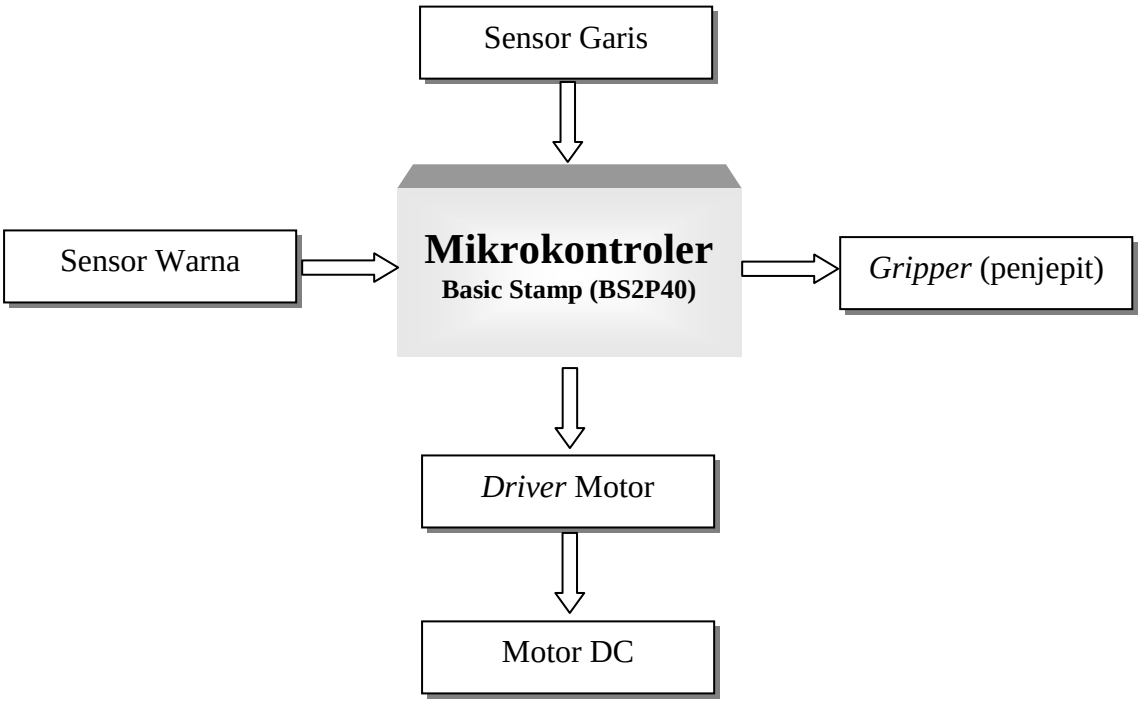
BAB III
PERANCANGAN SISTEM

Bab ini menjelaskan mengenai komponen yang membangun robot pengangkut *box* berdasarkan perbedaan warna berdasarkan landaasan teori. Gambar III.1 merupakan diagram blok keseluruhan dari sistem.



Gambar III.1. Diagram blok sistem

3.1 Perangkat Keras (Hardware)

3.1.1 Mikrokontroler Basic Stamp (BS2P40)

Mikrokontroler yang digunakan adalah 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 harganya yang relatif tidak terlalu mahal. Berikut konfigurasi *port*nya yang dipakai.

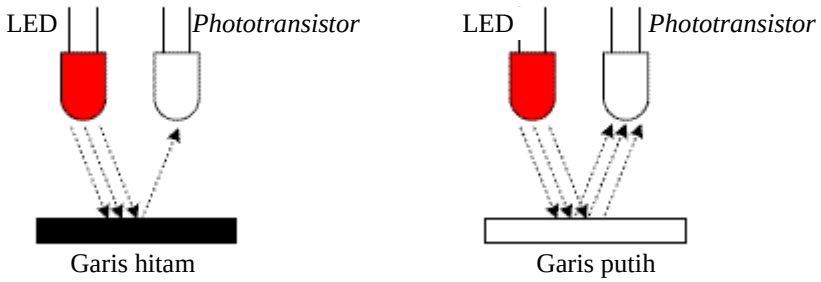
Tabel III.1. Konfigurasi port yang digunakan pada mikrokontroler basic stamp.

Nama Pin	No. Pin	Fungsi	Keterangan
MAIN I/O	0	*Input*	Sensor garis 1
MAIN I/O	1	*Input*	Sensor garis 2
MAIN I/O	2	*Input*	Sensor garis 3
MAIN I/O	3	*Input*	Sensor garis 4
MAIN I/O	4	-	-
MAIN I/O	5	-	-
MAIN I/O	6	-	-
MAIN I/O	7	-	-
MAIN I/O	8	*Output*	Gripper Servo 1
MAIN I/O	9	*Output*	Gripper Servo 2
MAIN I/O	10	*Output*	Driver motor (EN A)
MAIN I/O	11	*Output*	Driver motor (A +)
MAIN I/O	12	*Output*	Driver motor (A -)
MAIN I/O	13	*Output*	Driver motor (EN B)
MAIN I/O	14	*Output*	Driver motor (B +)
MAIN I/O	15	*Output*	Driver motor (B -)
AUX I/O	0	*Input*	Sensor warna (S0)
AUX I/O	1	*Input*	Sensor warna (S1)
AUX I/O	2	*Input*	Sensor warna (Out)
AUX I/O	3	*Input*	Sensor warna (Led)
AUX I/O	4	*Input*	Sensor warna (S2)
AUX I/O	5	*Input*	Sensor warna (S3)
AUX I/O	6	-	-
AUX I/O	7	-	-
AUX I/O	8	-	-
AUX I/O	9	-	-
AUX I/O	10	-	-
AUX I/O	11	-	-
AUX I/O	12	-	-
AUX I/O	13	-	-
AUX I/O	14	-	-
AUX I/O	15	-	-

3.1.2 Sensor Garis

Sensor garis berfungsi untuk mendeteksi warna dari permukaan yang berada dibawah robot *line follower* dengan maksud agar sensor garis ini dapat mengasilkan logika posisi dari robot *line follower* terhadap garis tepat berada dibawah robot.

Logika posisi yang dihasilkan oleh sensor garis ini kemudian akan dijadikan *input* ke mikrokontroler pada robot. Pada sensor garis, komponen yang digunakan yaitu *phototransistor* sebagai pendeteksi garis hitam dengan dasar putih yang menjadi jalur robot dan LED (*Light Emitting Diode*) sebagai pemancar cahaya ke lantai yang kemudian dipantulkan dan diterima oleh *phototransistor*.

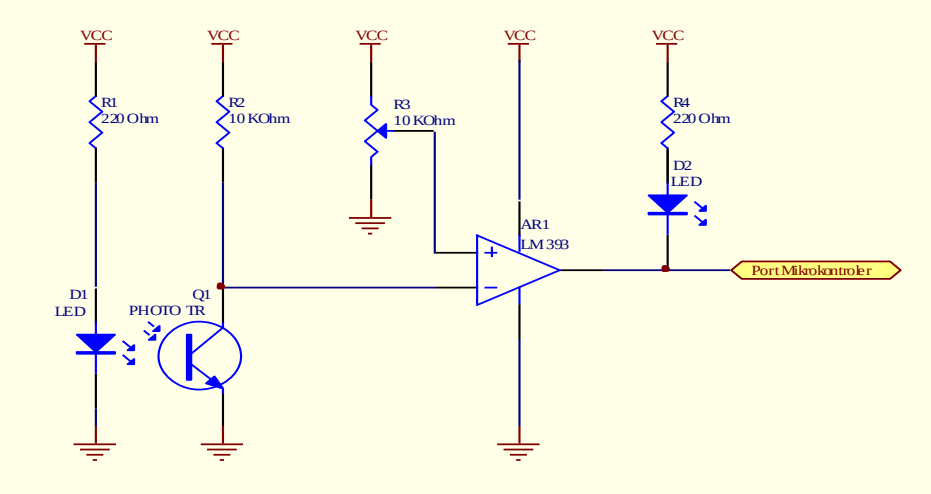


Gambar III.2. Prinsip kerja sensor garis

Phototransistor bekerja dengan cara menangkap emisi cahaya yang dikeluarkan oleh LED. Prinsip kerja dari *phototransistor* adalah ketika basis menangkap cahaya maka *collector* akan terhubung dengan *emitter* dalam hal ini transistor bekerja. *Phototransistor* memiliki *mode* aktif artinya transistor akan menghasilkan reaksi yang sebanding dengan besaran cahaya yang diterima sampai dengan tingkatan tertentu.

