BAB III
PERANCANGAN SISTEM

Bab ini akan membahas tentang perancangan sistem robot pengakut *box* berdasarkan perbedaan warna. Pada perancangan sistem ini dibagi menjadi tiga bagian penting, yaitu:

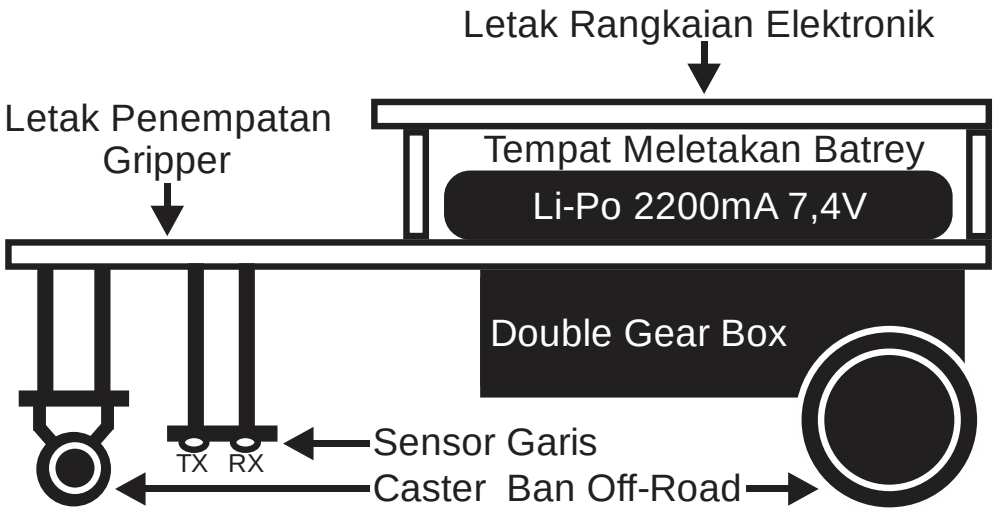
1. Perancangan mekanik.
2. Perancangan perangkat keras.
3. Perancangan perangkat lunak.

3.1 Perancangan Mekanik

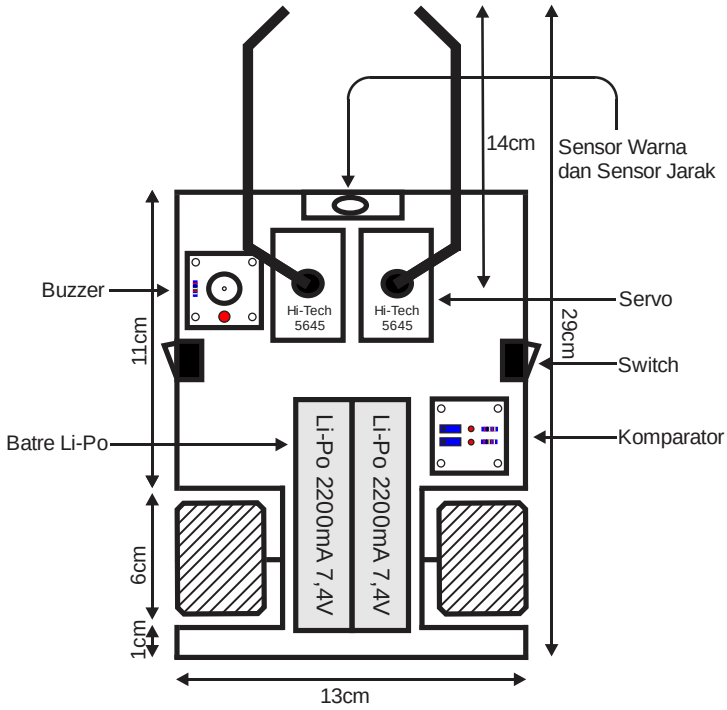
Perancangan mekanik robot, dirancang berdasarkan ukuran dan berat agar robot dapat mengankut *box* dengan beban 150 gram. Ukuran dan berat robot yang dirancang adalah sebagai berikut:

1. Panjang Robot : ± 29 cm
2. Lebar Robot : ± 13 cm
3. Berat Robot : ± 970 gram

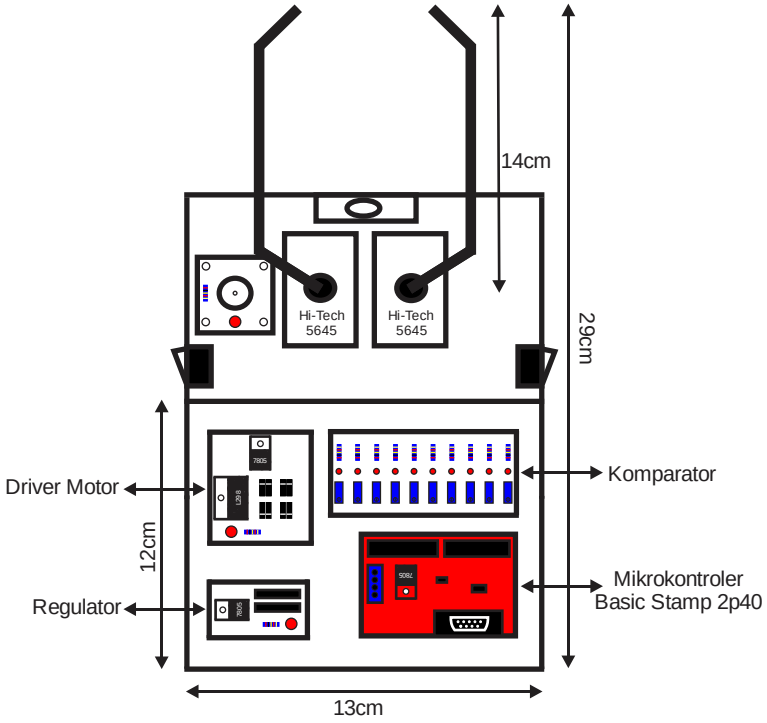
Gambar 3.1 menunjukkan robot tampak samping dengan penempatan komponen.



Gambar 3.1 Menunjukkan tata letak robot tampak samping.



Gambar 3.2 Menunjukkan tata letak robot tampak bagian tengah.



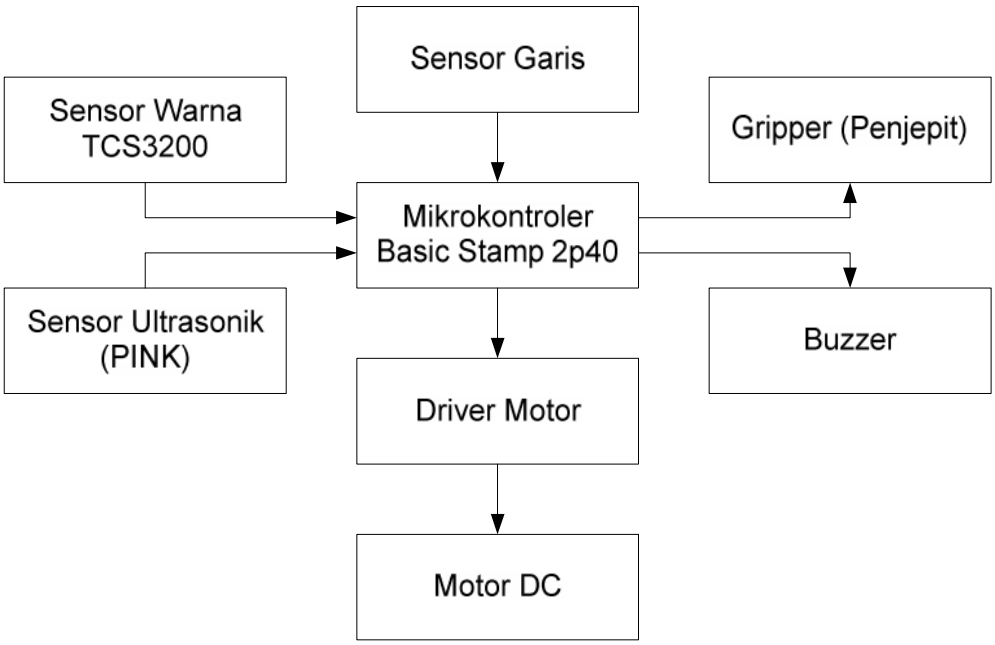
Gambar 3.3 Menunjukkan tata letak robot tampak bagian atas.

Penenmpatan Komponen:

1. Mikrokontroler ditempatkan pada bagian atas untuk menjangkau dan mempermudah pemberian input atau output kepada komponen yang lainnya.
2. Regulator ditempatkan pada bagian atas di samping mikrokontroler untuk mempermudah pemberian catu daya kepada komponen yang lainnya seperti mikrokontroler, komparator, servo dan yang lainnya.
3. Komparator ditempatkan pada bagian atas di depan mikrokontroler sejajar dengan sensor garis yang ditempatkan di bawah, sehingga kabel yang tersambung dari komparator dan sensor garis lurus.
4. *Driver* motor ditempatkan pada bagian atas di samping komparator.
5. Catu daya ditempatkan pada bagian tengah atau di bawah mikrokontroler, dikarenakan ukuran yang sangat besar sehingga ditempatkan di tengah dan untuk mempermudah pengecasan atau isi ulang batrey.
6. *Switch* atau saklar ditempatkan pada bagian tengah dan tataletak di samping kiri ujung dan kanan ujung. *Switch* pada bagian kiri untuk mengaktifkan *driver* motor dan *switch* pada bagian kanan adalah untuk mengaktifkan komponen regulator.
7. Servo ditempatkan pada bagian tengah depan batrey untuk membuat *gripper*, terdiri dari servo bagian kiri dan servo bagian kanan untuk mencapit *box*.
8. Buzzer ditempatkan pada bagian tengah samping servo kiri.
9. Sensor warna ditempatkan pada bagian tengah di depan servo agar mempermudah pembacaan warna.
10. Sensor ultrasonik ditempatkan pada bagian tengah di depan servo dan di atas sensor warna agar terjangkau pendeteksian *box*.
11. Sensor garis di letakan pada bagian bawah agar dapat terjangkau mendeteksi atau membedakan warna garis hitam dan dasar *track* warna putih

3.2 Perancangan Perangkat Keras (Hardware)

Komponen yang akan di bangun untuk pembuatan robot pengangkut *box* berdasarkan perbedaan warna. Gambar 3.4 merupakan gambar diagram blok secara keseluruhan dari sebuah sistem.



Gambar 3.4 Diagram blok sistem.

Penjelasan cara kerja gambar diagram blok sistem secara keseluruhan pada setiap bloknya yaitu:

1. Sensor garis berfungsi sebagai sensor untuk mendeteksi jalur atau garis sehingga robot dapat mengetahui arah jalur dan mengikuti garis tersebut. Setelah itu sensor garis dihubungkan dengan mikrokontroler basic stamp 2P40.
2. Mikrokontroler basic stamp 2P40 berfungsi sebagai otak dari robot yang mengendalikan semua komponen yang ada di dalam robot seperti sensor garis, sensor jarak, sensor warna dan yang lainnya. Setelah itu mikrokontroler dihubungkan dengan *driver* motor.

