam penelitian ini. Perbandingan tersebut dapat dilihat pada tabel Tabel 3.5.

Tabel 3.5 Perbandingan alat pengontrol motor

Keterangan	L298N	AN79
Harga (Berdasarkan di Tokopedia per Maret 2019)	Rp. 29.900,00	Rp. 26.000,00
Daya Masukan	5V – 12V	2V – 10V
Arus	2A	1.5A
Berat	30gram	20gram
Dimensi	43mm x 43mm x 27mm	24,7mm x 21mm x 5mm

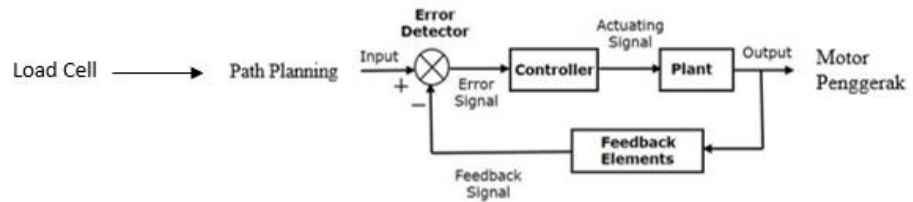
Pada penelitian ini modul L298N dipilih sebagai pengontrol gerak motor. Pemilihan modul ini dikarenakan modul L298N mampu memberikan arus keluaran yang stabil, sehingga lebih baik dalam mengontrol kecepatan motor.

3.2 Desain Sistem

Pada bagian ini penulis akan menjabarkan desain sistem yang akan dibuat. Berdasarkan Analisa yang sudah dilakukan maka penulis akan menggunakan Arduino nano sebagai pusat kendali dari robot, Sensor TCRT5000 sebagai alat

untuk mendeteksi garis, modul L298N sebagai modul untuk mengontrol motor DC, dan *load cell* untuk mengukur berat muatan pada robot.

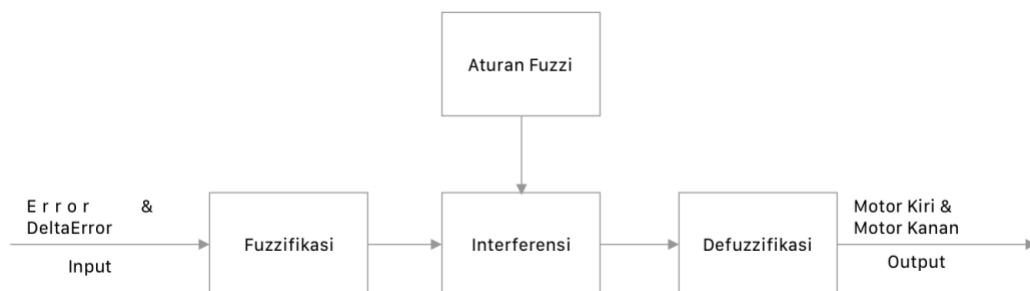
3.2.1 Desain Sistem Kontrol *Closed Loop*



Gambar 3.1 Desain closed loop

Pada Gambar 3.1 merupakan gambar sistem yang akan digunakan oleh robot. Pada sistem ini robot akan mendeteksi berat dari muatan, apabila beban muatan sudah sesuai maka robot akan membaca jalur yang sudah dibuat. Kemudian data ini akan diterjemahkan oleh *controller*. Pada penelitian ini peneliti menggunakan *fuzzy logic* sebagai *controller* robot. Data yang sudah diterjemahkan oleh *controller* akan diproses untuk menggerakkan *plant* atau motor. Gerakan motor kemudian akan memberikan umpan balik melalui sensor. Apabila terdapat selisih antara sensor dan jalur yang sudah dibuat maka akan terdeteksi oleh *error detector* sehingga robot akan menyesuaikan dan tidak melenceng dari jalur yang sudah dibuat.

