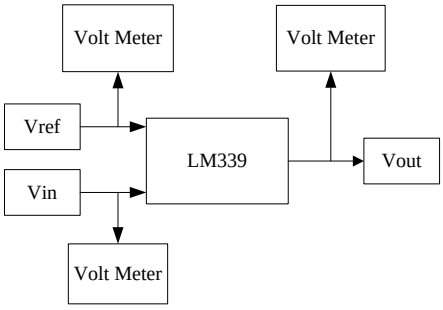
BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini akan diuraikan mengenai proses pengujian dari sistem yang dirancang. Dimana pengujian yang dilakukan meliputi pengukuran terhadap parameter-parameter komponen *input*/masukan, proses dan *output*/keluaran yang terdapat dalam sistem yang dirancang yang kemudian dilanjutkan dengan menganalisis hasil-hasil pengukuran tersebut. Hal ini bertujuan untuk melihat sistem yang dirancang berjalan sesuai dengan yang dirancang.

4.1. Pengujian dan Analisis *Infrared* dan Photodioda Pendeteksi Garis

Pengujian *infrared* dan photodioda pendeteksi garis ini bertujuan untuk mendapatkan nilai *output* photodioda saat menerima cahaya *infrared* melalui pantulan oleh garis lintasan warna hitam atau putih, dan mendapatkan nilai referensi yang dibutuhkan sebagai nilai perbandingan rangkaian komparator sehingga diharapkan mampu memperoleh nilai keluaran komparator sebagai masukan bagi mikrokontroler. Berikut *setup* pengukuran tegangan referensi, tegangan masukan dan tegangan keluaran IC LM339.



Gambar 4.1. *Setup* Pengukuran Tegangan Referensi,Tegangan Masukan dan Tegangan Keluaran IC LM339

Keterangan:
Vref = tegangan keluaran trimpot
Vin = tegangan keluaran photodioda.

Hasil dari *setup* pengukuran pada gambar 4.1 ditunjukan pada tabel 4.1 berikut.

Tabel 4.1 Pengukuran Tegangan Pendeteksian Garis Lintasan

Pendeteksi garis	**Tegangan Referensi (Volt)**	**Tegangan *Input* (Volt)**		**Tegangan *Output*(Volt)**	
		Vi1	**Vi2**	**Garis tidak terdeteksi (Vo1)**	**Garis Terdeteksi (Vo2)**
SR3	4,04	4,39	3,03	-2,93	4,60
SR2	3,43	4,26	0,65	-2,92	4,61
SR1	3,82	4,40	1,19	-2,94	4,62
SC1	3,80	4,42	1,75	-3,16	4,62
SC2	4,05	4,37	3,13	-3,15	4,61
SL1	4,00	4,37	0,84	-2,93	4,62
SL2	2,50	4,38	1.02	-3,05	4,62
SL3	3,67	4,40	0,92	-2,77	4,61

Keterangan:
SR3 : sensor kanan 3
SR2 : sensor kanan 2
SR1 : sensor kanan 1
SC1 : sensor tengah 1
SC2 : sensor tengah 2
SL1 : sensor kiri 1
SL2 : sensor kiri 2
SL3 : sensor kanan 3
Vi1 : tegangan *input* ketika mendeteksi warna putih
Vi2 : tegangan *input ketika* mendeteksi warna hitam
Vo1 : tegangan *output* ketika mendeteksi warna putih
Vo2 : tegangan *output* ketika mendeteksi warna hitam

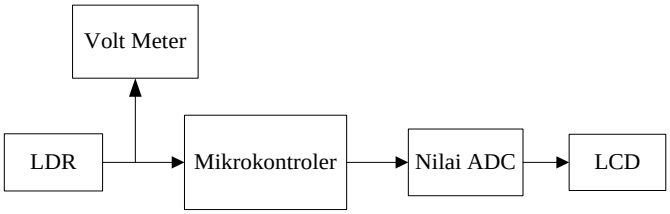
Analisis table 4.1 :

Cara kerja dari pendeteksi garis dalam perancangan ini adalah jika nilai referensi (Vref) lebih besar dari nilai keluaran photodioda (Vin) maka tegangan keluaran ICLM339 berlogika “1” atau garis hitam terdeteksi dan jika nilai referensi (Vref) lebih kecil dari nilai keluaran photodioda (Vin) maka tegangan keluaran ICLM339 berlogika “0” atau garis hitam tidak terdeteksi. dari data hasil pengukuran dan pengamatan menunjukan bahwa hasil perbandingan antara

tegangan referensi dengan 2 buah tegangan *input* yang menunjukan pendeteksian warna putih dan hitam menyimpulkan bahwa tegangan *output* yang mengarah pada tegangan rata-rata 4,6 V yang mewakili logika "1", Artinya sensor akan selalu mendeteksi warna hitam sebagai garis lintasan sedangkan tegangan *output* yang mengarah pada tegangan negatif yang mewakili logika "0" Artinya sensor tidak mendeteksi garis.

4.2. Pengujian dan Analisis Sensor *Light Dependent Resistor* (LDR) Pendeteksi Warna Barang

Light dependent resistor (LDR) adalah sensor yang mampu merubah besaran cayaha kedalam besaran listrik. Mengetahui jika perubahan resistansi LDR tergantung pada intensitas cahaya yang mengenai permukaannya maka dilakukan pengujian untuk mengetahui intensitas cahaya dari pantulan beberapa warna dengan cara mengarahkan cahaya kepermukaan warna sehingga pantulan cahaya dari warna tersebut akan mengenai permukaan sensor LDR. Pengujian ini bertujuan untuk mendapatkan nilai ADC sebagai parameter penggunaan 4 warna barang yang akan digunakan. Untuk mengetahui nilai ADC dan tegangan keluaran dari sensor warna ini maka dilakukan pengujian pada 7 warna yang berbeda, yaitu putih, kuning, merah, coklat, hijau, hitam dan biru dimana nantinya akan diambil 4 warna yang memiliki nilai ADC berjauhan sebagai warna yang akan digunakan. Berikut *setup* pengujian LDR yang hasilnya ditampilkan pada LCD.



Gambar 4.2. Pengujian Sensor *Dependent Resistor* (LDR)



(a)



(b)



(c)



(d)



(e)



(f)



(g)

Gambar 4.3. Pengujian *Light Dependent Resistor* (LDR)

Keterangan:
Gambar 4.3.(a) : pengujian mendeteksi warna putih
Gambar 4.3.(b) : pengujian mendeteksi warna kuning
Gambar 4.3.(c) : pengujian mendeteksi warna merah
Gambar 4.3.(d) : pengujian mendeteksi warna coklat
Gambar 4.3.(e) : pengujian mendeteksi warna hijau
Gambar 4.3.(f) : pengujian mendeteksi warna hitam
Gambar 4.3.(g) : pengujian mendeteksi warna biru.

Hasil dari *setup* pegujian pada gambar 4.2. diperoleh hasil pengujian seperti yang ditunjukan pada tabel 4.2 berikut.

Table 4.2 Pengujian Sensor *Dependent Resistor* (LDR) Pendeteksi Warna Barang

Warna	**Jarak (cm)**	**Tegangan (Vout)**	**Nilai ADC**	
			Pengukuran	**Perhitungan**
Putih	**2**	0,72	148	147
Kuning		0,88	182	180
Merah		1,07	222	218
Coklat		1,1	228	225
Hijau		1,79	370	366
Hitam		1,95	406	398
Biru		2,98	616	609

Nilai ADC hasil pengukuran pada table 4.2 diatas didapatkan dari *analog to decimal converter* pada mikrokontroler yang di tampilkan pada LCD dan nilai ADC hasil perhitungan didapat dari dibawah ini dimana nilai analog tegangan adalah nilai Vout pada tabel 4.2, tegangan AVCC adalah 5V dan 1023 adalah resolusi ADC pada Atmega 2560 yaitu 10 bit. Berikut rumus dan perhitungan nilai ADC.

$$Nilai\ digital = \frac{nilai\ analog\ tegangan}{tegangan\ AVCC} x1023$$

$$Nilai\ digital\ warna\ putih\ = \frac{0{,}72V}{5V} x1023 = 147$$

$$Nilai\ digital\ warna\ kuning\ = \frac{0{,}88}{5V} x1023 = 180$$

$$Nilai\ digital\ warna\ merah\ = \frac{1{,}07V}{5V} x1023 = 218$$

$$Nilai\ digital\ warna\ coklat\ = \frac{1{,}1V}{5V} x1023 = 225$$

$$Nilai\ digital\ warna\ hijau\ = \frac{1{,}79V}{5V} x1023 = 366$$

$$Nilai\ digital\ warna\ hitam = \frac{1{,}95V}{5V} x1023 = 398$$

$$Nilai\ digital\ warna\ biru\ = \frac{2{,}98V}{5V} x1023 = 609$$

Analisis:

Dari tabel 4.2 dapat dilihat jika nilai ADC hasil pengukuran dan perhitungan memiliki hasil nilai yang tidak jauh berbeda dimana perbedaan tersebut masih bisa ditoleransi. Dari hasil pengukuran dan perhitungan sensor LDR sebagai pendetekesi warna diperoleh beberapa nilai ADC yang berdekatan seperti nilai ADC pada warna putih dengan kuning, merah dengan coklat dan hijau dengan hitam. Dari tabel 4.2 diatas juga terdapat warna-warna yang bisa digunakan karena memiliki nilai ADC yang berjauhan seperti warna putih, merah, hitam dan biru kemudian warna kuning, coklat, hijau dan biru. Pemilihan 4 warna dilakukan secara bebas dengan nilai ADC 4 warna yang di pilih harus berjauhan Untuk mengurangi kemungkinan kesalahan pada saat pendeteksian warna. Dari hasil pengujian ini maka diambil 4 warna yaitu, putih, merah, hitam dan biru.