3. *Driver* motor berfungsi sebagai pengendali atau yang menggerakan motor DC. Setelah itu *driver* motor dihubungkan dengan motor DC.
4. Motor DC berfungsi sebagai penggerak roda robot.
5. Sensor jarak sebagai input dihubungkan dengan mikrokontroler, setelah itu mikrokontroler mengeluarakn output. Sensor jarak berfungsi sebagai pendeteksi benda, apabila di depan robot terdapat benda maka robot tidak akan mengambil objek tersebut.
6. Sensor warna sebagai input dihubungkan dengan mikrokontroler, setelah itu mikrokontroler mengeluarkan output. Sensor warna berfungsi sebagai pendeteksi objek berdasarkan perbedaan warna, sehingga robot dapat mengetahi objek warna tersebut.
7. *Gripper* atau penjepit sebagai output dihubungkan dengan mikrokontroler, setelah itu mikrokontroler mengeluarkan input. *Gripper* atau penjepit berfungsi sebagai pengambil objek.
8. *Buzzer* sebagai output dihubungkan dengan mikrokontroler, setelah itu mikrokontroler mengeluarkan input. *Buzzer* berfungsi sebagai alarm pemberitahu ketika di asal tidak ada *box* saat menjalankan robot pertama kali.

3.2.1 Mikrokontroler (Basic Stamp 2p40)

Mikrokontroler yang digunakan pada perancangan robot ini adalah mikrokontroler jenis *basic stamp 2P40* dengan 32 *port* yang dapat digunakan sebagi *input* atau *output*. Pemilihan mikrokontroler jenis ini didasari kemampuannya yang cukup handal dan pemrograman yang tidak terlalu sulit. Pada perancangan robot yang dibuat, penulis menggunakan satu buah mikrokontroler *basic stamp 2p40*.

Berikut ini merupakan konfigurasi *port* yang digunakan mikrokontroler, seperti yang terlihat pada tabel 3.1.

Tabel 3.1 Konfigurasi port yang digunakan pada mikrokontroler basic stamp.

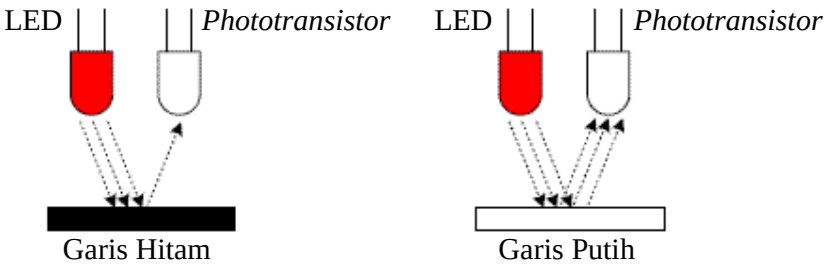
Nama Pin	No. Pin	Fungsi	Keterangan
MAIN I/O	0	*Input*	Sensor garis 1
MAIN I/O	1	*Input*	Sensor garis 2
MAIN I/O	2	*Input*	Sensor garis 3
MAIN I/O	3	*Output*	Driver motor (B -)
MAIN I/O	4	*Input*	Sensor garis 4
MAIN I/O	5	*Output*	Driver motor (B +)
MAIN I/O	6	*Input*	Sensor garis 5
MAIN I/O	7	*Output*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

AUX I/O	11	-	-
AUX I/O	12	*Output*	Gripper Servo 2
AUX I/O	13	-	-
AUX I/O	14	-	-
AUX I/O	15	-	-

3.2.2 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*) *superbright* sebagai pemancar cahaya ke lantai yang kemudian dipantulkan dan diterima oleh *phototransistor*. Untuk lebih jelasnya memahami prinsip kerja sensor garis dapat dilihat pada gambar 3.5.

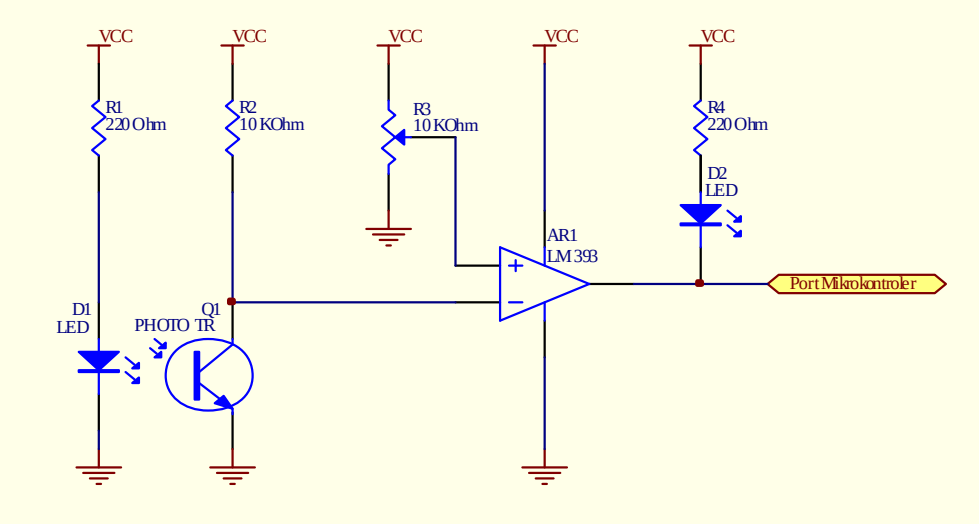


Gambar 3.5 Prinsip kerja sensor garis.

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.

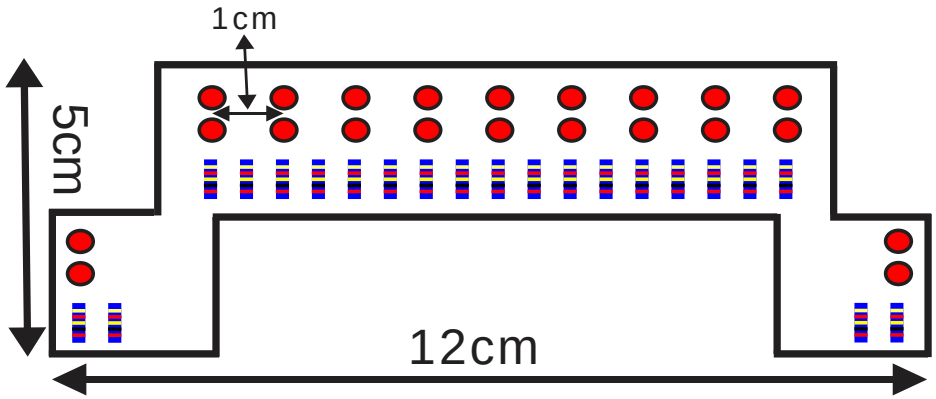
Output dari *phototransistor* yaitu berupa tegangan. Semakin besar intensitas cahaya yang didapat maka makin besar pula tegangan yang dihasilkan. Dengan memanfaatkan *mode* aktif dari *phototransistor* maka ditambahkan sebuah komparator sehingga nilai keluaran dari *phototransistor* yang berupa data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



Gambar 3.6 Rangkaian sensor garis.

Jumlah sensor yang digunakan adalah 10 sensor yang masing-masing output sensornya dihubungkan ke komparator dan mikrokontroler basic stamp 2p40.

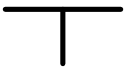
Jarak antar sensor sama dan tata letak berbeda, hal ini bertujuan agar memudahkan dalam mengenali bentuk persimpangan. Untuk lebih jelasnya dapat dilihat pada gambar 3.7.

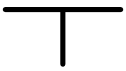
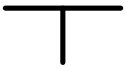
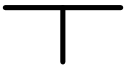


Gambar 3.7 Design bord sensor garis.

Pada sensor garis mempunyai fungsi yang berbeda-beda pada setiap sensor. Untuk lebih jelasnya dapat dilihat pada tabel 3.2

Tabel 3.2 Bentuk belokan dan persimpangan.

Nama	Bentuk Track	Sensor yang Digunakan	Keterangan
Kiri		Sensor 10	Slot 1 dan 3 pada program biru dan hijau
Kanan		Sensor 1	Slot 1 dan 3 pada program biru dan hijau
Lurus		Sensor 2 sampai dengan sensor 9	Line Follower
Putar 180°		Sensor 2 sampai dengan 9	Slot 0,1,2 dan 3 pada program master, biru, hijau dan merah

Kiri		Sensor 2 sampai dengan 9	Slot 0 dan 1 pada program master dan biru
Maju		Sensor 2 sampai dengan 9	Slot 2 pada program hijau
Kanan		Sensor 2 sampai dengan 9	Slot 3 pada program merah

3.2.3 Driver Motor

L298 adalah IC yang dapat digunakan sebagai *driver* motor DC. IC ini menggunakan prinsip kerja *H-Bridge*. Tiap *H-Bridge* dikontrol menggunakan level tegangan TTL yang berasal dari *output* mikrokontroler. L298 dapat mengontrol 2 buah motor DC. Tegangan yang dapat digunakan untuk mengendalikan robot bisa mencapai tegangan 46 V_{DC} dan arus 2 A untuk setiap kanalnya.

Power supply tegangan yang diberikan ke motor dc sebesar 7.4 V_{DC} yang berasal dari baterai Li-Po 2200mA, 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*.

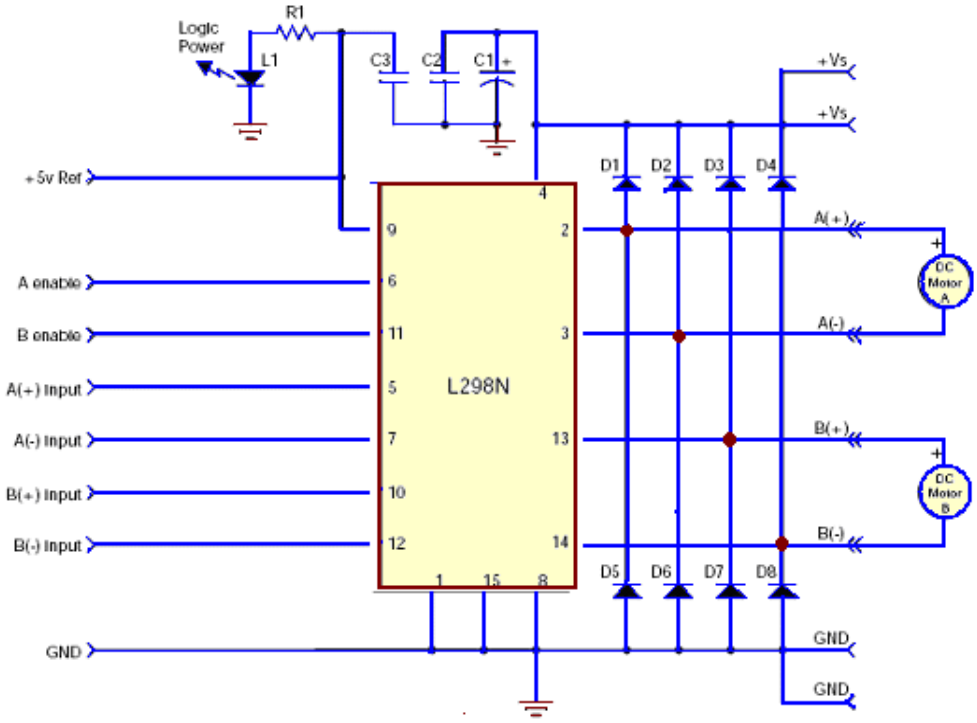
Satu motor dapat dikontrol dengan 3 pin yaitu A*enable*, A+ dan A-. Berikut ini adalah tabel kebenaran untuk salah satu motor.