3.2.2 Diagram Algoritma Fuzzy

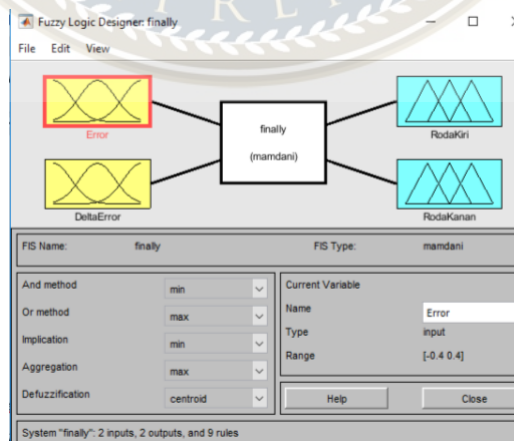


Gambar 3.2 Diagram fuzzy logic

Diagram dari algoritma Fuzzi yang akan digunakan oleh penulis dapat dilihat pada Gambar 3.2. Pada gambar dapat kita lihat *input* dari fuzzi berupa *Error* yaitu selisih pembacaan sensor saat ini dengan posisi ideal dari robot. Sedangkan *DeltaError* merupakan selisih *error* saat ini dengan *error* yang terjadi sebelum nya. *Input* ini akan masuk ke dalam proses fuzzifikasi untuk di rubah menjadi himpunan fuzzi. Himpunan fuzzi ini kemudian akan di proses melalui interferensi berdasarkan konsep aturan fuzzi yang sudah di terapkan. Hasil keluaran dari interferensi ini akan dilakukan proses defuzzifikasi untuk menghasilkan nilai sesungguhnya yang akan digunakan untuk mengatur kecepatan motorKiri dan motorKanan.

3.2.3 Desain *Fuzzy Logic Controller* pada Matlab

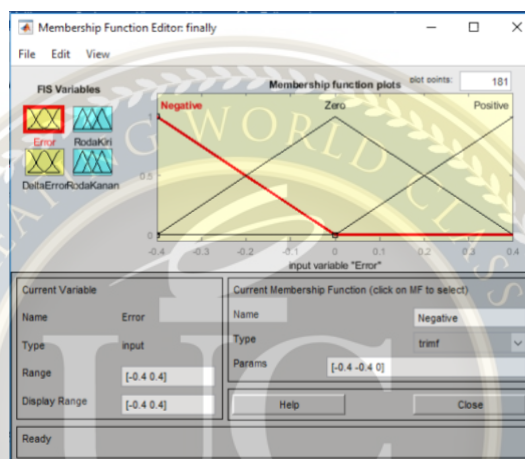
Desain *fuzzy logic controller* pada robot *line follower* dengan menggunakan *Matlab Fuzzy Logic Toolbox* dapat dilihat pada Gambar 3.3. Pada desain ini terdapat 2 *input* yaitu *error* dan *deltaError*. Dan untuk *output* terdapat 2 yaitu *output* untuk mengatur rodaKiri dan rodaKanan dari robot *line follower*.



Gambar 3.3 Desain *fuzzy logic controller*

3.2.3.1 Desain Input Membership Function Error

Desain dari *membership function error* yang merupakan *input* dari *fuzzy logic* dapat dilihat pada Gambar 3.4. Terdapat 3 *membership function* pada desain ini, yaitu *negative*, *zero*, dan *positive*. Pada penelitian ini, peneliti menetapkan nilai *error* berdasarkan selisih hasil pembacaan tiga sensor inframerah terhadap *set point* atau poisis ideal dari robot.



Gambar 3.4 Desain *input membership function error*

Detail nilai dari *input membership function error* berdasarkan Gambar 3.4 disajikan dalam bentuk tabel dapat dilihat pada Tabel 3.6.

Tabel 3.6 Nilai *input membership function error*

<i>Membership Function</i>	Nilai Grafik	Nilai <i>error</i> sebenarnya
<i>Negative</i>	$(-0.4) - 0$ Puncak di (-0.4)	$(-0.4) - 0$ Puncak di (-0.4)
<i>Zero</i>	$(-0.4) - 0.4$ Puncak di 0	$(-0.4) - 0.4$ Puncak di 0
<i>Positive</i>	$0 - 0.4$ Puncak di 0.4	$0 - 0.4$ Puncak di 0.4

3.2.3.2 Desain Input Membership Function DeltaError

Desain dari *membership function deltaError* yang merupakan *input* dari *fuzzy logic* dapat dilihat pada Gambar 3.5. Terdapat 3 *membership function* pada desain ini, yaitu *negative*, *zero*, dan *positive*. Pada penelitian ini, peneliti menetapkan nilai *deltaError* berdasarkan selisih *error* yang terjadi saat ini terhadap *error* yang terjadi sebelumnya.