Output dari *phototransistor* yaitu berupa tegangan. Semakin besar intensitas cahaya yang didapat maka makin besar pula tegangan yang dihasilkan. Dengan memanfaatkan *mode* aktif dari *phototransistor* maka ditambahkan sebuah komparator sehingga nilai keluaran dari *phototransistor* yang berupa data *analog* dirubah menjadi data *digital*. Hal

ini dilakukan karena mikrokontroler hanya bisa bekerja dengan data *digital*. Berikut ini adalah rangkaian sensor garis yang dipakai.



Gambar III.3. Rangkaian sensor garis

3.1.3 Penggerak Roda

3.1.3.1 Modul Penggerak Roda

L298 bisa men-*drive* dua buah motor dc sampai tegangan 46 V_{DC} dan arus sebesar 2A untuk tiap kanal, *supply* tegangan yang diberikan ke motor dc sebesar 7.2 V_{DC} yang berasal dari baterai Ni-MH 2700 mAh, 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*.

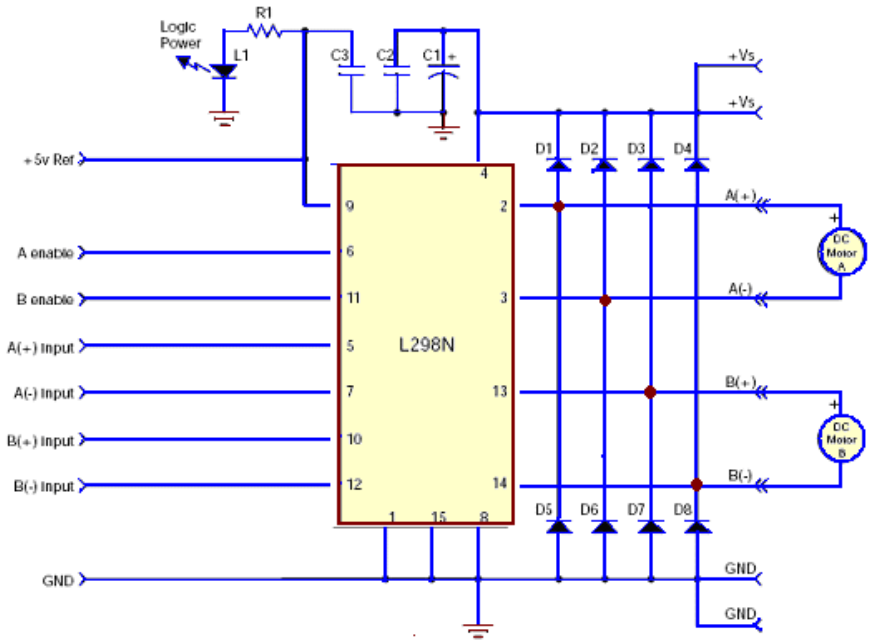
Satu motor dapat dikontrol dengan 3 pin yaitu A*enable*, A+ dan A-. Berikut ini adalah tabel kebenaran untuk salah satu motor.

Tabel III.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 III.4. Rangkaian driver motor L298

Pengontrolan kedua motor kiri dan kanan dibutuhkan 6 *port output* pada mikrokontroler, yaitu untuk motor kiri masing-masing A_{EN}, A+ , A- dan motor kanan B_{EN}, B+, B- seperti yang terlihat pada tabel berikut.

Tabel III.3. Tabel kebenaran untuk dua motor

Gerak	**A_{EN}**	**A+**	**A-**	**B_{EN}**	**B+**	**B-**
Maju	H	L	H	H	H	L
Belok kanan	H	L	H	H	L	H
Belok kiri	H	H	L	H	H	L

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

3.1.3.2 Pengaturan Gerak Robot Dengan Metode 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 7,4 Volt, dan jika PWM diberi kondisi *Low*, maka tegangan keluaran mendekati 0 Volt. Diantara kondisi itu maka nilai tegangan keluaran dapat kita hitung, yaitu :

(*Duty* ÷ 255) x 7,4 V.

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

(100 ÷ 255) x 7,4 V = 2,90 V, tegangan keluaran PWM sebesar 2,90 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 (0-15).
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.

Berikut adalah contoh pemrograman PWM untuk mengatur kecepatan gerak maju :

```
PWM AEN,150,40      ' pengaturan kecepatan dengan PWM untuk motor kanan
PWM BEN,200,40      ' pengaturan kecepatan dengan PWM untuk motor kiri
```

Gambar III.5. Syntax PWM

Pada *listing program* di atas, PWM adalah perintahnya, AEN / BEN adalah nama pin tujuan yang sudah dideklarasikan di

bagian variabel. Nilai 150 dan 200 adalah kecepatan *(duty cycle)* dan 40 adalah durasi PWM.

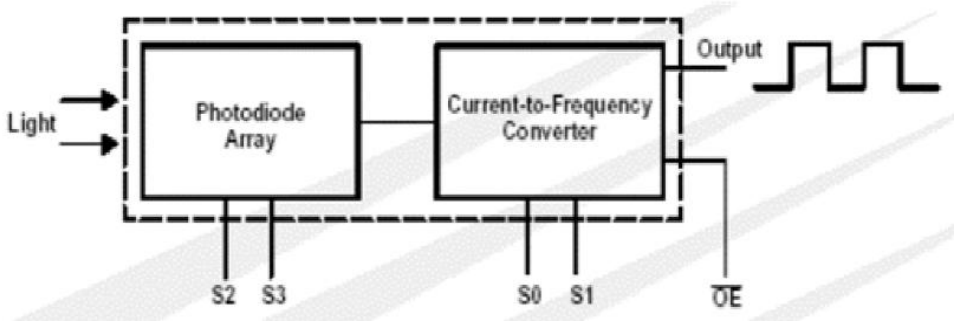
Nilai *duty cycle* atau kecepatan untuk motor kiri dan motor kanan nilainya berbeda. Hal ini dilakukan untuk mendapatkan gerak lurus, karena secara tipikalnya kedua motor tersebut berbeda, maka solusinya harus memberikan nilai kecepatan berbeda agar hasil geraknya sama / lurus.

3.1.4 Sensor Warna (TCS3200)

Modul sensor warna TCS3200 merupakan pendeteksi warna RGB menggunakan *array* fotodetektor berupa *chip* warna dari TAOS TCS3200. Aplikasinya antara lain pada pendeteksi obyek di industri.

Fitur:

- Tegangan Kerja : 2.7 - 5.5 V
- Sensor *Array* : 8 x 8 photo dioda, 16 photodioda filter merah, 16 photodioda *filter* hijau, 16 photodioda tanpa filter.
- Lensa : 5.3 mm (dalam jarak 25 mm mampu melihat area bujur sangkar dengan sisi 4 mm).