Tabel 3.3 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*

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



Gambar 3.8 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 3.4 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

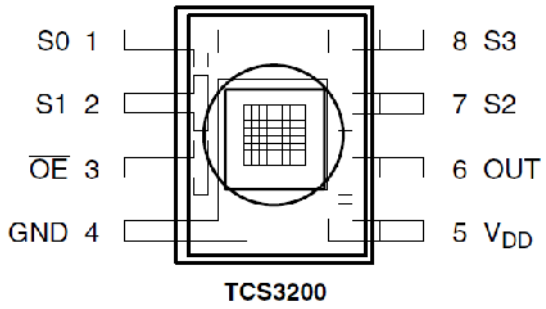
Keterangan: H = *High* = 1 dan L = *Low* = 0

3.2.4 Sensor Warna (TCS 3200)

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 Sensor Warna TCS 3200:

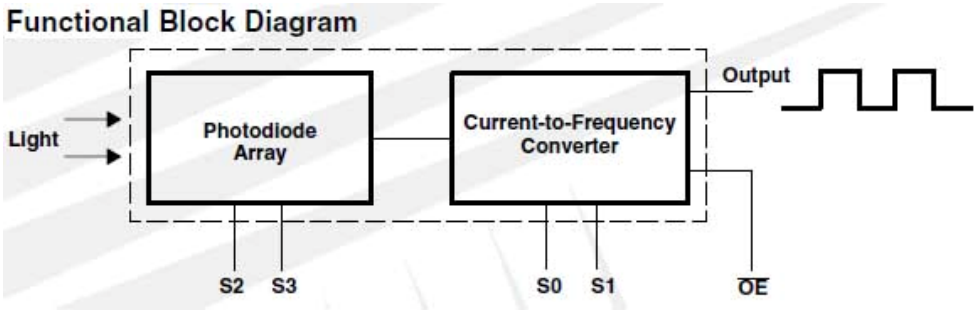
1. Tegangan Kerja : 2.7 - 5.5 V
2. Sensor *Array* : 8 x 8 photo dioda, 16 photodioda filter merah, 16 photodioda *filter* hijau, 16 photodioda tanpa filter.
3. Lensa : 5.3 mm (dalam jarak 25 mm mampu melihat area bujur sangkar dengan sisi 4 mm).



Gambar 3.9 Tataletak pin sensor warna TCS3200.

Tabel 3.5 Penjelasan pada pin sensor warna TCS3200.

Nama	PIN	I/O (Input Output)	Keterangan
GND	4		*Ground*
OE	3	*Input*	*Enable* untuk fo (*active low*)
OUT	6	*Output*	*Output* frekuensi (fo)
S0, S1	1, 2	*Input*	Pemilihan *input* skala frekuensi untuk *output* frekuensi
S2, S3	7, 8	*Input*	Pemilihan jenis *input photodiode*
V_{DD}	5		Sumber Tegangan



Gambar 3.10. 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 memfilter 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. Berikut ini adalah table kombinasi fungsi dari S2 dan S3.

Tabel 3.6 Kombinasi fungsi dari S2 dan S3.

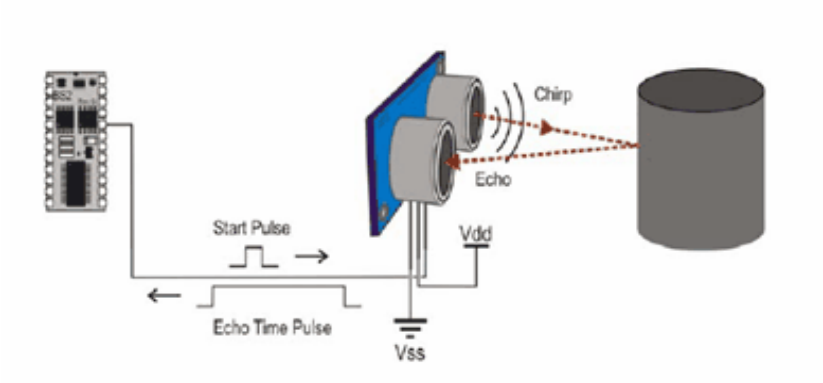
S2	S3	Photodiode yang aktif
0	0	Pemfilter Merah
0	1	Pemfilter Biru
1	0	Tanpa Filter
1	1	Pemfilter Hijau

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.

3.2.5 Sensor Ultrasonik

Alasan saya menggunakan sensor ping hanya mempunyai 3 buah pin yang digunakan. PIN sebelah kiri digunakan untuk *ground*, PIN di tengah digunakan untuk sinyal/ data dan PIN yang sebealah kanan digunakan untuk VCC (tempat masuknya arus).

Jarak dapat diketahui dengan menghitung waktu yang dibutuhkan oleh gelombang ultrasonik sejak awal pengiriman sampai kembali ke penerima gelombang. Berikut ini adalah gambar cara kerja sensor ultrasonic menerima dan mengirim sinyal.



Gambar 3.11 Cara kerja sensor ultrasonik.

Kelebihan sensor ini ialah hanya membutuhkan 1 sinyal (SIG), selain jalur 5V dan *ground*. Sensor ping mendeteksi jarak obyek dengan cara memancarkan gelombang ultrasonik (40 KHz) selama t_{BURST} (200 us) kemudian mendeteksi pantulannya. Sensor ping memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler pengendali (pulsa *trigger* dengan t_{OUT} min 2 us). Spesifikasi sensor ini:

1. Kisaran pengukuran 3cm-3m.
2. *Input trigger –positive* TTL *pulse*, 2 uS min., 5 uS tipikal.
3. *Echo hold off* 750uS dari *fall of trigger pulse.*
4. *Delay before next measurement* 200 uS.
5. *Burst indicator* LED menampilkan aktifitas sensor.

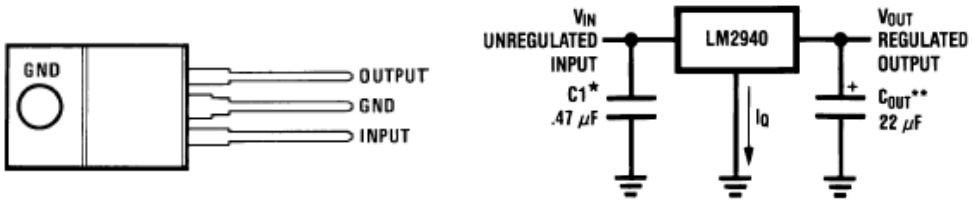
Gelombang ini melalui udara dengan kecepatan 344 m/s, lalu mengenai obyek dan memantul kembali ke sensor. Ping mengeluarkan pulsa *output high* pada pin SIG setelah memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi Ping akan membuat *output low* pada pin SIG. Lebar pulsa *High* (t_{IN}) akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk 2x jarak ukur dengan obyek. Keluaran dari pin SIG ini yang dihubungkan ke salah satu *port* di mikrokontroler.

3.2.6 Catu Daya

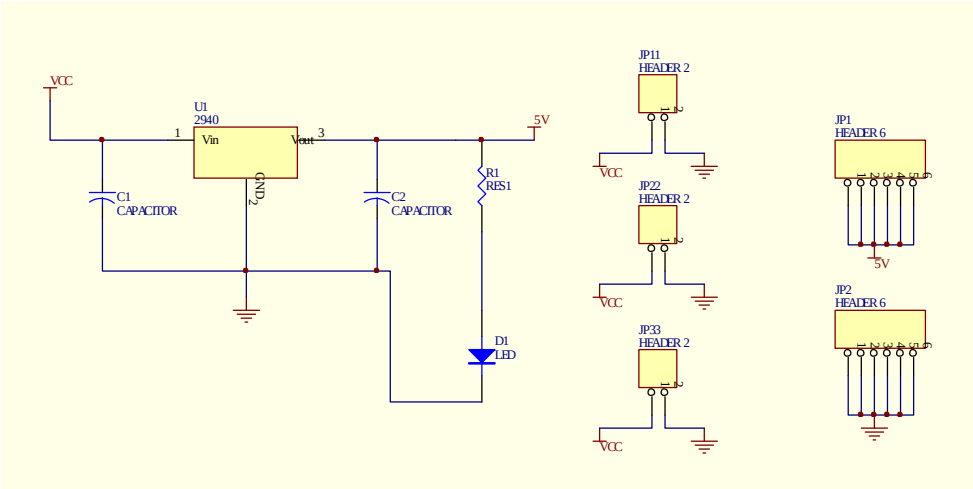
Catu daya merupakan faktor pendukung terpenting, karena hampir keseluruhan komponen *hardware* pada robot membutuhkan tegangan untuk mengoperasikannya. Catu daya yang dibutuhkan sebanyak 2 buah, pertama untuk mencatu mikrokontroler sensor garis, servo dan yang lainnya. Kedua, tegangan actuator berupa motor DC.

Tegangan yang digunakan berasal dari 1 buah baterai *rechargeable* berjenis *Lithium polymer* yang mempunyai nilai tegangan 7,4 volt dengan arus 2200 mAh. Untuk supply tegangan ke aktuator berupa motor DC, baterai dihungkan langsung ke motor DC sedangkan untuk supply tegangan ke mikrokontroler dan sensor garis diperlukan IC regulator untuk mendapatkan

tegangan 5 volt, IC tersebut yaitu IC LM2940. Berikut ini adalah gambar IC LM2940, skematik regulator dan rangkaian sumber tegangan.



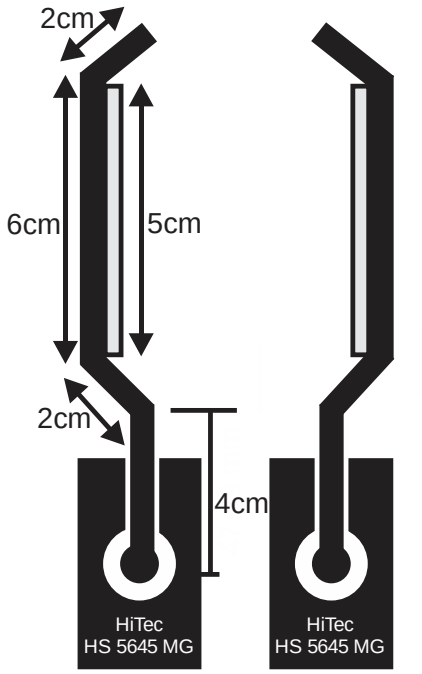
Gambar 3.12 (a) IC LM2940 *(b) Skematik regulator 5 volt*



Gambar 3.13 Rangkaian sumber tegangan robot.

3.2.7 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.

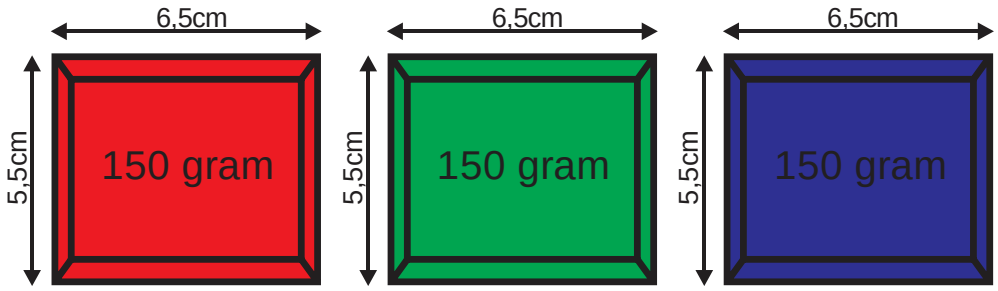


Gambar 3.14 Desaign gripper dengan dua servo.

Dari gambar diatas terlihat dimensi dari *gripper*. Pada *gripper* terdapat 2 mini servo sebagai penggerak *gripper*.

3.2.8 BOX

Box adalah objek yang akan digunakan untuk mengambil benda berupa objek *box* yang terdiri dari 3 warna *box* yaitu warna *box* merah, warna *box* hijau dan warna *box* biru. Box tersebut mempunyai beban berat sekitar ± 150 gram dan panjang *box* ± 6,5 cm serta lebar *box* yaitu ± 5,5 cm. Berikut gambar objek berupa *box* seperti gambar dibawah ini.