Gambar 3.5 Desain *input membership function deltaError*

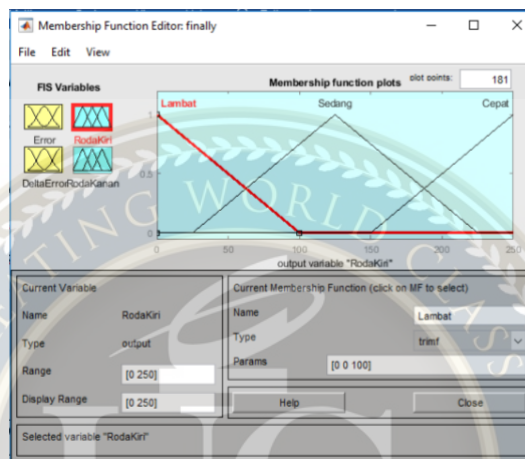
Detail nilai dari *input membership function deltaError* berdasarkan Gambar 3.5 disajikan dalam bentuk tabel dapat dilihat pada Tabel 3.7.

Tabel 3.7 Nilai *input membership function deltaError*

<i>Membership Function</i>	Nilai Grafik	Nilai <i>deltaError</i> sebenarnya
<i>Negative</i>	$(-0.8) - 0$ Puncak di (-0.8)	$0.8 - 0$ Puncak di 0.4
<i>Zero</i>	$(-0.8) - 0.8$ Puncak di 0	$(-0.8) - 0.8$ Puncak di 0
<i>Positive</i>	$0 - 0.8$ Puncak di 0.8	$0 - 0.8$ Puncak di 0.8

3.2.3.3 Desain Output Membership Function RodaKiri

Desain dari *membership function* rodaKiri yang merupakan *output* dari *fuzzy logic* dapat dilihat pada Gambar 3.6. Terdapat 3 *membership function* pada desain ini, yaitu lambat, sedang, dan cepat. Pada penelitian ini, peneliti menetapkan nilai rodaKiri berdasarkan kecepatan rodaKiri untuk mengatur gerak dari robot.



Gambar 3.6 Desain *output membership function* rodaKiri

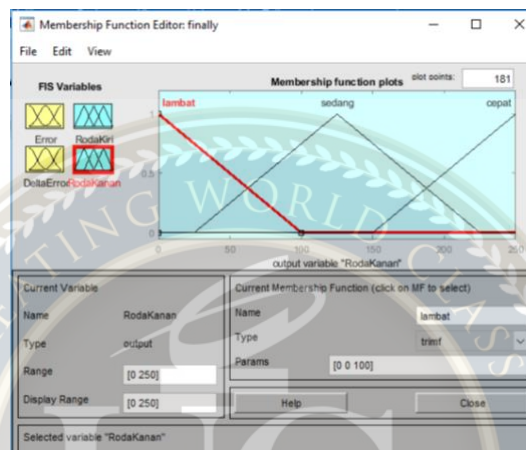
Detail nilai dari *output membership function* rodaKiri berdasarkan Gambar 3.6 disajikan dalam bentuk tabel bisa dilihat pada Tabel 3.8.

Tabel 3.8 Nilai *output membership function* rodaKiri

<i>Membership Function</i>	Nilai Grafik	Nilai rodaKiri sebenarnya
Lambat	0 – 100 Puncak di 0	0 – 100 Puncak di 0
Sedang	25 - 225 Puncak di 125	25 - 225 Puncak di 125
Cepat	150 – 250 Puncak di 250	150 – 250 Puncak di 250

3.2.3.4 Nilai Output Membership Function RodaKanan

Desain dari *membership function* rodaKanan yang merupakan *output* dari *fuzzy logic* dapat dilihat pada Gambar 3.7. Terdapat 3 *membership function* pada desain ini, yaitu lambat, sedang, dan cepat. Pada penelitian ini, peneliti menetapkan nilai rodaKanan berdasarkan kecepatan rodaKanan untuk mengatur gerak dari robot.



