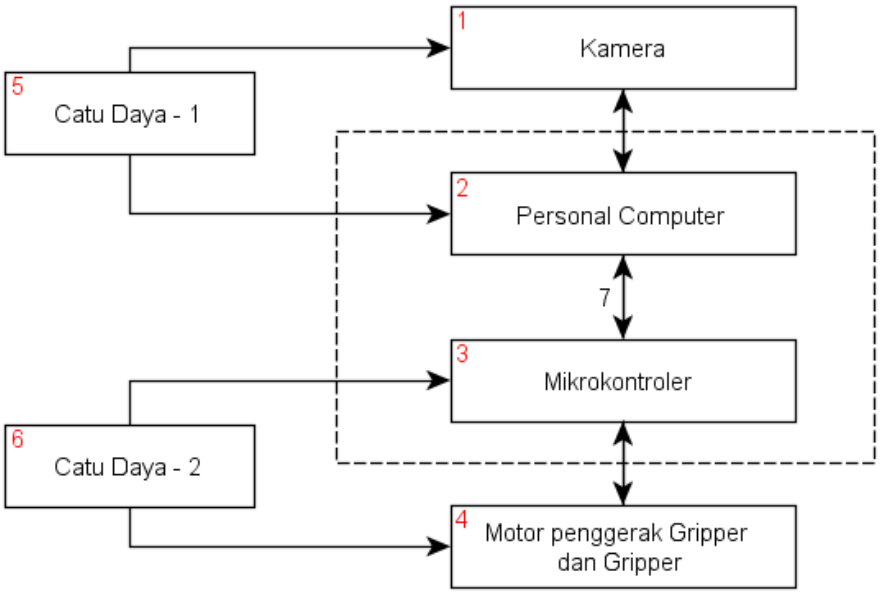
BAB III
PERANCANGAN SISTEM

Pada bab ini akan menjelaskan rancangan sistem pendeteksian objek dan pengambilan objek berupa piring maupun minuman yang berbentuk kotak. Perancangan sistem ini dibagi menjadi tiga bagian besar, yaitu perancangan mekanik, perancangan elektronik dan perancangan algoritma perangkat lunak. Pada Gambar 3.1 menunjukkan diagram blok sistem secara keseluruhan.



Gambar 3.1 Diagram blok sistem

Pada sistem pendeteksian objek dan pengambilan objek menggunakan beberapa komponen yang berhubungan satu dengan yang lainnya. Berikut adalah komponen-komponen yang digunakan pada sistem pendeteksian objek dan pengambilan objek:

1. Kamera

   Kamera digunakan untuk menangkap *(capture)* citra digital yang dijadikan sampel ciri objek yang harus dideteksi dan menangkap *(capture)* citra digital yang akan dideteksi keberadaan objek yang sesuai dengan sampel yang telah ditentukan.

2. *Personal Computer (PC)*

   Personal Computer (PC) digunakan untuk memproses algoritma pendeteksian ciri objek dari citra digital yang dihasilkan oleh kamera, kemudian *Personal Computer* akan mengirimkan data berupa posisi objek yang akan diambil kepada mikrokontroler.

3. Mikrokontroler

   Mikrokontroler memberi perintah ke *Personal Computer* untuk mendeteksi posisi objek, kemudian setelah menerima data posisi dari *Personal Computer* maka mikrokontroler akan mengendalikan sistem pengambil objek untuk menggerakkan *gripper* ke posisi yang telah dikirimkan oleh *Personal Computer*.

4. Motor penggerak *gripper* dan *gripper*

   Motor penggerak *gripper* dan *gripper* merupakan sistem pengambil objek yang terdiri dari motor DC dan motor servo, komponen-komonen ini akan bergerak berdasarkan perintah yang diberikan oleh mikrokontroler.