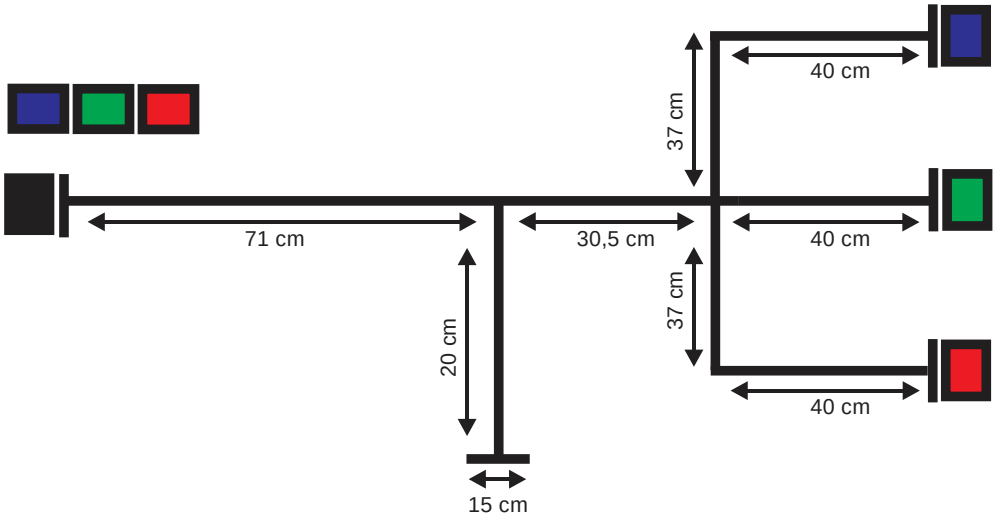
Gambar 3.15 Desaign box.

3.2.9 Miniatur Track Robot

Berikut ini adalah bentuk *track* yang akan digunakan dengan spesifikasi sebagai berikut:

1. Bahan yang digunakan dari kertas duplex dua buah dengan cara di tempelkan menggunakan solasi bening.
2. Untuk *track* atau garis dalam lintasan berwarna hitam menggunakan solasi warna hitam berukuran lebar 1,3 cm.
3. Panjang *track* ± 140,5 cm dan lebar *track* 84 cm.

Gambar 3.16 memperlihatkan miniature *track* yang akan digunakan dalam robot berjalan.



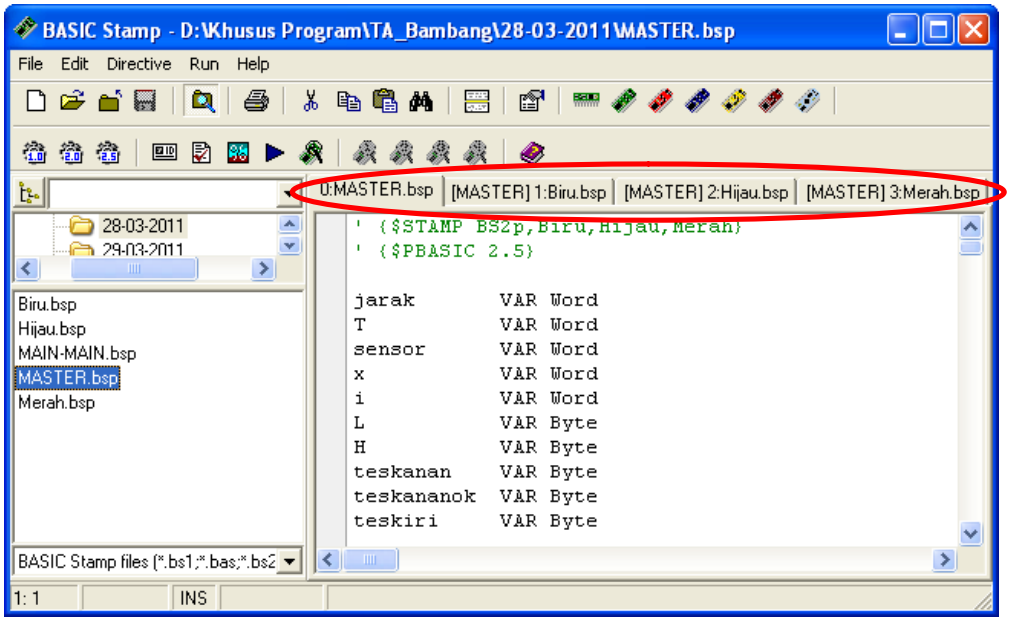
Gambar 3.16 Miniature track robot

3.3 Perancangan Perangkat Lunak (Software)

Setelah dalam pembuatan perangkat keras selesai, bagian yang paling penting dalam pembuatan tugas akhir ini yaitu dalam merancang suatu perangkat lunak. Perancangan algoritma sangat penting untuk setiap sistem kendali. Sehingga dibuat algoritma keseluruhan untuk sistem kendali robot agar dapat berjalan sesuai dengan keinginan. Dalam tugas akhir ini, perancangan flowchart dibagi menjadi empat bagian yaitu:

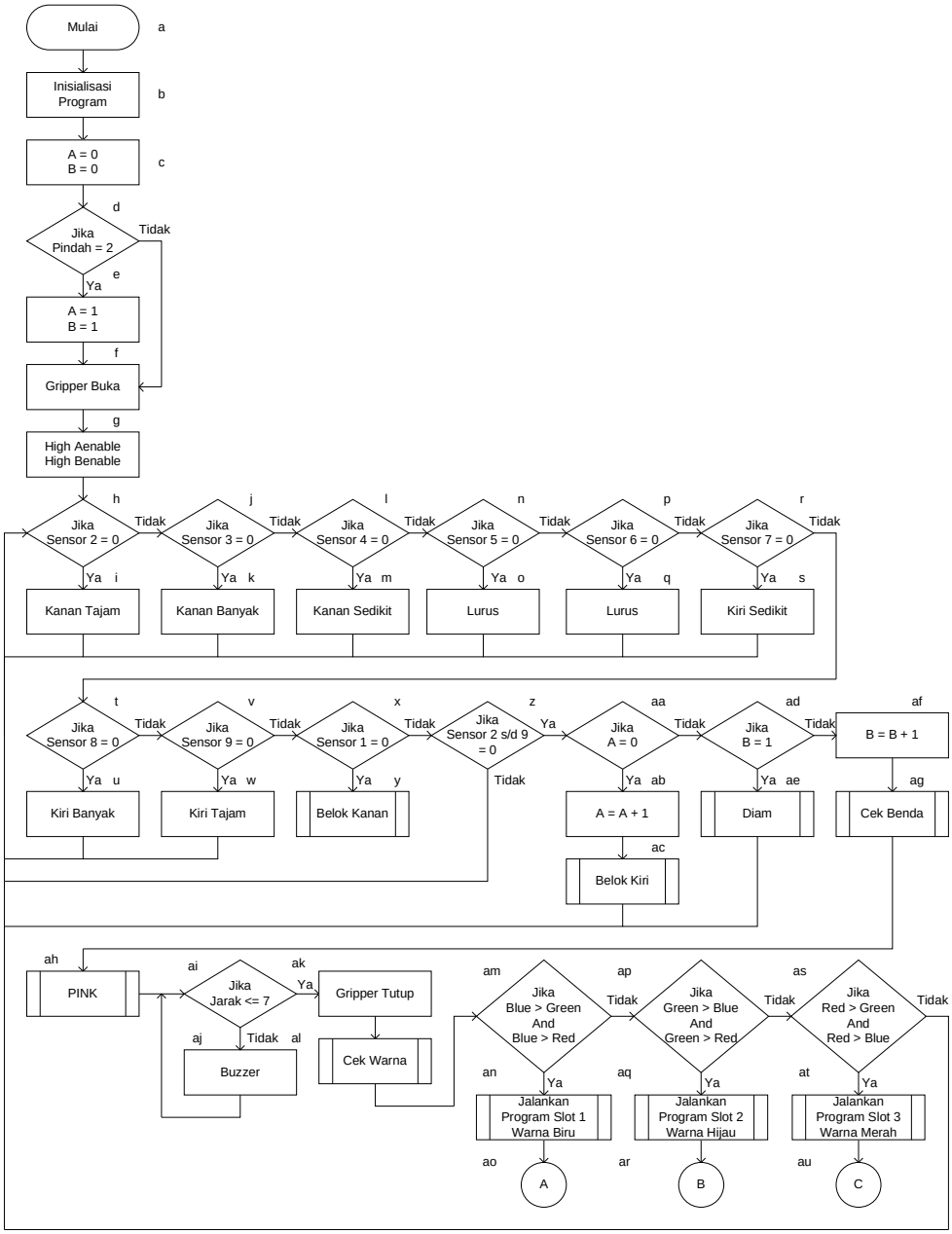
1. Program Master
2. Program Biru
3. Program Hijau
4. Program Merah

Berikut adalah program yang terdiri dari program master di slot 0, program biru di slot 1, program hijau di slot 2 dan program merah di slot 3.



Gambar 3.17 Tampilan program yang terdiri dari 4 slot.

Berikut adalah algoritma utama dalam bentuk *flowchart* yang terdiri dari 4 bagian *flowchart* seperti dibawah ini:



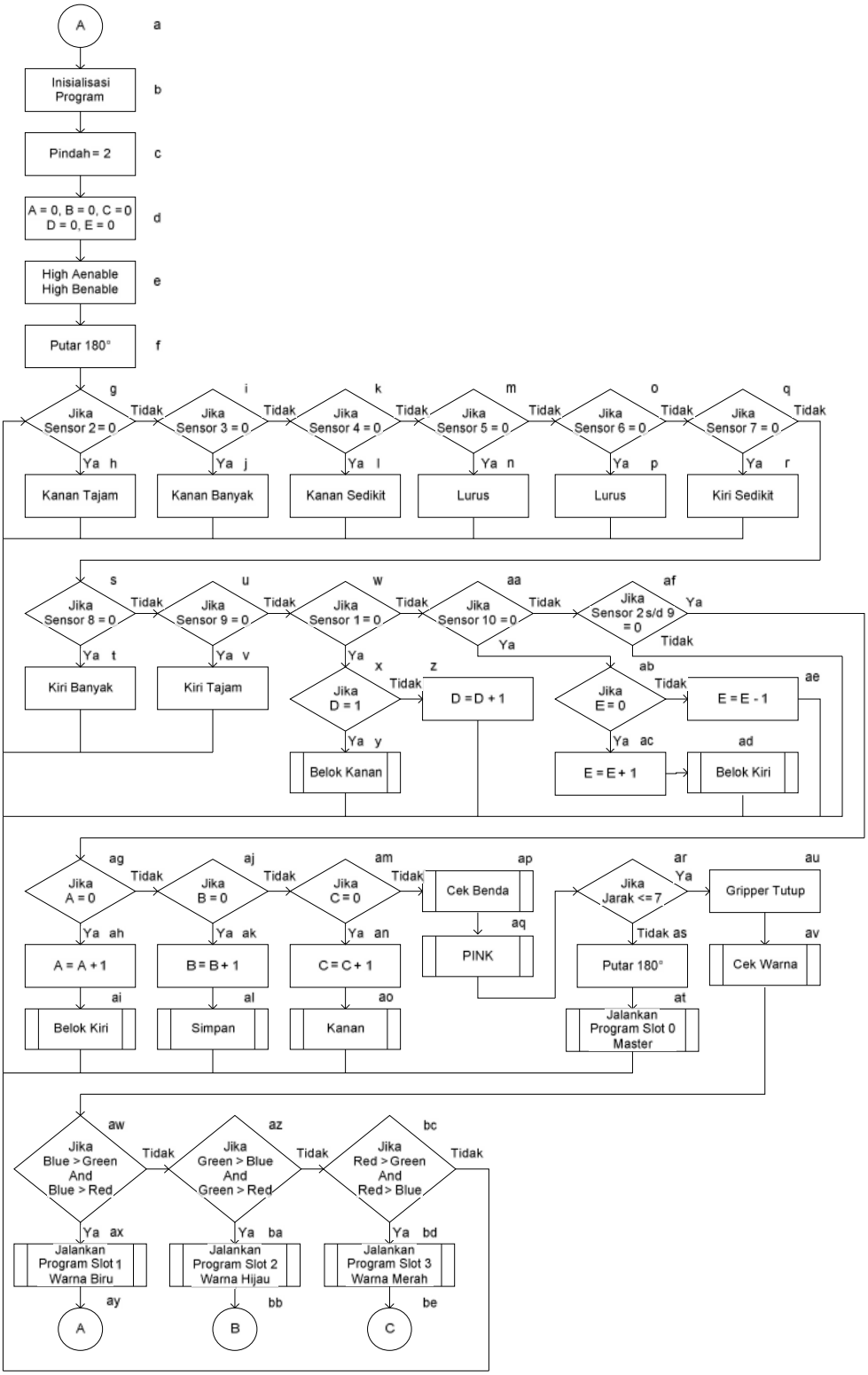
Gambar 3.18 Flowchart program master.

