BAB III
PERANCANGAN

3.1 Perancangan Perangkat Keras (*Hardware*)

3.1.1 Mikrokontroler BS2p40

Kemudahan dalam pengembangan program karena menggunakan bahasa tingkat tinggi menjadi faktor utama dalam pemilihan mikrokontroler BS2p40 ini, dan fasilitas yang ada di mikrokontroler ini hampir mendukung semua sensor yang digunakan.

Tabel 3.1. Penggunaan Pin I/O BS2p40

Pin	Tipe pin	Fungsi
5	P0	PWM kompas
6	P1	Motorkanan_aen
7	P2	SDA kompas
8	P3	Motorkanan_aplus
9	P4	SCL kompas
10	P5	Motorkiri_aplus
11	P6	Komunikasi ke Atmega 8
12	P7	Motorkiri_aen
13	P8	Jalur komunikasi ke Atmega8 (data2)
14	P9	Motorgripper_aen
15	P10	Jalur komunikasi ke Atmega8 (data1)
16	P11	Motorgripper_aplus
17	P12	LDR bawah
18	P13	Servocapit
19	P14	Tombol start
20	P15	Servonaikturun
21	X0	Sensor ultrasonik depankanan
22	X1	GPcapit
23	X2	Sensor ultrasonik depankiri
24	X3	GPmeja
25	X4	Sensor ultrasonik kanandepan
26	X5	GPdepantengah
27	X6	Sensor ultrasonik kananbelakang
28	X7	Cekpiring
29	X8	Sensor ultrasonik belakangkanan
30	X9	Switchkanan
31	X10	Sensor ultrasonik belakangkiri
32	X11	Switchkiri
33	X12	Sensor ultrasonik kiribelakang
34	X13	Posisitengah
35	X14	Sensor ultrasonik kiridepan
36	X15	Saklar mode start

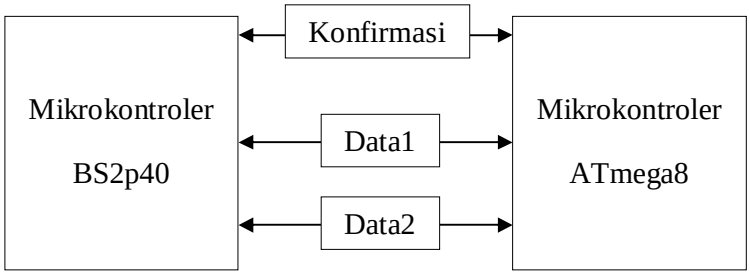
3.1.2 Mikrokontroler ATmega8

Mikrokontroler ATmega8 disini berfungsi sebagai mikrokontroler pendukung BS2p40, pemilihan mikrokontroler ini berdasarkan banyaknya kebutuhan sensor sehingga perlu mikrokontroler pendukung.

Tabel 3.2. Penggunaan Pin I/O ATmega8

Nama Port	No Pin	Pin I/O	Tipe Pin	Fungsi
PORT B	14	0	I/O	Komunikasi ke BS2p40 (data1)
	15	1	I/O	Komunikasi ke BS2p40 (data2)
	16	2	I/O	Sensor Lantai
	17	3	I/O	Konfirmasi komunikasi
	18	4	I/O	LED indikator
	19	5	I/O	LED indicator
	9	6	I/O	
	10	7	I/O	
PORT C	23	0	ADC	Sensor suara
	24	1	ADC	
	25	2	ADC	
	26	3	ADC	
	27	4	ADC	
	28	5	ADC	
	1	6	ADC	
PORT D	2	0	I/O	
	3	1	I/O	
	4	2	I/O	
	5	3	I/O	
	6	4	I/O	
	11	5	I/O	
	12	6	I/O	
	13	7	I/O	Saklar untuk mode kecepatan navigasi

Kedua mikrokontroler saling berkomunikasi dengan menggunakan 3 jalur komunikasi, pertama jalur konfirmasi berfungsi sebagai penanda sebuah intruksi sudah dieksekusi dan yang akan dieksekusi, dan dua jalur lainnya sebagai data dari intruksi.



Gambar 3.1. Blok Diagram Komunikasi Mikrokontroler

Komunikasi yang dilakukan dengan mengirimkan sinyal digital bernilai 1 dan 0 yang dikombinasikan dari data1 dan data2, setiap intruksi yang dikirim memiliki data yang berbeda. Mikrokontroler BS2p40 mengirimkan intruksi yang akan dikerjakan oleh ATmega8 dan jika sudah beres dikerjakan mikrokontroler ATmega8 akan mengirimkan data yang sama ke mikrokontroler BS2p40, untuk menghindari pengiriman data yang gagal maka digunakan satu jalur konfirmasi dimana sebelum proses pengiriman data dilakukan, pada bagian mikrokontroler pengirim akan menanyakan apakah data sudah siap untuk diterima atau belum pada bagian mikrokontroler penerima.

Tabel 3.3. Data Komunikasi

Mikrokontroler	Perintah	Data	
ATmega8	Mode piring atas	1	0
	Mode piring bawah	0	1

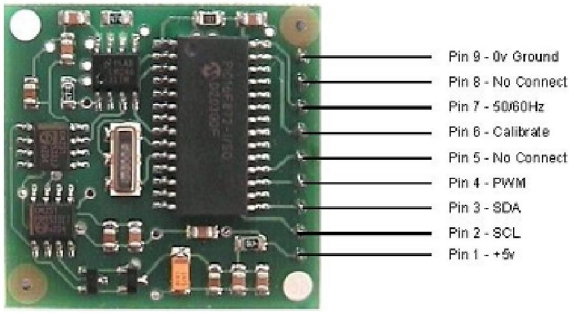
Pada awal program BS2p40 harus mengetahui piring mana yang harus diambil, perintah tersebut dikirim oleh mikrokontroler ATmega8 yang sudah tersabung dengan sensor suara. Untuk menentukan posisi piring mana yang harus diambil yaitu dengan suara yang keluar dari modul penghasil suara, jika suara yang keluar memiliki frekuensi 3500Hz maka mikrokontroler ATmega8 mengirimkan intruksi ke mikrokontroler BS2p40 untuk mengambil piring di

bawah, jika frekuensi 7500Hz maka mengambil pring di rak atas suara yang keluar memiliki toleransi ±10%.

3.1.3 Kalibrasi Kompas

Ada dua mode dalam start pertama mode start home standart dan mode start home arbitrary, untuk start standart sudah ditentukan di pojok kiri yang ditunjukan pada gambar 3.20, dan untuk start arbitrary ditentukan tidak akan lebih dekat dari posisi piring dan posisi home standart, untuk posisi home arbitrary diperkirakan didekat meja.

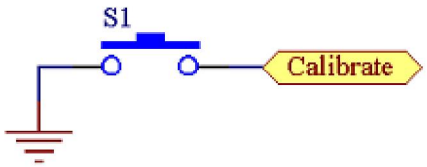
Dengan demikian robot harus bisa bergerak dari home standart dan home arbitrary, untuk mode start arbitrary diperlukan sebuah modul kompas digital untuk menentukan arah robot karena pada mode start arbitrary arah robot ditentukan oleh juri. Modul kompas yang digunakan yaitu CMPS03 *Magnetic Compass* buatan Devantech. Yang mempunyai ukuran 4x4 cm dan menggunakan sensor medan magnet Philips KMZ51, sensor magnet ini cukup sensitif untuk mendeteksi medan magnet bumi.



Gambar 3.2. CMPS03 *Magnetic Compass*

Terdapat 3 pin data yang masuk ke mikrokontroler yaitu PWM, SDA, dan SCL, sinyal PWM merupakan sebuah sinyal yang telah dimodulasi lebar pulsanya, pada CMPS03 lebar pulsa positif mempresentasikan sudut arah. Arah sudut mulai dari 0 - 259 derajat dan memiliki lebar pulsa mulai dari 1ms – 36.99ms, artinya lebar pulsa bisa berubah sebesar 100us setiap derajatnya. Sinyal

akan *low* selama 65ms di anatara pulsa, sehingga total periodenya adalah 65ms ditambah lebar pulsa positif antara 66ms sampai 102ms.