Gambar 3.7 Desain output *membership function* rodaKanan

Detail nilai dari *output membership function* rodaKanan berdasarkan Gambar 3.7 disajikan dalam bentuk tabel dapat dilihat pada Tabel 3.9.

Tabel 3.9 Nilai *output membership function* rodaKanan

<i>Membership Function</i>	Nilai Grafik	Nilai rodaKiri sebenarnya
Lambat	0 – 100 Puncak di 0	0 – 100 Puncak di 0
Sedang	25 - 225 Puncak di 125	25 - 225 Puncak di 125
Cepat	150 – 250 Puncak di 250	150 – 250 Puncak di 250

3.2.3.5 Desain Rule Fuzzy Logic

Rule yang akan digunakan pada fuzzy logic. Rule ini akan menggunakan *input error* dan *deltaError*, untuk menentukan hasil *output* untuk rodaKiri dan rodaKanan dapat dilihat pada Tabel 3.10. Rule ini menggunakan prinsip jika maka.

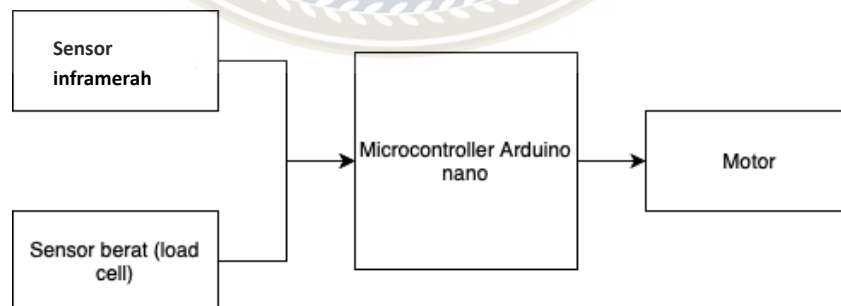
Tabel 3.10 Rule fuzzy logic

Rule	Input	Output	Hasil
Rule 1	IF <i>error = negative</i> AND <i>deltaError = negative</i>	THEN rodaKiri = cepat AND rodaKanan = lambat	Robot belok kanan
Rule 2	IF <i>error = negative</i> AND <i>deltaError = zero</i>	THEN rodaKiri = cepat AND rodaKanan = lambat	Robot belok kanan
Rule 3	IF <i>error = negative</i> AND <i>deltaError = positive</i>	THEN rodaKiri = sedang AND rodaKanan = lambat	Robot belok kanan
Rule 4	IF <i>error = zero</i> AND <i>deltaError = negative</i>	THEN rodaKiri = sedang AND rodaKanan = lambat	Robot belok kanan
Rule 5	IF <i>error = zero</i> AND <i>deltaError = zero</i>	THEN rodaKiri = sedang AND rodaKanan = sedang	Robot beregerak lurus
Rule 6	IF <i>error = zero</i> AND <i>deltaError = positive</i>	THEN rodaKiri = lambat AND rodaKanan = sedang	Robot belok kiri

Rule	Input	Output	Hasil
Rule 7	IF <i>error = positive AND deltaError = negative</i>	THEN rodaKiri = lambat AND rodaKanan = sedang	Robot belok kiri
Rule 8	IF <i>error = positive AND deltaError = zero</i>	THEN rodaKiri = lambat AND rodaKanan = cepat	Robot belok kiri
Rule 9	IF <i>error = positive AND deltaError = positive</i>	THEN rodaKiri = lambat AND rodaKanan = cepat	Robot belok kiri

