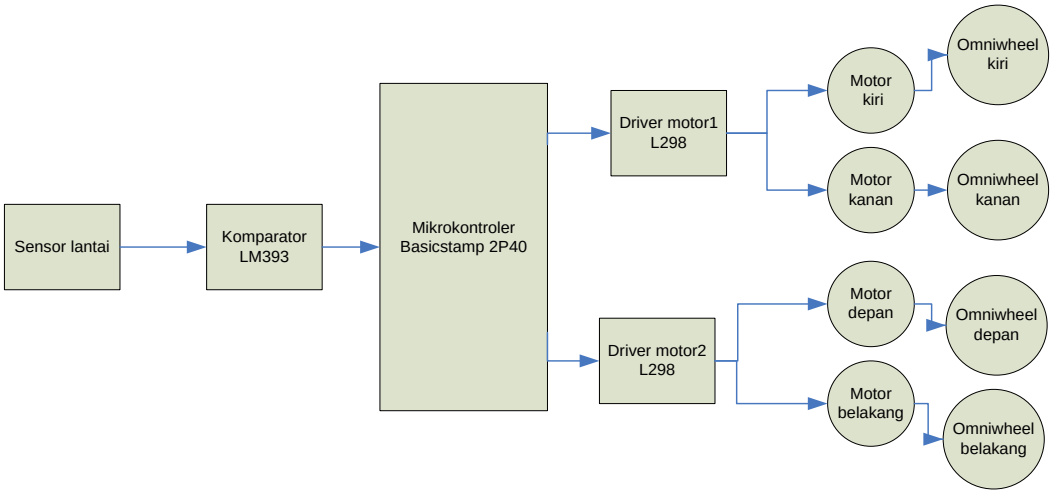
BAB III

PERANCANGAN SISTEM

Pada bab ini diterangkan tentang langkah-langkah dalam merancang atau membuat robot *line follower* tersebut. Perancangan ini meliputi perancangan keras (*hardware*) maupun perangkat lunak (*software*). Perancangan *hardware* meliputi perancangan skematik rangkaian, komponen elektronik yang dibutuhkan, desain PCB, dll. Perancangan *software* meliputi perancangan algoritma program, pembuatan program, dll.

3.1. Diagram Blok System

Bab ini membahas tentang perancangan mengenai komponen-komponen yang digunakan pada *robot line follower*.Sebelum masuk di perancangan *hardware* dan *software* maka akan di buat terlebih dahulu diagram blok sistem



Gambar 3.1 *Diagram Blok Syatem*

Berikut penjelasan tentang tiap-tiap blok sistem pada gambar 3.1.

1. **Mikrokontroler**

   Pada Robot *Line follower* ini digunakan *Besic stamp* 2P40 yang berfungsi sebagai pengontrolan pergerakan robot, pemeriksaan sensor serta pengaturan pergerakan motor yang diinginkan sehingga robot dapat

bergerak sesuai dengan pembacaan sensor agar robot tidak keluar jalur yang telah ditentukan.

2. Sensor Garis

Sensor garis sebagai pengirim menggunakan *lad* dan sebagai penerima menggunakan *pototransistor* yang masing-masing berjumlah 12 sensor. Sensor garis ini berfungsi untuk mendeteksi jalur yang ada, dan data yang ada kan di atur komparator, selanjutnya data digital di kirim ke mikrokontroler sehingga mikrokontroler dapat mengatur pergerakan robot agar tidak keluar dari jalur.

3. Komparator LM 393

Komparator disini menggunakan IC LM393,komparator berfungsi untuk mengatur sensor garis agar pembacaan sensor pada jalur selalu tepat, dan selanjutnya data pada sensor di kirim ke mikrokontroler. Karena IC LM393 memiliki dua buah op-amp,tiap op-amp mengontrol satu sensor, sedangkan sensor memiliki 12 sensor maka pada komparator ini menggunakan 6 buah IC LM393.

4. Driver Motor

Driver motor berfungsi sebagai sakelar untuk mengatur pergerakan motor, dalam robot *line follower* ini menggunakan dua driver motor tiap driver motor mengontrol dua motor dan dapat mengatur motor DC berputar atau tidak, driver motor pertama mengontrol motor pada roda kiri dan kanan dan pada driver motor ke dua mengontrol motor pada roda depan dan belakang.

5. Motor DC

Motor DC sebagai penggerak robot yang dapat bergerak maju mundur atau bergerak kiri kanan, pada *line follower* ini menggunakan empat penggerak motor DC yang terpasang pada ke empat sisi bagian robot yaitu pada bagian kiri, kanan, depan, dan belakang robot. Dimana motor DC pada roda kiri dan kanan bertujuan bergerak maju dan mundur, sedangkan motor DC pada roda depan dan belakang bertujuan untuk bergerakan kiri dan kanan. Motor DC terhubung pada *gearbox* yang

berfungsi untuk menambah tenaga putaran agar motor DC dapat berputar dengan beban yang ada pada robot

6. Roda *Omniwheel*

Omniwheel roda adalah roda dengan 2 derajat kebebasan, *omniwheel* tidak seperti roda biasa tetapi sebuah roda khusus. *Omniwhee*l terdiri dari roda inti besar dan sepanjang periferal ada terdapat banyak roda kecil tambahan yang mempunyai poros tegak lurus pada roda inti. Roda *ohmniwheel* ini dipasang pada motor DC yang telah memiliki *gearbox* sehingga mengikuti putaran dan gerakan motor DC. Pada perancangan ini menggunakan 4 buah roda *omniwheel* yang terdapat pada sebelah kiri dan kanan terdapat masing-masing satu buah,sebelah kiri 1 buah sebelah kanan 1 buah dan tardapat pada depan dan belakang masing-masing 1 buah, bagian depan 1 buah dan bagian belakang 1 buah jadi keseluruhannya menggunakan 4 buah roda omniwheel.

3.2. Perangkat Keras (*Hardware*)

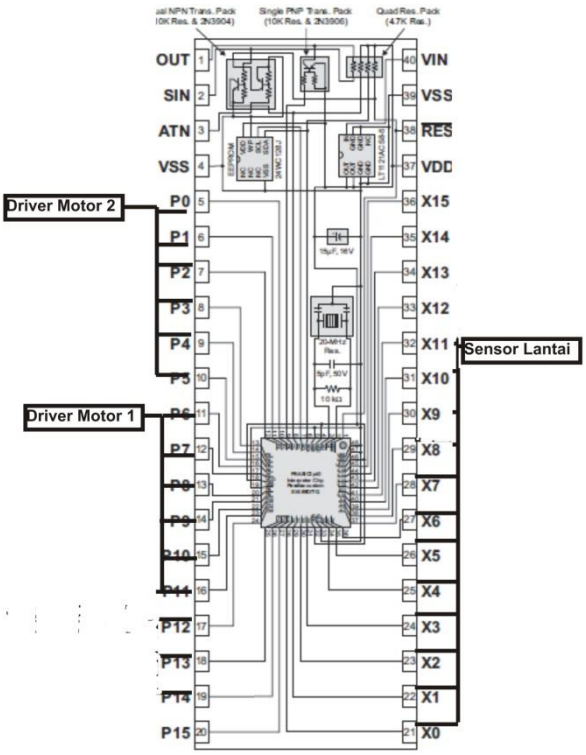
3.2.1. Perancangan Mekanik

Pada robot *line follower* ini dipasang empat motor DC, yang dipasang pada tiap samping robot yang mana saat maju keempat motor akan bergarak bersamaan tetapi memiliki putaran yang berbeda sehingga menghasilkan putaran maju, semua komponen sperti mikro, driver motor,komparator dan baterai smuanya terletak di atas robot, sedangkan sensor, motor DC dan *Gearbox* diletakkan pada bawah robot. Tinggi robot 10cm, panjang 13cm dan lebar 13cm.