4.3. Pengujian dan Analisis *Keypad* 4x4 dan *Liquid Crystal Display* (LCD) 16x2

Pengujian *keypad* 4x4 dan *Liquid Crysal Display* (LCD) *Character* 16x2 dilakukan untuk mendapatkan tampilan berupa karakter (huruf, angka, dsb) yang sesuai dengan penekanan *keypad*, dimana karakter-karakter tersebut akan tampil

pada layar LCD tersebut. Karena itu, Agar *user* dapat melihat kode rak pengalamatan barang akan disimpan dan LCD dapat menampilkan karakter (huruf, angka, dsb) tersebut maka perlu dilakukan pemrograman mikrokontroler. Adapun *listing* programnya ditunjukan sebagai berikut.

```
//.........Definisi 16 karakter untuk tampil.........
  char tampil [16] = {
  ' ',' ',' ',' ',
  ' ',' ',' ',' ',
  ' ',' ',' ',' ',
  ' ',' ',' ',' '};
//...................Pin yang digunakan keypad..........
byte rowPins[ROWS] ={39, 41, 43, 45};
byte colPins[COLS] ={47, 49, 51, 53};
//...................inisialisasi pembacaan keypad......
Keypad myKeypad = Keypad(makeKeymap(Key), rowPins, coll
byte k=4, n=0;
String jawab,kunci = "A01";
 String jawab1,kunci1 = "A02";
 String jawab2,kunci2 = "A03";
 String jawab3,kunci3 = "A04";
 String jawab4,kunci4 = "A05";
 String jawab5,kunci5 = "A06";
 String jawab6,kunci6 = "A07";
 String jawab7,kunci7 = "A08";
 String jawab8,kunci8 = "A09";
//........ Penambahan File LCD..............
#include <LiquidCrystal.h>
//........ Penambahan File keypad...........
#include <Keypad.h>
//.........Pin yang digunakn LCD............
LiquidCrystal lcd (16, 17, 18, 19, 20, 21);
//.........Penentuan Matrix Keypad
const byte ROWS = 4;
const byte COLS =4;
//.........Defini tombol keypad....
char Key[ROWS] [COLS] = {
  {'1', '4', '7', '*'},
  {'2', '5', '8', '0'},
  {'3', '6', '9', '>'},
  {'A', 'B', 'C', 'D'}
};
//.........Definisi 16 karakter untuk berita.........
char berita [16] = {
  ' ',' ',' ',' ',
  ' ',' ',' ',' ',
  ' ',' ',' ',' ',
  ' ',' ',' ',' '};
 String jawab9,kunci9 = "B01";
 String jawab10,kunci10 = "B02";
 String jawab11,kunci11 = "B03";
 String jawab12,kunci12 = "B04";
 String jawab13,kunci13 = "B05";
 String jawab14,kunci14 = "B06";
 String jawab15,kunci15 = "B07";
 String jawab16,kunci16 = "B08";
 String jawab17,kunci17 = "B09";

 String jawab18,kunci18 = "C01";
 String jawab19,kunci19 = "C02";
 String jawab20,kunci20 = "C03";
 String jawab21,kunci21 = "C04";
 String jawab22,kunci22 = "C05";
 String jawab23,kunci23 = "C06";
 String jawab24,kunci24 = "C07";
 String jawab25,kunci25 = "C08";
 String jawab26,kunci26 = "C09";
 String jawab27,kunci27 = "D01";
 String jawab28,kunci28 = "D02";
 String jawab29,kunci29 = "D03";
 String jawab30,kunci30 = "D04";
 String jawab31,kunci31 = "D05";
 String jawab32,kunci32 = "D06";
 String jawab33,kunci33 = "D07";
 String jawab34,kunci34 = "D08";
 String jawab35,kunci35 = "D09";

void setup () {
  lcd.begin(16,2);
  lcd.setCursor (0,0);
  lcd.print("INSERT KEY");
  delay(2500);
  lcd.clear();
}

void loop(){
//...........Proses Scanning dan ditampilkan pada LCD ........
  char Key = myKeypad.getKey();
  if (Key != NO_KEY){
    berita[n]=Key;
    tampil[n]=Key;
    n++;
    if (n>15){
      for(n=1;n<=15;n++)berita[n-1]=berita[n];
      n=15;
    }
      berita[n]=' ';
      lcd.setCursor(0,0);
      lcd.print (tampil);

 }if (Key=='>'){
        for(n=0;n<3;n++)jawab+=berita[n];
        for(n=0;n<3;n++)jawab1+=berita[n];
        for(n=0;n<3;n++)jawab2+=berita[n];
        for(n=0;n<3;n++)jawab3+=berita[n];
         for(n=0;n<3;n++)jawab4+=berita[n];
        for(n=0;n<3;n++)jawab5+=berita[n];
        for(n=0;n<3;n++)jawab6+=berita[n];
        for(n=0;n<3;n++)jawab7+=berita[n];
        for(n=0;n<3;n++)jawab8+=berita[n];
        for(n=0;n<3;n++)jawab9+=berita[n];
        for(n=0;n<3;n++)jawab10+=berita[n];
        for(n=0;n<3;n++)jawab11+=berita[n];
         for(n=0;n<3;n++)jawab12+=berita[n];
        for(n=0;n<3;n++)jawab13+=berita[n];
        for(n=0;n<3;n++)jawab14+=berita[n];
        for(n=0;n<3;n++)jawab15+=berita[n];
        for(n=0;n<3;n++)jawab16+=berita[n];
        for(n=0;n<3;n++)jawab17+=berita[n];
        for(n=0;n<3;n++)jawab18+=berita[n];
        for(n=0;n<3;n++)jawab19+=berita[n];
         for(n=0;n<3;n++)jawab20+=berita[n];
        for(n=0;n<3;n++)jawab21+=berita[n];
        for(n=0;n<3;n++)jawab22+=berita[n];
        for(n=0;n<3;n++)jawab23+=berita[n];
```

```
        for(n=0;n<3;n++)jawab24+=berita[n];
        for(n=0;n<3;n++)jawab25+=berita[n];
        for(n=0;n<3;n++)jawab26+=berita[n];
        for(n=0;n<3;n++)jawab27+=berita[n];
         for(n=0;n<3;n++)jawab28+=berita[n];
        for(n=0;n<3;n++)jawab29+=berita[n];
        for(n=0;n<3;n++)jawab30+=berita[n];
        for(n=0;n<3;n++)jawab31+=berita[n];
        for(n=0;n<3;n++)jawab32+=berita[n];
        for(n=0;n<3;n++)jawab33+=berita[n];
        for(n=0;n<3;n++)jawab34+=berita[n];
        for(n=0;n<3;n++)jawab35+=berita[n];


jawab == kunci ? rak_A1():
jawab1== kunci1? rak_A2():
jawab2==kunci2? rak_A3():
jawab3==kunci3? rak_A4():
jawab4== kunci4? rak_A5():
jawab5== kunci5? rak_A6():
jawab6==kunci6? rak_A7():
jawab7==kunci7? rak_A8():
jawab8== kunci8 ? rak_A9():

jawab9== kunci9? rak_B1():
jawab10==kunci10? rak_B2():
jawab11==kunci11? rak_B3():
jawab12 == kunci12 ? rak_B4():
jawab13== kunci13? rak_B5():
jawab14==kunci14? rak_B6():
jawab15==kunci15? rak_B7():
jawab16== kunci16? rak_B8():
jawab17== kunci17? rak_B9():
```

```
jawab18==kunci18? rak_C1():
jawab19==kunci19? rak_C2():
jawab20== kunci20? rak_C3():
jawab21== kunci21? rak_C4():
jawab22==kunci22? rak_C5():
jawab23==kunci23? rak_C6():
jawab24== kunci24? rak_C7():
jawab25== kunci25? rak_C8():
jawab26==kunci26? rak_C9():

jawab27==kunci27? rak_D1():
jawab28== kunci28? rak_D2():
jawab29== kunci29? rak_D3():
jawab30==kunci30? rak_D4():
jawab31==kunci31? rak_D5():
jawab32== kunci32 ? rak_D6():
jawab33== kunci33? rak_D7():
jawab34==kunci34? rak_D8():
jawab35==kunci35? rak_D9(): software_reset();

}if (digitalRead (SMA) == HIGH){goto manual;}
else{lcd.clear(); delay (2000); software_reset();}}
```

Gambar 4.4 *Listing* Program Menampilkan karakter pada LCD yang sesuai dengan penekanan *keypad*



(a)



(c)



(b)



(d)

Gambar 4.5. Pengujian langsung *keypad* dan LCD

Keterangan:
Gambar 4.5. (a) : tampilan LCD saat terjadi penekanan pada *keypad* A-0-1-#
Gambar 4.5. (b) : tampilan LCD saat terjadi penekanan pada *keypad* B-0-5-#
Gambar 4.5. (c) : tampilan LCD saat terjadi penekanan pada *keypad* C-0-9-#
Gambar 4.5. (d) : tampilan LCD saat terjadi penekanan pada *keypad* D-0-7-#

Tabel 4.3 Keterangan Pengalamatan Barang yang ditampilkan oleh LCD sesuai penekanan *Keypad* dari *listing* program diatas.