Tabel 3.7 Penjelasan flowchart program master.

No	Keterangan
a	Memulai awal program.
b	Pendeklarasian program. Pengaturan nama kompiler, konfigurasi pin-pin dan deklarasi variabel.
c	Memberikan nilai awal mulai dari A = 0 dan B = 0.
d	Pengecekan apakah variable pindah = 2.
e	Apabila pengecekan variable pindah = 2 benar, maka nilai A = 1 dan nilai B = 1.
f	Kondisi awal robot membuka *gripper*.
g	Kondisi driver motor jalan, terdiri dari Aen dan Ben.
h	Pengecekan jika sensor garis yang di Pin 0 = 0 .
i	Pengecekan jika sensor garis yang di Pin 0 = 0 maka robot jalan ke kiri tajam.
j	Pengecekan jika sensor garis yang di Pin 2 = 0.
k	Pengecekan jika sensor garis yang di Pin 2 = 0 maka robot jalan ke kiri banyak.
l	Pengecekan jika sensor garis yang di Pin 4 = 0.
m	Pengecekan jika sensor garis yang di Pin 4 = 0 maka robot jalan ke kiri sedikit.
n	Pengecekan jika sensor garis yang di Pin 6 = 0.
o	Pengecekan jika sensor garis yang di Pin 6 = 0 maka robot jalan lurus.
p	Pengecekan jika sensor garis yang di Pin 8 = 0.
q	Pengecekan jika sensor garis yang di Pin 8 = 0 maka robot jalan lurus.
r	Pengecekan jika sensor garis yang di Pin 10 = 0.
s	Pengecekan jika sensor garis yang di Pin 10 = 0 maka robot jalan ke kanan sedikit.

t	Pengecekan jika sensor garis yang di Pin 12 = 0.
u	Pengecekan jika sensor garis yang di Pin 12 = 0 maka robot jalan ke kanan banyak.
v	Pengecekan jika sensor garis yang di Pin 14 = 0.
w	Pengecekan jika sensor garis yang di Pin 14 = 0 maka robot jalan ke kanan tajam.
x	Pengecekan jika sensor garis yang di Pin 1 = 0
y	Pengecekan jika sensor garis yang di Pin 1 = 0 maka robot jalan belok kanan 90º.
z	Pengecekan jika sensor garis yang di Pin 00000000=0
aa	Pengecekan jika nilai A = 0.
ab	Pengecekan jika nilai A = 0 maka A = A + 1.
ac	Setelah itu robot berjalan belok kiri 90º.
ad	Pengecekan jika nilai B = 0.
ae	Pengecekan jika nilai B = 0 maka panggil prosedur diam.
af	Pengecekan jika nilai B ≠ 0. Maka nilai B = B + 1.
ag	Panggil prosedur cek benda.
ah	Panggil prosedur PINK.
ai	Pengecekan jika nilai jarak <= 7.
aj	Pengecekan jika nilai jarak > 7. Maka buzzer bunyi member peringatan.
ak	Pengecekan jika nilai jarak <= 7. Maka *gripper* mencapit *box.*
al	Panggil prosedur cek warna.
am	Pengecekan jika nilai blue > green dan blue > red

an	Pengecekan jika nilai blue > green dan blue > red. Maka jalankan program slot 1. Program jalannya robot menuju tujuan warna biru.
ao	Lompat ke flowchart selanjutnya A.
ap	Pengecekan jika nilai green > blue dan green > red
aq	Pengecekan jika nilai green > blue dan green > red. Maka jalankan program slot 2. Program jalannya robot menuju tujuan warna hijau.
ar	Lompat ke flowchart selanjutnya B.
as	Pengecekan jika nilai red > blue dan red > green.
at	Pengecekan jika nilai red > blue dan red > green. Maka jalankan program slot 3. Program jalannya robot menuju tujuan warna merah.
au	Lompat ke flowchart selanjutnya C.



Gambar 3.19 Flowchart program biru.

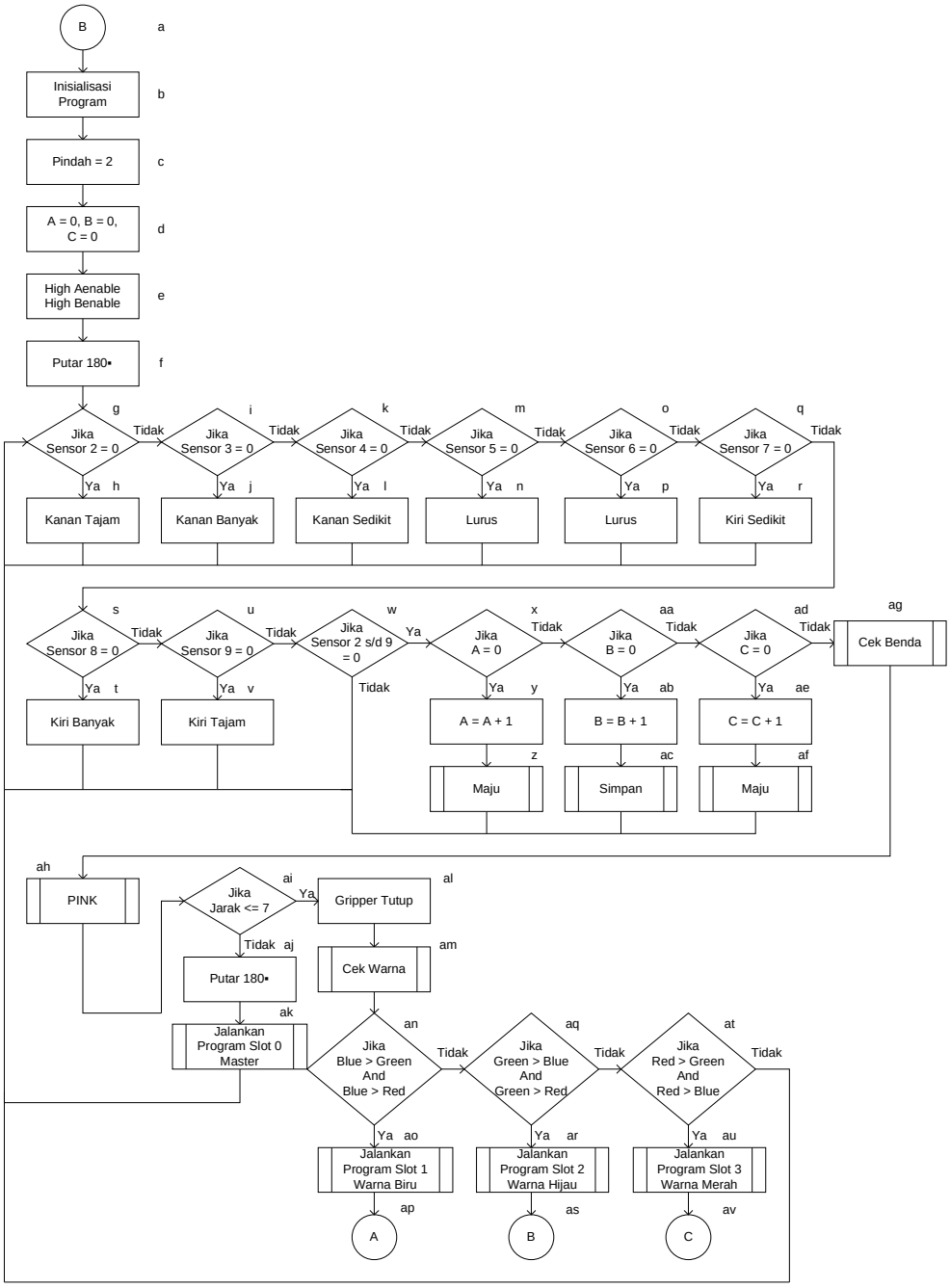
Tabel 3.8 Penjelasan flowchart program biru.

No	Keterangan
a	Memulai awal program A.
b	Pendeklarasian program. Pengaturan nama kompiler, konfigurasi pin-pin dan deklarasi variabel.
c	Memberikan nilai variable pindah = 2.
d	Memberikan nilai awal mulai dari A = 0, B = 0, C = 0, D = 0 dan E = 0.
e	Kondisi driver motor jalan, terdiri dari Aen dan Ben.
f	Setelah itu robot berputar 180º ke kanan.
g	Pengecekan jika sensor garis yang di Pin 0 = 0 .
h	Pengecekan jika sensor garis yang di Pin 0 = 0 maka robot jalan ke kiri tajam.
i	Pengecekan jika sensor garis yang di Pin 2 = 0.
j	Pengecekan jika sensor garis yang di Pin 2 = 0 maka robot jalan ke kiri banyak.
k	Pengecekan jika sensor garis yang di Pin 4 = 0.
l	Pengecekan jika sensor garis yang di Pin 4 = 0 maka robot jalan ke kiri sedikit.
m	Pengecekan jika sensor garis yang di Pin 6 = 0.
n	Pengecekan jika sensor garis yang di Pin 6 = 0 maka robot jalan lurus.
o	Pengecekan jika sensor garis yang di Pin 8 = 0.
p	Pengecekan jika sensor garis yang di Pin 8 = 0 maka robot jalan lurus.
q	Pengecekan jika sensor garis yang di Pin 10 = 0.
r	Pengecekan jika sensor garis yang di Pin 10 = 0 maka robot jalan ke kanan sedikit.
s	Pengecekan jika sensor garis yang di Pin 12 = 0.