3.2.2. Perancangan Sistem Minimum

Mikrokontroler yang digunakanadalah mikrokontroler jenis *basic stamp* BS2P40 dengan dilengkapi 40 *port* yang bisa digunakan sebagi *input* ataupun *output*. Pemilihan mikrokontroler jenis ini didasari kemampuannya yang cukup handal, permrograman yang tidak terlalu sulit dan dalam

pengaturan *pulse width modulation* (PWM) dapat dengan mudah di program.



Gambar 3.2. *Rangkanan Mikrokontroler Basic stamp 2P40*

Tabel 3.1. *Konfigurasi port yang digunakan pada mikrokontroler basic stamp*

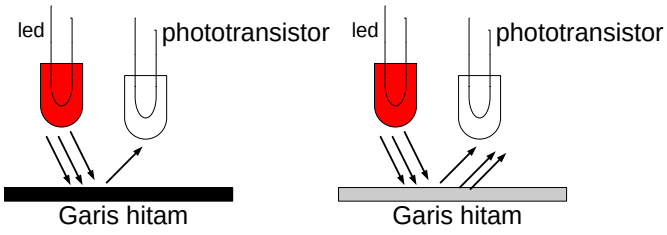
Nama Pin	No. Pin	Fungsi	Keterangan
MAIN I/O	0	*input*	Sensor1
MAIN I/O	1	*input*	Sensor2
MAIN I/O	2	*input*	Sensor3
MAIN I/O	3	*input*	Sensor4
MAIN I/O	4	*input*	Sensor5
MAIN I/O	5	*input*	Sensor6
MAIN I/O	6	*input*	Sensor7
MAIN I/O	7	*input*	Sensor8
MAIN I/O	12	*input*	Sensor9
MAIN I/O	9	*input*	Sensor10
MAIN I/O	10	*input*	Sensor11
MAIN I/O	11	*input*	Sensor12

AUX I/O	0	*Output*	Driver motor2 (A2 EN)
AUX I/O	1	*Output*	Driver motor2 (A2 +)
AUX I/O	2	*Output*	Driver motor2 (A2 -)
AUX I/O	3	*Output*	Driver motor2 (B2 EN)
AUX I/O	4	*Output*	Driver motor2 (B2 +)
AUX I/O	5	*Output*	Driver motor2 (B2 -)
AUX I/O	6	*Output*	Driver motor1 (A1 EN)
AUX I/O	7	*Output*	Driver motor1 (A1 +)
AUX I/O	8	*Output*	Driver motor1 (A1 -)
AUX I/O	9	*Output*	Driver motor1 (B1 EN)
AUX I/O	10	*Output*	Driver motor1 (B1 +)
AUX I/O	11	*Output*	Driver motor1 (B1 -)

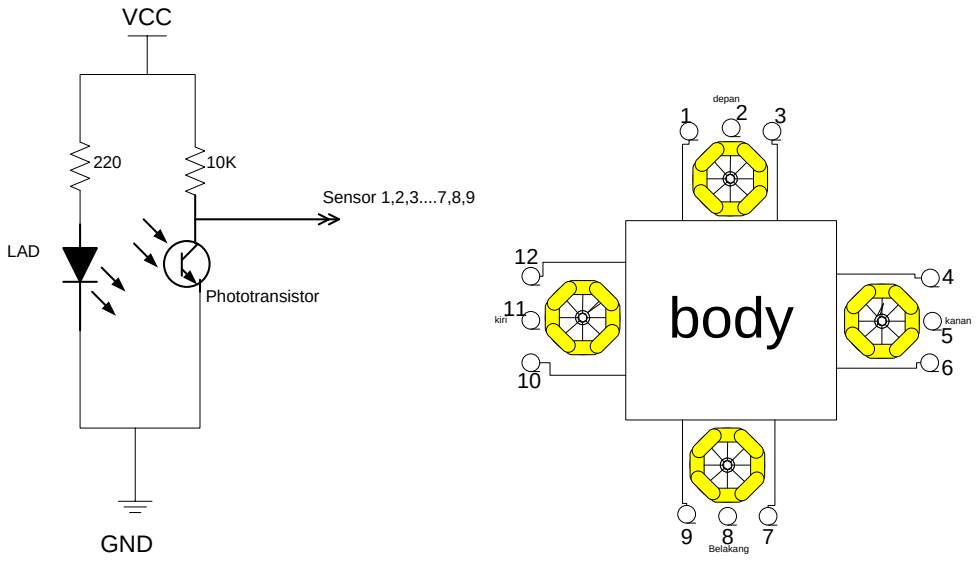
3.2.3. Perancangan sensor garis

Sensor garis berfungsi untuk mendeteksi warna yang berada di bagian bawah robot *line follower* dengan maksud agar robot dapat bergerak sesuai dengan jalur yang telah di tentuin agar robot tidak keluar dari jalur.

Pada sensor garis menggunakan *lad* sebagai pengirim sedangkan penerimanya menggunakan *phototransistor* sebagai penerima



Gambar 3.3. *Prinsip kerja sensor garis*



Gambar 3.4. *A.Rangkaian sensor B. posisi tiap-tiap sensor dan roda*

Arus yang boleh mengalir di dalam lad super brigh yaitu minimal 10mA dan maximal 30mA. Untuk mengetahui resistor yang digunakan pada rangkaian maka menggunakan rumus persamaan 3.1

$$R = \frac{V - Vir}{Imax} \quad \text{..... 3.1}$$

$$R = \frac{V - Vir}{Imin} \quad \text{..... 3.2}$$

Ket :

R = nilai hambatan

V = tegangan sumber

Vir =tegangan pada led

Imax = arus maximal pada led

Imin = arus minimal pada led

Jadi untuk mengetahui batas maximal dan minimal penggunaan resistor maka masukan nilai pada rumus.

R= belum di ketahui

V= 5V

Vir= 2,2V

Imax= 30mA = 0,03A

Imin= 10mA = 0,01A

$$R = \frac{5-2,2}{0,01} = \frac{2,8}{0,01} = 280\Omega$$

$$R = \frac{5-2,2}{0,03} = \frac{2,8}{0,03} = 93,3\Omega$$

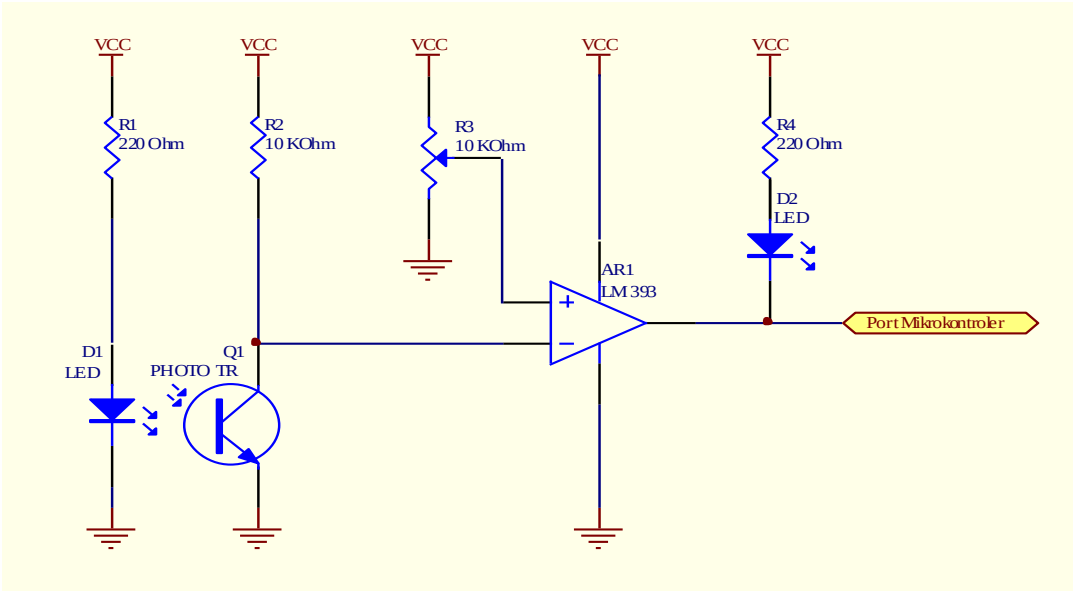
Maka batas minimal resistor yang boleh digunakan pada lad super brigh yaitu 93,3Ω sedangkan batas maksimum digunakan 280Ω jadi diperbolehkan menggunakan resistor diantara 93,3Ω sampai 280Ω. Jika menggunakan resistor kekecilan tahanannya maka akan berakibat rusaknya lad super brigh karena arusnya yng di hasilkan akan berlebilah dari yang seharusnya dan jika menggunakan resistor melebihi 280Ω maka rangkaian tidak akan berfungsi karena tidak ada arus yang melewati rangkaian lad super brigh.