Keypad	LCD	Keterangan
A – 0 – 1 – #	A01>	Simpan barang ke rak A no. 01
A – 0 – 2 – #	A02>	Simpan barang ke rak A no. 02
A – 0 – 3 – #	A03>	Simpan barang ke rak A no. 03
A – 0 – 4 – #	A04>	Simpan barang ke rak A no. 04
A – 0 – 5 – #	A05>	Simpan barang ke rak A no. 05
A – 0 – 6 – #	A06>	Simpan barang ke rak A no. 06
A – 0 – 7 – #	A07>	Simpan barang ke rak A no. 07
A – 0 – 8 – #	A08>	Simpan barang ke rak A no. 08
A – 0 – 9 – #	A09>	Simpan barang ke rak A no. 09
B – 0 – 1 – #	B01>	Simpan barang ke rak B no. 01
B – 0 – 2 – #	B02>	Simpan barang ke rak B no. 02
B – 0 – 3 – #	B03>	Simpan barang ke rak B no. 03
B – 0 – 4 – #	B04>	Simpan barang ke rak B no. 04
B – 0 – 5 – #	B05>	Simpan barang ke rak B no. 05
B – 0 – 6 – #	B06>	Simpan barang ke rak B no. 06
B – 0 – 7 – #	B07>	Simpan barang ke rak B no. 07
B – 0 – 8 – #	B08>	Simpan barang ke rak B no. 08
B – 0 – 9 – #	B09>	Simpan barang ke rak B no. 09
C – 0 – 1 – #	C01>	Simpan barang ke rak C no. 01
C – 0 – 2 – #	C02>	Simpan barang ke rak C no. 02
C – 0 – 3 – #	C03>	Simpan barang ke rak C no. 03
C – 0 – 4 – #	C04>	Simpan barang ke rak C no. 04
C – 0 – 5 – #	C05>	Simpan barang ke rak C no. 05
C – 0 – 6 – #	C06>	Simpan barang ke rak C no. 06
C – 0 – 7 – #	C07>	Simpan barang ke rak C no. 07
C – 0 – 8 – #	C08>	Simpan barang ke rak C no. 08
C – 0 – 9 – #	C09>	Simpan barang ke rak C no. 09
D – 0 – 1 – #	D01>	Simpan barang ke rak D no. 01
D – 0 – 2 – #	D02>	Simpan barang ke rak D no. 02
D – 0 – 3 – #	D03>	Simpan barang ke rak D no. 03
D – 0 – 4 – #	D04>	Simpan barang ke rak D no. 04
D – 0 – 5 – #	D05>	Simpan barang ke rak D no. 05
D – 0 – 6 – #	D06>	Simpan barang ke rak D no. 06
D – 0 – 7 – #	D07>	Simpan barang ke rak D no. 07
D – 0 – 8 – #	D08>	Simpan barang ke rak D no. 08
D – 0 – 9 – #	D09>	Simpan barang ke rak D no. 09

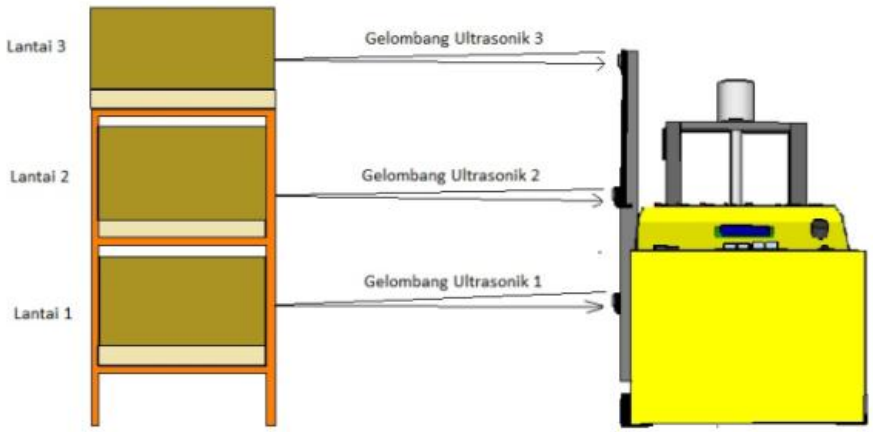
Analisis:

a) Dari hasil pengujian *keypad* dan LCD diketahui ketika terjadi penekanan pada *keypad* maka LCD akan menampilkan karakter sesuai dengan karakter *keypad* yang ditekan. Jika kode rak tidak sesuai seperti pada tabel 4.3 diatas maka sistem akan *restart* kembali ke kondisi awal.

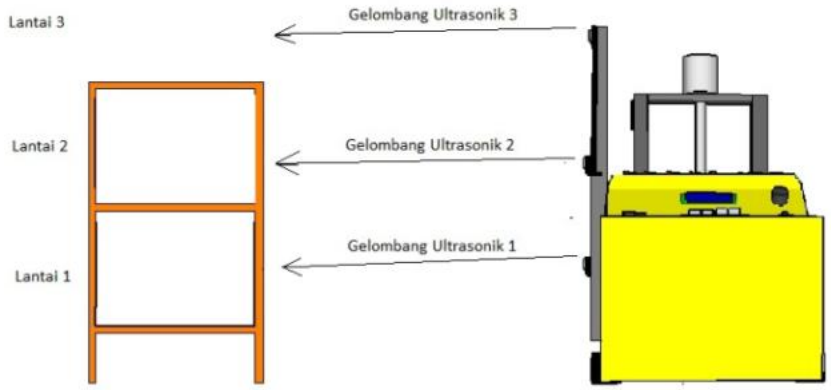
b) Berdasarkan hasil pengujian, *keypad* mampu menjadi masukan pengalamatan penyimpanan barang sedangkan LCD mampu menampilkan alamat sesuai dengan penekanan *keypad* sehingga dapat disimpulkan *keypad* dan LCD bekerja dengan baik sesuai dengan harapan.

4.4. Pengujian dan Analisis Sensor Ultrasonik HCSR-04

Sensor Ultrasonik HCSR-04 adalah sensor jarak berbasis gelombang ultrasonik. Pengujian sensor ultrasonik pada perancangan ini bertujuan untuk memperoleh jarak robot ke rak ketika kondisi kosong dan berisi barang. *Setup* pengukuran jarak robot ke rak kosong dan rak isi ditunjukan sebagai berikut.



(a)



(b)

Gambar 4.6. *Setup* Pengukuran Jarak Robot ke Rak

Keterangan:
Gambar 4.6. (a) : *setup* pengukuran jarak robot ke rak isi
Gambar 4.6. (b) : *setup* pengukuran jarak robot ke rak kosong

Dari gambar 4.6. (a) masing-masing gelombang ultrasonik yang dipancarkan terpantulkan kembali ke masing-masing sensor ultrasonik oleh barang. Pada gambar 4.6. (b) ini pendeteksian rak kosong dimungkinkan gelombang ultrasonik terpantul ke bagian celah-celah tengah rak sehingga selisih jarak rak ketika kondisi kosong dan terisi barang tidak terlalu jauh. Berikut tabel hasil pengukuran jarak robot ke rak kosong dan rak isi.

Tabel 4.4 Pengukuran Menggunakan Sensor Ultrasonik HCSR-04

Rak Lantai	**Kondisi Rak Kosong (cm)**	**Kondisi Rak isi (cm)**
1	47 sampai 48	45 sampai 46
2	48 sampai 49	45 sampai 46
3	49 sampai 51	46 sampai 47

Dari jarak diatas dapat di cari waktu tempuh gelombang ultrasonik saat dipancarkan sampai diterima kembali menggunakan rumus berikut.

$$S = \frac{v.t}{2}$$

$t = \frac{S.2}{v}$

dimana v = 344 m/s dan S = jarak pada tabel 4.4 diatas. Sehingga waktu tempuh gelombang ultrasonik untuk jarak pada tabel 4.4 diatas dapat diketahui seperti pada tabel berikut.

Tabel 4.5. Perhitungan Waktu Tempuh Gelombang Ultrasonik

Jarak (S / m)	Perhitungan	Waktu tempuh gelombang ultrasonik (t)
0,45	$t = \frac{0{,}45\ m.2}{344\ {}^{m}/_{s}}$	$0{,}00261\ s$
0,46	$t = \frac{0{,}46\ m.2}{344\ {}^{m}/_{s}}$	$0{,}00267\ s$
0,47	$t = \frac{0{,}47\ m.2}{344\ {}^{m}/_{s}}$	$0{,}00273\ s$
0,48	$t = \frac{0{,}48\ m.2}{344\ {}^{m}/_{s}}$	$0{,}00279\ s$
0,49	$t = \frac{0{,}49\ m.2}{344\ {}^{m}/_{s}}$	$0{,}00284\ s$
0,50	$t = \frac{0{,}50\ m.2}{344\ {}^{m}/_{s}}$	$0{,}00290\ s$
0,51	$t = \frac{0{,}51\ m.2}{344\ {}^{m}/_{s}}$	$0{,}00296\ s$

Analisis:

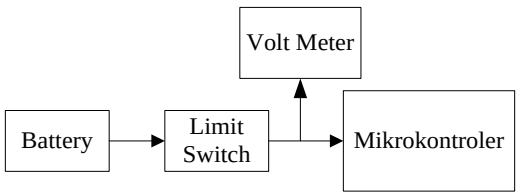
a) Hasil dari pengukuran jarak dan perhitungan waktu tempuh gelombang ultrasonik menggunakan sensor HCSR-04 didapatkan bahwa jarak yang di dapat tergantung dari waktu tempuh gelombang ultrasonik, semakin lama waktu tempuh berarti jarak semakin jauh.

b) Jarak dari tabel 4.4 didapatkan dari pengujian sensor ultrasonik menggunakan *setup* pengukuran seperti pada gambar 4.6. (a) dan (b) dan hasil jarak yang didapat ditampilkan pada LCD. Dari tabel 4.4 didapatkan sensor jarak ultrasonik mendeteksi rak kosong dengan jarak lebih jauh

dibanding dengan rak ketikan kondisi terisi barang. tetapi selisih jarak ketika kondisi rak kosong dan ketika kondisi rak terisi barang terlihat tidak terlalu jauh. Hal tersebut dikarenakan gelombang ultrasonik yang di pancarkan mengenai celah-celah bagian tengah rak sehingga sensor akan mendeteksi rak yang kosong tersebut sebagai jarak dari kedudukan robot ke bagian celah-celah rak.