3.2.4 Diagram Blok Sistem

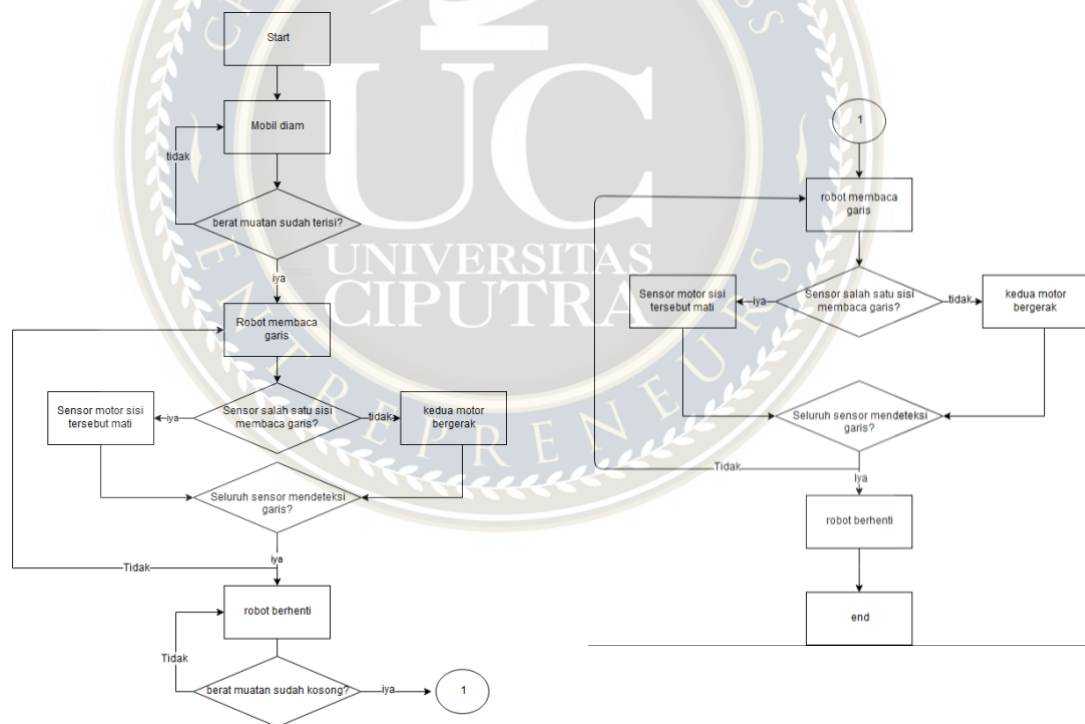
Diagram blok dari sistem dapat dilihat pada Gambar 3.8. Pada gambar dapat dilihat data dari sensor inframerah dan dari sensor berat akan dikirimkan ke arduino nano yang berfungsi sebagai mikrokontroler. Data tersebut akan diolah untuk menggerakkan robot. Gerak robot ini akan bergerak mengikuti garis yang terbaca oleh sensor inframerah



Gambar 3.8 Diagram Blok Sistem

3.2.5 Diagram Alir Sistem

Diagram yang menjelaskan kerja sistem dapat dilihat pada Gambar 3.9. Tahap awal dari robot adalah membaca sensor berat. Apabila muatan dari beban sudah terisi dan sudah sesuai maka robot akan mulai membaca garis. Pada saat pembacaan garis, apabila sensor salah satu sisi membaca garis maka motor dari sisi sensor tersebut mati sehingga dia bergerak ke arah tersebut. Apabila kedua sensor mendeteksi garis, maka motor kedua robot akan bergerak sehingga robot bergerak maju. Ketika semua sensor mendeteksi garis, maka robot akan berhenti dan menunggu beban muatan menjadi kosong, kemudian robot akan bergerak lagi hingga semua sensor membaca garis dan akhirnya berhenti.



Gambar 3.9 Diagram alir sistem

3.2.5.1 Penentuan Berat Muatan Pada Robot

Pada sistem yang telah dirancang penulis, robot akan mulai membaca garis ketika beban muatan sudah terisi. Pada penelitian ini penulis akan menggunakan nilai 50gram sebagai nilai yang akan menjadi acuan pada sensor berat sebelum robot mulai membaca garis. Penggunaan nilai 50gram didasari dari jumlah muatan yang dapat terisi pada alat pengukur berat robot.