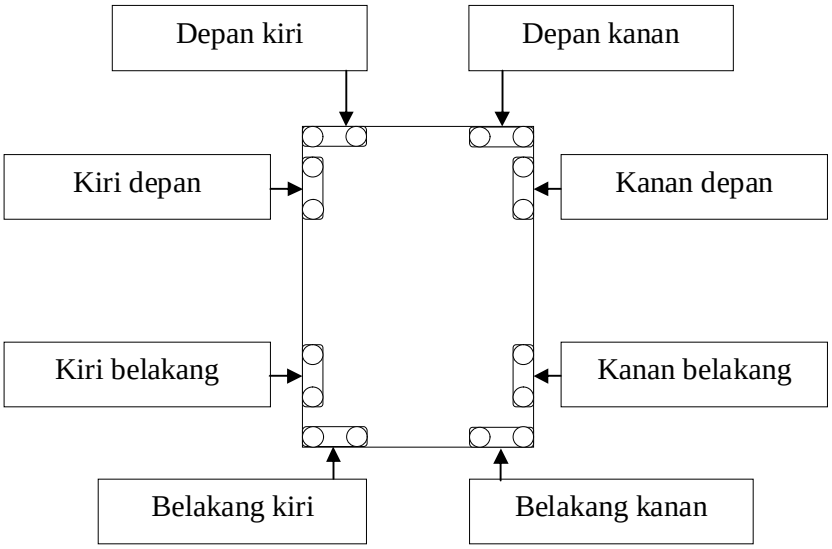


Gambar 3.3. *Tactile Switch* Untuk Proses Kalibrasi

Sebelumnya modul CMPS03 telah dikalibrasi oleh pabriknya dengan sudut utara bumi, jika ingin menentukan sudut utara yang kita inginkan maka modul ini harus dikalibrasi ulang dengan cara *I2C* atau dengan menggunakan pin manual. Untuk mempermudah mengkalibrasi maka menggunakan teknik pin maunua karena lebih efisien, dengan menggunakan rangkaian *tactile switch* seperti pada gambar 3.2 yang langsung terhubung ke pin 4 pada modul CMPS03. Dengan langkah memposisikan orientasi utara dari CMPS03 ke arah utara yang diinginkan lalu tekan tactile switch, dan begitu seterusnya untuk arah selatan, timur, dan barat. Dengan demikian robot mempunyai arah utara sendiri sehingga bisa menentukan arah gerak robot, dan tidak tergantung pada arah utara bumi.

3.1.4 Sensor Ultrasonik PING)))

Pemilihan sensor Ultrasonik dilakukan berdasarkan kebutuhan yang diperlukan robot dalam bernavigasi, program untuk sensor Ultrasonik PING))) sudah ada dalam datasheet sensor tersebut, sehingga memudahkan dalam pengembangan program. Sensor yang digunakan sebanyak delapan buah, yang digunakan untuk navigasi robot.



Gambar 3.4. Penempatan Sensor Ultrasonik

Pada bagian robot sensor sebelah kanan dan sebelah kiri berfungsi untuk bergerak maju menyusuri dinding, dengan cara membandingkan jarak sensor kanan depan dan sensor kanan belakang, jika sensor kanan depan lebih besar dari pada sensor kanan belakang maka kecepatan motor kanan berkurang, dan jika sebaliknya maka kecepatan motor kiri berkurang. Begitu juga pada sensor sebelah kiri bekerja, sama seperti sensor pada bagian sebelah kanan. Untuk sensor pada bagian depan dan belakang, berfungsi untuk meluruskan posisi robot dengan dinding sesudah mengerjakan gerakan belok.

Berdasarkan data sheet pada sensor ultrasonic PING))), memiliki kecepatan suara 1130 kaki/detik jika dikonversi ke centi meter setara dengan 34442,4 cm/detik. Untuk mengetahui besarnya jarak antara sensor dengan objek dapat ditulis program seperti dibawah.

PULSOUT Pin, Duration

Keterangan :

- PULSOUT : Pulsa yang dihasilkan oleh Pin selama Duration yang akan menjadi *input* sensor.

- Pin : Sebuah variabel/konstanta yang menetapak pin I/O yang digunakan, pin ini diset untuk mode *Output*.
- Duration : Lamanya waktu pulsa yang dihasilkan antara 1-65535.

Setelah pulsa *trigger* dikirim, sensor menunggu selama 750us sebelum mengirimkan gelombang ultrasonik pendek. Waktu ini juga digunakan oleh sensor untuk mengrimkan gelombang ultrasonik selama 200us pada frekuensi 40Khz dan mikrokontroler menyiapkan intruksi selanjutnya. Berikut intruksi yang digunakan untuk memulai pengukuran, setelah proses pengiriman pulsa dari sensor.

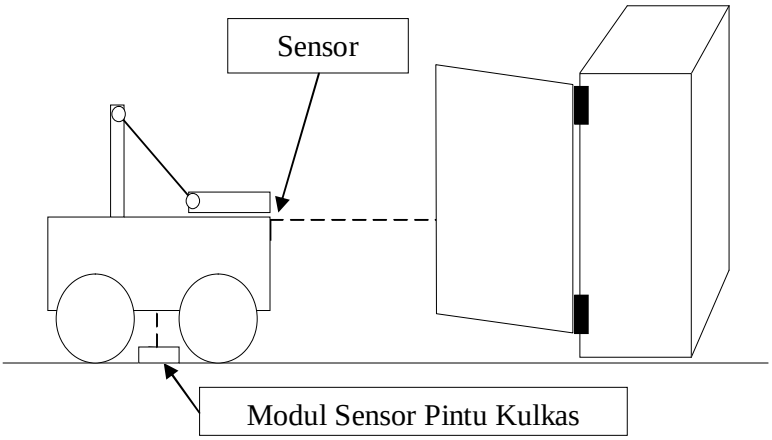
PULSIN Pin, State, Variable

Keterangan:

- PULSIN : Mengukur lebar pulsa pada *Pin* yang digambarkan oleh *State* dan menyimpan hasilnya ke dalam Variabel.
- Pin : Sebuah variabel/konstanta yang menetapak pin I/O yang digunakan, pin ini diset untuk mode *Input*.
- State : Sebuah variabel/konstanta (0-1) yang menetapkan apakah pulsa yang diukur *low* (0) atau *high* (1).
- Variable : Sebuah variabel yang mana pulsa yang diukur akan disimpan.

3.1.5 Sensor Inframerah (GP2D12 dan GP2D15)

Sensor infra merah yang digunakan adalah GP2D15 dan GP2D12 yang dipasang pada bagian depan robot dan dibagian bawah *gripper*, pada bagian depan robot menggunakan dua buah sensor GP2D12 untuk mendeteksi pintu kulkas pada saat hendak membuka dan menutup, dan sensor GP2D15 untuk menentukan jarak antara robot dengan bagian dalam kulkas.



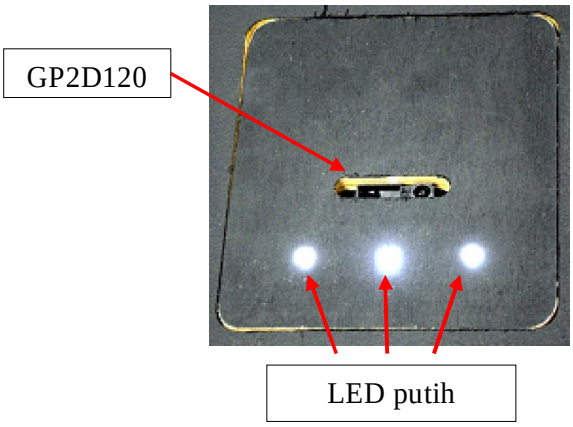
Gambar 3.5. Fungsi Sensor GP2D12 Untuk Mendeteksi Pintu Kulkas

Robot harus mengetahui jika pintu kulkas sudah terbuka penuh dan sudah tertutup rapat, cara yang dilakukan adalah dengan mendeteksi jarak sensor dengan pintu kulkas. Robot menunggu pintu kulkas hendak membuka sampai sensor mendeteksi jarak yang ditentukan pada saat pintu kulkas membuka setengah, dan sensor menunggu sampai jarak menjadi jauh pada saat pintu kulkas dalam keadaan membuka lebar. Begitu juga pada saat kondisi pintu akan hendak menutup, sensor akan menunggu sampai jarak sensor dekat dengan pintu kulkas, dan menunggu jarak dari pintu kulkas menjadi jauh kembali pada saat kondisi pintu kulkas tertutup rapat.

Pin data dari sensor GP2D12 tidak langsung masuk ke pin mikrokontroller, kita menggunakan komparator agar sinyal yang diterima mikrokontroller berupa sinyal digital. Prinsip kerja komparator hanya membandingkan tegangan Vin dengan Vref, jika Vin lebih kecil maka *output* sama dengan Vcc (*high*), jika Vin lebih besar maka *output* sama dengan Vee (*low*). Gambar 3.5 diatas menunjukan posisi sensor terhadap pintu kulkas, jarak antara sensor dengan dinding sudah ditentukan sebelumnya.

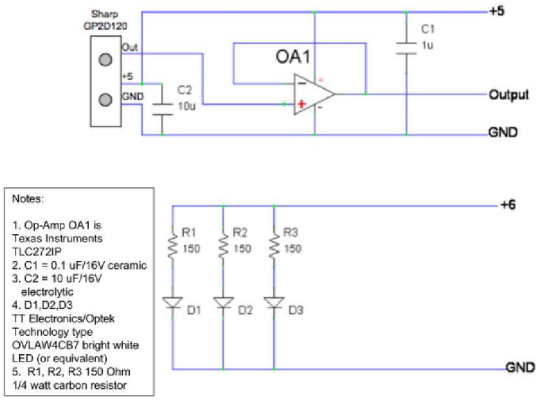
Pada gambar 3.6 terdapat sebuah modul sensor pintu kulkas yang berfungsi mendeteksi robot, jika robot berada diatas modul sensor tersebut maka pintu kulkas akan terbuka dan jika robot berada diatas modul sensor untuk kedua kalinya maka pintu akan menutup. Pada peraturannya pintu kulkas akan berfungsi hanya satu kali membuka dan satu kali menutup dalam satu kali percobaan, jika

robot berada diatas modul sensor untuk ketiga kalinya maka pintu kulkas tidak akan melakukan gerakan membuka atau menutup .