4.5. Pengujian dan Analisis *Limit Switch*

Pengujian sensor pembatas *limit switch* ini bertujuan untuk mendapatkan gambaran kinerja sensor sebagai pembatas pergerakan garpuh pada saat kondisi garpu maju, mundur, naik dan turun serta posisi garpu pada saat melakukan pengambilan barang, penyimpanan barang pada rak lantai 1, rak lantai 2 atau rak lantai 3. *Setup* pengukuran *limit switch* seperti yang ditunjukan gambar berikut.



Gambar 4.7. *Setup* Pengukuran *Limit Switch*



(a)



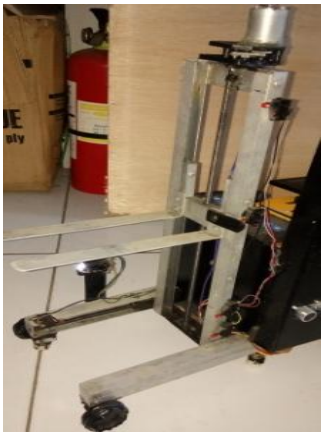
(b)



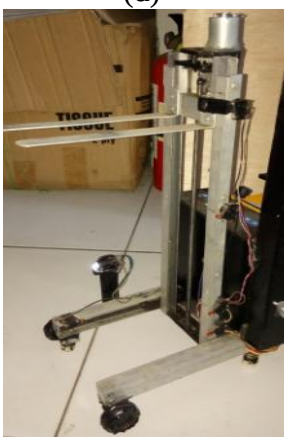
(c)



(d)



(e)



(f)

Gambar 4.8. Pengujian *Limit Switch*

Keterangan:
Gambar 4.8. (a) : kondisi *limit Switch* belakang aktif
Gambar 4.8. (b) : kondisi *limit Switch* depan aktif
Gambar 4.8. (c) : kondisi *limit Switch* bawah aktif
Gambar 4.8. (d) : kondisi *limit Switch* lantai 1 aktif
Gambar 4.8. (e) : kondisi *limit Switch* lantai 2 aktif
Gambar 4.8. (f) : kondisi *limit Switch* lantai 3 aktif

Adapun hasil pengujiannya dinyatakan dalam tabel berikut.

Tabel 4.6. Pengujian *Limit Switch* Pembatas Maju dan Mundur Garpu

Posisi Garpuh	**Tengangan Keluaran (V)**		**Status *Limit Switch***	
	***Limit Switch* Depan**	***Limit Switch* Belakang**	**Depan**	**Belakang**
Depan	4,71	0	Aktif	Tidak
Belakang	0	4,68	Tidak	Aktif

Tabel 4.7. Pengujian *Limit Switch* Pembatas Lantai rak

Posisi Garpuh	**Tegangan Keluaran (V)**				**Status *Limit Switch***			
	***Limit Switch* Bawah**	***Limit Switch* Lantai1**	***Limit Switch* Lantai2**	***Limit Switch* Lantai3**	**Bawah**	**Lantai1**	**Lantai2**	**Lantai3**
Bawah	4,70	0	0	0	Aktif	Tidak	Tidak	Tidak
Lantai1	0	4,71	0	0	Tidak	Aktif	Tidak	Tidak
Lantai2	0	0	4,68	0	Tidak	Tidak	Aktif	Tidak
Lantai3	0	0	0	4,70	Tidak	Tidak	Tidak	Aktif

Analisis:

1. Dari tabel 4.6 dan gambar 4.8.(a) dan (b) dapat dilihat bahwa apabila garpu berada di posisi depan maka tegangan keluaran dari *limit Switch* depan sebagai masukan ke mikrokontorler sebesar 4,71V sedangkan tegangan keluaran *limit switch* belakang sebesar 0V, ini artinya *limit switch* depan aktif dan *limit switch* belakang tidak aktif atau posisi garpu berada di depan. Sebaliknya apabila garpu berada di posisi belakang maka tegangan keluaran dari *limit Switch* belakang sebagai masukan ke mikrokontroler sebesar 4,68V dan tegangan keluaran *limit switch* depan sebesar 0V, ini artinya *limit switch* belakang aktif dan *limit switch* depan tidak aktif atau posisi garpu berada dibelakang.
2. Pada tabel 4.7 dan gambar 4.8. (c), (d), (e) dan (f) saat garpu berada di posisi paling bawah maka tegangan keluaran dari *limit Switch* bawah sebagai masukan ke mikrokontroler sebsesar 4,70V dan tegangan keluaran *limit switch* lantai 1, lantai 2 dan lantai 3 bernilai 0V, artinya *limit switch* bawah aktif, *limit switch* lantai 1, *limit switch* lantai 2 dan *limit switch* lantai 3 tidak aktif. apabila garpu berada di posisi lantai 1 maka tegangan keluaran dari *limit Switch* lantai 1 sebagai *inputan* ke mikrokontorler adalah 4,71V dan

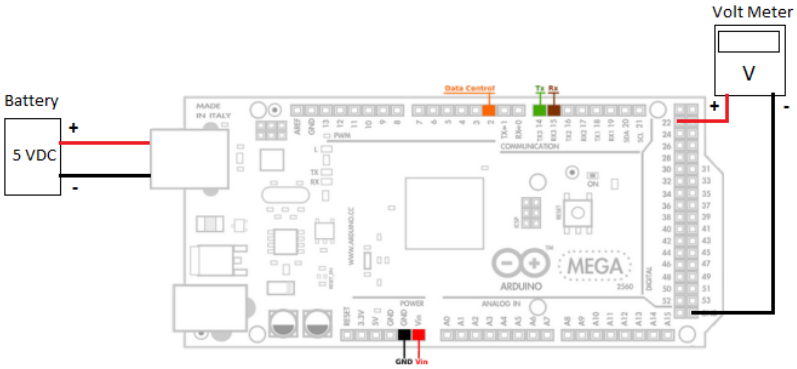
tegangan keluaran *limit switch* bawah , *limit switch* lantai 2 dan *limit switch* lantai 3 sebesar 0V, artinya *limit switch* lantai 1 aktif, *limit switch* bawah, *limit switch* lantai 2 dan *limit switch* lantai 3 tidak aktif. Jika garpu berada di posisi lantai 2 maka tegangan keluaran dari *limit Switch* lantai 2 sebagai masukan ke mikrokontorler adalaha 4,68V dan tegangan keluaran *limit switch* bawah , *limit switch* lantai 1 dan *limit switch* lantai 3 sebesar 0V, artinya *limit switch* lantai 2 aktif, *limit switch* bawah, *limit switch* lantai 1 dan *limit switch* lantai 3 tidak aktif. Kemudian Jika garpu berada di posisi lantai 3 maka tegangan keluaran dari *limit Switch* lantai 3 sebagai masukan ke mikrokontorler adalah 4,70V dan tegangan keluaran *limit switch* bawah , *limit switch* lantai 1 dan *limit switch* lantai 2 sebesar 0V, yang artinya *limit switch* lantai 3 aktif, *limit switch* bawah, *limit switch* lantai 1 dan *limit switch* lantai 2 tidak aktif.

3. Kondisi aktif dan tidak aktif dari tabel pengukuran dan pengujian diatas mewakili keberadaan posisi garpu. Ketika *limit Switch* aktif artinya garpuh berada pada posisi *limit switch* tersebut.

4.6. Pengujian dan Analisis Arduino Mega2560

Tujuan dari pengujian papan mikrokontoroler arduino mega 2560 adalah untuk mengetahui apakah keluaran papan mikrokontoroler arduino mega 2560 mampu untuk menggerakan motor DC. Pengujian dilakukan dengan memberi tegangan 5V dan ground melalui konektor printer yang ada pada papan mikrokontroler arduino mega 2560 . Program yang dimasukan kedalam papan mikrokontroler arduino mega 2560 adalah program dalam bahasa *basic* dengan menggunakan program Arduino 1.0 dimana terdapat 2 perintah yang digunakan

untuk mengeluarkan logika “1” dan logika “0” pada *port* I/O mikrokontroler, yaitu perintah *digitalWrite (Pin, HIGH)* digunakan untuk mengeluarkan logika “1” atau *high* dan *digitalWrite (Pin, LOW)* digunakan untuk mengeluarkan logika “0”. Berikut setup pengukuran pada papan mikrokontroler arduino mega 2560



Gambar 4.9. Setup Pengukuran Papan Arduino Mega 2560

Meggunakan *setup* pengukuran seperti pada gambar 4.9 untuk mengetahui tegangan keluaran papan arduino mega 2560 saat berlogika *high* untuk menggerakan motor dan *low* untuk menghentikan gerakan motor maka didapat hasil pengukuran seperti pada tabel berikut ini.