t	Pengecekan jika sensor garis yang di Pin 12 = 0 maka robot jalan ke kanan banyak.
u	Pengecekan jika sensor garis yang di Pin 14 = 0.
v	Pengecekan jika sensor garis yang di Pin 14 = 0 maka robot jalan ke kanan tajam.
w	Pengecekan jika sensor garis yang di Pin 1 = 0.
x	Pengecekan jika nilai D = 1.
y	Pengecekan jika nilai D = 1. Maka setelah itu robot akan belok 90º ke kanan.
z	Pengecekan jika nilai D ≠ 1. Maka nilai D = D + 1.
aa	Pengecekan jika sensor garis yang di Pin 15 = 0.
ab	Pengecekan jika nilai E = 1.
ac	Pengecekan jika nilai E = 1. Maka nilai E = E + 1.
ad	Pengecekan jika nilai E = 1. Maka setelah itu robot akan belok 90º ke kiri.
ae	Pengecekan jika nilai E ≠ 1. Maka nilai E = E – 1.
af	Pengecekan jika sensor garis yang di Pin 00000000=0.
ag	Pengecekan jika nilai A = 0.
ah	Pengecekan jika nilai A = 0. Maka nilai A = A + 1
ai	Pengecekan jika nilai A = 0. Maka setelah itu robot akan belok 90º ke kiri.
aj	Pengecekan jika nilai B = 0.
ak	Pengecekan jika nilai B= 0. Maka nilai B = B + 1.
al	Pengecekan jika nilai B = 0. Maka setelah itu robot akan panggil prosedur simpan.

am	Pengecekan jika nilai C = 0.
an	Pengecekan jika nilai C= 0. Maka nilai C = C + 1.
ao	Pengecekan jika nilai C = 0. Maka setelah itu robot akan belok 90º ke kanan.
ap	Pengecekan jika nilai C ≠ 0. Maka panggil prosedur cek benda.
aq	Pengecekan jika nilai C ≠ 0. Maka panggil prosedur PINK.
ar	Pengecekan jika nilai jarak <= 7.
as	Pengecekan jika nilai C ≠ 0. Maka setelah itu robot akan berputar 180º ke kanan.
at	Jalankan program slot 0. Kembali lagi ke program master atau ke program utama.
au	Pengecekan jika nilai jarak <= 7. Maka *gripper* mencapit *box.*
av	Panggil prosedur cek warna.
aw	Pengecekan jika nilai blue > green dan blue > red
ax	Pengecekan jika nilai blue > green dan blue > red. Maka jalankan program slot 1. Program jalannya robot menuju tujuan warna biru.
ay	Lompat ke flowchart selanjutnya A.
az	Pengecekan jika nilai green > blue dan green > red
ba	Pengecekan jika nilai green > blue dan green > red. Maka jalankan program slot 2. Program jalannya robot menuju tujuan warna hijau.
bb	Lompat ke flowchart selanjutnya B.
bc	Pengecekan jika nilai red > blue dan red > green.

bd	Pengecekan jika nilai red > blue dan red > green. Maka jalankan program slot 3. Program jalannya robot menuju tujuan warna merah.
be	Lompat ke flowchart selanjutnya C.



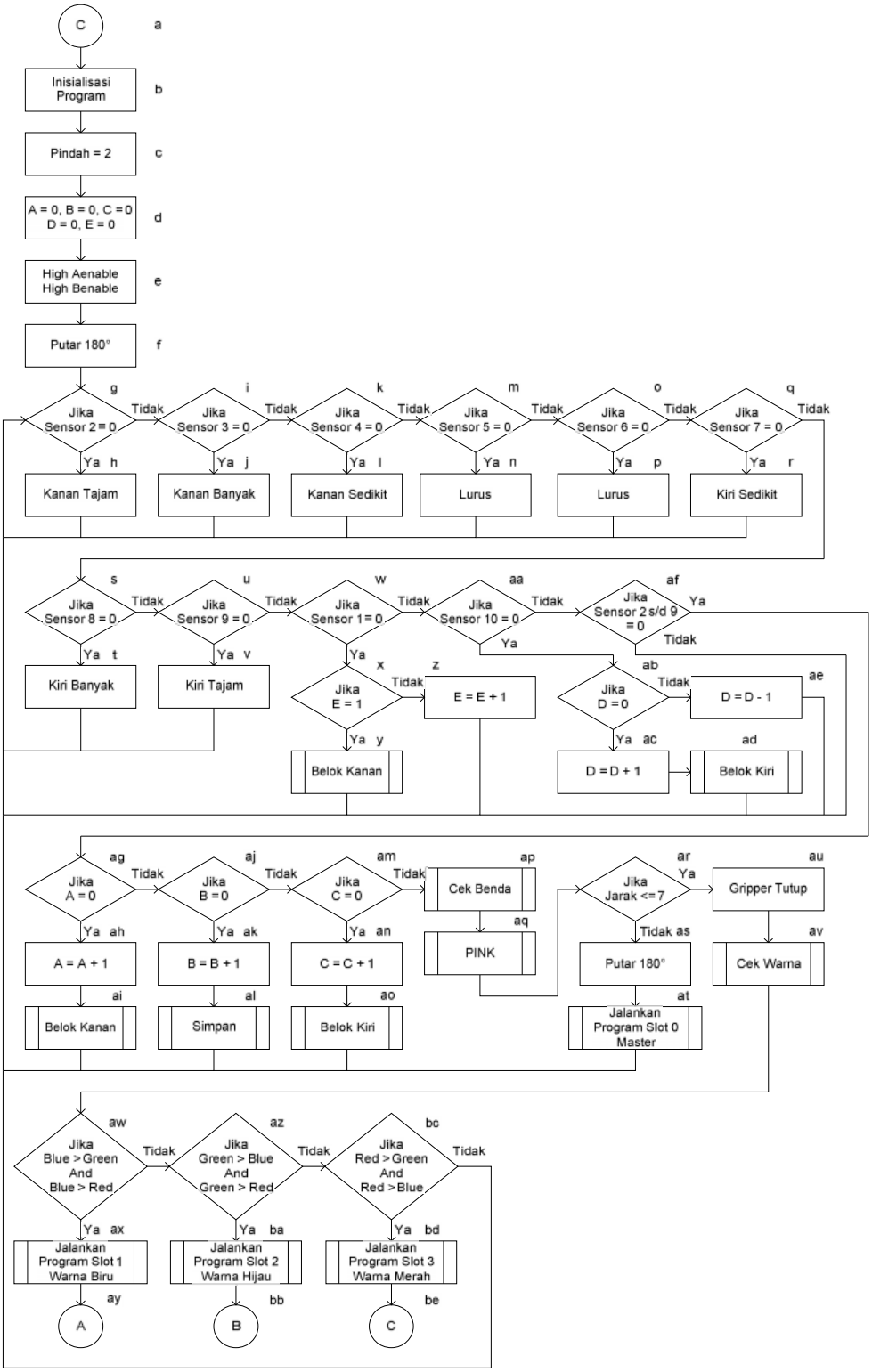
Gambar 3.20 Flowchart program hijau.

Tabel 3.9 Penjelasan flowchart program hijau.

No	Keterangan
a	Memulai awal program B.
b	Pendeklarasian program. Pengaturan nama kompiler, konfigurasi pin-pin dan deklarasi variabel.
c	Memberikan nilai variable pindah = 2.
d	Memberikan nilai awal mulai dari A = 0, B = 0 dan C = 0
e	Kondisi driver motor jalan, terdiri dari Aen dan Ben.
f	Setelah itu robot berputar 180º ke kanan.
g	Pengecekan jika sensor garis yang di Pin 0 = 0 .
h	Pengecekan jika sensor garis yang di Pin 0 = 0 maka robot jalan ke kiri tajam.
i	Pengecekan jika sensor garis yang di Pin 2 = 0.
j	Pengecekan jika sensor garis yang di Pin 2 = 0 maka robot jalan ke kiri banyak.
k	Pengecekan jika sensor garis yang di Pin 4 = 0.
l	Pengecekan jika sensor garis yang di Pin 4 = 0 maka robot jalan ke kiri sedikit.
m	Pengecekan jika sensor garis yang di Pin 6 = 0.
n	Pengecekan jika sensor garis yang di Pin 6 = 0 maka robot jalan lurus.
o	Pengecekan jika sensor garis yang di Pin 8 = 0.
p	Pengecekan jika sensor garis yang di Pin 8 = 0 maka robot jalan lurus.
q	Pengecekan jika sensor garis yang di Pin 10 = 0.
r	Pengecekan jika sensor garis yang di Pin 10 = 0 maka robot jalan ke kanan sedikit.
s	Pengecekan jika sensor garis yang di Pin 12 = 0.

t	Pengecekan jika sensor garis yang di Pin 12 = 0 maka robot jalan ke kanan banyak.
u	Pengecekan jika sensor garis yang di Pin 14 = 0.
v	Pengecekan jika sensor garis yang di Pin 14 = 0 maka robot jalan ke kanan tajam.
w	Pengecekan jika sensor garis yang di Pin 00000000=0.
x	Pengecekan jika nilai A = 0.
y	Pengecekan jika nilai A = 0. Maka nilai A = A + 1
z	Pengecekan jika nilai A = 0. Maka setelah itu robot akan maju.
aa	Pengecekan jika nilai B = 0.
ab	Pengecekan jika nilai B= 0. Maka nilai B = B + 1.
ac	Pengecekan jika nilai B = 0. Maka setelah itu robot akan panggil prosedur simpan.
ad	Pengecekan jika nilai C = 0.
ae	Pengecekan jika nilai C= 0. Maka nilai C = C + 1.
af	Pengecekan jika nilai C = 0. Maka setelah itu robot akan maju.
ag	Pengecekan jika nilai C ≠ 0. Maka panggil prosedur cek benda.
ah	Pengecekan jika nilai C ≠ 0. Maka panggil prosedur PINK.
ai	Pengecekan jika nilai jarak <= 7.
aj	Pengecekan jika nilai C ≠ 0. Maka setelah itu robot akan berputar 180º ke kanan.
ak	Jalankan program slot 0. Kembali lagi ke program master atau ke program utama.

al	Pengecekan jika nilai jarak <= 7. Maka *gripper* mencapit *box.*
am	Panggil prosedur cek warna.
an	Pengecekan jika nilai blue > green dan blue > red
ao	Pengecekan jika nilai blue > green dan blue > red. Maka jalankan program slot 1. Program jalannya robot menuju tujuan warna biru.
ap	Lompat ke flowchart selanjutnya A.
aq	Pengecekan jika nilai green > blue dan green > red
ar	Pengecekan jika nilai green > blue dan green > red. Maka jalankan program slot 2. Program jalannya robot menuju tujuan warna hijau.
as	Lompat ke flowchart selanjutnya B.
at	Pengecekan jika nilai red > blue dan red > green.
au	Pengecekan jika nilai red > blue dan red > green. Maka jalankan program slot 3. Program jalannya robot menuju tujuan warna merah.
av	Lompat ke flowchart selanjutnya C.



Gambar 3.21 Flowchart program merah.

Tabel 3.10 Penjelasan flowchart program merah.

No	Keterangan
a	Memulai awal program C.
b	Pendeklarasian program. Pengaturan nama kompiler, konfigurasi pin-pin dan deklarasi variabel.
c	Memberikan nilai variable pindah = 2.
d	Memberikan nilai awal mulai dari A = 0, B = 0, C = 0, D = 0 dan E = 0.
e	Kondisi driver motor jalan, terdiri dari Aen dan Ben.
f	Setelah itu robot berputar 180º ke kanan.
g	Pengecekan jika sensor garis yang di Pin 0 = 0 .
h	Pengecekan jika sensor garis yang di Pin 0 = 0 maka robot jalan ke kiri tajam.
i	Pengecekan jika sensor garis yang di Pin 2 = 0.
j	Pengecekan jika sensor garis yang di Pin 2 = 0 maka robot jalan ke kiri banyak.
k	Pengecekan jika sensor garis yang di Pin 4 = 0.
l	Pengecekan jika sensor garis yang di Pin 4 = 0 maka robot jalan ke kiri sedikit.
m	Pengecekan jika sensor garis yang di Pin 6 = 0.
n	Pengecekan jika sensor garis yang di Pin 6 = 0 maka robot jalan lurus.
o	Pengecekan jika sensor garis yang di Pin 8 = 0.
p	Pengecekan jika sensor garis yang di Pin 8 = 0 maka robot jalan lurus.
q	Pengecekan jika sensor garis yang di Pin 10 = 0.
r	Pengecekan jika sensor garis yang di Pin 10 = 0 maka robot jalan ke kanan sedikit.
s	Pengecekan jika sensor garis yang di Pin 12 = 0.