Gambar 3.6. Modul Sensor Pintu Kulkas

Sensor yang digunakan berupa sensor infra merah GP2D120, sensor ini di pasang dilantai lapangan dengan posisi menghadap keatas dengan dilengkapi tiga buah LED berwana putih. Berikut rangkaian modul sensor pintuk kulkas yang ditujukan pada gambar 3.7.



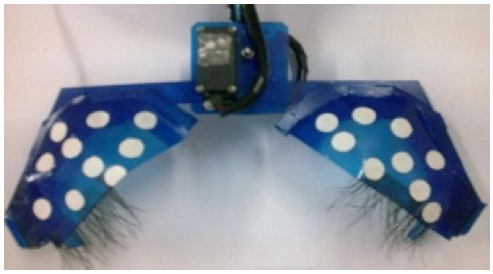
Gambar 3.7. Rangkain Modul Sensor Pintu Kulkas

Setelah robot berhasil membuka pintu kulkas maka robot harus mengambil piring yang sudah ditentukan, robot bergerak maju kebagian dalam kulkas sampai jarak sensor GP2D15 dengan rak bagian bawah kurang dari 24cm. Posisi

pemasangan sensor tersebut agak miring keatas, karena robot mendeteksi kebagian bawah rak kulkas.

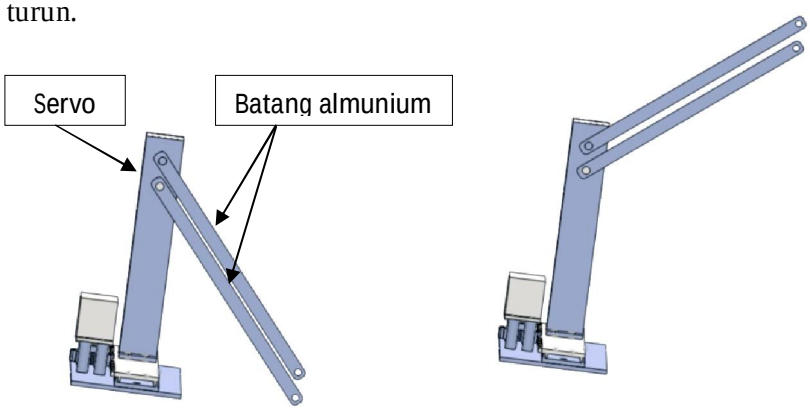
3.1.6 Mekanisme *Gripper*

Robot memiliki satu buah *gripper* yang berfungsi untuk mengambil piring, dengan menggunakan 1 buah servo, 2 buah *gear* berdiameter 1,5 cm dan potongan *acrylic* sebagai pencapit piring. Bentuk *gripper* dirancang sesuai dengan ukuran dari piring yang digunakan, ukuran piring tersebut berdiameter 10cm maka robot hanya bisa mengambil piring dengan ukuran yang sudah ditentukan. Pada gambar 3.8 bentuk *gripper* dengan penutup makanan terlihat tampak atas.



Gambar 3.8. *Gripper* tampak atas

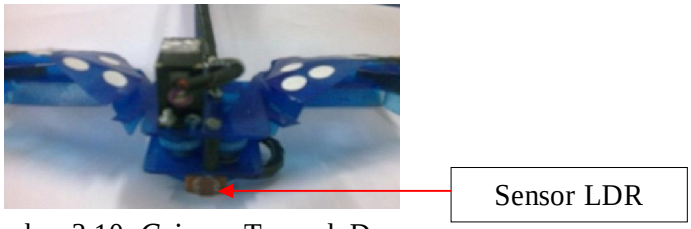
Mekanisme yang dilakukan supaya bisa mengambil piring pada rak atas dan piring pada rak bawah, ditunjukan pada gambar 3.9 dengan menggunakan 1 buah servo dan 2 buah batang almunium sebagai tuasnya supaya *gripper* bisa bergerak naik dan turun.



a **b**

Gambar 3.9. **a.** Mekanisme *Gripper* Bergerak Turun **b.** Mekanisme *Gripper* Bergerak Naik

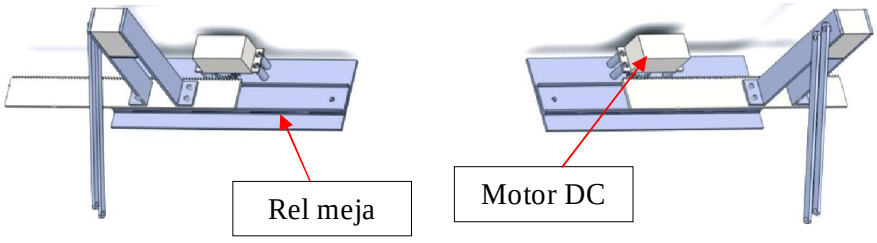
Pada bagian depan *gripper* dipasang satu buah sensor cahaya menggunakan LDR untuk mendeteksi posisi piring didalam rak, gambar dibawah menunjukan *gripper* tampak depan.



Gambar 3.10. *Gripper* Tampak Depan

Posisi *gripper* pada saat mengambil piring tidak selalu berada pada posisi lurus dengan piring, sehingga pada bagian bawah *gripper* ditambahkan mekanisme agar *gripper* bisa bergeser kekanan dan kekiri. Untuk mengetahui poisisi robot pada saat akan mengambil piring, robot mendeteksi cahaya LED dengan sensor LDR dan mengukur jarak dari sensor Ultrasonik sebelah kanan belakang ke pintu kulkas. Pada gambar 3.10 menunjukan penempatan sensor LDR.

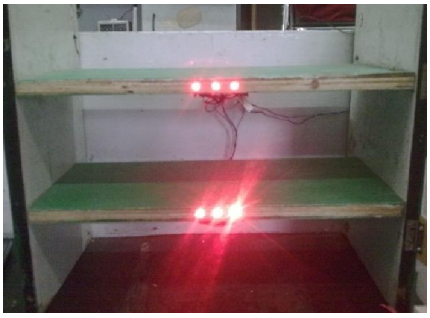
Jika jarak sensor Ultrasonik lebih besar dari 20 cm dan sensor LDR tidak mendeteksi adanya cahaya, maka *gripper* bergeser kesebelah kanan sampai sensor LDR mendeteksi adanya cahaya, dan jika jarak sensor Ultrasonik lebih kecil dari 10 dan sensor LDR tidak mendeteksi adanya cahaya maka *gripper* bergeser ke kiri sampai sensor LDR mendeteksi cahaya dari LED, berikut gambar yang menunjukan mekanisme gerak geser *gripper* berikut menunjukan mekanisme gerak *gripper* pada gambar 3.11.



a. **b.**

Gambar 3.11. **a.** Mekanisme *Gripper* Bergerak Kekanan, **b.** Mekanisme *Gripper* Bergerak Kekiri

Untuk bisa menggerakan *gripper* kekiri dan kekanan digunakan satu buah motor dc sebagai penggerak, dan juga menggunakan sebuah rel meja yang sebagai penopang *gripper*.



Gambar 3.12. Bagian Dalam Kulkas

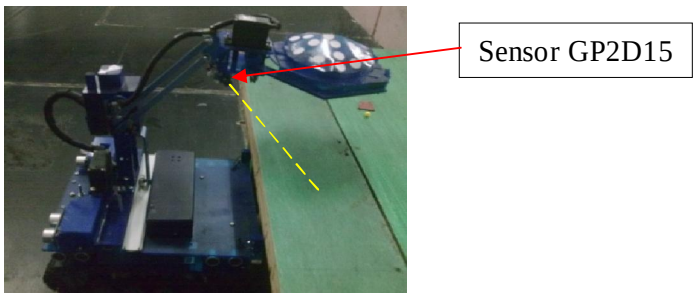
Terdapat 2 buah piring didalam kulkas dibagian rak atas dan rak bawah, robot harus mengambil salah satu dari kedua piring tersebut sesuai ketentuan pada gambar 3.12 menunjukan bagian dalam kulkas.



Gambar 3.13. Bentuk Piring Dengan Berisikan Makanan

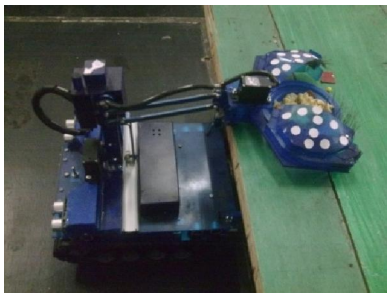
Pada gambar diatas menunjukan bentuk piring dengan berisikan makanan yang disimulasikan dengan makanan sereal, agar makanan didalam piring tidak berjatuhan, karena bisa mengurangi penilaian dalam pertandingan sehingga pada bagian atas *gripper* diberi penutup agar makanan didalam piring tidak akan jatuh.

Robot akan menyimpan piring dimeja setelah mengambil piring didalam kulkas, pemasangan sensor GP2D15 pada *gripper* yang ditunjukan pada gambar 3.14, tujuannya untuk mendeteksi jarak robot dengan meja pada saat robot bergerak maju kearah meja.