Tabel 4.8. Pengukuran tegangan *output* Papan Arduino Mega 2560

Tegangan *input* Arduino Mega 2560 (V)	**Tegangan *Output* (V)**	
	Ketika Berlogika *High*	**Ketika Berlogika *Low***
5 V	4,87 sampai 4,91	0,00 sampai 0,01

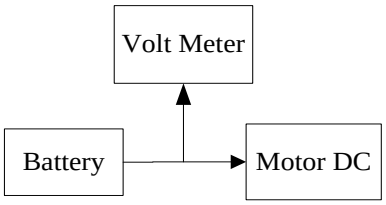
Analisis

Dari hasil pengujian yang dilakukan dapat dilihat bahwa tegangan saat logika “1” pada pin mikrokontroler berkisar antara 4,87V-4,91V dan saat logika”0” tegangan pada pin mikrokontroler diperoleh sebesar 0,00V-0,001V. Sehingga motor akan bergerak saat pin mikrokontroler berlogika “1” dan akan berhenti saat berlogika “0”. Karena itu, merujuk pada kondisi seperti tersebut diatas , maka

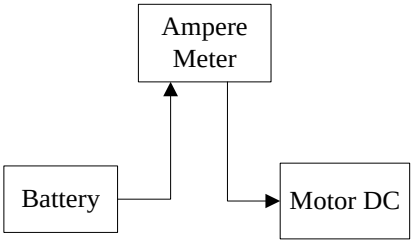
dapat dinyatakan bahwa papan mikrokontroler arduino mega 2560 bekerja dengan baik.

4.7. Pengujian dan Analisis Motor DC

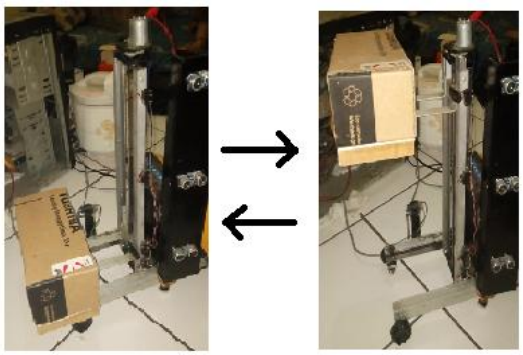
Tujuan pengujian motor DC ini untuk mengetahui nilai tegangan dan arus saat motor bergerak. Pada percobaan ini akan diuji 4 motor dc yang digunakan untuk maju dan mudur garpu, naik dan turun garpu, dan 2 motor untuk pergerakan robot maju, mundur dan berbelok. Berikut *setup* pengukuran motor DC.



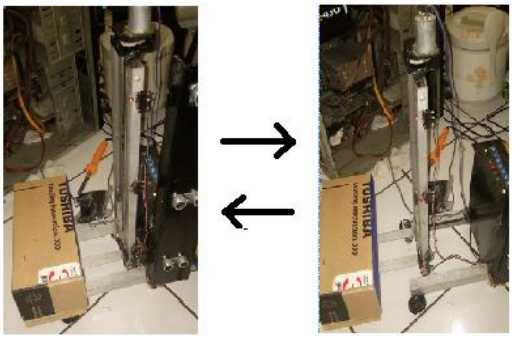
Gambar 4.10. *Setup* Pengukuran Tegangan Motor DC



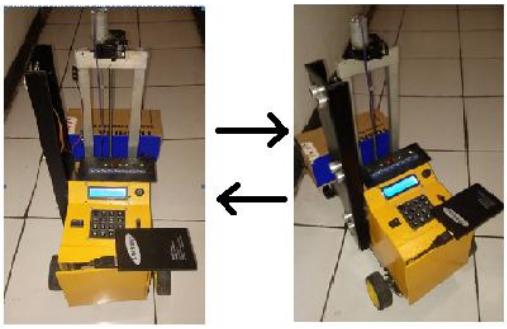
Gambar 4.11 *Setup* Pengukuran Arus Motor DC



(a)



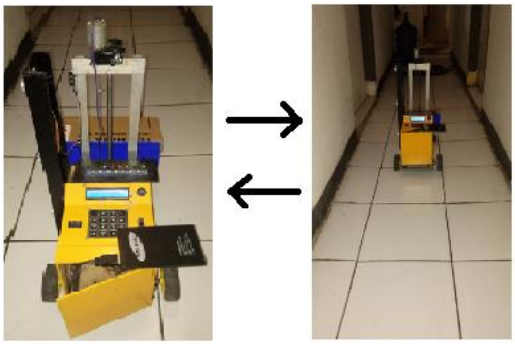
(b)



(c)



(d)



(e)

Gambar 4.12. Pengujian Motor DC

Keterangan:
Gamabar 4.12. (a) : kondisi motor menggerakan garpu maju dan mundur

Gamabar 4.12. (b) : kondisi motor menggerakan garpu naik dan turun
Gamabar 4.12. (c) : kondisi motor kiri penggerakan robot mundur dan maju
Gamabar 4.12. (d) : kondisi motor kanan penggerakan robot mundur dan maju
Gamabar 4.12. (e) : kondisi motor kiri dan kanan penggerakan robot maju dan Mundur.



Gambar 4.13. Pemasangan Roda Kiri Robot Kurang Baik

Dengan *setup* pengukuran pada gambar 4.10 dan gambar 4.11 dengan berat robot 850 gram ditambah 150 gram beban maksimal barang yang dibawa dan tegangan masukan motor 18 V, maka didapat hasil pengukuran seperti pada tabel berikut.

Tabel 4.9. Pengukuran Tegangan dan Arus Motor DC

Motor	Kondisi	Berat Beban (gram)	Tegangan (V)	Arus (mA)	Daya (W) P=V.I
Kiri	**Maju**	3000	16,56	47	0,77
	Mundur		16,75	34	0,56
Kanan	**Maju**		16,92	34	0,57
	Mundur		16,94	33	0,55
Kiri dan Kanan	**Kiri Maju**		16,66	41	0,68
	Kanan Maju		17,13	28	0,47
	Kiri Mundur		16,68	31	0,51
	Kanan Mundur		17,24	25	0,40
Garpuh	**Maju**	150	16,35	103	1,68
	Mundur		16,43	100	1,64
	Naik		16,90	93	1,57
	Turun		17,15	53	0,90

Analisis:

a) Dari hasil pengujian motor DC penggerak garpu naik, turun, maju dan mundur maka diketahui untuk keamanan motor DC beban maksimal barang yang dapat dibawa robot seberat 150 gram, jika berat barang mencapai 250

gram motor DC penggerak garpu naik, turun mampu menaikan dan menurunkan barang tetapi motor DC penggerak maju dan mundur garpu tidak bergerak dengan baik saat mendorong barang ke rak 3 dikarenakan dengan beban 250 gram tiang garpuh condong kedepan yang mengakibatkan gesekan tiang garpuh dan rell tidak sempurna sehingga terjadi hentakan – hentakan dan tertahannya gerakan motor DC saat garpu bergerak maju.

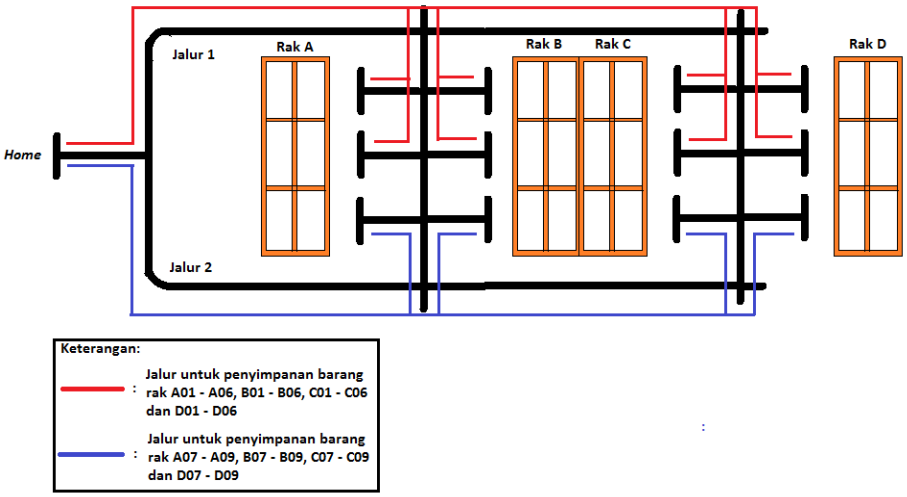
b) Dari table 4.9 diketahui jika konsumsi daya motor kiri dan kanan berbeda cukup jauh, motor kiri mengkonsumsi daya lebih besar dibandingkan motor kanan hal tersebut dimungkinkan karena pemasangan motor roda kiri kurang baik seperti ditunjukan pada gambar 4.13 sehingga pergerakan motor kiri lebih berat dan memerlukan daya yang lebih besar dibandingkan motor kanan.

4.8. Pengkondisian Jalur Penyimpanan Barang

Agar robot *forklift* mampu menyimpan barang sesuai pada tempatnya pada saat berjalan secara manual maupun secara otomatis, maka dilakukan pengkondisian jalur utama ketika jalur masih utuh atupun jalur alternatif ketika terjadi kerusakan pada jalur utama untuk melakukan penyimpanan barang dan kembali ke *home.*

4.8.1. Jalur Utama

Untuk kondisi jalur utama penyimpanan barang ketika jalur masih utuh maka pengkondisian jalur seperti pada gambar 4.14 berikut.



Gambar 4.14. Pengkondisian Jalur Utama

Berdasarkan hasil pengujian, waktu tempuh robot dari *home* ke rak tujuan tanpa menghitung waktu pengambilan dan penyimpanan barang dimana waktu penyimpanan sama dengan waktu kembali ke *home* ditunjukan seperti pada tabel berikut ini.