Gambar III.6. Sketsa fisik dan blok fungsional TCS3200

Setiap warna bisa disusun dari warna dasar. Untuk cahaya, warna dasar penyusunnya adalah warna Merah, Hijau dan Biru, atau lebih dikenal dengan istilah RGB (*Red-Green-Blue*).

Photodiode pada IC TCS3200 disusun secara *array* 8x8 dengan konfigurasi: 16 *photodiode* untuk menfilter warna merah, 16 *photodiode* untuk memfilter warna hijau, 16 *photodiode* untuk memfilter warna biru, dan 16 *photodiode* tanpa *filter*. Kelompok photodiode mana yang akan dipakai bisa diatur melalui kaki selektor S2 dan S3.

Tabel III.4. Kombinasi fungsi dari S2 dan S3

S2	S3	Photodiode yang aktif
0	0	Pemfilter Merah
0	1	Pemfilter Biru
1	0	Tanpa Filter
1	1	Pemfilter Hijau

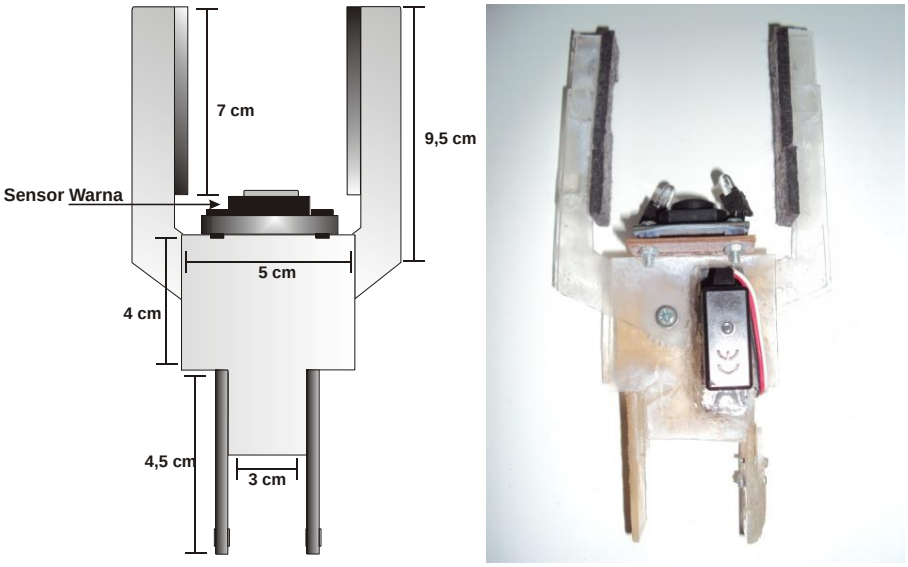
Photodiode akan mengeluarkan arus yang besarnya sebanding dengan kadar warna dasar cahaya yang menimpanya. Arus ini kemudian dikonversikan menjadi sinyal kotak dengan frekuensi sebanding dengan besarnya arus. Frekuensi *Output* ini bisa diskala dengan mengatur kaki selektor S0 dan S1.

Tabel III.5. Penskalaan Output

S0	S1	Skala frekuensi Output
0	0	*Power Down*
0	1	2%
1	0	20%
1	1	100%

3.1.5 Design Gripper

Gripper digunakan untuk mengangkat box yang ada di asal yang akan di pindahkan ke tujuan. *Gripper* di *design* disesuaikan dengan bentuk objek yaitu *box*. berikut ini adalah gambar design gripper yang digunakan.



Gambar III.7. *Design gripper*

Dari gambar diatas terlihat dimensi dari *gripper*. Pada *gripper* terdapat 2 mini servo sebagai penggerak *gripper* dan sensor warna sebagai mendeteksi warna *box*.

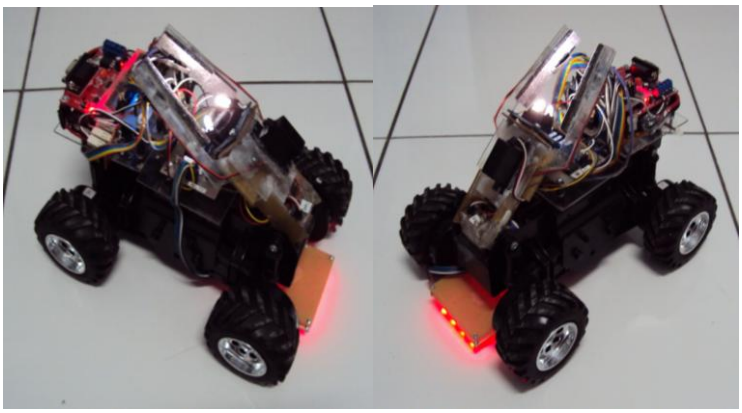
3.1.6 Design Robot

Berikut ini adalah robot yang dipakai dengan spesifikasi sebagai berikut:

1. Jenis robot merupakan robot beroda yang memiliki 4 buah roda yang digerakkan dengan 2 motor dc di roda bagian kiri dan kanan.
2. Bentuk robot berdimensi 25 cm x 18 cm.
3. Sensor garis dipasang di bawah bagian depan.
4. *Gripper* sebagai pengangkat *box* dipasang pada bagian depan robot.
5. Sensor warna dipasang pada bagian depan *gripper*.



(a)



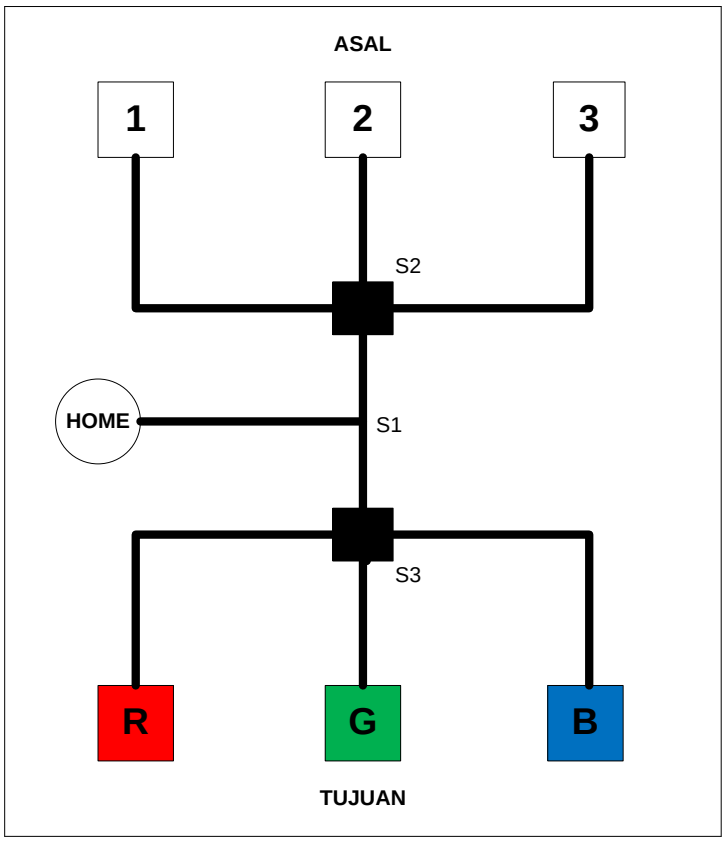
(b) *(c)*

Gambar III.8. (a) Robot tampak depan (b) Robot tampak samping kiri (c) Robot tampak samping kanan

3.2 Perangkat Lunak (Software)

3.2.1 Algoritma dasar

Perancangan algoritma dasar sangat penting untuk sistem kendali pada robot, karena algoritma yang akan menjadi panduan dalam mengembangkan algoritma kendali keseluruhan dari track yang telah ditentukan. Berikut adalah miniatur *track* robot.