3.2.5.2 Penentuan Pembacaan Sensor Inframerah

Pembacaan nilai referensi dari sensor inframerah dapat dilihat pada Tabel 3.11, pada sensor nilai 0 berarti sensor tidak mendeteksi objek dan nilai 1 berarti sensor mendeteksi objek. Referensi ini nantinya akan memberikan nilai yang akan digunakan untuk menghitung *input* pada *fuzzy logic*. Pada tabel terdapat kolom kondisi, *value*, *error*, *minError* dan *maxError*. Kolom kondisi menjelaskan kondisi dari kemungkinan pembacaan sensor. *Value* adalah pemberian nilai berdasarkan hasil pembacaan sensor. *Error* adalah nilai selisih antara *value* hasil pembacaan sensor dengan kondisi ideal atau *set point*. Sedangkan *minError* merupakan selisih *error* minimal yang mungkin terjadi bila dibandingkan dengan *error* sebelumnya. Sedangkan *maxError* merupakan selisih *error* maksimal yang mungkin terjadi bila dibandingkan dengan *error* sebelumnya.

Tabel 3.11 Hasil pembacaan sensor inframerah

Kondisi	Sensor	Sensor	Sensor	Value	Error	MinError	MaxError
	Kiri	Tengah	Kanan				
1	0	0	1	0.2	-0.4	-0.8	0
2	0	1	1	0.4	-0.2	-0.6	0.2
3	1	1	1	0.6	0	-0.4	0.4
4	0	1	0	0.6	0	-0.4	0.4
5	1	1	0	0.8	0.2	-0.2	0.6
6	1	0	0	1	0.4	0	0.8

3.2.5.3 Penentuan Kecepatan Motor

Pada penelitian ini, peneliti menggunakan modul L298N sebagai pengontrol kecepatan motor. Penggunaan modul L298N memiliki kisaran nilai 0 hingga 255. Berdasarkan nilai tersebut, penulis membagi nilai tersebut menjadi tiga kelompok kecepatan. Kecepatan lambat untuk nilai PWM dibawah 100. Kecepatan sedang untuk nilai PWM antara 100 dan 200. Dan kecepatan cepat untuk nilai PWM diatas 200.

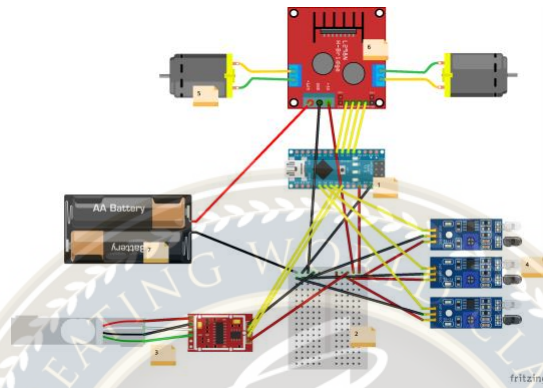
3.2.6 Diagram Breadboard

Desain diagram breadboard Arduino dapat dilihat pada Gambar

3.10. Rangkaian breadboard dari Arduino terdiri dari:

1. Arduino Nano
2. Mini Breadboard

3. Load Cell & HX711
4. Sensor Inframerah
5. Motor
6. Modul L298N
7. Baterai



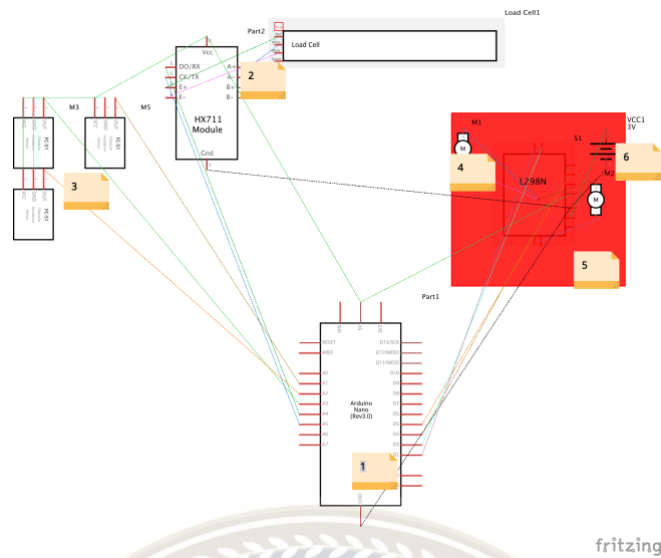
Gambar 3.10 Diagram breadboard

3.2.7 Diagram Skematik

Desain diagram skematik Arduino dapat dilihat pada Gambar 3.11.