Gambar 3.14. Posisi Robot Akan Menyimpan Piring

Robot akan bergerak maju sampai kondisi sensor terpenuhi dengan jarak kurang dari 24cm dari meja. Kondisi ini bertujuan agar posisi piring tidak terlalu berada dibibir meja, yang bisa menyebabkan piring terjatuh kelantai. Gambar 3.14 menunjukan posisi robot pada saat meletakan piring.



Gambar 3.15. Posisi Robot Meletakan Piring Dimeja

3.1.7 Modul sensor suara

Robot hanya akan aktif jika diberikan suara dengan frekuensi 3500Hz dan 7500Hz, nilai frekuensi ini menentukan piring yang akan diambil oleh robot. Jika robot menerima frekuensi 3500Hz maka robot harus mengambil piring di rak bawah, dan jika robot menerima frekuensi 7500Hz maka mengambil piring dirak atas.



Gambar 3.16. Modul Sensor Penghasil Suara

Modul penghasil suara menggunakan mikrokontroler BS2sx, karena sudah ada *sintak* program untuk suara frekuensi. *Output* dari modul penerima suara berupa sinyal analog, maka dibutuhkan sebuah komponen ADC untuk mengkonversi ke sinyal digital. Mikrokontroler ATmega8 memiliki ADC internal sehingga tidak perlu menggunakan modul rangkaian ADC lain, *output* dari modul penerima suara masuk ke pin ADC ATmega8. Setelah dikonversi mikrokontroler ATmega8 mengirimkan data digital ke BS2p40, sebagai data aktifasi robot dan menentukan piring mana yang harus diambil.

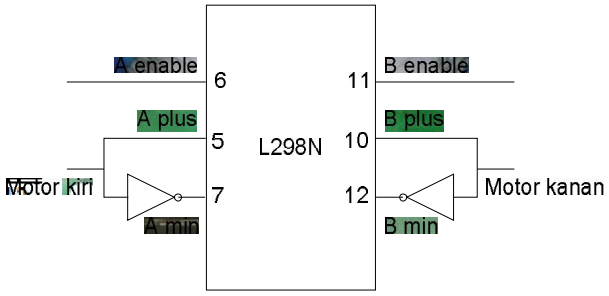
3.1.8 Kendali Gerak Motor

Robot bergerak menggunakan 2 buah motor DC, yang dikendalikan menggunakan IC L298 yang merupakan komponen *Duall Full-bridge Driver*. Komponen ini bisa mendrive dua buah motor dc sampai tegangan 46 Vdc dan arus 2 A untuk tiap kanal. Satu motor dapat dikontrol dengan 3 pin yang terdiri dari pin *Aenable, A*+, dan *A*- untuk motor A dan untuk motor B terdiri dari *Benable, B*+, dan *B*-. Arah gerak robot langsung dikendalikan oleh mikrokontroler BS2p40, untuk kecepatan robot menggunakan PWM yang sudah ada sintak program pada BS2p40.

Tabel 3.4. Tabel Kebenaran Driver Motor

Aen	A(+)	A(-)	Status Motor
Low	X	X	Motor *off*
High	Low	Low	Stop (rem)
High	High	Low	Berputar kedepan
High	Low	High	Berputar kebelakang
High	High	High	Stop (rem)
Pulsa	High	Low	Berputar kedepan
Pulsa	Low	High	Berputar kebelakang

Keterbatasan pada port I/O mikrokontroler BS2p40 mengaharuskan dalam mengoptimalkan port tersebut, maka untuk pin *A+*, *A-* dan pin *B+*,*B-* menjadi disatukan dengan menggunakan gerbang NOT IC4704 sehingga hanya membutuhkan 2 pin untuk satu motor dc.



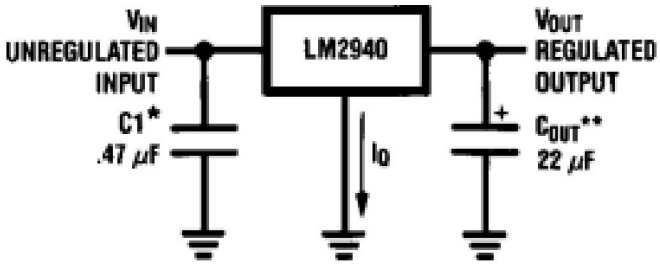
Gambar 3.17. Konfigurasi Rangkaian Driver Motor

Pin Aen merupakan variabel nilai dari kecepatan motor PWM (*Pulse Width Modulation*), dan pin A+ dan A- merupakan menentukan arah gerak motor. Untuk menentukan tegangan yang keluar dengan menggunakan PWM bisa menggunakan rumus (*Duty* ÷ 255) x tegangan baterai, sebagai contoh jika nilai duty 100 dan tegangan baterai 19,5 Vdc maka (100 ÷ 255) x 19,5 Vdc = 7,64 Vdc dari tegangan baterai. Maka jika kondisi pin Aen *high* sama dengan (255 ÷ 255) x 19,5Vdc = 19,5 Vdc, dan jika *low* sama dengan 0 Vdc.

3.1.9 Catu Daya Regulator

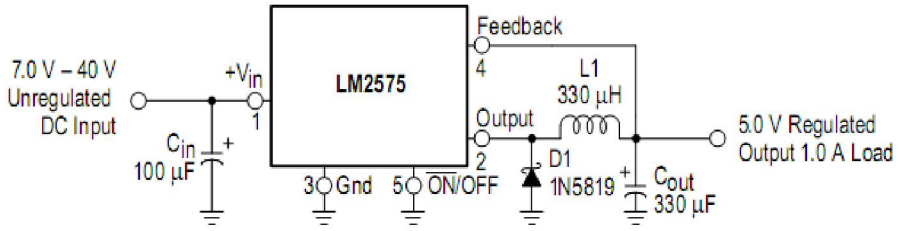
Faktor pendukung yang sangat penting adalah catu daya, karena mikrokontroler, sensor, dan aktuator bisa bekerja karena adanya tegangan. *Supply* tegangan menggunakan 3 sumber tegangan yaitu tegangan untuk mikrokontroler dan sensor dengan 5 Vdc, tegangan motor dc memakai 19,5 Vdc, dan tegangan 5 Vdc untuk mencatu kendali motor *driver*. Pemisahan tegangan ini dimaksudkan agar tegangan yang masuk ke mikrokontroler tidak terganggu oleh tegangan dari motor.

Tegangan yang digunakan berasal dari *baterai rechargeable* yang mempunyai nilai tegangan 1,2 volt per satu baterai, dengan penggunaan baterai yang diserikan sehingga untuk mendapatkan nilai tegangan untuk motor dc 19,5 Vdc dibutuhkan sebanyak 16 buah baterai, dan untuk mencatu tegangan mikrokontroler dibutuhkan 8 buah baterai dengan nilai tegangan 9,5 Vdc. Catu daya untuk tegangan *output* 5Vdc didapat dari *output* sebuah regulator, yang menggunakan IC LM2940 sebagai regulator tegangan positif yang dihasilkan dari tegangan sumber sebesar 9.6 Vdc.



Gambar 3.18. Rangkaian Regulator LM2940

Untuk mendapatkan tegangan 5 Volt sebagai tegangan *logic* bagi motor *driver*, tidak menggunakan *IC* regulator LM2940 karena arus yang dibutuhkan untuk motor *driver* kurang lebih 3A. Pemilihan *IC* regulator LM2575 didasari dengan kemampuan bekerja dari tegangan *input* sebesar 40 volt dengan arus sebesar 3A.



Gambar 3.19. Rangkaian Regulator LM2575

3.2 Perancangan Perangkat Lunak (*Software*)

3.2.1 Algoritma Dasar

Perancangan algoritma dasar ini merupkan sebagai patokan dalam perancangan algoritma secara keseluruhan, maka perancangan algoritma ini sangatlah penting yang dijelaskan sebagai berikut:

a. Gerak maju dan mundur

Gerakan maju dilakukan dengan menggerakan kedua roda secara bersamaan ke arah depan, begitu juga gerakan mundur dengan menggerakan kedua roda secara bersamaan kea rah belakang. Untuk menggerakan robot dengan arah maju dan mundur, pada mikrokontroler diberikan intruksi sebagai berikut:

```
HIGH motorkanan          ; motor arah maju
HIGH motorkiri           ; motor arah maju
PWM, Aen,speedr,40       ; kecepatan putar motor kanan
PWM, Ben,speedl,40       ; kecepatan putar motor kiri
```

Kecepatan motor kiri dan kanan tergantung pada nilai yang dimasukan pada variabel speedr dan speedl, dengan tujuan agar kecepatan motor dapat diatur sesuai dengan kebutuhan antara 0 – 255 untuk kecepatan penuh motor sebanding dengan nilai kecepatan 255 yang dimasukan dalam intruksi PWM. Untuk menggerakan robot dengan arah mundur sama seperti halnya arah maju, dengan cara mengganti arah putaran motor pada intruksi maju, berikut intruksi yang digunakan:

```
LOW motorkanan          ; motor arah mundur
LOW motorkiri           ; motor arah mundur
PWM, Aen,speedr,40      ; kecepatan putar motor kanan
PWM, Ben,speedl,40      ; kecepatan putar motor kiri
```

Intruksi maju dan mundur sering digunakan dalam navigasi robot, untuk menghindari penulisan listing program maka intruksi untuk gerak maju dan mundur dijadikan sebuah prosedur untuk menghemat pemakaian memori pada mikrokontroler, dan mempermudah dalam pengembangan algoritma.