Tabel 4.10. Waktu Tempuh Robot Ketika bekerja dengan jalur utama

Tanda Jalur	**Rak Tujuan**	**Waktu total (detik)**
Merah	A01 sampai A03 B01 sampai B03	140
	A04 sampai A06 B04 sampai B06	150
	C01 sampai C03 D01 sampai D03	180
	C04 sampai C06 D04 sampai D06	190
Biru	A07 sampai A09 B07 sampai B09	140
	C07 sampai C09 D07 sampai D09	180

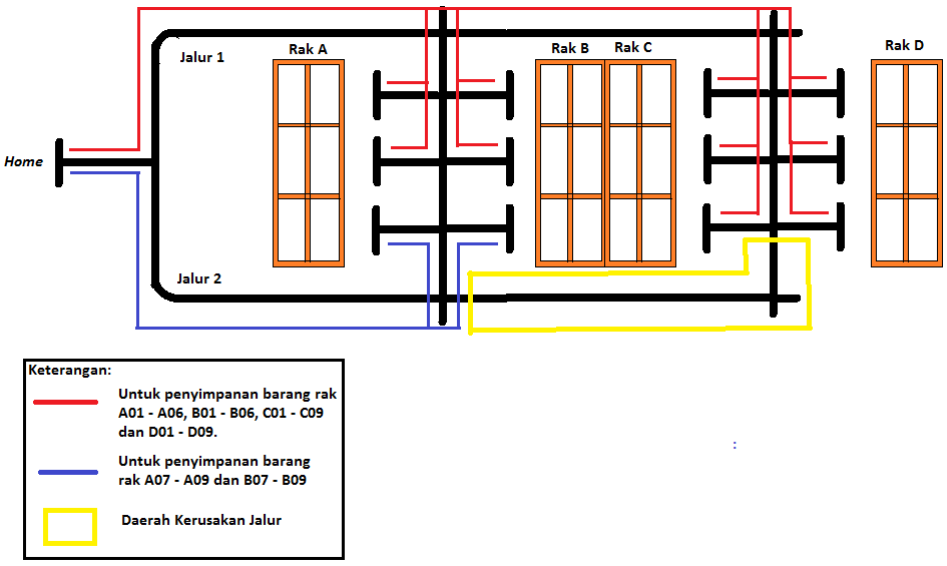
Pada gambar 4.14 garis merah menunjukan jalur penyimpanan barang dan kembali ke *home* untuk alamat rak A01 sampai rak A06, B01 sampai B06, C01 sampai C06 dan D01 sampai D06. Sedangkan garis biru menunjukan jalur

penyimpanan barang dan kembali ke *home* untuk alamat rak A07 sampai rak A09, B07 sampai B09, C07 sampai C09 dan D07 sampai D09. Pergerakan robot saat bergerak pada jalur yang di tandai garis merah, pertama robot mengambil barang didaerah *home* kemudian mundur sampai simpangan pertama kemudian belok kanan lalu bergerak mengikuti garis jika tujuan rak A01 sampai A06 atau B01 sampai B06 di persimpangan kedua robot akan belok kanan kemudian mengukuti garis, jika tujuan rak A robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Jika tujuan rak B robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. tujuan rak C01 sampai C06 atau D01 sampai D06 di persimpangan ketiga robot akan belok kanan kemudian maju mengukuti garis, jika tujuan rak C robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Jika tujuan rak D robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Setelah barang tersimpan robot akan kembali ke *home* dengan jalur yang sama seperti saat penyimpanan. Pergerakan robot saat bergerak pada jalur yang di tandai garis biru, pertama robot mengambil barang didaerah *home* kemudian mundur sampai simpangan pertama kemudian belok kiri lalu bergerak mengikuti garis jika tujuan rak A07 sampai A09 atau B07 sampai B09 di persimpangan kedua robot akan belok kiri kemudian mengukuti garis, jika tujuan rak A robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti

garis dan simpan barang saat berada di depan rak. Jika tujuan rak B robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. tujuan rak C07 sampai C09 atau D07 sampai D09 di persimpangan ketiga robot akan belok kiri kemudian maju mengukuti garis, jika tujuan rak C robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Jika tujuan rak D robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Setelah barang tersimpan robot akan kembali ke *home* dengan jalur yang sama seperti saat penyimpanan.

4.8.2. Jalur Alternatif 1

Jalur alternatif 1 digunakan apabila terjadi kerusakan pada jalur 2 seperti pada gambar 4.15 dibawah ini maka pengkondisian jalur ditunjukan seperti pada gambar 4.15 berikut.



Gambar 4.15. Pengkondisian Jalur Alternatif 1

Berdasarkan hasil pengujian, waktu tempuh robot dari *home* ke rak tujuan tanpa menghitung waktu pengambilan dan penyimpanan barang dimana waktu penyimpanan sama dengan waktu kembali ke *home* ditunjukan seperti pada tabel berikut ini.

Tabel 4.11. Waktu Tempuh Robot Ketika bekerja dengan jalur Alternatif 1

Tanda Jalur	**Rak Tujuan**	**Waktu total (detik)**
Merah	A01 sampai A03 B01 sampai B03	140
	A04 sampai A06 B04 sampai B06	150
	C01 sampai C03 D01 sampai D03	180
	C04 sampai C06 D04 sampai D06	190
	C07 sampai C09 D07 sampai D09	200
Biru	A07 sampai A09 B07 sampai B09	140

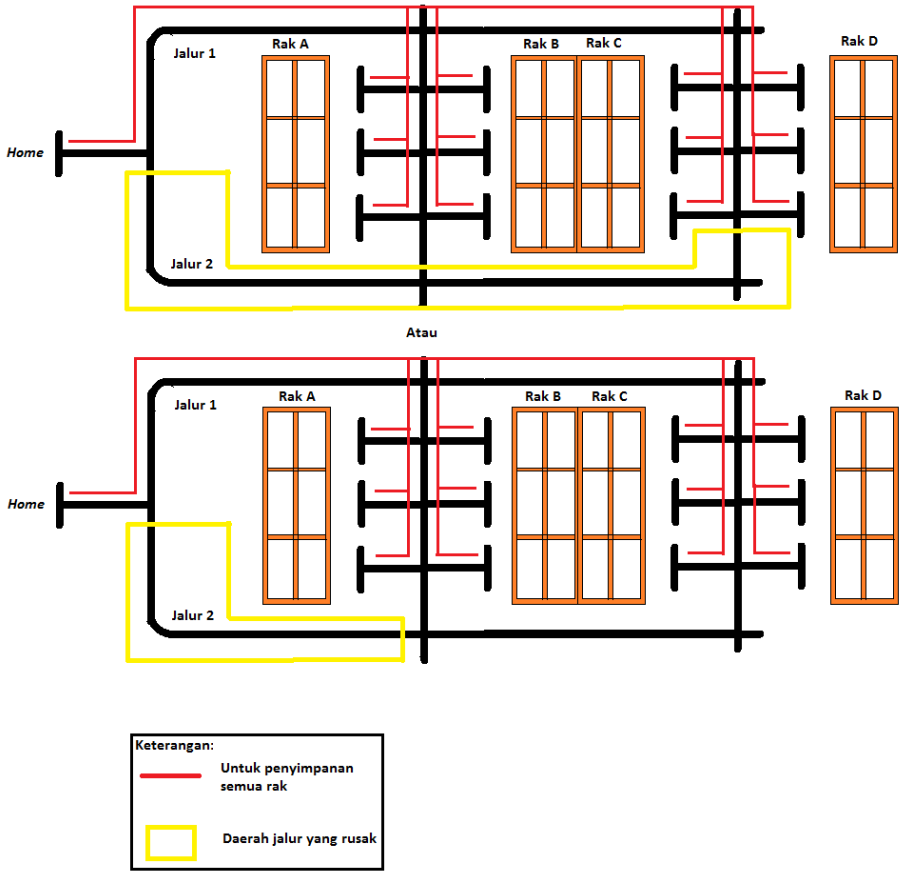
Apabila terjadi kerusakan jalur pada daerah kuning seperti pada gambar 4.15 diatas maka untuk alamat rak A01 sampai rak A06, B01 sampai B06, C01 sampai C09 dan D01 sampai D09 ditunjukan oleh garis merah dan untuk alamat rak A07 sampai rak A09 dan B07 sampai B09 ditunjukan oleh garis biru. Pergerakan robot saat bergerak pada jalur yang di tandai garis merah, pertama robot mengambil barang didaerah *home* kemudian mundur sampai simpangan pertama kemudian belok kanan lalu bergerak mengikuti garis jika tujuan rak A01 sampai A06 atau B01 sampai B06 di persimpangan kedua robot akan belok kanan kemudian mengukuti garis, jika tujuan rak A robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan

simpan barang saat berada di depan rak. Jika tujuan rak B robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. untuk tujuan rak C01 sampai C09 atau D01 sampai D09 di persimpangan ketiga robot akan belok kanan kemudian maju mengukuti garis, jika tujuan rak C robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Jika tujuan rak D robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Setelah barang tersimpan robot akan kembali ke *home* dengan jalur yang sama seperti saat penyimpanan. Pergerakan robot saat bergerak pada jalur yang di tandai garis biru, pertama robot mengambil barang didaerah *home* kemudian mundur sampai simpangan pertama kemudian belok kiri lalu bergerak mengikuti garis jika tujuan rak A07 sampai A09 atau B07 sampai B09 di persimpangan kedua robot akan belok kiri kemudian mengukuti garis, jika tujuan rak A robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Jika tujuan rak B robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Setelah barang tersimpan robot akan kembali ke *home* dengan jalur yang sama seperti saat penyimpanan. Dari tabel 4.11 terlihat jika waktu tempuh robot saat menyimpan ke rak dan kembali dari rak C07 sampai C09 dan rak D07 sampai D09 dengan menggunakan jalur alternatif 1 lebih lama 20 detik daripada saat menggunakan jalur utama dimana saat menggunakan jalur utama adalah 180 detik dan saat

menggunakan jalur alternatif 1 200 detik. Hal tersebut dikarenakan jarak tempuh robot lebih jauh saat mggunakan jalur alternatif 1 dibanding saat menggunakan jalur utama.

4.8.3. Jalur Alternatif 2

Jalur alternatif 2 digunakan jika terjadi kerusakan jalur 2 seperti gambar 4.16 maka pengkondisisan jalur ditunjukan seperti pada gambar 4.16 berikut.