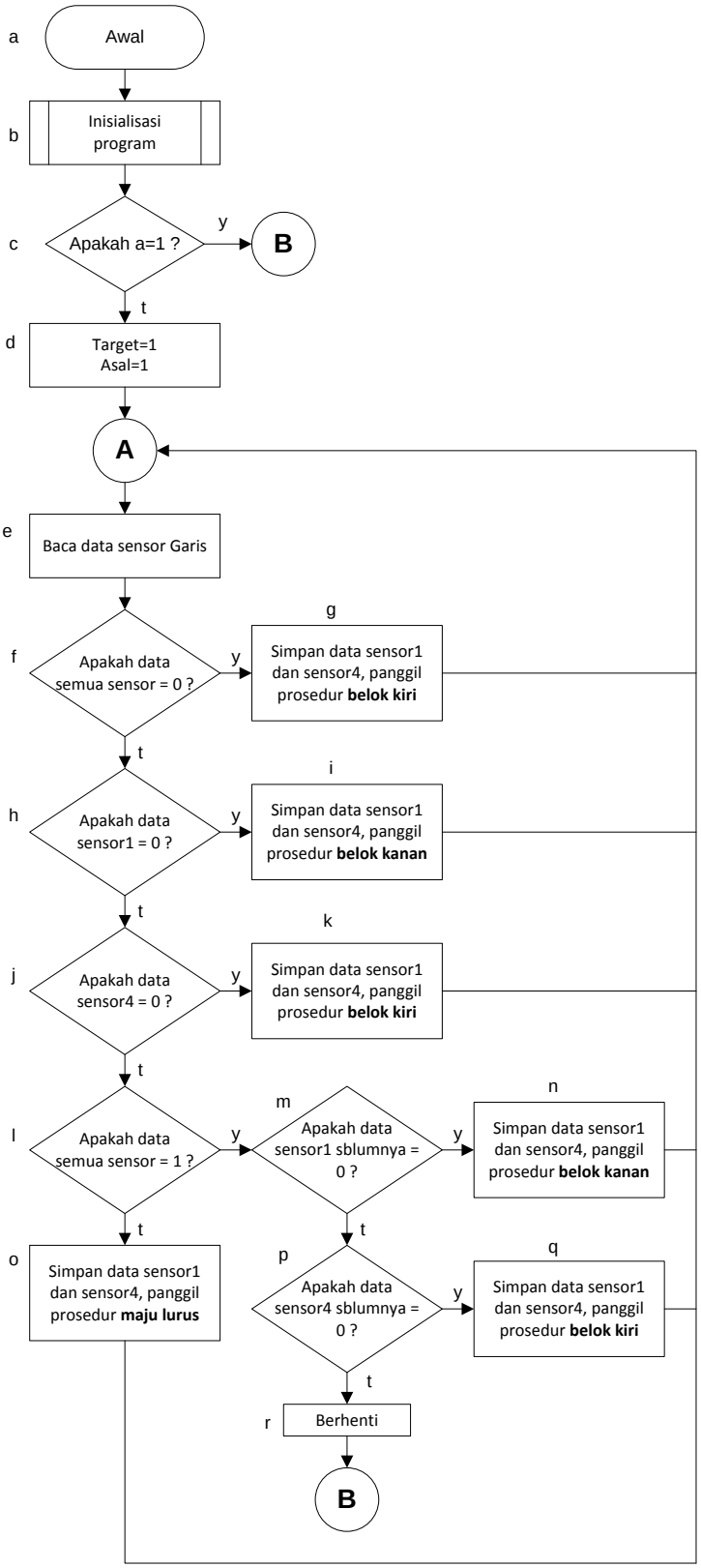


Gambar III.9. Miniatur track robot

Pencarian *box* dimulai dari *home*, *box* yang akan dipindahkan ke tujuan berada di 3 sumber asal, *box* akan diacak dan dan dipindahkan ke tujuan penyimpanan sesuai dengan warna. Pencarian *box* di asal terurut dari asal 1 sampai dengan 3, dengan proses pencarian warna merah, hijau dan biru. Setelah ketiga warna dasar dipindahkan robot kemudian kembali ke *home*.

3.2.2 Algoritma Seluruh Kendali Robot

Pada bagian ini adalah algoritma jalan robot dengan tugas memindahkan *box* ke tujuan pada tiga bagian asal *box* yang di *random* berdasarkan warna.



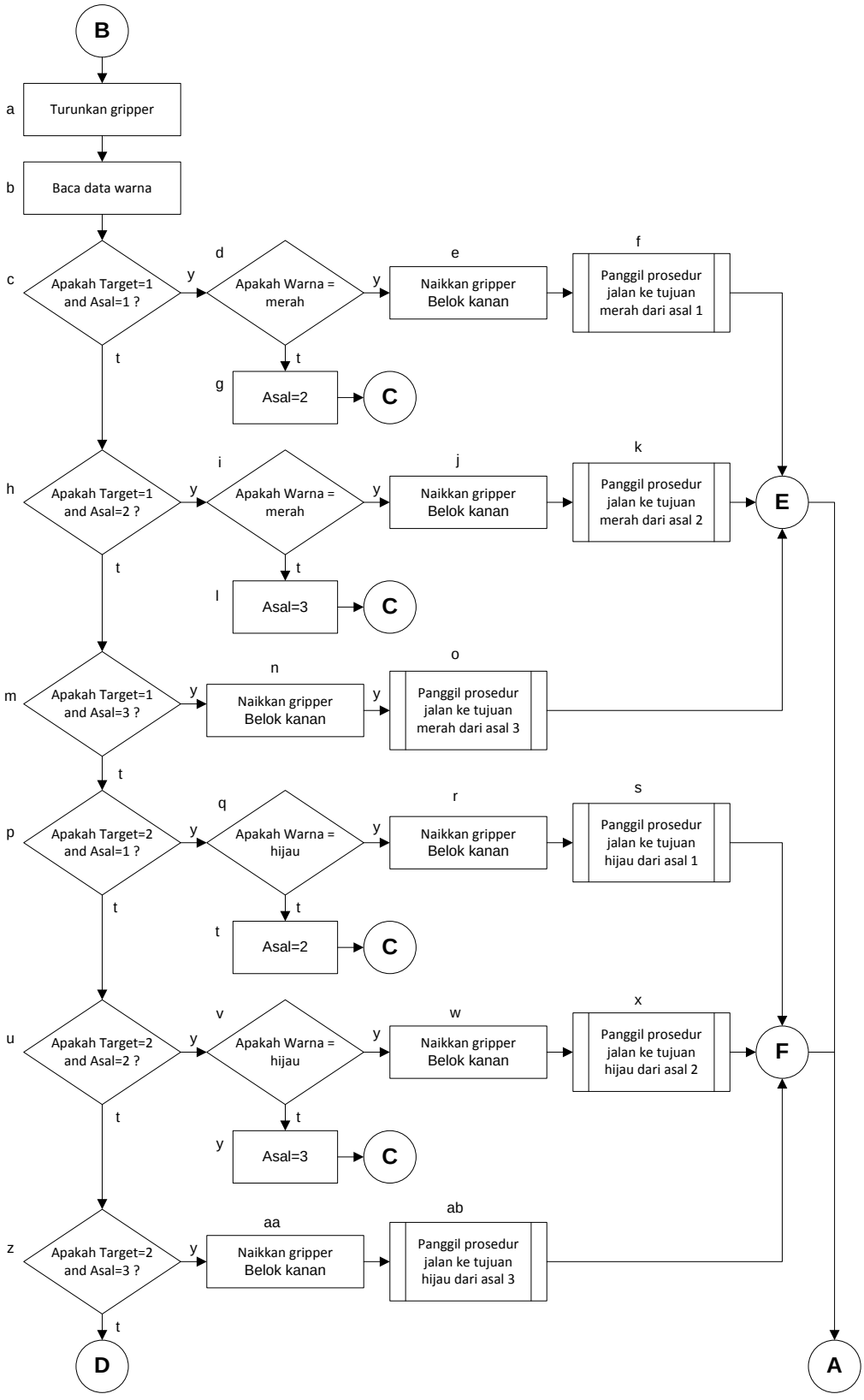
Gambar III.10. Flowchart program utama

Algoritma diatas adalah algoritma untuk program awal jalan robot yang *start* di home, jalan mengikuti garis. Robot selalu akan menuju ke asal 1 untuk mendeteksi warna awal d asal 1. Selama proses jalan ke asal 1, setiap persimpangan yang dilewatinya selalu belok kiri karena di sesuaikan dengan track jalan ke asal 1 dan akan berhenti jika semua sensor garis pada robot berada pada permukaan putih. Berikut ini adalah penjelasan *flowchart* program:

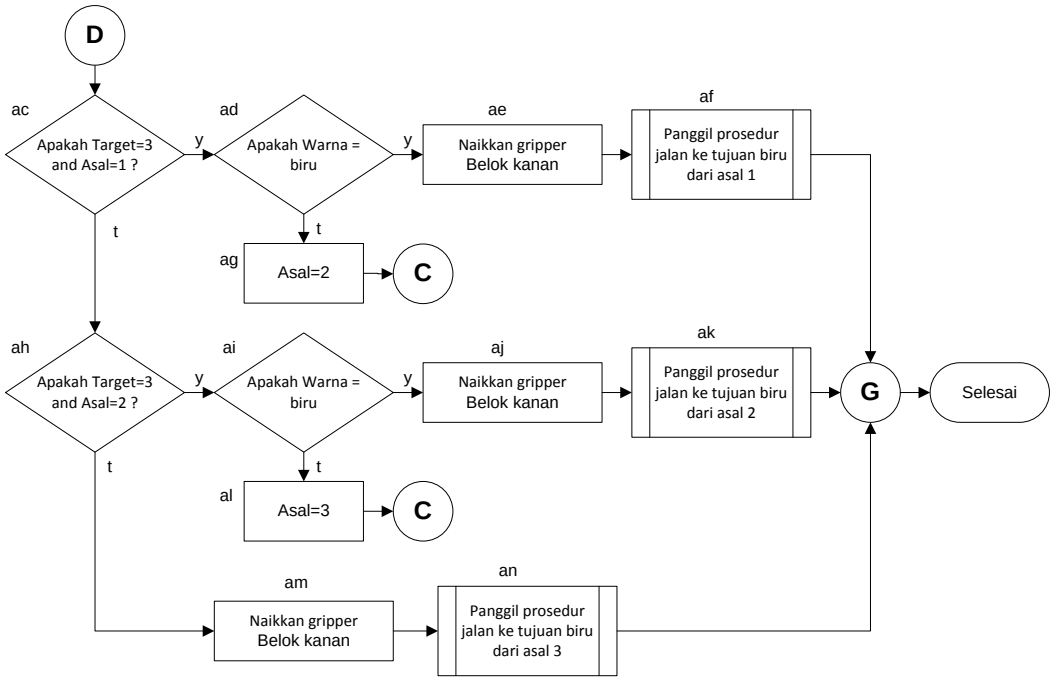
Tabel III.6. Penjelasan flowchart program utama

No	Keterangan
a	Memulai awal program.
b	Pendeklarasian program. Pengaturan nama prosesor,nama kompiler, konfigurasi pin-pin dan deklarasi variabel.
c	Membandingkan nilai a, jika a=1 maka siap untuk baca warna
d	Jika a tidak bernilai 1, maka nilai target dan asal=1
e	Baca sensor garis, untuk mengetahui posisi robot terhadap garis.
f	Membandingkan semua sensor dengan nilai nol. Sensor bernilai 0, berarti sensor berada pada garis hitam. Jika semua sensor hitam, robot berada pada persimpangan.
g	Jika sensor berada di persimpangan, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
h	Jika semua sensor tidak bernilai nol, maka lakukan lagi perbandingan data sensor 1 dengan nilai nol.
i	Jika sensor 1 bernilai nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kanan.

j	Jika sensor 1 tidak bernilai nol, maka lakukan lagi perbandingan data sensor 4 dengan nilai nol.
k	Jika sensor 4 bernilai nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
l	Jika sensor 4 tidak bernilai nol, maka lakukan lagi perbandingan data semua sensor dengan nilai satu. Sensor bernilai 1, berarti sensor berada pada permukaan putih. Jika semua sensor putih, robot berada di luar jalur hitam.
m	Jika semua sensor bernilai satu, lakukan perpandingan lagi terhadap kondisi bebelum data semua sensor, sensor 1 bernilai nol.
n	Jika kondisi sensor 1 sebelumnya nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kanan.
o	Jika data semua sensor tidak bernilai satu, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur maju lurus.
p	Jika kondisi sensor 1 sebelumnya tidak nol, lakukan perbandingan lagi dengan data sensor 4 sebelumnya nol.
q	Jika data sensor 4 sebelumnya adalah nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
r	Jika data sensor 4 sebelumnya tidak nol, maka robot berhenti.



Gambar III.11. Flowchart program utama lanjutan1



Gambar III.12. Flowchart program utama lanjutan2

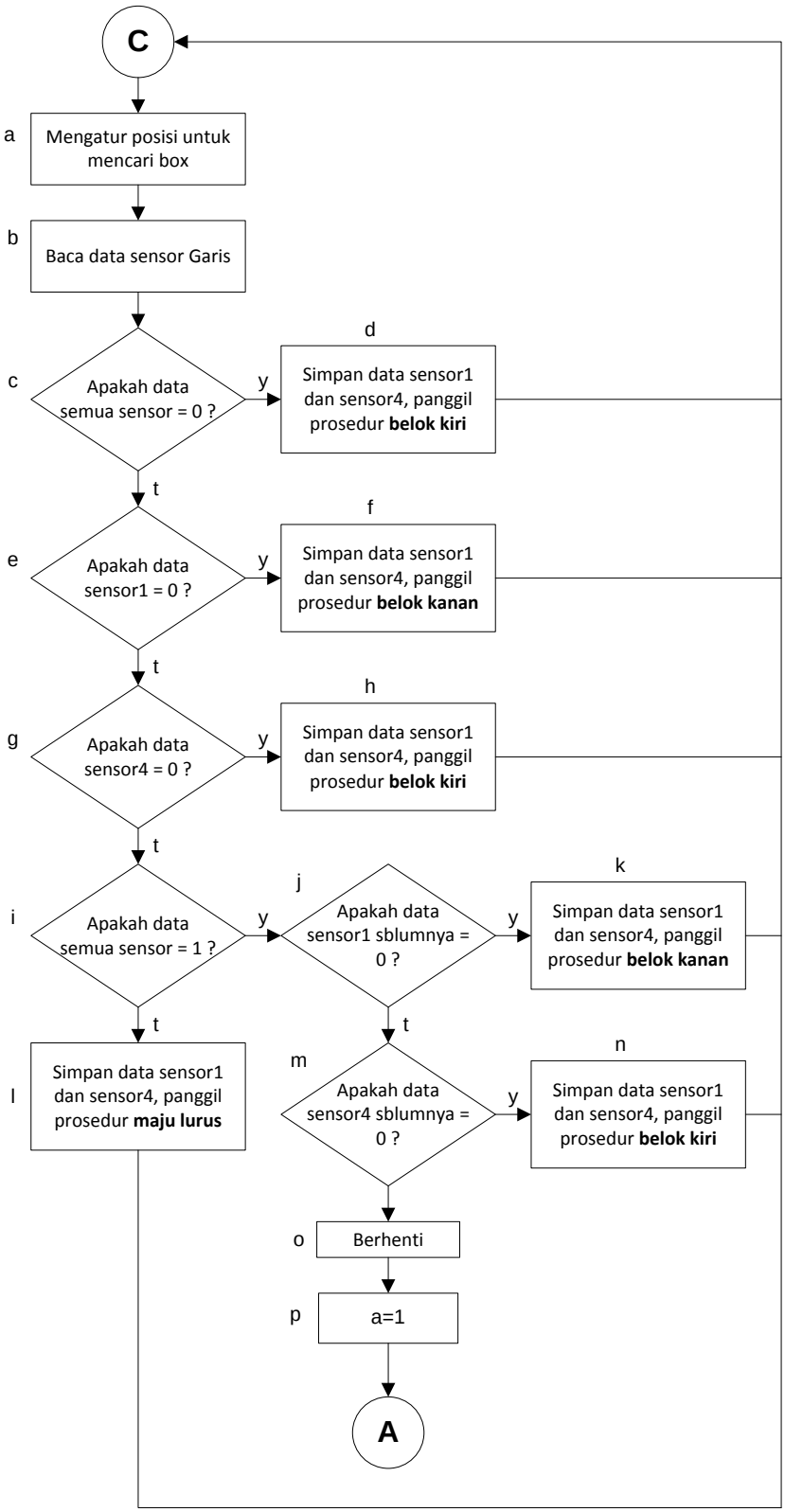
Algoritma di atas adalah algoritma program untuk proses pendeteksian warna box, pencarian box di asal, memindahkan box ke tujuan sesuai dengan target warna sampai tugas robot selesai kembali ke home. Berikut ini adalah penjelasan diagram alir program:

Tabel III.7. Penjelasan flowchart program utama lanjutan

No	Keterangan
a	Robot berada di posisi asal 1, baca data sensor jarak
b	Turunkan gripper, maka baca data warna box dengan menggunakan sensor warna.
c	Apakah target pencarian warna box merah di asal 1
d	Bandingkan warna box dengan target merah
e	Jika target sesuai dengan warna box, maka angkat box dan belok kanan sampai ketemu garis hitam

f	Panggil prosedur jalan ke tujuan merah dari asal 1
g	Jika target dengan warna box tidak sama, maka cari di asal 2
h	Apakah target pencarian warna box merah di asal 2
i	Bandingkan warna box dengan target merah
j	Jika target sesuai dengan warna box, maka angkat box dan belok kanan sampai ketemu garis hitam
k	Panggil prosedur jalan ke tujuan merah dari asal 2
l	Jika target dengan warna box tidak sama, maka cari di asal 3
m	Apakah target pencarian warna box merah di asal 3
n	Angkat box dan belok kanan sampai ketemu garis hitam
o	Panggil prosedur jalan ke tujuan merah dari asal 3
p	Apakah target pencarian warna box hijau di asal 1
q	Bandingkan warna box dengan target hijau
r	Jika target sesuai dengan warna box, maka angkat box dan belok kanan sampai ketemu garis hitam
s	Panggil prosedur jalan ke tujuan hijau dari asal 1
t	Jika target dengan warna box tidak sama, maka cari di asal 2
u	Apakah target pencarian warna box hijau di asal 2
v	Bandingkan warna box dengan target hijau
w	Jika target sesuai dengan warna box, maka angkat box dan belok kanan sampai ketemu garis hitam
x	Panggil prosedur jalan ke tujuan hijau dari asal 2
y	Jika target dengan warna box tidak sama, maka cari di asal 3
z	Apakah target pencarian warna box hijau di asal 3
aa	Angkat box dan belok kanan sampai ketemu garis hitam
ab	Panggil prosedur jalan ke tujuan hijau dari asal 3

ac	Apakah target pencarian warna box biru di asal 1
ad	Bandingkan warna box dengan target biru
ae	Jika target sesuai dengan warna box, maka angkat box dan belok kanan sampai ketemu garis hitam
af	Panggil prosedur jalan ke tujuan biru dari asal 1
ag	Jika target dengan warna box tidak sama, maka cari di asal 2
ah	Apakah target pencarian warna box biru di asal 2
ai	Bandingkan warna box dengan target biru
aj	Jika target sesuai dengan warna box, maka angkat box dan belok kanan sampai ketemu garis hitam
ak	Panggil prosedur jalan ke tujuan biru dari asal 2
al	Jika target dengan warna box tidak sama, maka cari di asal 3
am	Jika target sesuai dengan warna box, maka angkat box dan belok kanan sampai ketemu garis hitam
an	Panggil prosedur jalan ke tujuan biru dari asal 3