b. Gerak belok kanan dan belok kiri

Gerakan belok dilakukan dengan menggerakan roda kiri dan roda kanan dengan arah yang berlawanan, jika ingin melakukan gerakan belok kanan maka motot kiri digerakan dengan arah maju dan motor kanan digerakan dengan arah mundur. Jika ingin melakukan gerakan belok kiri maka kebalikan dari gerakan belok kanana, untuk membuat gerakan belok kiri dan belok kanan, pada mikrokontroler diberikan intruksi sebagai berikut:

Gerakan belok kanan:

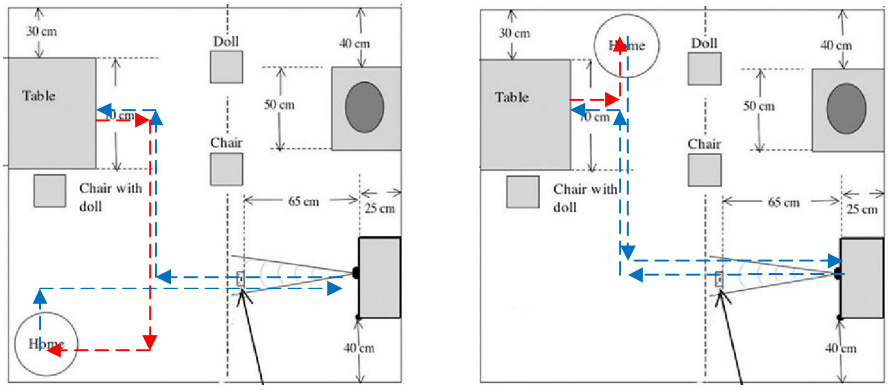
```
LOW motorkanan          ; motor arah mundur
HIGH motorkiri          ; motor arah maju
FOR ulang = 0 TO belok  ; perulangan sampai belok
PWM, Aen,speedr,40      ; kecepatan putar motor kanan
PWM, Ben,speedl,40      ; kecepatan putar motor kiri
NEXT
```

Gerakan belok kiri:

```
HIGH motorkanan         ; motor arah maju
LOW motorkiri           ; motor arah mundur
FOR ulang = 0 TO belok  ; perulangan sampai belok
PWM, Aen,speedr,40      ; kecepatan putar motor kanan
PWM, Ben,speedl,40      ; kecepatan putar motor kiri
NEXT
```

3.2.2 Algoritma Keseluruhan

Algoritma keseluruhan dalam mengambil piring didalam kulkas dan menyimpannya diatas meja untuk mode start di home standart dan mode start di home arbitrary. Untuk posisi start arbitrary diperkirakan diantara daerah tersebut, karena sesuai dengan peraturan posisi start tidak akan lebih dekat dengan posisi start standart dan posisi kulkas.



a **b**

Gambar 3.20. **a.** Mode Posisi Start Standart **b.** Mode Posisi Start Arbitrary

Konfigurasi lapangan diambil dari peraturan pertandingan Robot Waiter di Trinity College tahun 2011, dengan ketentuan secara garis besar sebagai berikut:

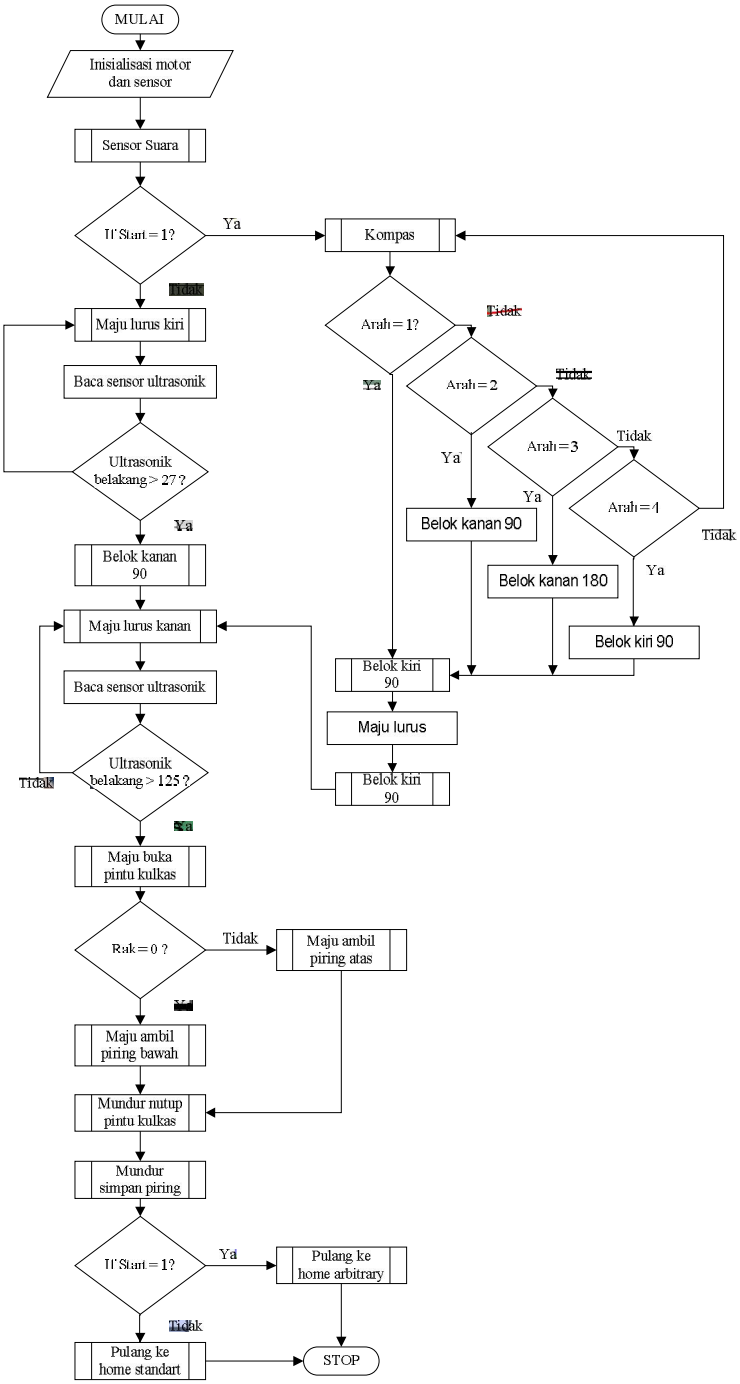
1. Ukuran robot, yang dibatasi panjang maksimal 30cm, lebar maksimal 30cm, dan tinggi maksimal 30cm

2. Ukuran arena lapangan yang berukuran 2,5m x 2,5m.
3. Terdapat sebuah kulkas yang sudah berisi piring pada rak yang berbeda, dengan ukuran rak bawah 14cm dari lantai dan rak atas 28cm dari lantai. Ukuran kulkasnya sendiri memiliki tinggi 42cm, panjang 45cm, dan lebar 25cm, yang harus dibuka dan ditutup secara otomatis oleh robot.
4. Terdapat sebuah meja tempat menyimpan piring dengan ukuran panjang 70cm, lebar 45cm, dan tinggi 21cm.
5. Tugas utama robot yaitu:

a. Robot harus membuka pintu kulkas secara otomatis, agar bisa mengambil piring yang ada di dalam kulkas.
b. Robot harus mengambil piring yang tepat sesuai suara frekuensi yang diberikan.
c. Setelah mengambil piring, robot harus bisa memastikan bahwa pintu sudah tertutup dengan benar.
d. Robot harus mengantarkan piring ke meja.

6. Beberapa kondisi yang membuat robot gagal percobaan diantaranya:
   a. Robot menyentuh pintu kulkas, boneka nenek, dan kursi.
   b. Robot salah mengambil piring.
   c. Menjatuhkan piring.
7. Posisi boneka nenek dan kursi acak pada garis lurus yang sudah ditentukan, tetapi tidak mungkin berada menghalangi sensor kulkas.
8. Mode start.
   a. Mode start standart tempatnya di pojok kiri bawah pada gambar 1 dan arah robot acak.
   b. Mode start arbitrary posisi start random tidak akan lebih dekat dari tempat penyimpanan piring dan arah robot acak.
9. Waktu yang diberikan dalam menyelesaikan tugas 4 menit, dan penambahan waktu 2 menit untuk kembali ke home.