Phototransistor adalah merupakan sebuah transistor yang akan saturasi pada saat menerima sinar *lad super brigh* dan *cut off* pada saat tidak ada sinar *lad super brigh*. *led module* adalah sebuah rangkaian yang terdiri dari sebuah *phototransistor* dan filter yang terbentuk dalam satu modul di mana kolektor dari *phototransistor* merupakan *output* dari modul ini. Pada saat *phototransistor cut off* maka tidak terjadi aliran arus dari kolektor menuju ke emitter sehingga kolektor yang merupakan output dari *led module* akan berkondisi *high*. Apabila *phototransistor* saturasi maka arus mengalir dari kolektor ke emitter dan output dari *led module*akan berkondisi low.

Transmisi data dilakukan dengan menggunakan prinsip aktif dan non-aktifnya LED Super brigh sebagai kondisi logic 0 dan logic 1. Pada rangkaian ini menggunakan 12 pasang sensor yang didesain sesuai pergerkan robot. Dan data dari sensor akan di kirim ke komparator selanjutnya data digital yang dari komparator dikirim ke mikrokontroler agar dapat mengatur pergerakan robot sesuai dengan jalur yang ada

Sebagai pengatur sensor agar pembacaan sensor menghasilkan nilai yang sama maka menggunakan komparator IC LM393. Pembacaan sensor bila terkena garis akan menghasilkan nilai berlogika low (0), sedangkan bila tidak berada dijalur akan menghasilkan logika high (1).

Pada gambar 3.5. merupakan skematik rangkaian sensor garis menggunakan komparator menggunakan IC LM393.



Gamabar 3.5. Rangkaian sensor garis

3.2.4. Motor Driver

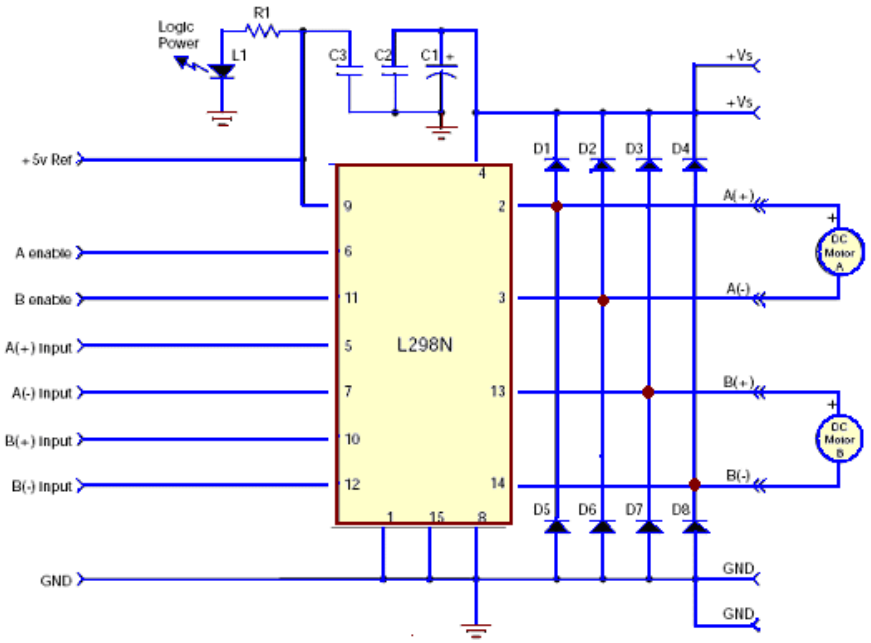
L298 bisa men-*drive* dua buah motor dc sampai tegangan maksimal 46 V_{DC} dan arus sebesar 2A untuk tiap kanal, *supply* tegangan yang diberikan ke motor dc sebesar 14,8 V_{DC} yang berasal dari baterai *litium polimer (lipo)* 2200mAh, sedangkan untuk tegangan *logic* pada *driver* motor diberikan tegangan sebesar 5 V_{DC} yang berasal dari mikrokontroler. Penggunaan dioda pada perancangan *driver* motor ini ditujukan agar *driver* motor dapat menahan arus balik yang datang dari motor *dc*.

Satu motor dapat dikontrol dengan 3 pin yaitu A*enable*, A+ dan A-. Dan dikarenakan robot ini menggunaka empat buah motor dc maka dibutuhkan dua buah driver motor yang tiap driver motor mengatur dua buah motor dc. Berikut ini adalah tabel kebenaran untuk salah satu motor.

***Tabel 3.2**. Tabel kebenaran untuk salah satu motor*

A_{EN}	A+	A-	Reaksi Motor
L	L	L	Motor *off*
L	L	H	Motor *off*
L	H	L	Motor *off*
L	H	H	Motor *off*
H	L	L	*Stop*
H	H	L	Berputar searah jarum jam
H	L	H	Berputar berlawanan arah jarum jam
H	H	H	*Stop*

Ket: H = *High* = 1 dan L = *Low* = 0



***Gambar 3.6.** Rangkaian driver motor L298*

Pengontrolan empat buah motor kiri, kanan, depan dan belakang dibutuhkan 12 *port output* pada mikrokontroler, yaitu untuk motor kiri masing-masing A_{EN1}, A+1, A-1, motor kanan B_{EN1}, B+1, B-1, dan untuk mengontrol motor depan A_{EN2}, A+2 , A-2, motor belakang B_{EN2}, B+2, B-1 seperti yang terlihat pada tabel berikut.

Tabel 3.3. *Tabel kebenaran untuk empat motor*

Gerak	**A_{EN1}**	A+1	**A-1**	**B_{EN1}**	**B+1**	**B-1**	**A_{EN2}**	A+2	**A-2**	**B_{EN2}**	**B+2**	**B-2**
Maju	H	H	L	H	H	L	L	L	L	L	L	L
Belok kanan	L	L	L	L	L	L	H	H	L	H	H	L
Belok kiri	L	L	L	L	L	L	H	L	H	H	L	H
Mundur	H	L	H	H	L	H	L	L	L	L	L	L
stop	L	L	L	L	L	L	L	L	L	L	L	L

Ket: H = *High* = 1 dan L = *Low* = 0

1. Pengaturan gerak robot dengan detode PWM

Pada dasarnya PWM yang dihasilkan mikrokontroler akan berubah menjadi tegangan analog yang akan diterima oleh bagian motor *driver*. Jika PWM diberi kondisi *High*, maka tegangan keluaran mendekati 14,8 Volt, dan jika PWM diberi kondisi *Low*, maka tegangan keluaran mendekati 0 Volt. Diantara kondisi itu maka nilai tegangan keluaran dapat dihitung, yaitu :

(*Duty* ÷ 255) x 14,8 V.