Gambar 4.16. Pengkondisian Jalur Alternatif 2

Berdasarkan hasil pengujian, waktu tempuh robot dari *home* ke rak tujuan tanpa menghitung waktu pengambilan dan penyimpanan barang dimana waktu penyimpanan sama dengan waktu kembali ke *home* ditunjukan seperti pada tabel 4.12 berikut ini.

Tabel 4.12. Waktu Tempuh Robot Ketika bekerja dengan jalur Alternatif 2

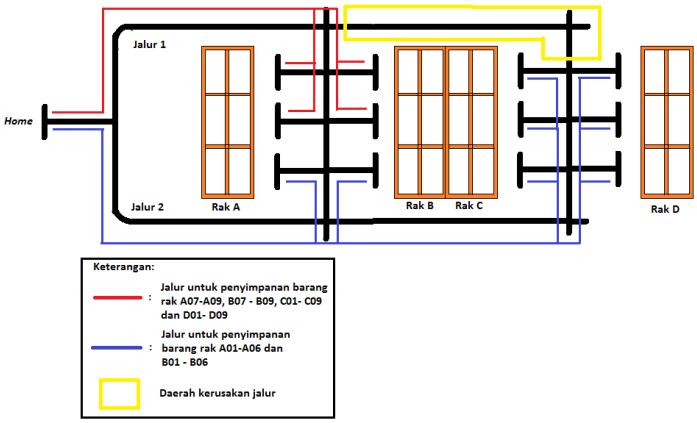
Tanda Jalur	**Rak Tujuan**	**Waktu total (detik)**
Merah	A01 sampai A03 B01 sampai B03	140
	A04 sampai A06 B04 sampai B06	150
	C01 sampai C03 D01 sampai D03	180
	C04 sampai C06 D04 sampai D06	190
	C07 sampai C09 D07 sampai D09	200
	A07 sampai A09 B07 sampai B09	160

Apabila terjadi kerusakan jalur pada daerah kuning seperti pada gambar 4.16 diatas maka semua jalur penyimpanan barang ditunjukan oleh garis merah. Pergerakan robot saat bergerak pada jalur yang di tandai garis merah, pertama robot mengambil barang didaerah *home* kemudian mundur sampai simpangan pertama kemudian belok kanan lalu bergerak mengikuti garis jika tujuan rak A01 sampai A09 atau B01 sampai B09 di persimpangan kedua robot akan belok kanan kemudian mengukuti garis, jika tujuan rak A robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Jika tujuan rak B robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. untuk tujuan rak C01 sampai C09 atau D01 sampai D09 di persimpangan ketiga robot akan belok kanan kemudian maju mengukuti garis, jika tujuan rak C robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti

garis dan simpan barang saat berada di depan rak. Jika tujuan rak D robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Setelah barang tersimpan robot akan kembali ke *home* dengan jalur yang sama seperti saat penyimpanan. Dari tabel 4.12 terlihat jika waktu tempuh robot saat menyimpan ke rak dan kembali dari rak C07 sampai C09, rak D07 sampai D09, rak C07 sampai C09 dan rak D07 sampai D09 dengan menggunakan jalur alternatif 2 lebih lama 20 detik daripada saat menggunakan jalur utama dimana saat menggunakan jalur utama untuk rak C07 sampai C09, rak D07 sampai D09 adalah 180 detik dan saat menggunakan jalur alternatif 200 detik, untuk rak A07 sampai A09, rak B07 sampai B09. Hal tersebut dikarenakan jarak tempuh robot lebih jauh saat mggunakan jalur alternatif 2 dibanding saat menggunakan jalur utama.

4.8.4. Jalur Alternatif 3

Jalur alternatif 3 digunakan jika terjadi kerusakan pada jalur 1 seperti pada gambar 4.17 dibawah ini maka pengkondisian jalur ditunjukan seperti pada gambar berikut.



Gambar 4.17. Pengkondisian Jalur Alternatif 3

Berdasarkan hasil pengujian, waktu tempuh robot dari *home* ke rak tujuan tanpa menghitung waktu pengambilan dan penyimpanan barang dimana waktu penyimpanan sama dengan waktu kembali ke *home* ditunjukan seperti pada tabel berikut ini.

Tabel 4.13. Waktu Tempuh Robot Ketika bekerja dengan jalur Alternatif 3

Tanda Jalur	**Rak Tujuan**	**Waktu total (detik)**
Merah	A01 sampai A03 B01 sampai B03	140
	A04 sampai A06 B04 sampai B06	150
Biru	C01 sampai C03 D01 sampai D03	200
	C04 sampai C06 D04 sampai D06	190
	A07 sampai A09 B07 sampai B09	140
	C07 sampai C09 D07 sampai D09	180

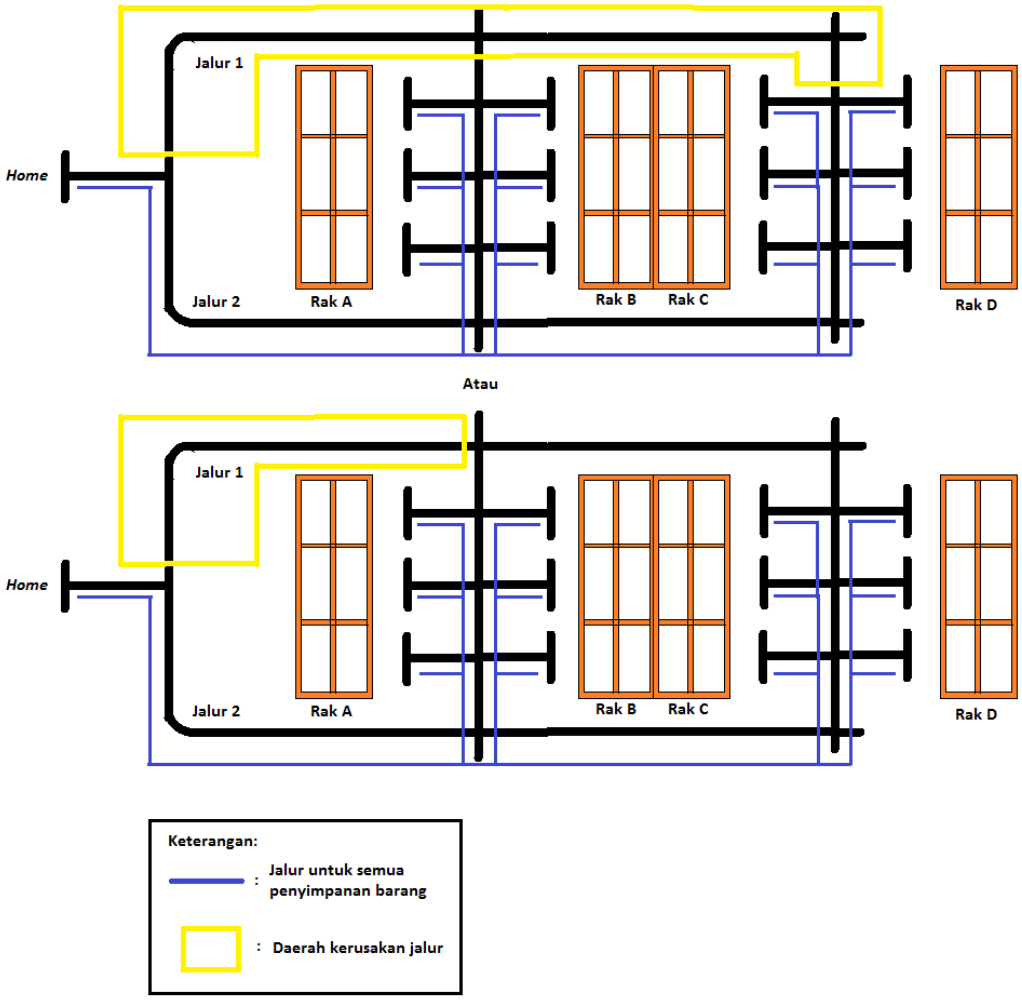
Apabila terjadi kerusakan jalur pada daerah kuning seperti pada gambar 4.17 diatas maka untuk alamat rak A01 sampai rak A06 dan B01 sampai B06 ditunjukan oleh garis merah dan untuk alamat rak A07 sampai rak A09, B07 sampai B09, C01 sampai C09 dan D01 sampai D09 ditunjukan oleh garis biru. Pergerakan robot saat bergerak pada jalur yang di tandai garis merah, pertama robot mengambil barang didaerah *home* kemudian mundur sampai simpangan pertama kemudian belok kanan lalu bergerak mengikuti garis jika tujuan rak A01 sampai A06 atau B01 sampai B06 di persimpangan kedua robot akan belok kanan kemudian mengukuti garis, jika tujuan rak A robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Jika tujuan rak B robot akan belok

kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Setelah barang tersimpan robot akan kembali ke *home* dengan jalur yang sama seperti saat penyimpanan. Pergerakan robot saat bergerak pada jalur yang di tandai garis biru, pertama robot mengambil barang didaerah *home* kemudian mundur sampai simpangan pertama kemudian belok kiri lalu bergerak mengikuti garis jika tujuan rak A07 sampai A09 atau B07 sampai B09 di persimpangan kedua robot akan belok kiri kemudian mengukuti garis, jika tujuan rak A robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Jika tujuan rak B robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. tujuan rak C01 sampai C09 atau D01 sampai D09 di persimpangan ketiga robot akan belok kiri kemudian maju mengukuti garis, jika tujuan rak C robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Jika tujuan rak D robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Setelah barang tersimpan robot akan kembali ke *home* dengan jalur yang sama seperti saat penyimpanan. Dari tabel 4.13 terlihat jika waktu tempuh robot saat menyimpan ke rak dan kembali dari rak C01 sampai C03 dan rak D01 sampai D03 dengan menggunakan jalur alternatif 3 lebih lama 20 detik daripada saat menggunakan jalur utama dimana saat menggunakan jalur utama adalah 180 detik dan saat menggunakan jalur alternatif 3 200 detik. Hal tersebut dikarenakan jarak tempuh

robot lebih jauh saat mggunakan jalur alternatif 3 dibanding saat menggunakan jalur utama.