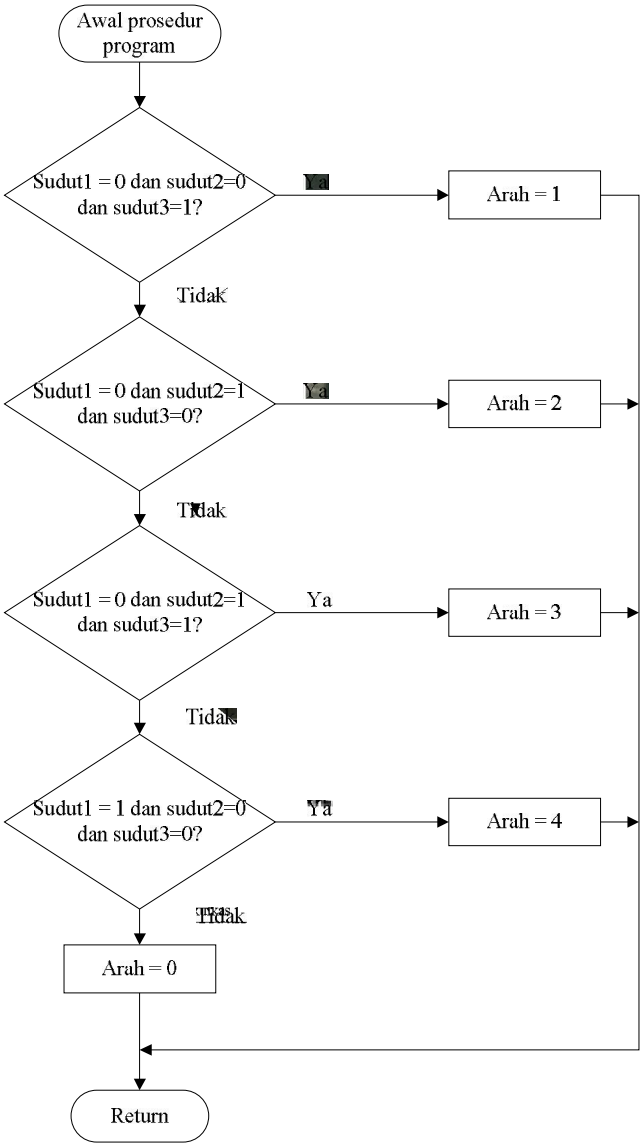
Robot harus mengetahui dari posisi home yang mana dia harus bergerak, sebelumnnya kita yang menentukan dengan cara menekan saklar toggle untuk menentukan posisi start. Algoritma gerak secara keseluruhan untuk mode start standart dan arbitrary ditunjukan pada Gambar 3.21.



Gambar 3.21. Diagram Alir Untuk Seluruh Algoritma

Berikut prosedur – prosedur dari algoritma keselurahan pada posisi start mode standart dan mode arbitrary:

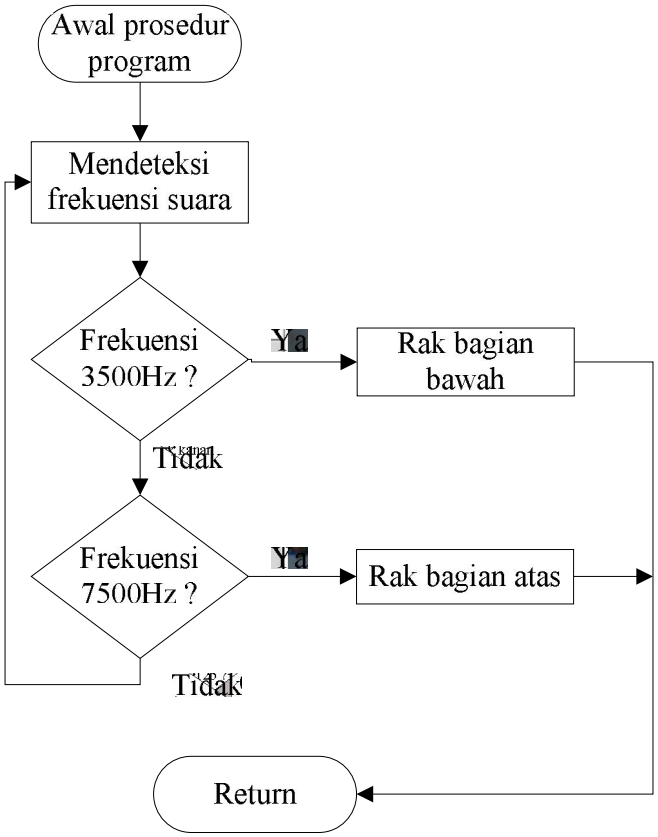
a. Prosedur Arah Kompas



Gambar 3.22. Diagram alir pembagian arah kompas

Dalam start arbitrary robot harus dikalibrasi untuk menentukan arah gerak, dengan menggunakan 4 arah sudut dengan membandingkan nilai dari sudut posisi kompas.

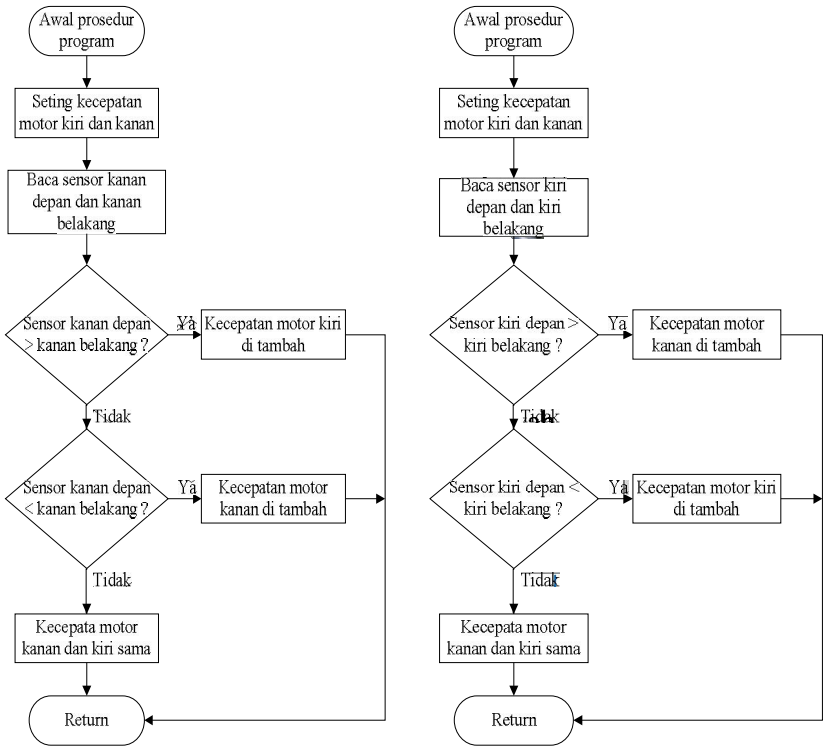
b. Prosedur sinyal suara untuk menentukan posisi piring



Gambar 3.23. Diagram Alir Sinyal Suara Untuk Menentukan Posisi Piring

Sebelum mengambil piring, robot harus mengetahui dibagian rak mana piring harus di ambil. Sinyal suara tinggi dan rendah yang membedakan posisi piring tersebut, pada gambar 3.23 merupakan diagram alir sinyal suara dan juga sebagai aktifasi robot.

c. Prosedur susur dinding kanan dan susur dinding kiri

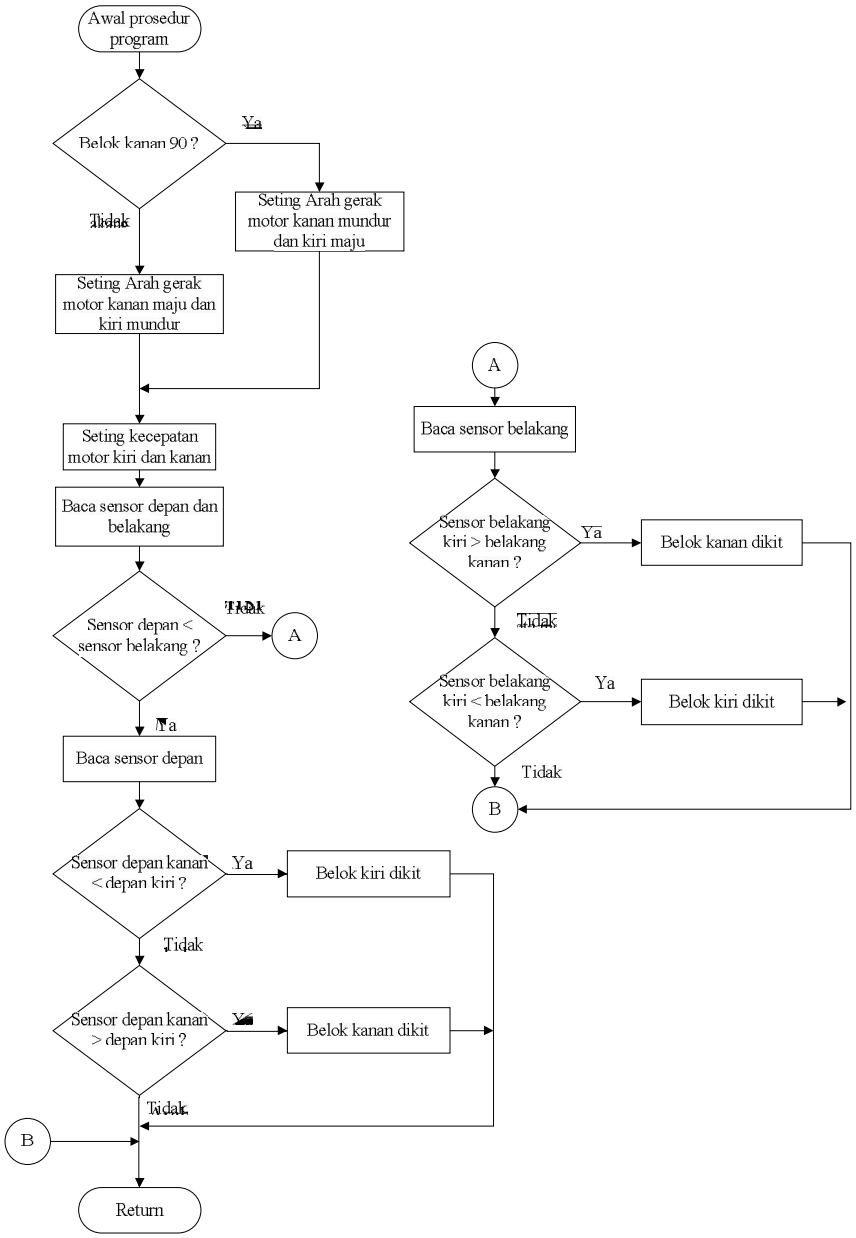


a **b**

Gambar 3.24. **a.** Diagram Alir Susur Dinding Kanan **b.** Diagram Alir Susur Dinding Kiri