Sebagai contoh, ketika *Duty* adalah 100, maka :

(100 ÷ 255) x 14,8 V = 5,8 V, tegangan keluaran PWM sebesar 5,8 V.*Duty cycle* sebagai pengaturan kecepatan, mulai (0-255)

Hal-hal yang harus diperhatikan dalam pemrogramannya adalah nama pin tujuan, lamanya PWM diberikan dan kecepatannya *(duty cycle).*Perintah yang digunakan dalam pemrograman bahasa basic untuk mikrokontroler BS2P40 adalah sebagai berikut :

Syntax: PWM *Pin, Duty, Duration*

Keterangan :

1. *Pin* adalah variable, dengan rentang nomor pin dari (0-11).
2. *Duty* adalah variabel, (0-255), menentukan besarnya tegangan analog, yang oleh motor driver akan diterima untuk mengatur kecepatan putarannya.
3. *Duration* adalah varibel (0-255) untuk durasi PWM tersebut dikerjakan.

Pada tabel 3.4, tabel 3.5, tabel 3.6, dan tabel 3.7 adalah hasil uji coba pada pengaturan gerak roda menggunakan PWM :

Tabel.3.4. *pengaturan gerak motor roda kiri dan kanan bergerak maju*

Roda	**Enable**	**Duty**	**Durasi**	**gerak**	**keterangan**
Kiri	Aen2	100	60	lambat	Robot bergerak ke arah kiri dengan cepat
Kanan	Ben2	110	60	cepat	
Kiri	Aen2	105	60	lambat	Robot bergerak ke arah kanan mulai pelan
Kanan	Ben2	90	60	cepat	
Kiri	Aen2	105	60	cepat	Robot bergerak ke arah kiri
Kanan	Ben2	95	60	lambat	
Kiri	Aen2	110	60	lambat	Robot bergerak lurus, motor kiri dan kanan bergerak sama cepatnya
Kanan	Ben2	90	60	lambat	

Tabel 3.5. *pengaturan gerak motor roda kiri dan kanan bergerak mundur*

Roda	**Enable**	**Duty**	**Durasi**	**gerak**	**keterangan**
Kiri	Aen2	110	60	cepat	Robot bergerak ke arah kanan dengan cepat
Kanan	Ben2	110	60	lambat	
Kiri	Aen2	95	60	lambat	Robot bergerak ke arah kiri mulai pelan
Kanan	Ben2	115	60	cepat	
Kiri	Aen2	95	60	lambat	Robot bergerak ke arah kiri
Kanan	Ben2	115	60	cepat	
Kiri	Aen2	93	60	lambat	Robot bergerak lurus, motor kiri dan kanan bergerak sama cepatnya
Kanan	Ben2	108	60	lambat	

Tabel 3.6. *pengaturan gerak motor roda depan dan belakang bergerak kanan*

Roda	**Enable**	**Duty**	**Durasi**	**gerak**	**keterangan**
depan	Ben1	120	60	lambat	Robot bergerak ke arah depan cepat
belakang	Aen1	120	60	cepat	
depan	Ben1	110	60	lambat	Robot bergerak ke arah depan dengan pelan
belakang	Aen1	115	60	cepat	
depan	Ben1	109	60	cepat	Robot bergerak ke arah

belakang	Aen1	120	60	lambat	belakang
depan	Ben1	109	60	lambat	Robot bergerak lurus, motor depan dan belakang bergerak sama cepatnya
belakang	Aen1	102	60	lambat	

Tabel 3.7. *pengaturan gerak motor roda depan dan belakang bergerak kiri*

Roda	**Enable**	**Duty**	**Durasi**	**gerak**	**keterangan**
depan	Ben1	100	60	cepat	Robot bergerak ke arah depan dengan cepat
belakang	Aen1	115	60	lambat	
depan	Ben1	110	60	lambat	Robot bergerak ke arah belakang mulai pelan
belakang	Aen1	105	60	cepat	
depan	Ben1	120	60	cepat	Robot bergerak ke arah belakang
belakang	Aen1	100	60	lambat	
depan	Ben1	100	60	lambat	Robot bergerak lurus, motor depan dan belakang bergerak sama cepatnya
belakang	Aen1	102	60	lambat	

Pada hasil percobaan pengaturan PWM di atas maka di dapat hasil yang cocok untuk digunakan pada pemrograman pada PWM. Berikut hasil *duty* dan durasi yang digunakan pada pemrograman motor DC

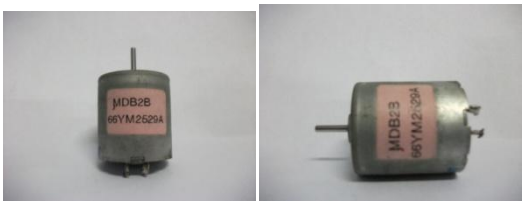
Tabel 3.8. *duty dan durasi yang digunakan pada PWM*

Roda	**Enable**	**Duty**	**Durasi**	**gerak**	**keterangan**	**Arah gerak robot**
Kiri	Aen2	108	60	lambat	Robot bergerak lurus, motor kiri dan kanan bergerak sama cepatnya	Bergerak maju
Kanan	Ben2	92	60	lambat		
Kiri	Aen2	93	60	lambat	Robot bergerak lurus, motor kiri dan kanan	Bergerak mundur
Kanan	Ben2	108	60	lambat		

					bergerak sama cepatnya	
depan	Ben1	109	60	lambat	Robot bergerak lurus, motor depan dan belakang bergerak sama cepatnya	Bergerak Ke kanan
belakang	Aen1	102	60	lambat		
depan	Ben1	109	60	lambat	Robot bergerak lurus, motor depan dan belakang bergerak sama cepatnya	Bergerak ke kiri
belakang	Aen1	112	60	lambat		

3.2.5. Motor DC

Motor DC digunakan sebagai penggerak bebuah robot yang di kendalikan oleh driver motor yang berfungsi sebagai sakelar untuk mengatur gerakan motor yang dikontrol oleh mikrokontroler. Pada robot ini menggunakan empat buah motor dc yang menggunakan motor dc 12 Volt. Menggunakan motor dc 12 volt dikarenakan berat beban robot dan memiliki putaran yang kuat sehingga diharapkan motor dc mampu berputar dengan beban yang cukup besar



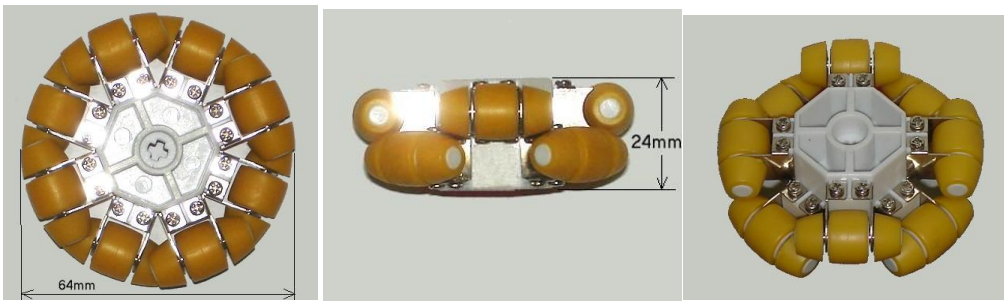
Gambar 3.7. *Motor DC 12 Volt*

Untuk membantu pergerakan gerak robot agar dapat membawa beban maka digunakan *gearbox*. *Gearbox* yang digunakan yang umumnya terdapat di pasaran ataupun bisa juga di ambil *gearbox* mobil-mobilan yang biasa

dijual di pasaran. *Gearbox* yang digunakan mempunyai skala 1:78 yang artinya 1 putaran pada roda sebanding dengan 78 kali putaran pada motor dc.