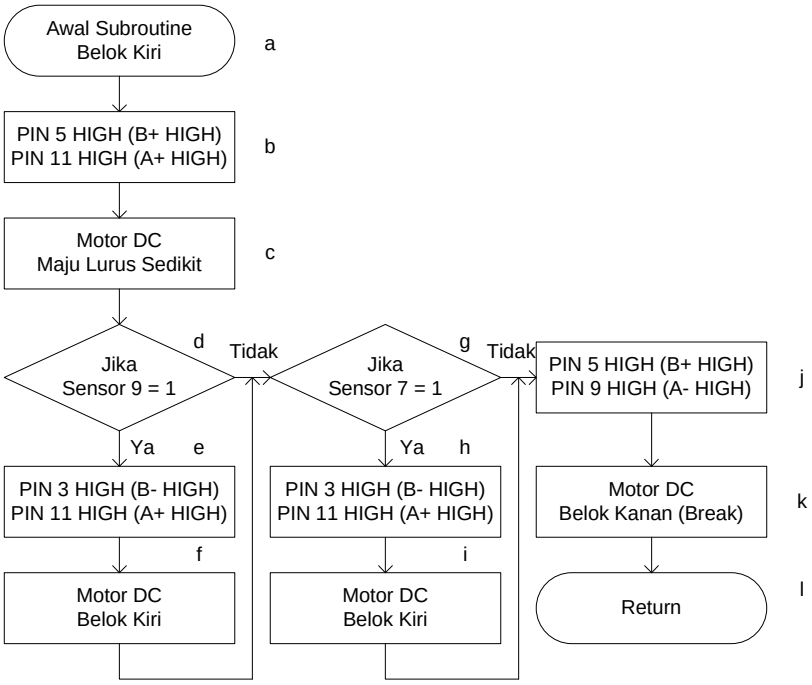
t	Pengecekan jika sensor garis yang di Pin 12 = 0 maka robot jalan ke kanan banyak.
u	Pengecekan jika sensor garis yang di Pin 14 = 0.
v	Pengecekan jika sensor garis yang di Pin 14 = 0 maka robot jalan ke kanan tajam.
w	Pengecekan jika sensor garis yang di Pin 1 = 0.
x	Pengecekan jika nilai E = 1.
y	Pengecekan jika nilai E = 1. Maka setelah itu robot akan belok 90º ke kanan.
z	Pengecekan jika nilai E ≠ 1. Maka nilai E = E + 1.
aa	Pengecekan jika sensor garis yang di Pin 15 = 0.
ab	Pengecekan jika nilai D = 0.
ac	Pengecekan jika nilai D = 0. Maka nilai D = D + 1.
ad	Pengecekan jika nilai D = 0. Maka setelah itu robot akan belok 90º ke kiri.
ae	Pengecekan jika nilai D ≠ 0. Maka nilai D = D – 1.
af	Pengecekan jika sensor garis yang di Pin 00000000=0.
ag	Pengecekan jika nilai A = 0.
ah	Pengecekan jika nilai A = 0. Maka nilai A = A + 1
ai	Pengecekan jika nilai A = 0. Maka setelah itu robot akan belok 90º ke kanan.
aj	Pengecekan jika nilai B = 0.
ak	Pengecekan jika nilai B= 0. Maka nilai B = B + 1.
al	Pengecekan jika nilai B = 0. Maka setelah itu robot akan panggil prosedur simpan.

am	Pengecekan jika nilai C = 0.
an	Pengecekan jika nilai C= 0. Maka nilai C = C + 1.
ao	Pengecekan jika nilai C = 0. Maka setelah itu robot akan belok 90º ke kiri.
ap	Pengecekan jika nilai C ≠ 0. Maka panggil prosedur cek benda.
aq	Pengecekan jika nilai C ≠ 0. Maka panggil prosedur PINK.
ar	Pengecekan jika nilai jarak <= 7.
as	Pengecekan jika nilai C ≠ 0. Maka setelah itu robot akan berputar 180º ke kanan.
at	Jalankan program slot 0. Kembali lagi ke program master atau ke program utama.
au	Pengecekan jika nilai jarak <= 7. Maka *gripper* mencapit *box.*
av	Panggil prosedur cek warna.
aw	Pengecekan jika nilai blue > green dan blue > red
ax	Pengecekan jika nilai blue > green dan blue > red. Maka jalankan program slot 1. Program jalannya robot menuju tujuan warna biru.
ay	Lompat ke flowchart selanjutnya A.
az	Pengecekan jika nilai green > blue dan green > red
ba	Pengecekan jika nilai green > blue dan green > red. Maka jalankan program slot 2. Program jalannya robot menuju tujuan warna hijau.
bb	Lompat ke flowchart selanjutnya B.
bc	Pengecekan jika nilai red > blue dan red > green.

bd	Pengecekan jika nilai red > blue dan red > green. Maka jalankan program slot 3. Program jalannya robot menuju tujuan warna merah.
be	Lompat ke flowchart selanjutnya C.

3.3.1 Subroutine Pergerakan Robot

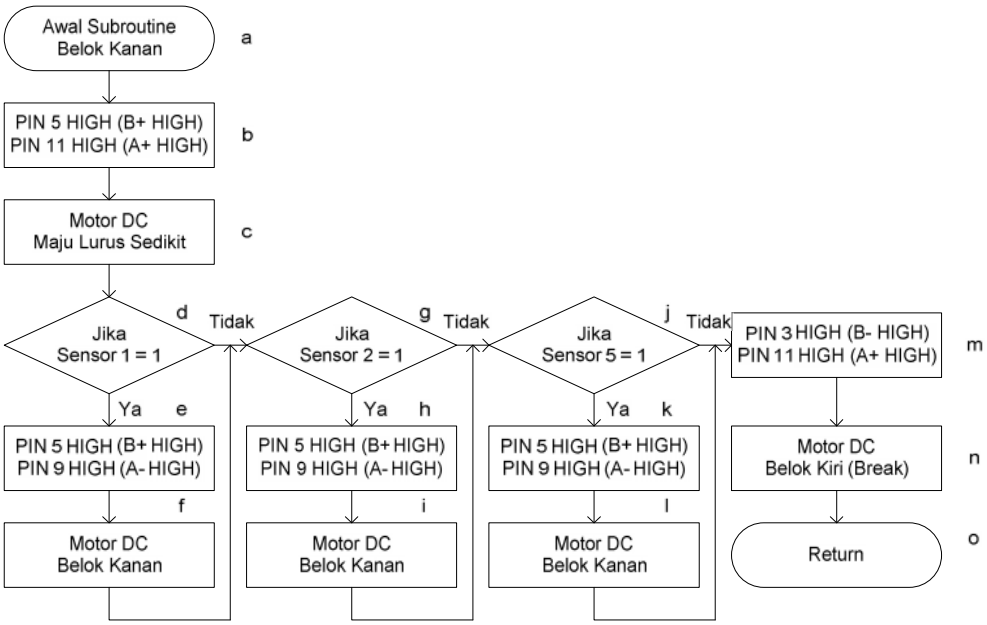
Pada *Subroutine* ini terdiri beberapa instuksi agar robot dapat berputar atau belok, maju dan berhenti. *Subroutine* ini dipanggil ketika robot menemui belokan dan persimpangan. Berikut adalah gambar flowchart subroutine:



Gambar 3.22 Diagram alir subroutine belok kiri.

Tabel 3.11 Penjelasan diagram alir subroutine belok kiri.

No	Keterangan
a	Memulai awal *subroutine* belok kiri.
b	Inisialisasi pin yang dihubungkan ke B+ dan A+.
c	Inisialisasi pin yang dihubungkan ke B+ dan A+ agar roda berputar maju lurus.
d	Jika sensor 9 = 1 maka berhenti belok kiri.
e	Inisialisasi pin yang dihubungkan ke B- dan A+.
f	Inisialisasi pin yang dihubungkan ke B- dan A+ agar roda berputar belok kiri.
g	Jika sensor 7 = 1 maka berhenti belok kiri.
h	Inisialisasi pin yang dihubungkan ke B- dan A+.
i	Inisialisasi pin yang dihubungkan ke B- dan A+ agar roda berputar belok kiri.
j	Inisialisasi pin yang dihubungkan ke B- dan A+
k	Inisialisasi pin yang dihubungkan ke B+ dan A- agar roda berputar belok kanan memberhentikan belok kiri (break).
l	Keluar dari *subroutine* dan kembali mengeksekusi instruksi berikutnya pada program utama.

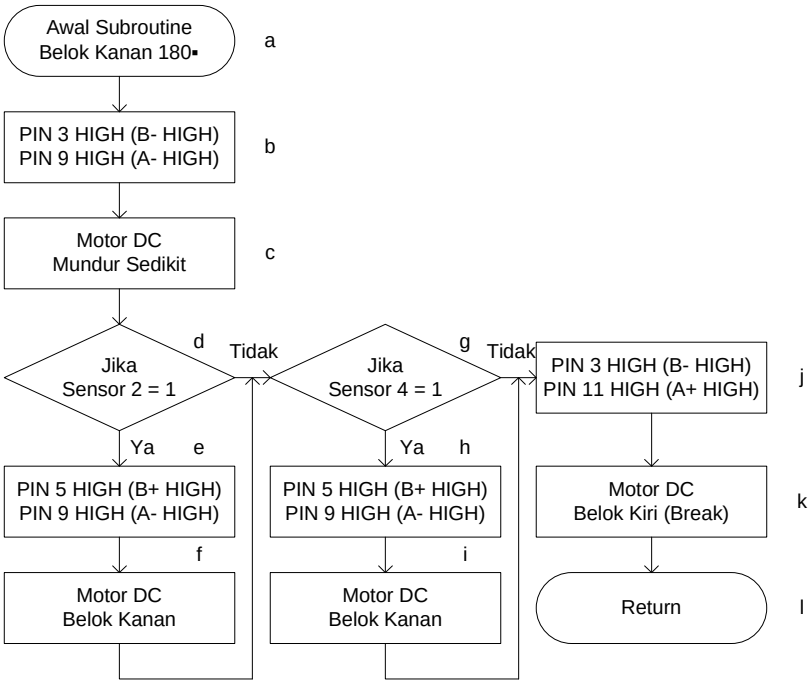


Gambar 3.23 Diagram alir subroutine belok kanan.

Tabel 3.12 Penjelasan diagram alir subroutine belok kanan.

No	Keterangan
a	Memulai awal *subroutine* belok kanan.
b	Inisialisasi pin yang dihubungkan ke B+ dan A+.
c	Inisialisasi pin yang dihubungkan ke B+ dan A+ agar roda berputar maju lurus.
d	Jika sensor 1 = 1 maka berhenti belok kanan.
e	Inisialisasi pin yang dihubungkan ke B+ dan A-.
f	Inisialisasi pin yang dihubungkan ke B+ dan A- agar roda berputar belok kanan.
g	Jika sensor 2 = 1 maka berhenti belok kanan.
h	Inisialisasi pin yang dihubungkan ke B+ dan A-.
i	Inisialisasi pin yang dihubungkan ke B+ dan A- agar roda berputar belok kanan.
j	Jika sensor 5 = 1 maka berhenti belok kanan.