Keterangannya sebagai berikut:

1. Arduino Nano
2. Load Cell & HX711
3. Sensor Inframerah
4. Motor
5. Modul L298N
6. Baterai

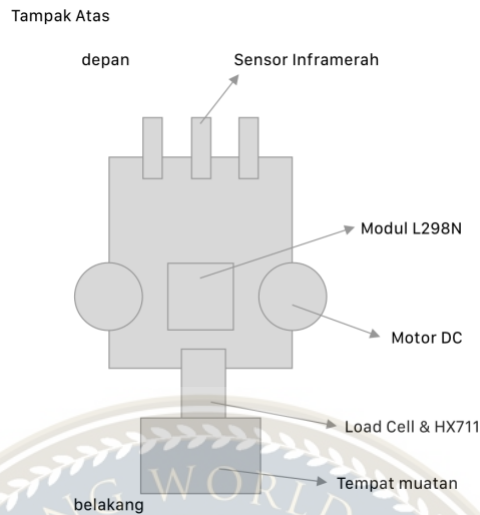


Gambar 3.11 Diagram skematik

3.2.8 Perancangan Robot

Gambar rancangan robot tampak atas dapat dilihat pada Gambar 3.12. Pada bagian depan terdapat tiga buah sensor inframerah yang digunakan untuk mendeteksi garis berwarna hitam. Pada bagian tengah robot terdapat modul L298N yang digunakan untuk mengatur kecepatan motor DC yang terletak pada kiri dan kanan robot. Pada bagian belakang digunakan untuk meletakkan sensor berat *load cell* dan modul HX711. *Load cell* digunakan untuk mengukur berat dari muatan sedangkan modul HX711

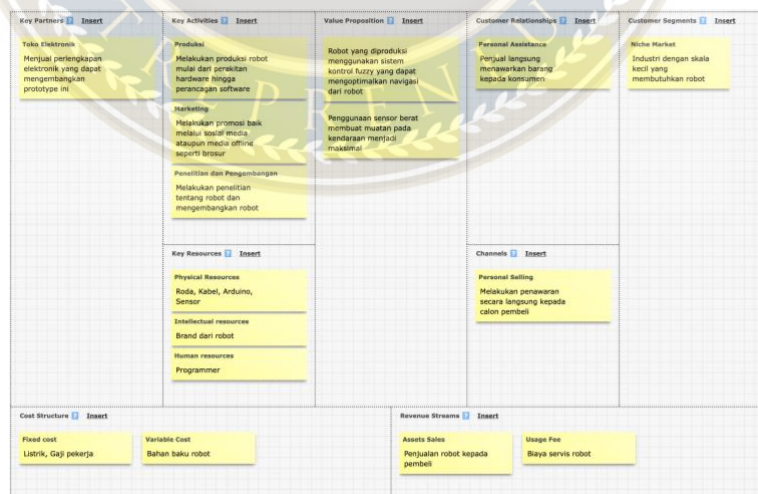
berfungsi mengkonversi hasil bacaan dari *load cell* untuk diproses oleh *microcontroller*.



Gambar 3.12 Gambar rancangan robot

3.3 Nilai Entrepreneurship

Dalam pembuatan sistem ini, peneliti tidak hanya membuat sistem dari permasalahan yang ditemukan, tetapi juga membuat BMC (*Business Model Canvas*) untuk berjalannya bisnis penelitian ini. Rancangan *Business Model Canvas* dapat dilihat pada Gambar 3.13.



Gambar 3.13 BMC

3.3.1 Oportunity Creation

Peluang yang ada pada penelitian ini adalah robot yang dapat melakukan navigasi secara otomatis, dimana robot dapat mendeteksi beban muatan sebelum melakukan sebuah perjalanan agar kerja robot menjadi lebih optimal. Robot akan membantu mengoptimalkan kerja dari sebuah industri dan menghemat biaya pengeluaran.

3.3.2 Market Sensitivity

Dari sisi market, robot yang dibuat ini menargetkan industri-industri kecil yang sudah membutuhkan penggunaan robot namun belum menggunakan robot. Selain itu *electric engineering* juga merupakan target market dari produk ini karena *electric engineering* nanti nya mampu mengembangkan robot ini menjadi lebih baik.