3.2.6. Roda Ohmniwheel

Holonomic roda adalah roda dengan 2 derajat kebebasan, omniwheel tidak seperti roda biasa tetapi sebuah roda khusus. Omniwheel terdiri dari roda inti besar dan sepanjang periferal ada terdapat banyak roda kecil tambahan yang mempunyai poros tegak lurus pada roda inti. [5]



Gambar 3.8. *Roda Ohmniwheel*

3.2.7. Sumber Tegangan

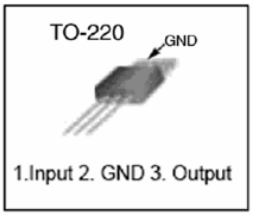
A. Baterai

Pada robot ini menggunakan dua buah baterai lipo yang memiliki tegangan 14,8 Volt 2200mAh yang ada di pasaran. Sumber tegangan ini digunakan untuk memberi sumber tegangan pada semua komponen yang digunakan. [13]

B. Reglator LM7805

Dan untuk menstabilkan tegangan pada mikro maka digunakan regulator. Regulator tegangan yang digunakan dalam hal ini adalah regulator tegangan 5 Volt dengan tegangan keluaran yang dihasilkannya ±5 Volt.IC regulator yang digunakan adalah LM7805 seperti ditunjukan pada gambar 2.20. Yang mana keluaran tegangan IC ini adalah ± 5 Volt, dengan tegangan masukan antara 7,2 V sampai 25 V.

Pada regulator ini sudah terdapat pada besicstamp 2P40,sehingga tidak pelu membuat rangkaian regulator lagi.



Gambar 3.9. *LM7805*

3.3. Perancangan *Software*

Perangkat lunak merupakan faktor penting dalam tahap perancangan robot.Perangkat lunak ini merupakan algoritma gerak dan tugas robot dalam bentuk listing program yang ditanamkan kedalam mikrokontroler.Program dapat bermacam-macam bentuk versi dan bahasa pemrogramannya, sesuai dengan spesifikasi dari mikrokontroler yang digunakan.

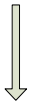
Sistem perangkat lunak merupakan suatu perangkat yang dibuat untuk mengendalikan sistem kerja dari masing-masing komponen. Sistem ini dikendalikan oleh mikrokontroler yang juga mengendalikan perangkat keras yang juga mengendalikan *input* dan *output* data serta pertukaran informasi.

Sebelum pembuatan program terlebih dahulu dibuat algoritma programnya dalam bentuk *flowchart*. Ini bertujuan agar pada saat pemrograman jika terjadi *error* maka dapat dengan mudah mengetahui letak kesalahan pemrogramannya