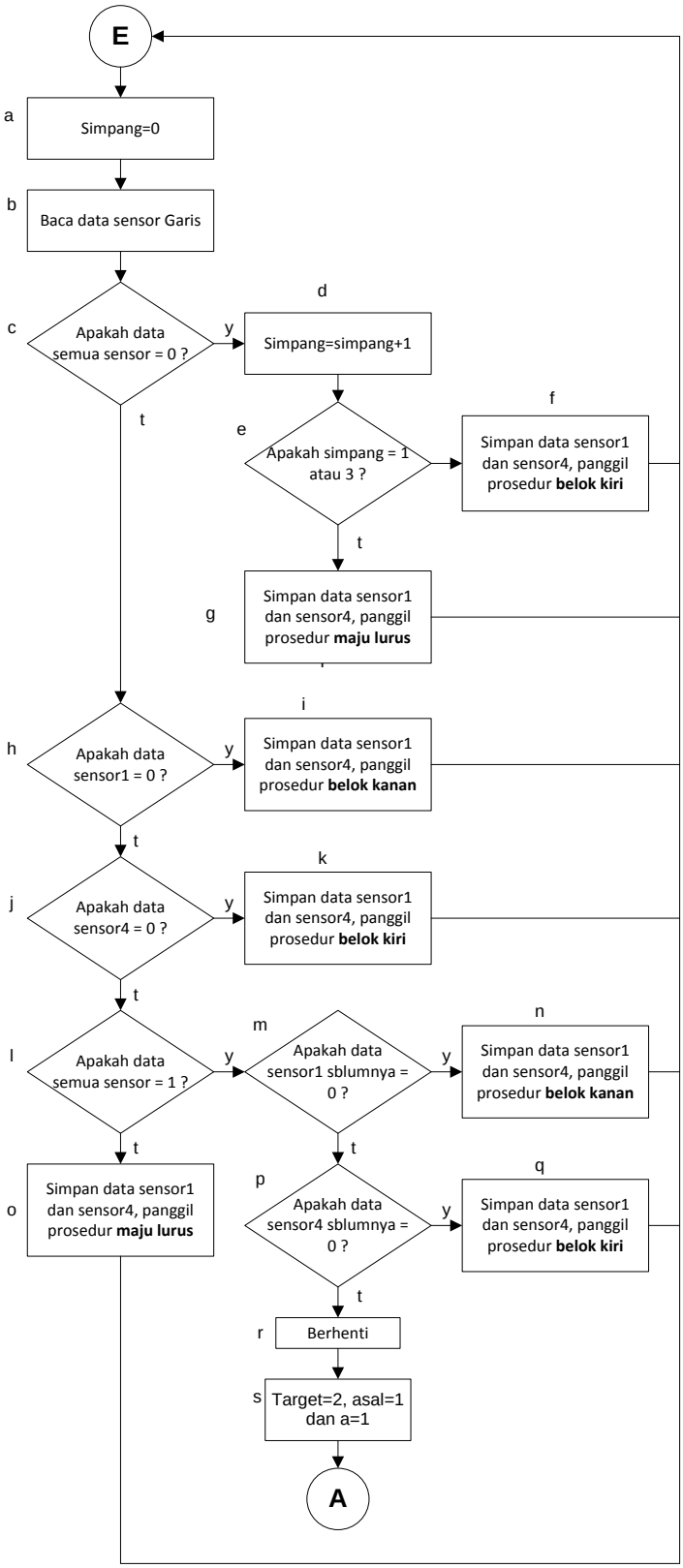


Gambar III.13. Flowchart program pencarian box di asal

Tabel III.8. Penjelasan flowchart program pencarian box di asal

No	Keterangan
a	Mengatur posisi robot untuk siap mencari box di asal selanjutnya
b	Baca sensor garis, untuk mengetahui posisi robot terhadap garis.
c	Membandingkan semua sensor dengan nilai nol. Sensor bernilai 0, berarti sensor berada pada garis hitam. Jika semua sensor hitam, robot berada pada persimpangan.
d	Jika sensor berada di persimpangan, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
e	Jika semua sensor tidak bernilai nol, maka lakukan lagi perbandingan data sensor 1 dengan nilai nol.
f	Jika sensor 1 bernilai nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kanan.
g	Jika sensor 1 tidak bernilai nol, maka lakukan lagi perbandingan data sensor 4 dengan nilai nol.
h	Jika sensor 4 bernilai nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
i	Jika sensor 4 tidak bernilai nol, maka lakukan lagi perbandingan data semua sensor dengan nilai satu. Sensor bernilai 1, berarti sensor berada pada permukaan putih. Jika semua sensor putih, robot berada di luar jalur hitam.
j	Jika semua sensor bernilai satu, lakukan perpandingan lagi terhadap kondisi bebelum data semua sensor, sensor 1 bernilai nol.
k	Jika kondisi sensor 1 sebelumnya nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok

	kanan.
l	Jika data semua sensor tidak bernilai satu, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur maju lurus.
m	Jika kondisi sensor 1 sebelumnya tidak nol, lakukan perbandingan lagi dengan data sensor 4 sebelumnya nol.
n	Jika data sensor 4 sebelumnya adalah nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
o	Jika data sensor 4 sebelumnya tidak nol, maka robot berhenti.
p	Nilai variabel a=1

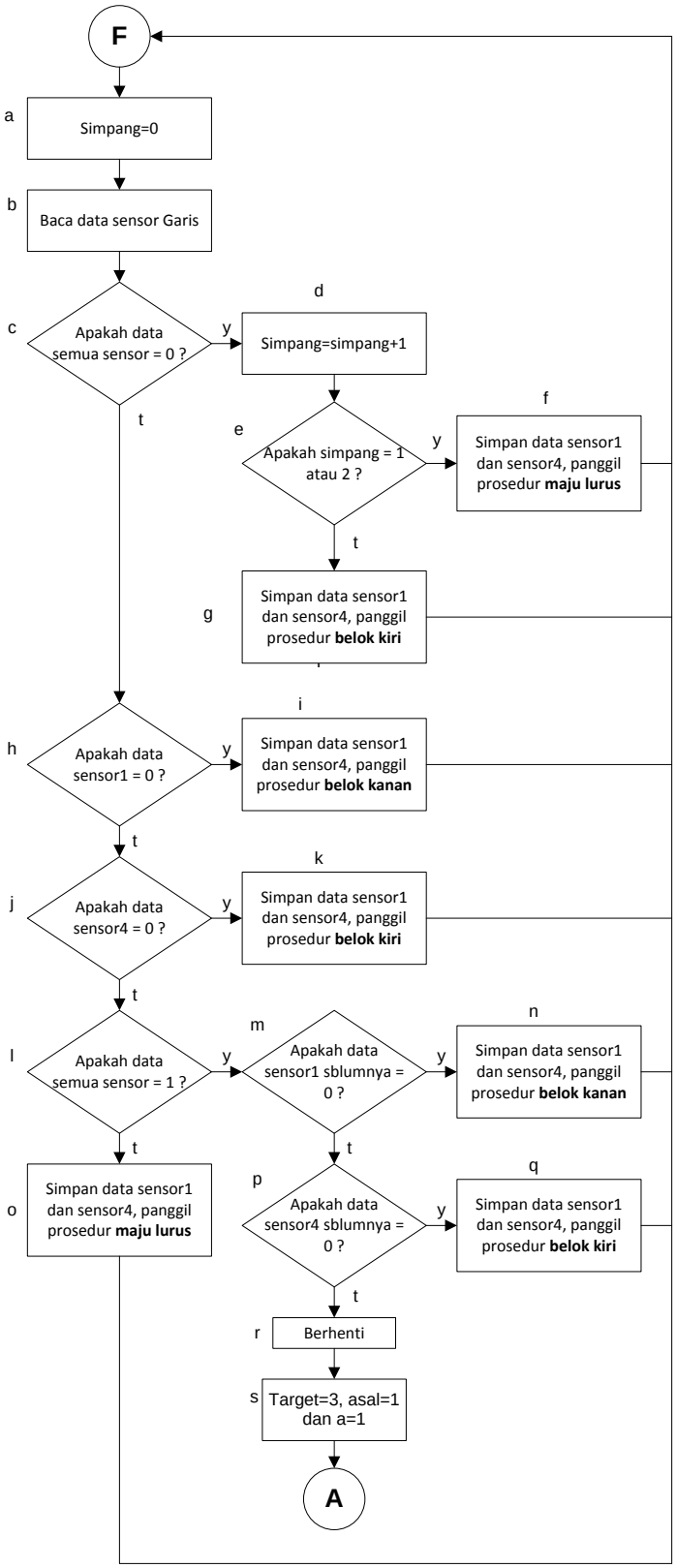


Gambar III.14. Flowchart program jalan robot menuju asal 1 dari tujuan R

Tabel III.9. Penjelasan flowchart program jalan menuju asal 1 dari tujuan R

No	Keterangan
a	Nilai simpang=0
b	Baca sensor garis, untuk mengetahui posisi robot terhadap garis.
c	Membandingkan semua sensor dengan nilai nol. Sensor bernilai 0, berarti sensor berada pada garis hitam. Jika semua sensor hitam, robot berada pada persimpangan.
d	Nilai simpang ditambah 1
e	Apakah nilai simpang 1 atau 3
f	Jika nilai simpang 1 atau 3, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
g	Jika nilai simpang tidak 1 atau 3, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur maju lurus.
h	Jika semua sensor tidak bernilai nol, maka lakukan lagi perbandingan data sensor 1 dengan nilai nol.
i	Jika sensor 1 bernilai nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kanan.
j	Jika sensor 1 tidak bernilai nol, maka lakukan lagi perbandingan data sensor 4 dengan nilai nol.
k	Jika sensor 4 bernilai nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
l	Jika sensor 4 tidak bernilai nol, maka lakukan lagi perbandingan data semua sensor dengan nilai satu. Sensor bernilai 1, berarti sensor berada pada permukaan putih. Jika semua sensor putih, robot berada di luar jalur hitam.
m	Jika semua sensor bernilai satu, lakukan perpandingan lagi terhadap kondisi bebelum data semua sensor, sensor 1 bernilai nol.

n	Jika kondisi sensor 1 sebelumnya nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kanan.
o	Jika data semua sensor tidak bernilai satu, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur maju lurus.
p	Jika kondisi sensor 1 sebelumnya tidak nol, lakukan perbandingan lagi dengan data sensor 4 sebelumnya nol.
q	Jika data sensor 4 sebelumnya adalah nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
r	Jika data sensor 4 sebelumnya tidak nol, maka robot berhenti.
s	Nilai target=2, asal=1 dan a=1.