4.8.5. Jalur Alternatif 4

Jalur alternatif 4 digunakan jika terjadi kerusakan jalur 1seperti gambar 4.18 dibawah ini maka pengkondisisan jalur ditunjukan seperti pada gambar berikut.



Gambar 4.18. Pengkondisian Jalur Alternafi 4

Berdasarkan hasil pengujian, waktu tempuh robot dari *home* ke rak tujuan tanpa menghitung waktu pengambilan dan penyimpanan barang dimana waktu

penyimpanan sama dengan waktu kembali ke *home* ditunjukan seperti pada tabel berikut ini.

Tabel 4.14. Waktu Tempuh Robot Ketika bekerja dengan jalur Alternatif 1

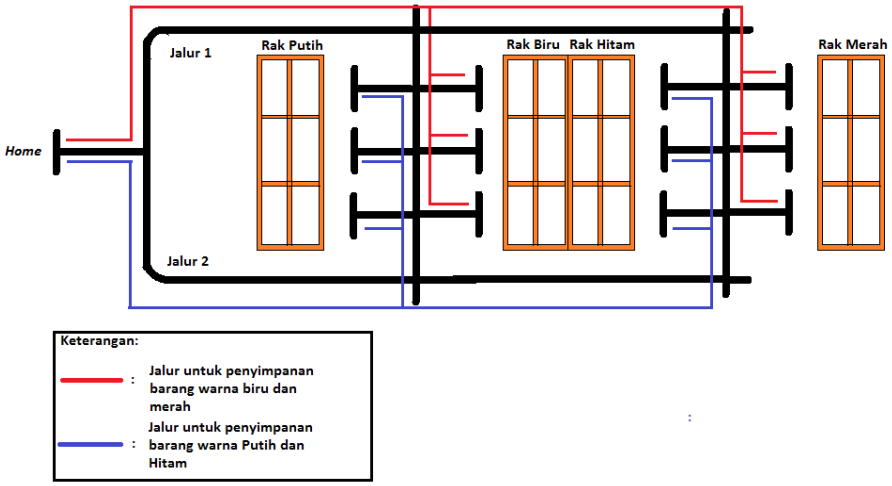
Tanda Jalur	**Rak Tujuan**	**Waktu total (detik)**
Biru	A01 sampai A03 B01 sampai B03	160
	A04 sampai A06 B04 sampai B06	150
	C01 sampai C03 D01 sampai D03	200
	C04 sampai C06 D04 sampai D06	190
	C07 sampai C09 D07 sampai D09	180
	A07 sampai A09 B07 sampai B09	140

Apabila terjadi kerusakan jalur pada daerah kuning seperti pada gambar 4.18 diatas maka semua jalur penyimpanan barang ditunjukan oleh garis biru. Pergerakan robot saat bergerak pada jalur yang di tandai garis biru, pertama robot mengambil barang didaerah *home* kemudian mundur sampai simpangan pertama kemudian belok kiri lalu bergerak mengikuti garis jika tujuan rak A01 sampai A09 atau B01 sampai B09 di persimpangan kedua robot akan belok kiri kemudian mengukuti garis, jika tujuan rak A robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Jika tujuan rak B robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. tujuan rak C07 sampai C01 atau D07 sampai D09 di persimpangan ketiga robot akan belok kiri kemudian maju mengukuti garis, jika tujuan rak C robot akan belok kiri saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti

garis dan simpan barang saat berada di depan rak. Jika tujuan rak D robot akan belok kanan saat berada di persimpangan yang sesuai dengan rak tujuan kemudian maju mengikuti garis dan simpan barang saat berada di depan rak. Setelah barang tersimpan robot akan kembali ke *home* dengan jalur yang sama seperti saat penyimpanan. Pengkondisian jalur robot untuk menyimpan barang apabila terjadi kerusakan jalur dilakukan dengan cara mengubah jumlah *Counter* dan arah belokan robot pada programnya. Dari tabel 4.14 terlihat jika waktu tempuh robot saat menyimpan ke rak dan kembali dari rak A01 sampai A03, rak B01 sampai B03, rak C01 sampai C03 dan rak D01 sampai D03 dengan menggunakan jalur alternatif 4 lebih lama 20 detik daripada saat menggunakan jalur utama dimana saat menggunakan jalur utama untuk rak C01sampai C03, rak D01 sampai D03 adalah 180 detik dan saat menggunakan jalur alternatif 200 detik, untuk rak A01 sampai A03, rak B01 sampai B03. Hal tersebut dikarenakan jarak tempuh robot lebih jauh saat mggunakan jalur alternatif 4 dibanding saat menggunakan jalur utama. Namun perubahan jalur robot digunakan hanya ketika jalur robot perlu perbaikan yang memakan waktu lama seperti adanya permukaan jalur yang tidak rata atau berlubang sehingga mengganggu laju robot dan adanya jalur yang hilang dibagian jalur yang dapat terlindas ban robot.

4.8.6. Jalur Robot Saat Bekerja Otomatis

Untuk pengkondisian jalur penyimpanan barang saat robot bekerja secara otomatis ditunjukan seperti pada gambar 4.19 berikut.



Gambar 4.19. Pengkondisian Jalur Saat Robot Bekerja Secara Otomatis

Berdasarkan hasil pengujian, waktu tempuh robot dari *home* ke rak tujuan tanpa menghitung waktu pengambilan dan penyimpanan barang dimana waktu penyimpanan sama dengan waktu kembali ke *home* ditunjukan seperti pada tabel berikut ini.

Tabel 4.15. Waktu Tempuh Robot Ketika bekerja Secara Otomatis

Tanda Jalur	**Rak Tujuan**	**Waktu total (detik)**
Biru	Putih 1 sampai Putih 3	140
	Putih 4 sampai Putih 6	150
	Putih 7 sampai Putih 9	160
Merah	Biru 1 sampai Biru 3	140
	Biru 4 sampai Biru 6	150
	Biru 7 sampai Biru 9	160
Biru	Hitam 1 sampai Hitam 3	180
	Hitam 4 sampai Hitam 6	190
	Hitam 7 sampai Hitam 9	200
Merah	Merah1 sampai Merah 3	180
	Merah1 sampai Merah 3	190
	Merah1 sampai Merah 3	200

Garis merah menunjukan jalur untuk penyimpanan barang warna biru dan merah sedangkan garis biru menunjukan jalur untuk penyimpanan barang warna putih dan hitam. Pergerakan robot saat bergerak pada jalur yang di tandai garis merah,

pertama robot mengambil barang didaerah *home* kemudian mundur sampai simpangan pertama kemudian belok kanan lalu bergerak mengikuti garis jika tujuan rak biru di persimpangan kedua robot akan belok kanan kemudian mengukuti garis, saat berada di persimpangan robot akan mendeteksi rak, jika rak kosong robot belok kiri kemudian maju mengikuti garis dan simpan barang saat berada di depan rak jika tidak robot akan maju mendeteksi garis sampai persimpangan berikutnya dan mendeteksi kembali sampai menemukan rak kosong. jika tujuan rak merah di persimpangan ketiga robot akan belok kanan kemudian mengukuti garis, saat berada di persimpangan robot akan mendeteksi rak, jika rak kosong robot belok kiri kemudian maju mengikuti garis dan simpan barang saat berada di depan rak jika tidak robot akan maju mendeteksi garis sampai persimpangan berikutnya dan mendeteksi kembali sampai menemukan rak kosong. Pergerakan robot saat bergerak pada jalur yang di tandai garis biru, pertama robot mengambil barang didaerah *home* kemudian mundur sampai simpangan pertama kemudian belok kiri lalu bergerak mengikuti garis jika tujuan rak putih di persimpangan kedua robot akan belok kiri kemudian mengukuti garis, saat berada di persimpangan robot akan mendeteksi rak, jika rak kosong robot belok kiri kemudian maju mengikuti garis dan simpan barang saat berada di depan rak jika tidak robot akan maju mendeteksi garis sampai persimpangan berikutnya dan mendeteksi kembali sampai menemukan rak kosong. jika tujuan rak hitam di persimpangan ketiga robot akan belok kiri kemudian mengukuti garis, saat berada di persimpangan robot akan mendeteksi rak, jika rak kosong robot belok kiri kemudian maju mengikuti garis dan simpan barang saat berada di depan rak jika tidak robot akan maju mendeteksi garis sampai persimpangan berikutnya dan

mendeteksi kembali sampai menemukan rak kosong. Setelah barang tersimpan robot akan kembali ke *home* dengan jalur yang sama seperti saat penyimpanan. Dari rabel 4.15 diketahui waktu tempuh robot untuk beberapa alamat rak lebih lama dibanding saat robot bekerja secara manual dengan menggunakan jalur utama dikarenakan saat robot bekerja secara otomatis maka hanya jalur seperti yang ditunjukan pada gambar 4.19 diatas yang bisa digunakan karena sensor ultrasonik pendeteksi rak kosong hanya berada pada bagian sisi kiri robot sehingga rak pada sisi kanan robot tidak dapat terdeteksi. Karena kekurangan robot saat bekerja secara otomatis hanya jalur seperti yang ditunjukan pada gambar 4.19 yang bisa dilalui maka dibuat sistem agar robot dapat bekerja secara manual.