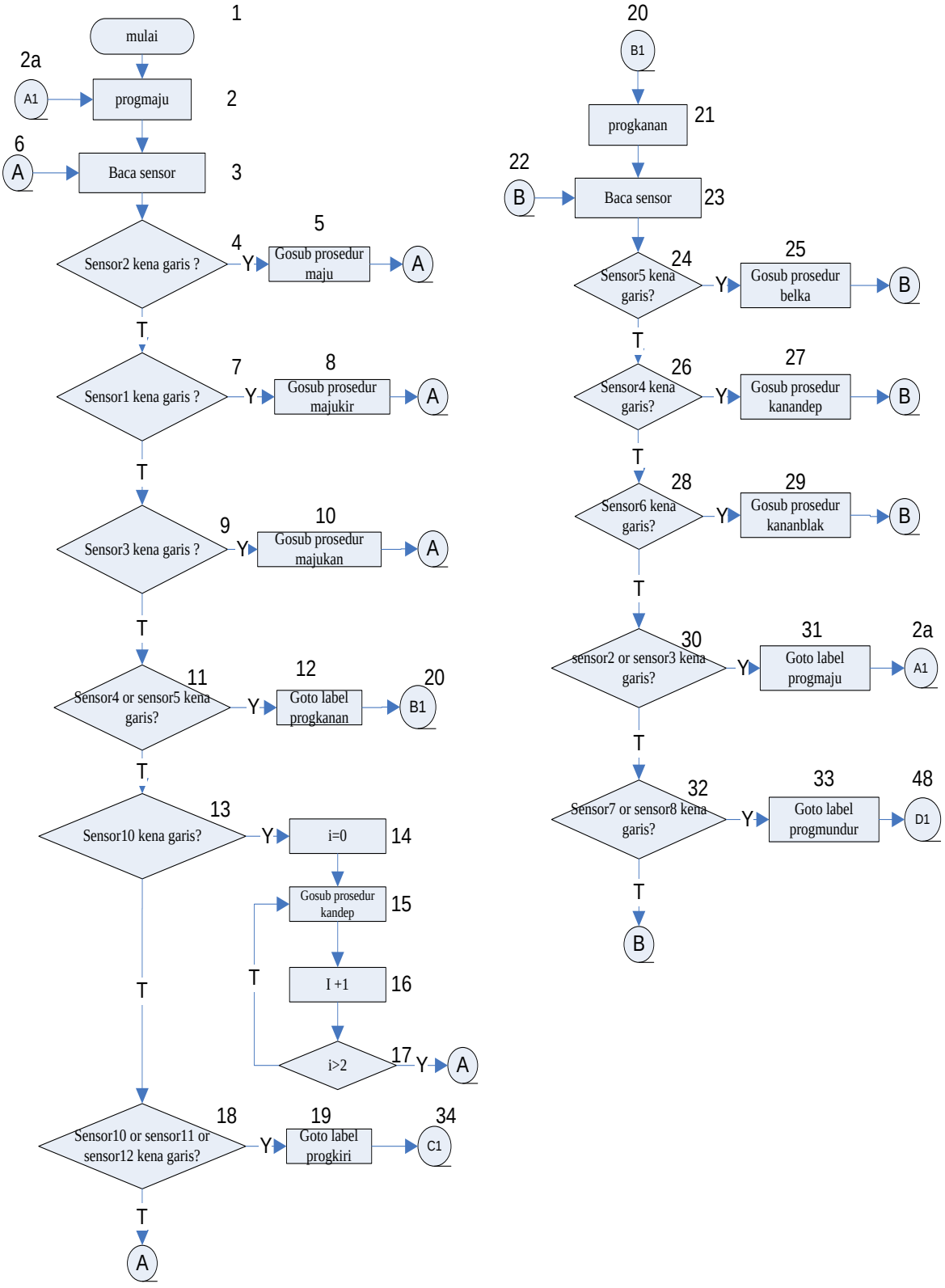
3.3.1. Perancangan algoritma sistem

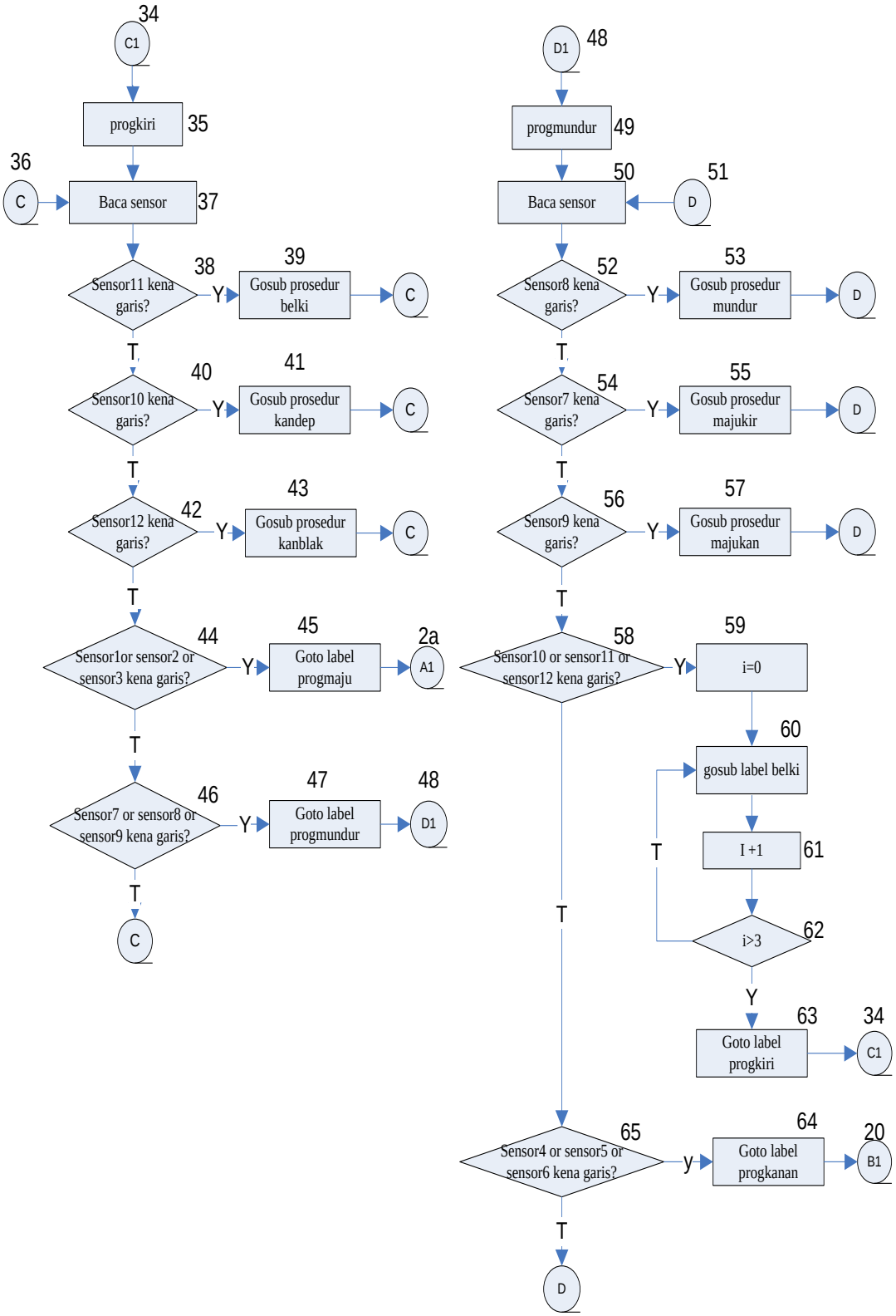
Tabel 3.9. *pola gerak utama robot dan kondisi sensor*

Pola gerak utama	Terdeteksi sensor	Arah gerak
↑	Sensor2	maju
→	Sensor5	kanan

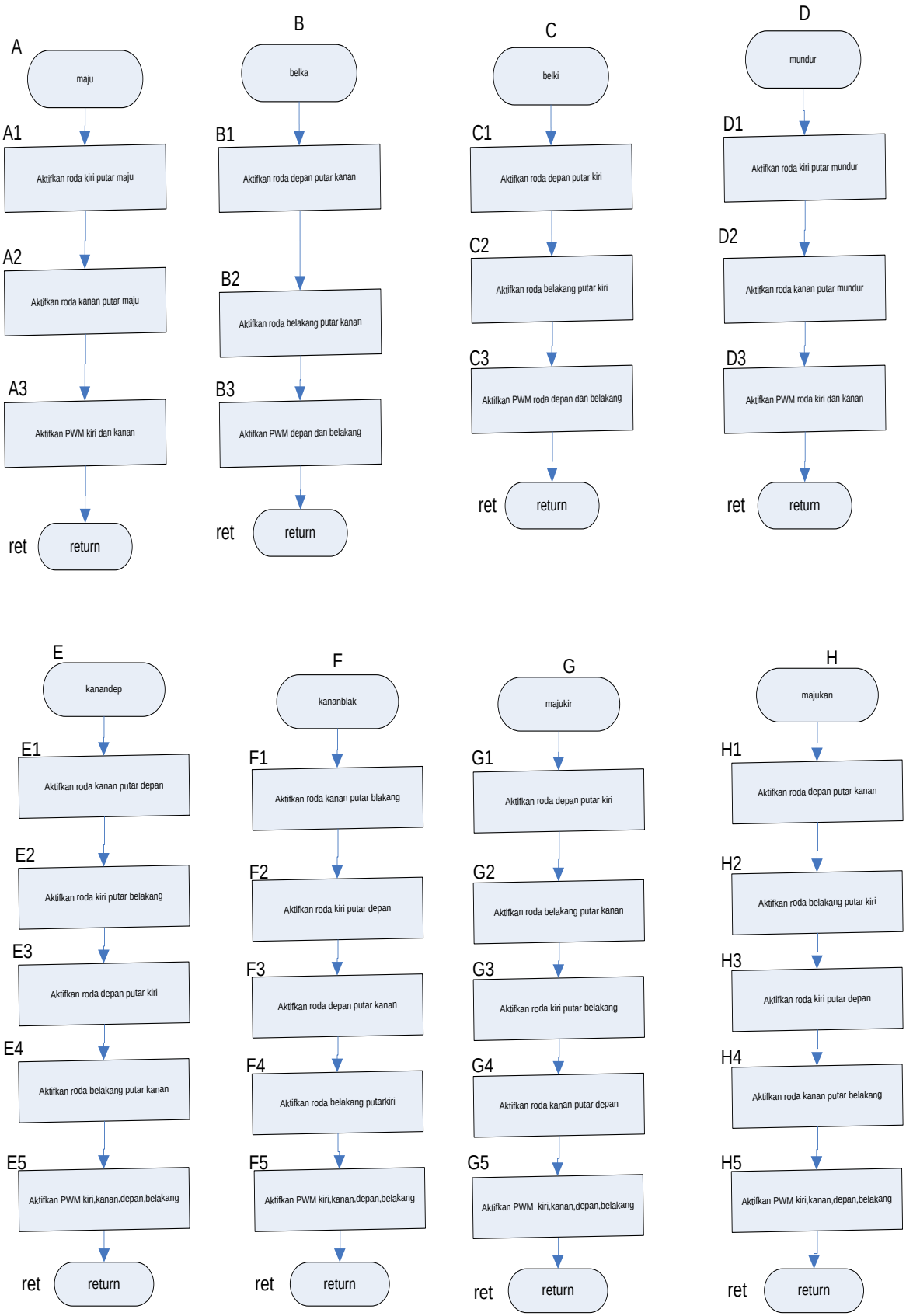
	Sensor11	kiri
	Sensor2	Maju 45
	Sensor8	mundur

Dari pola arah gerak robot pada tabel 3.9 maka didapat algoritma flowchart sistem pada pola gerak utama. Dapat di lihat pada gambar 3.10