Untuk prosedur susur dinding ini merupakan gerak untuk meluruskan robot dalam berjalan, dengan cara membandingkan dua buah sensor di sebelah kanan atau di sebelah kiri. Pada diagram alir diatas menunjukan bagaimana kedua sensor tersebut saling membandingkan, contoh untuk susur dinding kanan yang membandingkan antara sensor sebelah kanan bagian depan dengan sensor sebelah kanan bagian belakang. Jika sensor kanan bagian depan lebih besar jaraknya dari sensor kanan bagian belakang maka kecepatan motor sebelah kiri ditambah, jika kebalikannya maka kecepatan motor sebelah kanan yang ditambah. Dengan demikian diharapkan robot dapat berjalan dengan lurus, tidak menabrak dinding yang disusurinya.

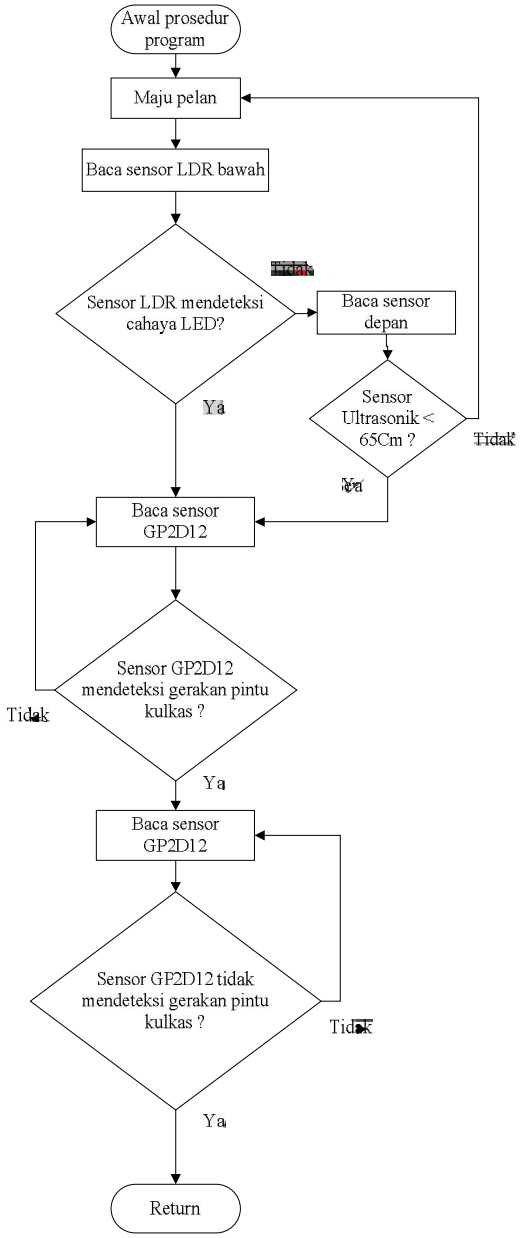
d. Prosedur belok kanan dan belok kiri



Gambar 3.25. Diagram Alir Belok Kanan Dan Belok Kiri

Pada algoritma belok kanan dan belok kiri diberikan program meluruskan ke dinding agar posisi robot tetap lurus pada saat bergerak maju, digram alir diatas menunjukan gabungan belok kanan dan kiri karena sintak programnya sama hanya membalikan arah gerak motor saja.

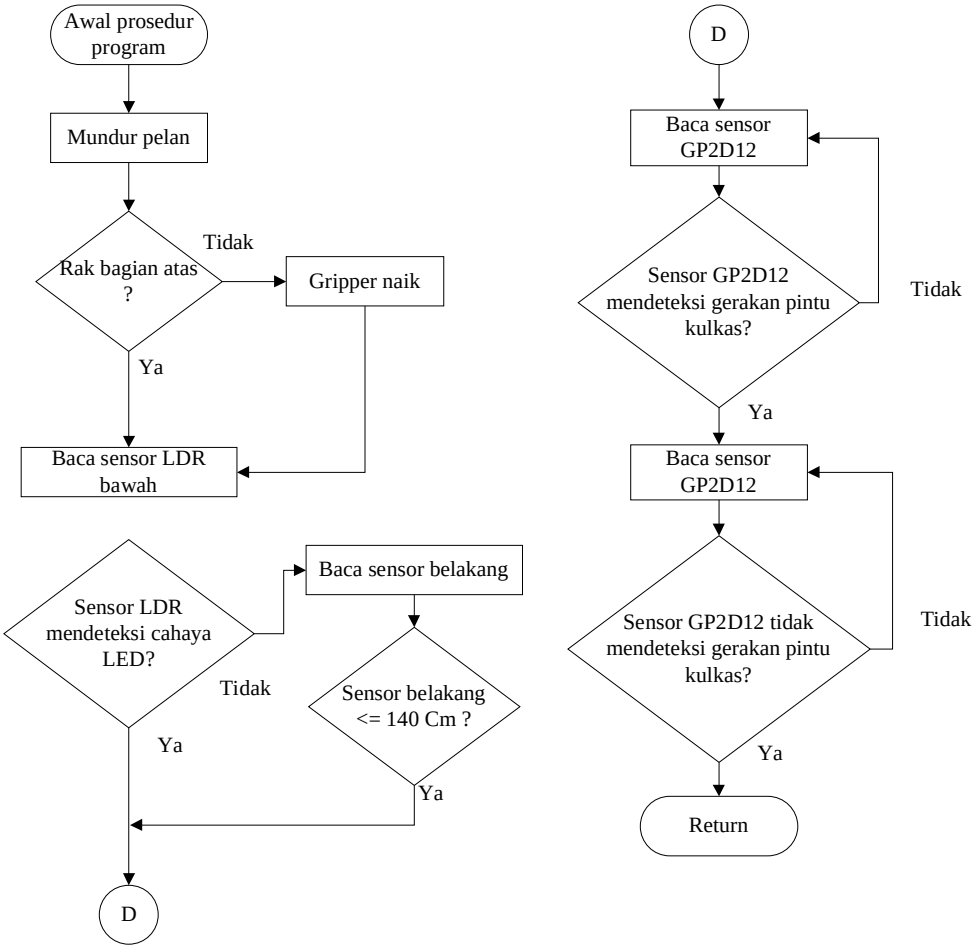
e. Prosedur maju dan membuka pintu kulkas



Gambar 3.26. Diagram Alir Maju Dang Membuka Pintu Kulkas

Untuk membuka pintu kulkas robot harus berdiri diatas modul sensor pintu kulkas yang ditanam dilantai, cara yang dilakukan dengan mendeteksi cahaya LED dengan menggunakan sensor LDR. Setelah pintu terbuka robot mendeteksi gerakan pintu untuk menentukan pintu sudah terbuka lebar.

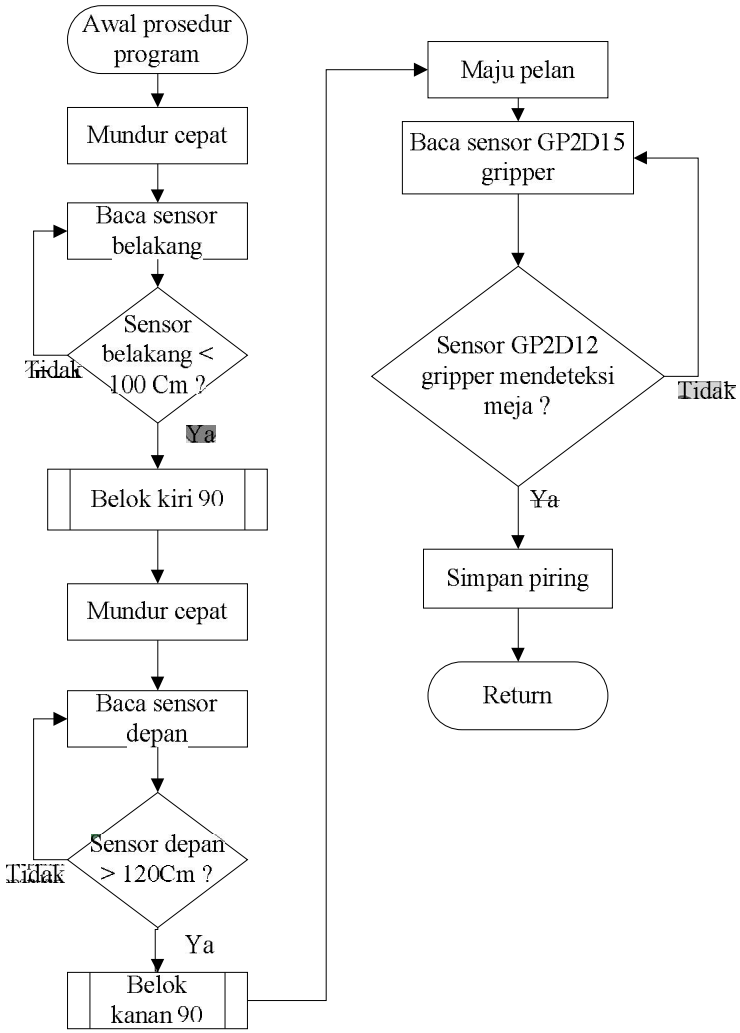
f. Prosedur maju dan menutup pintu kulkas