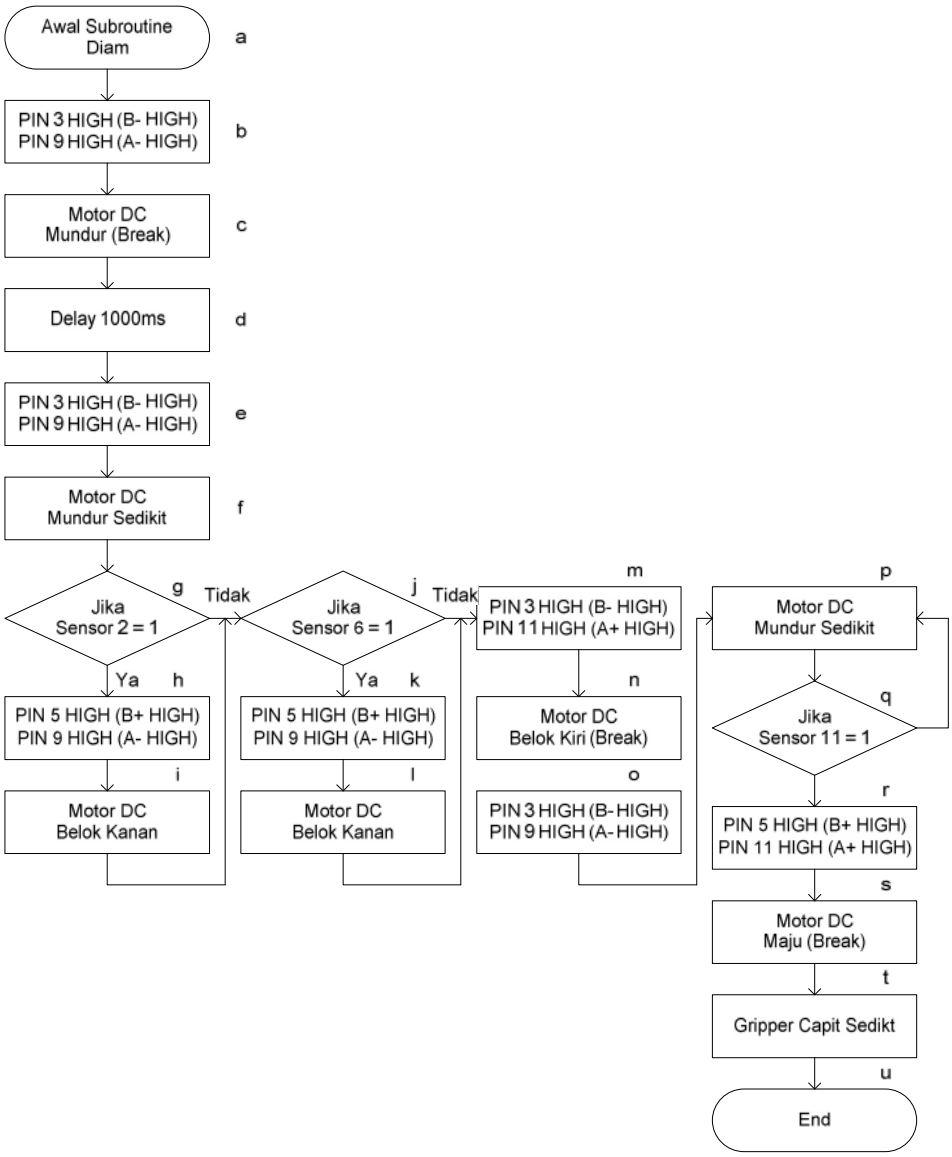
k	Inisialisasi pin yang dihubungkan ke B+ dan A-.
l	Inisialisasi pin yang dihubungkan ke B+ dan A- agar roda berputar belok kanan.
m	Inisialisasi pin yang dihubungkan ke B- dan A+
n	Inisialisasi pin yang dihubungkan ke B- dan A+ agar roda berputar belok kiri memberhentikan belok kanan (break).
o	Keluar dari *subroutine* dan kembali mengeksekusi instruksi berikutnya pada program utama.



Gambar 3.24 Diagram alir subroutine belok kanan 180°.

Tabel 3.13 Penjelasan diagram alir subroutine belok kanan180°.

No	Keterangan
a	Memulai awal *subroutine* belok kanan 180°.
b	Inisialisasi pin yang dihubungkan ke B- dan A-.
c	Inisialisasi pin yang dihubungkan ke B- dan A- agar roda berputar mundur.
d	Jika sensor 2 = 1 maka berhenti belok kanan.
e	Inisialisasi pin yang dihubungkan ke B+ dan A-.
f	Inisialisasi pin yang dihubungkan ke B+ dan A- agar roda berputar belok kanan.
g	Jika sensor 4 = 1 maka berhenti belok kanan.
h	Inisialisasi pin yang dihubungkan ke B+ dan A-.
i	Inisialisasi pin yang dihubungkan ke B+ dan A- agar roda berputar belok kanan.
j	Inisialisasi pin yang dihubungkan ke B- dan A+
k	Inisialisasi pin yang dihubungkan ke B- dan A+ agar roda berputar belok kiri memberhentikan belok kanan (break).
l	Keluar dari *subroutine* dan kembali mengeksekusi instruksi berikutnya pada program utama.

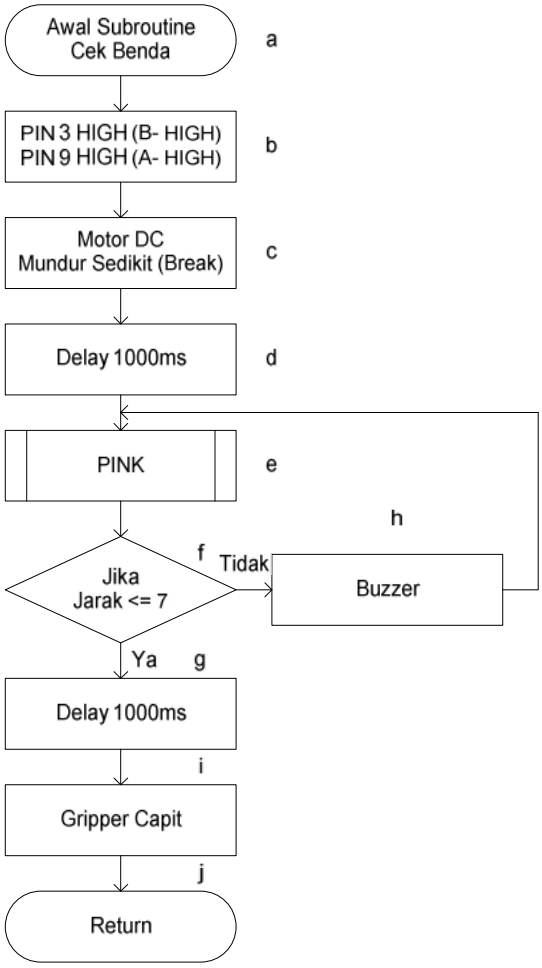


Gambar 3.25 Diagram alir subroutine diam.

Tabel 3.14 Penjelasan diagram alir subroutine diam.

No	Keterangan
a	Memulai awal *subroutine* diam.
b	Inisialisasi pin yang dihubungkan ke B- dan A-.
c	Inisialisasi pin yang dihubungkan ke B- dan A- agar roda berputar mundur (break).

d	Robot berhenti delay 1000ms.
e	Inisialisasi pin yang dihubungkan ke B- dan A-.
f	Inisialisasi pin yang dihubungkan ke B- dan A- agar roda berputar sedikit.
g	Jika sensor 2 = 1 maka berhenti belok kanan.
h	Inisialisasi pin yang dihubungkan ke B+ dan A-.
i	Inisialisasi pin yang dihubungkan ke B+ dan A- agar roda berputar belok kanan.
j	Jika sensor 6 = 1 maka berhenti belok kanan.
k	Inisialisasi pin yang dihubungkan ke B+ dan A-.
l	Inisialisasi pin yang dihubungkan ke B+ dan A- agar roda berputar belok kanan.
m	Inisialisasi pin yang dihubungkan ke B- dan A+.
n	Inisialisasi pin yang dihubungkan ke B- dan A+ agar roda berputar belok kiri (break).
o	Inisialisasi pin yang dihubungkan ke B- dan A-.
p	Inisialisasi pin yang dihubungkan ke B- dan A- agar roda berputar mundur.
q	Jika sensor 11 = 1 maka berhenti maju.
r	Inisialisasi pin yang dihubungkan ke B+ dan A+.
s	Inisialisasi pin yang dihubungkan ke B+ dan A+ agar roda berputar maju (break).
t	Gripper menutup capit sedikit.
u	Berakhir

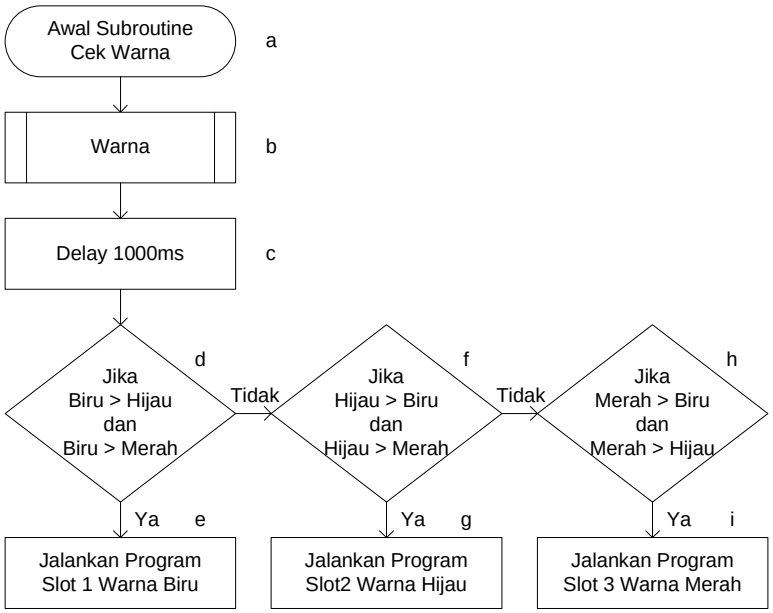


Gambar 3.26 Diagram alir subroutine cek benda.

Tabel 3.15 Penjelasan diagram alir subroutine cek benda.

No	Keterangan
a	Memulai awal *subroutine* cek benda.
b	Inisialisasi pin yang dihubungkan ke B- dan A-.
c	Inisialisasi pin yang dihubungkan ke B- dan A- agar roda berputar mundur.
d	Delay 1000ms.
e	Pemanggilan prosedur PINK.
f	Jika jarak sensor PINK <= 7 maka.

g	Delay 1000ms.
h	Buzzer bunyi ketika jarak lebih dari 7cm.
i	Gripper capit ketika jarak kurang sama dengan 7cm.
l	Keluar dari *subroutine* dan kembali mengeksekusi instruksi berikutnya pada program utama.



Gambar 3.27 Diagram alir subroutine cek warna.

Tabel 3.16 Penjelasan diagram alir subroutine cek warna.

No	Keterangan
a	Memulai awal *subroutine* cek warna.
b	Pemanggilan prosedur warna.
c	Delay 1000ms
d	Jika biru > hijau dan biru > merah maka.
e	Jalankan program slot 1 warna biru.
f	Jika hijau > biru dan hijau > merah maka.
g	Jalankan program slot 2 warna hijau.

h	Jika merah > biru dan merah > hijau maka.
i	Jalankan program slot 3 warna merah.