1
mulai
2a
A1
progmaju
2
6
A
Baca sensor
3
Sensor2 kena garis ?
4
Y
5
Gosub prosedur maju
A
T
Sensor1 kena garis ?
7
Y
8
Gosub prosedur majukir
A
T
Sensor3 kena garis ?
9
Y
10
Gosub prosedur majukan
A
T
Sensor4 or sensor5 kena garis?
11
Y
12
Goto label progkanan
20
B1
T
Sensor10 kena garis?
13
Y
i=0
14
Gosub prosedur kandep
15
T
I +1
16
i>2
17
Y
A
T
Sensor10 or sensor11 or sensor12 kena garis?
18
Y
19
Goto label progkiri
34
C1
T
A
20
B1
progkanan
21
22
B
Baca sensor
23
Sensor5 kena garis?
24
Y
25
Gosub prosedur belka
B
T
Sensor4 kena garis?
26
Y
27
Gosub prosedur kanandep
B
T
Sensor6 kena garis?
28
Y
29
Gosub prosedur kananblak
B
T
sensor2 or sensor3 kena garis?
30
Y
31
Goto label progmaju
2a
A1
T
Sensor7 or sensor8 kena garis?
32
Y
33
Goto label progmundur
48
D1
T
B



Gambar 3.10. *flowchart sistem*



Gambar 3.11. Prosedur pemutaran arah gerak roda

Penjelasan diagram alir flowchart system pada tabel 3.10 :

Tabel 3.10. *penjelasan alir flowchart system*

nomor	**Keterangan**
1	Awal pemrograman.
2	Label maju sebagai awal program maju.
2a	Sebagai inisialisasi lompatan program dari label program lain.
3	Baca kondisi sensor.
4	Apakah sensor2 deteksi garis?jika ya maka kerjakan program pada nomor 5. jika tidak, maka lakukan yang ada di nomor 7.
5	Jika benar maka panggil prosedur maju untuk gerak robot maju,jika telah maju maka lompat pada no 6 sebagai lompatan untuk baca sensor.
6	Sebagai inisialisasi lompatan menuju baca kondisi sensor.
7	Apakah sensor1 deteksi garis?jika ya,maka kerjakan program pada nomor 8. Jika tidak maka kerjakan program pada nomor 8.
8	Jika benar maka panggil prosedur majukir untuk gerak robot memutar ke arah kiri sebagai koreksi posisi jika robot akan keluar jalur,dan lompat ke baca sensor pada nomor 3.
9	Apakah sensor3deteksi garis?jika ya, maka kerjakan program pada nomor 10. Jika tidak maka kerjakan program di nomor 11.
10	Jika ya maka panggil prosedur majukan untuk memutar arak ke kanan sebagai koreksi posisi ke arak kanan jika robot akan keluar jalur. Dan lompat pada nomor 3 untuk pengecekan sensor.
11	Apakah sensor4 atau sensor5 terdeteksi garis? Jika ya, maka kerjakan pada program no 12. Jika tidak maka cek lagi pada program nomor 13
12	Jika iya maka lompat ke label progkanan untuk melakukan program arah gerak kanan pada nomor 20 sebagai label progkanan.
13	Apakah sensor10 deteksi garis?jika ya, maka lakukan pada nomor 14. Dan jika tidak maka lakukan program pada nomor 18
14	Masukan nilai 0 pada variabel i
15	Panggil prosedur kandep sebagai gerak putar arah
16	Nilai variabel i di tambah 1

17	Apakah variabel i>2? Jika ya maka lompat lagi ke nomor 6 sebagai baca sensor. Jika i<2 maka panggil lagi prosedur kandep pada nomor 15,ulangi terus sampai i>2.
18	Apakah sensor10 atau sensor11 atau sensor12 deteksi garis?jika iya,maka lakukan program di nomor19. Jika tidak maka kembali cek sensor pada nomor 3
19	Lompat pada label progkiri pada nomor 34.
20	Inisialisasi lompatan ke label progkanan sebagai awal label program untuk pergerakan kanan.
21	Label progkanan sebagai awal label program untuk pergerakan kanan
22	Inisialisasi untuk menandakan lompatan ke baca sensor di program kanan.
23	Pembacaan kondisi sensor.
24	Apakah sensor5 deteksi garis?jika ya,maka lakukan perinta pada nomor 25. Jika tidak,maka lakukan pada nomor 26.
25	Jika ya,maka panggil prosedur putar motor arah kanan untuk pergerakan motor gerak kanan.jika sudah selesai pemanggilan maka lompat pada nomor 23.
26	Apakah sensor4 deteksi garis?jika iya,maka lakukan pada program di nomor 27. Jika tidak maka cek lagi pada nomor 28.
27	Jika ya,maka panggil prosedur kandep,untuk memutar arah gerak ke depan sebagai koreksi posisi jika robot akan keluar garis.setelah perika posisi selesai maka lompat pada baca kondisi sensor lagi.
28	Apakah sensor6 deteksi garis?jika ya,maka kerjakan pada nomor29.jika tidak,maka lakukan pada program di nomor 30.
29	Jika ya,maka panggil prosedur kanblak sebagai putar arah robot untuk koreksi posisi jika robot keluar garis.dan lompat pada pembacaan sensor.
30	Apakah sensor2 atau sensor3 deteksi garis?jika ya,maka lakukan program nomor 31. Jika tidak,maka lakukan proram di nomor32.
31	Jika ya,maka lompat ke label progmaju untuk menjalankan program gerak maju yang ada pada nomor 2a.
32	Apakah sensor7 atau sensor8 deteksi garis?jika ya,maka lakukan pada

	nomor 33. Jika tidak ,maka lompat pada nomor 23 untuk pembacaan kondisi sensor lagi.
33	Jika ya,lompat pada label progmundur untuk melakukan arah gerak mundur yang ada pada program nomor 48.
34	Inisialisasi untuk lompatan dari program sebelumya menuju progkiri untuk melakukan arah gerak kiri.
35	Label progkiri sebagai awal program arah gerak kiri.
36	Sebagai inisialisasi lompatan dari program sebslumnya menuju baca kondisi sensor.
37	Cek kondisi sensor untuk arah gerak kiri.
38	Apakah sensor11 deteksi garis?jika ya,maka lakukan program yang ada di nomor 39. Jika tidak,maka lakukan program di nomor 40.
39	Jika ya, maka panggil prosedur belki untuk arah gerak kiri,dan lompat pada pembacaan sensor pada nomor 37.
40	Apakah sensor10 kena garis?jika ya,maka lakukan program di nomor 41. Jika tidak,maka lakukan perintah di nomor 42.
41	Jika ya,maka panggil prosedur kandep untuk memutar arah ke depan sebagai koreksi posisi jika robot keluar jalur.dan lompat ke pembacaan sensor.
42	Apakah sensor12 deteksi garis?jika ya,maka lakukan perintah di nomor 43. Jika tidak maka lakukan program di nomor 44.
43	Jika ya,maka panggil prosedur kanblak untuk memutar arah belakang sebagai koreksi posisi jika robot mulai keluar garis.dan lompat ke pembacaan kondisi sensor.
44	Apakah sensor1 atau sensor2 atau sensor3 deteksi garis?jika ya,maka lakukan perintah di nomor 45.jika tidak maka lakukan perintah di nomor 45.
45	Jika ya,maka lompat ke label progmaju untuk melakukan pergerakan maju yang ada pada label di nomor 2a.
46	Apakah sensor4 atau sensor5 atau sensor6 deteksi garis?jika ya maka lakukan perintah pada nomor 47. Jika tidak maka lompat pada pembacaan sensor di nomor 37