Gambar 3.27. Diagram Alir Maju Dan Menutup Pintu Kulkas

Seperti halnya membuka pintu kulkas robot harus berada diatas modul sensor pintu kulkas untuk kedua kalinya, sehingga pintu bergerak menutup dan mendeteksi gerakan pintu dengan menggunakan sensor inframerah GP2D12 untuk menentukan pintu kulkas sudah tertutup rapat.

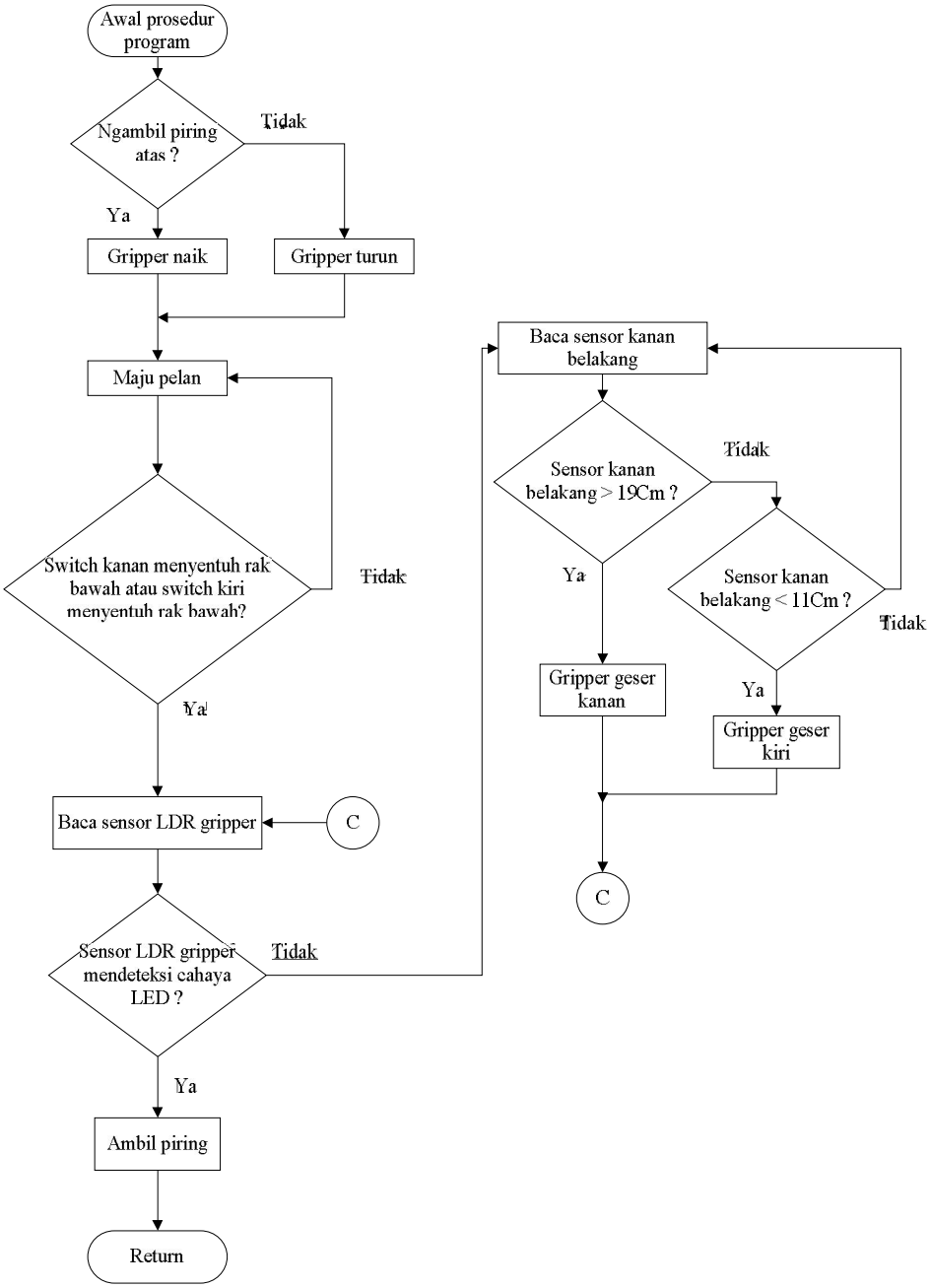
g. Prosedur bergerak untuk menyimpan piring di meja



Gambar 3.28. Diagram Alir Bergerak Untuk Menyimpan Piring

Setelah robot mengambil piring dan menutup pintu kulkas, kemudian robot bergerak ke arah meja sebagai tempat untuk menyimpan piring yang disimulasikan sebagai meja boneka kakek. Agar pada saat menyimpan piring dimeja posisi piring tidak terlalu pinggir maka dipasang sensor dibagian bawah *gripper*, untuk mendeteksi jarak robot terhadap meja pada saat sebelum menyimpan piring.

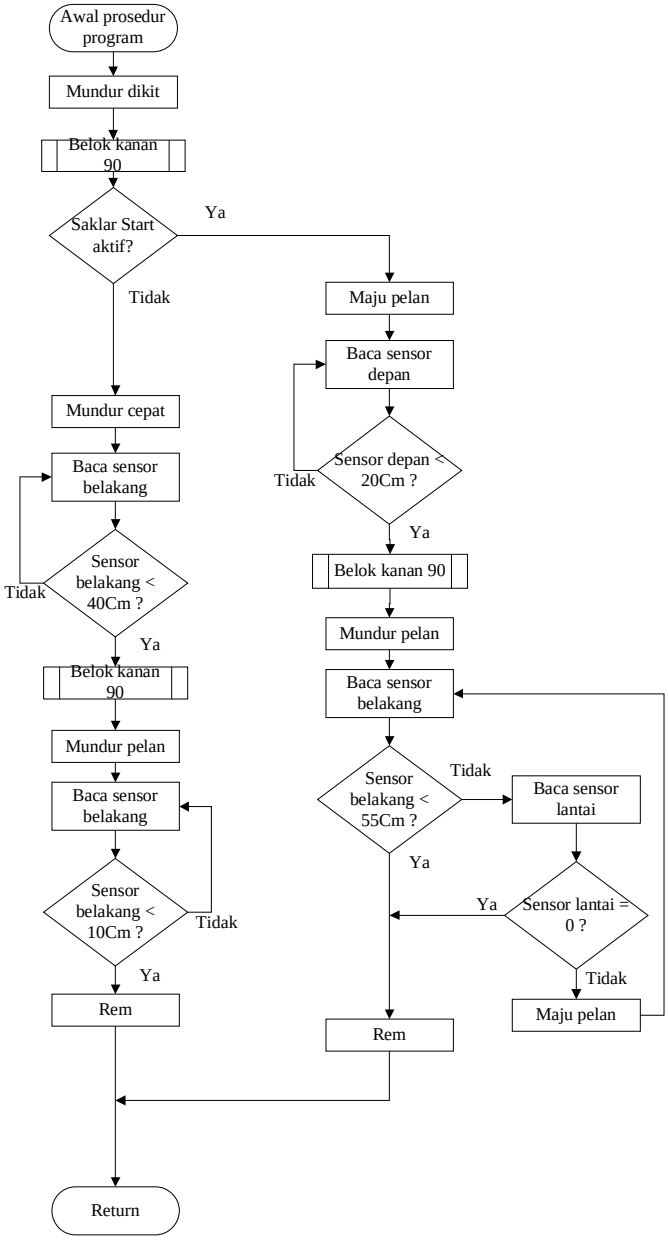
h. Prosedur mengambil piring di rak atas dan piring rak bawah



Gambar 3.29. Diagram Alir Mengambil Piring Di Rak Atas Dan Rak Bawah

Algoritma mengambil piring atas dan piring bawah sama saja, hanya yang membedakan dari gerak servo akan naik jika mengambil dirak atas dan akan turung jika mengambil dirak bawah.

i. Prosedur kembali ke posisi awal



Gambar 3.30. Diagram Alir Kembali Kembali Keawal

Setelah menyimpan piring robot harus kembali ke posisi awal, sesuai dengan posisi start awal pada saat robot mulai bergerak. Pada diagram alir diatas menunjukan robot bergerak kembali ke posisi awal.