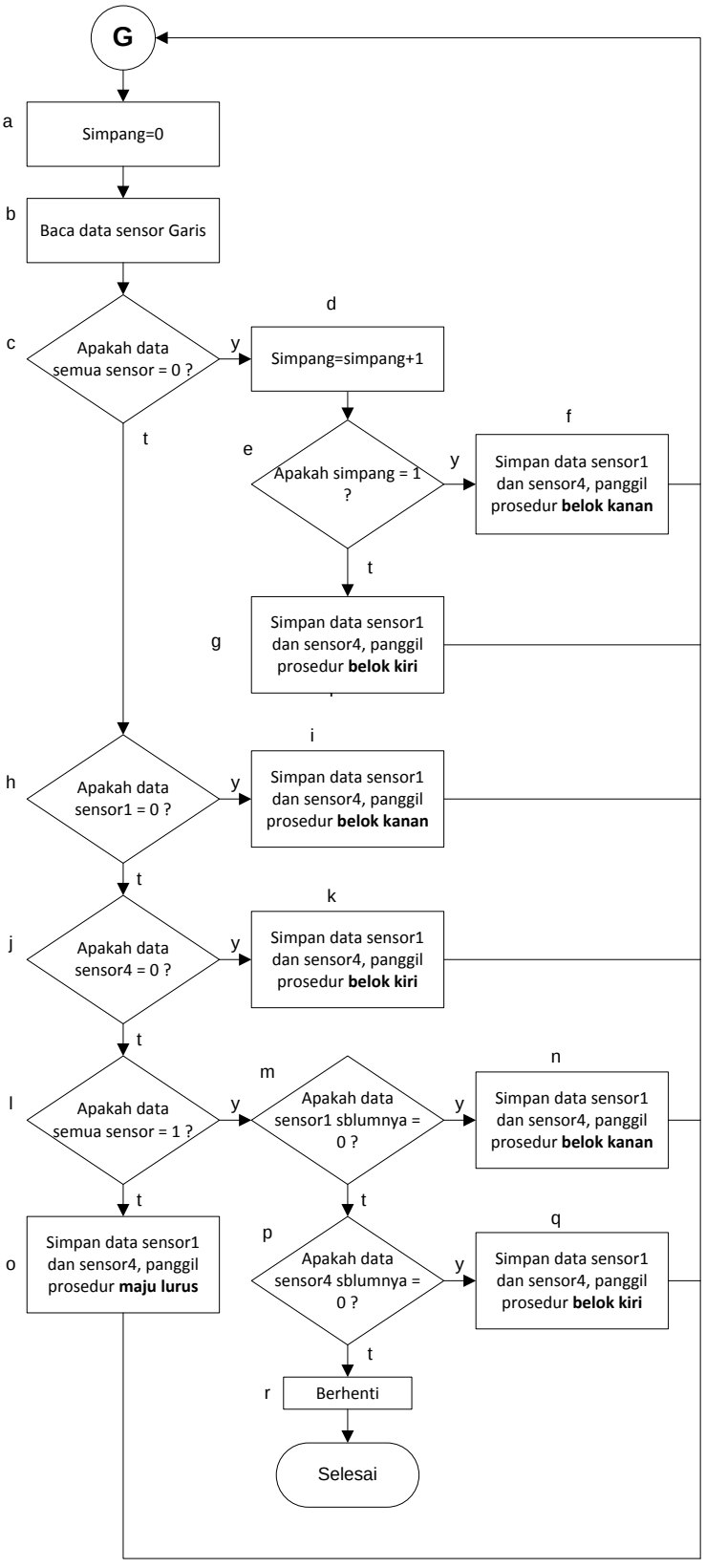


Gambar III.15. Flowchart program jalan robot menuju asal 1 dari tujuan G

Tabel III.10. Penjelasan flowchart program jalan menuju asal 1 dari tujuan G

No	Keterangan
a	Nilai simpang=0
b	Baca sensor garis, untuk mengetahui posisi robot terhadap garis.
c	Membandingkan semua sensor dengan nilai nol. Sensor bernilai 0, berarti sensor berada pada garis hitam. Jika semua sensor hitam, robot berada pada persimpangan.
d	Nilai simpang ditambah 1
e	Apakah nilai simpang 1 atau 2
f	Jika nilai simpang 1 atau 2, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur maju lurus.
g	Jika nilai simpang tidak 1 atau 2, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
h	Jika semua sensor tidak bernilai nol, maka lakukan lagi perbandingan data sensor 1 dengan nilai nol.
i	Jika sensor 1 bernilai nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kanan.
j	Jika sensor 1 tidak bernilai nol, maka lakukan lagi perbandingan data sensor 4 dengan nilai nol.
k	Jika sensor 4 bernilai nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
l	Jika sensor 4 tidak bernilai nol, maka lakukan lagi perbandingan data semua sensor dengan nilai satu. Sensor bernilai 1, berarti sensor berada pada permukaan putih. Jika semua sensor putih, robot berada di luar jalur hitam.
m	Jika semua sensor bernilai satu, lakukan perpandingan lagi terhadap kondisi bebelum data semua sensor, sensor 1 bernilai nol.

n	Jika kondisi sensor 1 sebelumnya nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kanan.
o	Jika data semua sensor tidak bernilai satu, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur maju lurus.
p	Jika kondisi sensor 1 sebelumnya tidak nol, lakukan perbandingan lagi dengan data sensor 4 sebelumnya nol.
q	Jika data sensor 4 sebelumnya adalah nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
r	Jika data sensor 4 sebelumnya tidak nol, maka robot berhenti.
s	Nilai target=3, asal=1 dan a=1.



Gambar III.16. Flowchart program jalan robot menuju home dari tujuan B

Tabel III.11. Penjelasan flowchart program jalan menuju home dari tujuan G

No	Keterangan
a	Nilai simpang=0
b	Baca sensor garis, untuk mengetahui posisi robot terhadap garis.
c	Membandingkan semua sensor dengan nilai nol. Sensor bernilai 0, berarti sensor berada pada garis hitam. Jika semua sensor hitam, robot berada pada persimpangan.
d	Nilai simpang ditambah 1
e	Apakah nilai simpang 1
f	Jika nilai simpang 1, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kanan.
g	Jika nilai simpang tidak 1, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
h	Jika semua sensor tidak bernilai nol, maka lakukan lagi perbandingan data sensor 1 dengan nilai nol.
i	Jika sensor 1 bernilai nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kanan.
j	Jika sensor 1 tidak bernilai nol, maka lakukan lagi perbandingan data sensor 4 dengan nilai nol.
k	Jika sensor 4 bernilai nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
l	Jika sensor 4 tidak bernilai nol, maka lakukan lagi perbandingan data semua sensor dengan nilai satu. Sensor bernilai 1, berarti sensor berada pada permukaan putih. Jika semua sensor putih, robot berada di luar jalur hitam.
m	Jika semua sensor bernilai satu, lakukan perpandingan lagi terhadap kondisi bebelum data semua sensor, sensor 1 bernilai nol.

n	Jika kondisi sensor 1 sebelumnya nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kanan.
o	Jika data semua sensor tidak bernilai satu, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur maju lurus.
p	Jika kondisi sensor 1 sebelumnya tidak nol, lakukan perbandingan lagi dengan data sensor 4 sebelumnya nol.
q	Jika data sensor 4 sebelumnya adalah nol, simpan data terakhir sensor 1 dan sensor 4 kemudian panggil prosedur belok kiri.
r	Jika data sensor 4 sebelumnya tidak nol, maka robot berhenti.