Pada tahun 2017 jumlah industri kecil di Indonesia mencapai 4,59 juta unit usaha. Dari jumlah ini penulis menargetkan 1% dari jumlah tersebut yaitu sebesar 40 ribu sebagai pengguna robot ini.

Perhitungan biaya pada bisnis ini akan dibagi menjadi tiga bagian yaitu Modal, *fixed cost*, dan *variable cost*. Pada bisnis ini diperlukan modal sebesar Rp. 350.000,00. Modal tersebut terdiri dari pembelian solder dan meja. Rincian dapat dilihat pada Tabel 3.12.

Tabel 3.12 Perhitungan modal

Modal	Biaya
Solder	Rp. 100.000,00
Meja	Rp. 250.000,00

Fixed cost pada bisnis ini merupakan biaya tetap yang akan dikeluarkan setiap bulan nya. Setiap bulan nya biaya tetap yang harus dikeluarkan sebesar Rp. 15.500.000,00. *Fixed cost* dalam pembuatan bisnis ini meliputi biaya sewa rumah, gaji marketing, gaji teknisi, listrik, dan internet. Rincian dari *fixed cost* dapat dilihat pada Tabel 3.13.

Tabel 3.13 Perhitungan fixed cost

Fixed Cost	Biaya
Sewa Rumah	Rp. 10.000.000,00
Gaji Marketing	Rp. 2.000.000,00
Gaji Teknisi	Rp. 2.500.000,00
Listrik	Rp. 500.000,00
Internet	Rp. 500.000,00

Variable cost pada bisnis ini merupakan biaya yang akan digunakan untuk proses produksi sebuah robot. Biaya yang akan digunakan untuk setiap produksi robot adalah sebesar Rp. 1.986.500,00. *Variable cost* dalam setiap proses produksi meliputi akrilik, Arduino, 3 buah sensor inframerah, baterai, sensor berat 500kg, modul HX711, 3 buah roda, modul L298N, dan Timah. Rincian dari *variable cost* dapat dilihat pada Tabel 3.14.

Tabel 3.14 Perhitungan variable cost

Variable Cost	Jumlah	Total Biaya
Akrilik	1	Rp. 250.000,00
Arduino	1	Rp. 38.900,00
Sensor Inframerah	3	Rp. 17.700,00

Variable Cost	Jumlah	Total Biaya
Baterai	1	Rp. 250.000,00
Sensor Berat 500kg	1	Rp. 800.000,00
Modul HX711	1	Rp. 50.000,00
Roda	3	Rp. 500.000,00
Modul L298N	1	Rp. 29.900,00
Timah	1	Rp. 50.000,00

Berdasarkan rincian biaya modal, *fixed cost*, dan *variable cost* penulis menetapkan harga jual berdasarkan *variable cost* atau biaya produksi sebuah produk ditambah dengan keuntungan. Keuntungan yang akan diambil dari sebuah produk adalah sebesar 100% dari biaya pembuatan produk sehingga harga jual menjadi Rp. 3.973.000,00. Berdasarkan perhitungan tersebut maka titik balik modal atau BEP dapat dihitung melalui rumus:

$$\begin{aligned}
 \text{BEP} &= \text{fixed cost} / (\text{Harga Jual} - \text{Variable Cost}) \\
 &= \text{Rp. 15.500.000} / (\text{Rp. 3.973.000} - \text{Rp. 1.986.500}) \\
 &= \text{Rp. 15.500.000} / \text{Rp. 1.986.500} \\
 &= 7,802
 \end{aligned}$$

Berdasarkan perhitungan tersebut maka setiap bulan nya harus dilakukan penjualan sebanyak 8 buah produk robot untuk mencapai titik balik modal.

3.3.3 Creativite Innovation

Robot ini inovatif dikarenakan robot ini menawarkan sistem kendali menggunakan Fuzzi yang dapat membuat navigasi robot menjadi lebih baik. Selain itu robot ini dilengkapi dengan pendeteksi berat sehingga pekerjaan pada perindustrian menjadi lebih optimal.