47	Jika ya,maka lompat ke label progmundur untuk melakukan gerakan mundur yang ada pada nomor 48
48	Inisialisasi label untuk lompatan dari program lain menuju label progmundur.
49	Sebagai awal program untuk program arah gerak mundur.
50	Baca kondisi sensor utuk gerak mundur.
51	Inisialisasi untuk lompatan dari program untuk menuju pembacaan kondisi sensor
52	Apakah sensor8 deteksi garis?jika ya,maka lakukan perintah pada nomor 53. Jika tidak maka lakukan perintah pada nomor 54.
53	Jika ya,maka panggil prosedur mundur untuk arah gerak mundur dan lompat ke pembacaan kondisi sensor pada nomor 50.
54	Apakah sensor7 deteksi garis? Jika ya,maka lakukan perintah pada nomor 55.jika tidak maka lakukan perintah pada 56.
55	Jika ya,maka panggil prosedur mundur untuk arah gerak mundur dan lompat ke pembacaan sensor pada nomor 50.
56	Apakah sensor9 deteksi garis?jika ya,maka lakukan perintah pada nomor 57. Jika tidak maka lakukan perintah pada nomor 58.
57	Jika ya,maka panggil prosedur majukan untuk memutar arah robot ke arah kanan dikarenakan robot mulai keluar jalur sehingga pergerakan ini digunakan untuk koreksi posisi,dan lompat pada nomor 50 untuk pengecekan sensor.
58	Apakah sensor10 atau sensor11 atau sensor12 deteksi garis?jika ya,maka lakukan perintah selanjutnya.jika tidak maka lakukan perintah pada nomor 65.
59	Masukan variabel i=0.
60	Panggil prosedur belki untuk belok arah kiri.
61	Variabel i+1
62	Apakah nlai i>3?jika tidak maka lompat lagi ke pemanggilan pada nomor 60. Jika ya maka lakukan perintah selanjutnya pada nomor 63.
63	Jika sudah terpenuhi nilai i>3 maka lompat ke label progkiri untuk melakukan pergerakan arah kiri yang berada pada nomor 34.

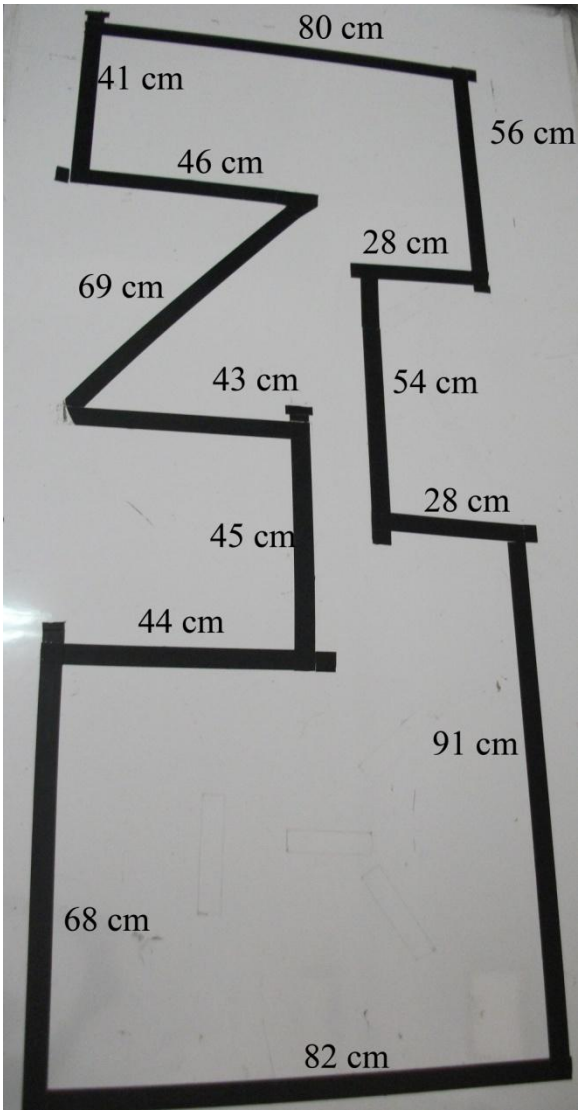
64	Jika ya,lompat ke label progkiri untuk menjalankan program dengan gerak kiri yang berada pada label di nomor 34.
65	Apakah sensor4 atau sensor5 atau sensor6 deteksi garis?jika ya maka lakukan perintah pada no 64.jika tidak,maka lompat pada pembacaan sensor pada nomor 50 untuk pengecekan sensor

Tabel 3.11. *penjelasan flowchart prosedur putaran roda*

Label	**Keterangan**
A	Awal prosedur maju,untuk menghasilkan arah gerak maju
A1	Aktifkan roda kiri putar kedepan.
A2	Aktifkan roda kanan putar kedepan.
A3	Aktifkan PWM kiri dan PWM kanan untuk pengaturan kecepatan roda.
ret	Kembali ke pemanggilan pada program utama.
B	Awal prosedur gerak kanan,untuk menghasilkan arah gerak kanan.
B1	Aktifkan roda depan putar kanan.
B2	Aktifkan roda belakang putar kanan.
B3	Aktifkan PWM depan dan PWM belakang untuk pengaturan kecepatan roda
ret	Kembali ke pemanggilan pada program utama.
C	Awal prosedur gerak kiri,untuk menghasilkan arah gerak kiri.
C1	Aktifkan roda depan putar kiri.
C2	Aktifkan roda belakang putar kiri.
C3	Aktifkan PWM depan dan PWM belakang untuk pengaturan kecepatan roda
ret	Kembali ke pemanggilan pada program utama.
D	Awal prosedur mundur,untuk menghasilkan arah gerak mundur
D1	Aktifkan roda kiri putar ke belakang.
D2	Aktifkan roda kanan putar ke belakang.
D3	Aktifkan PWM kiri dan PWM kanan untuk pengaturan kecepatan roda.
ret	Kembali ke pemanggilan pada program utama.
E	Awal prosedur putar arah untuk putar arah ke depan

E1	Aktifkan roda kanan putar arah depan.
E2	Aktifkan roda kiri putar belakang.
E3	Aktifkan roda depan putar kiri.
E4	Aktifkan roda belakang putar kanan.
E5	Aktifkan PWM roda kiri,PWM roda kanan,PWM roda depan, PWM roda belakang untuk mengatur kecepatan masing-masing roda.
ret	Kembali ke pemanggilan pada program utama.
F	Awal prosedur putar arah untuk putar arah ke belakang
F1	Aktifkan roda kanan putar arah belakang.
F2	Aktifkan roda kiri putar depan.
F3	Aktifkan roda depan putar kanan.
F4	Aktifkan roda belakang putar kiri.
F5	Aktifkan PWM roda kiri,PWM roda kanan,PWM roda depan, PWM roda belakang untuk mengatur kecepatan masing-masing roda.
ret	Kembali ke pemanggilan pada program utama.
G	Awal prosedur untuk pergerakan putar arah ke kiri.
G1	Aktifkan roda depan putar kiri.
G2	Aktifkan roda belakang putar kanan.
G3	Aktifkan roda kiri putar belakang.
G4	Aktifkan roda kanan putar depan
G5	Aktifkan PWM roda kiri,PWM roda kanan,PWM roda depan, PWM roda belakang untuk mengatur kecepatan masing-masing roda.
ret	Kembali ke pemanggilan pada program utama.
H	Awal prosedur untuk pergerakan putar arah ke kanan.
H1	Aktifkan roda depan putar kanan.
H2	Aktifkan roda belakang putar kiri.
H3	Aktifkan roda kiri putar depan.
H4	Aktifkan roda kanan putar belakang
H5	Aktifkan PWM roda kiri,PWM roda kanan,PWM roda depan, PWM roda belakang untuk mengatur kecepatan masing-masing roda.
ret	Kembali ke pemanggilan pada program utama.

3.4 Perancangan Maket Lapangan



Gambar 3.12. Jalur lintasan robot

Jalur yang digunakan memiliki lebar garis 2cm,yang dasar lintasan berwarna putih dan jalurnya berwarna hitam serta bahan yang digunakan sebagai dasar jalur yaitu berbahan melamin