5. Catu daya -1

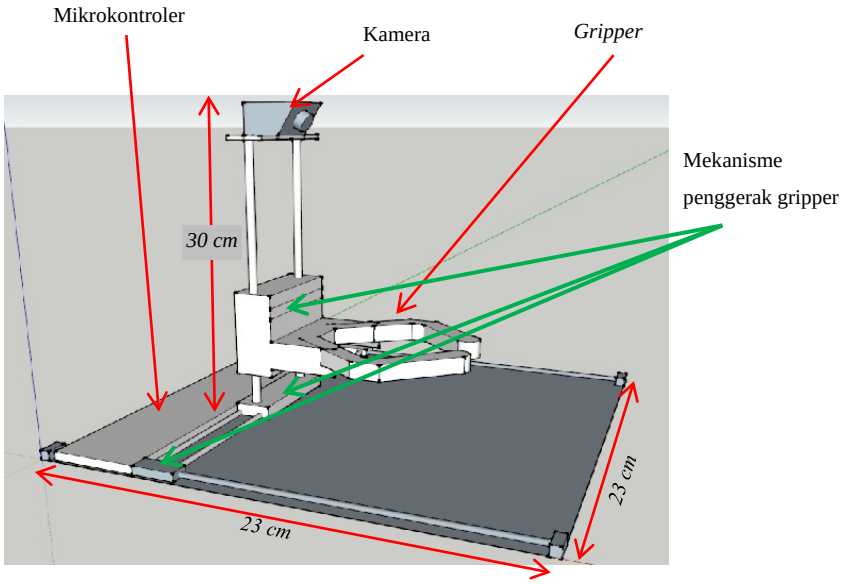
   Catu daya -1 berasal dari baterai *Personal Computer*, catu daya ini digunakan untuk mengaktifkan *Personal Computer* dan kamera.

6. Catu daya -2

   Catu daya -2 berasal dari baterai *LiPo 2200 mAh* dengan tegangan 11,1 Vdc. Catu daya-2 dibagi menjadi dua bagian, bagian pertama digunakan untuk mengaktifkan mikrokontroler dan kendali logika pada *driver* motor, dan bagian kedua digunakan untuk mengaktifkan motor DC serta motor servo.

3.1 Perancangan Mekanisme Pengambilan Objek

Perancangan mekanisme pengambil objek pada sistem ini harus menyesuaikan pada benda yang akan diambil. Mekanisme ini memiliki dimensi $23cm\ x\ 23cm\ x\ 30cm$ dengan berat ± 1.5 Kg. Bahan yang digunakan adalah plastik, nilon, aktrilik, alumunium dan besi.



Gambar 3.2 Rancangan mekanik sistem

Pada Gambar 3.2 menunjukkan penempatan posisi kamera, mikrokontroler, *gripper* dan mekanisme penggerak *gripper*. Mekasisme *Gripper* dirancang agar dapat bergeser ke kanan ke kiri, bergerak maju mundur dan bergerak naik turun. Posisi kamera diletakkan di atas *gripper*, hal ini bertujuan agar ketika *gripper* bergeser atau bergerak maju maka kamera akan bergerak mengikuti perubahan pergerakan *gripper*.

3.2 Perancangan Elektronik

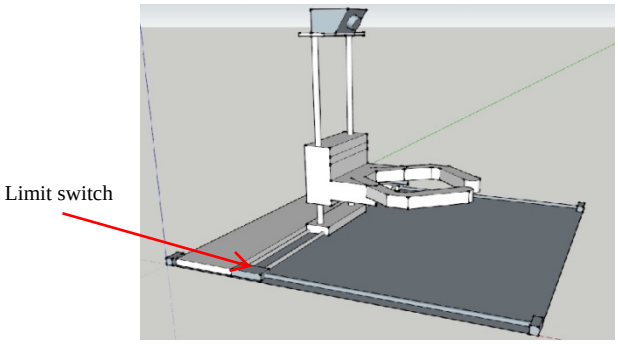
Perancangan elektronik berupa perancangan elektronik pada sensor, mikrokontroler serta sistem penggerak.

3.2.1 *Limit Switch*

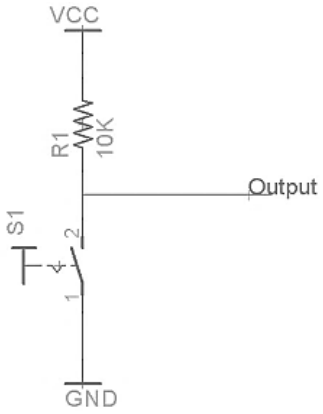
Limit switch digunakan untuk membatasi pergerakan *gripper*. Peletakan *limit switch* berada pada ujung mekanisme penggerak *gripper*, hal ini bertujuan agar *gripper* berhenti ketika mencapai ujung mekanisme penggerak.



Gambar 3.3 Limit switch



(a)



(b)

Gambar 3.4 (a) Posisi peletakan limit switch (b) Rangkaian limit switch

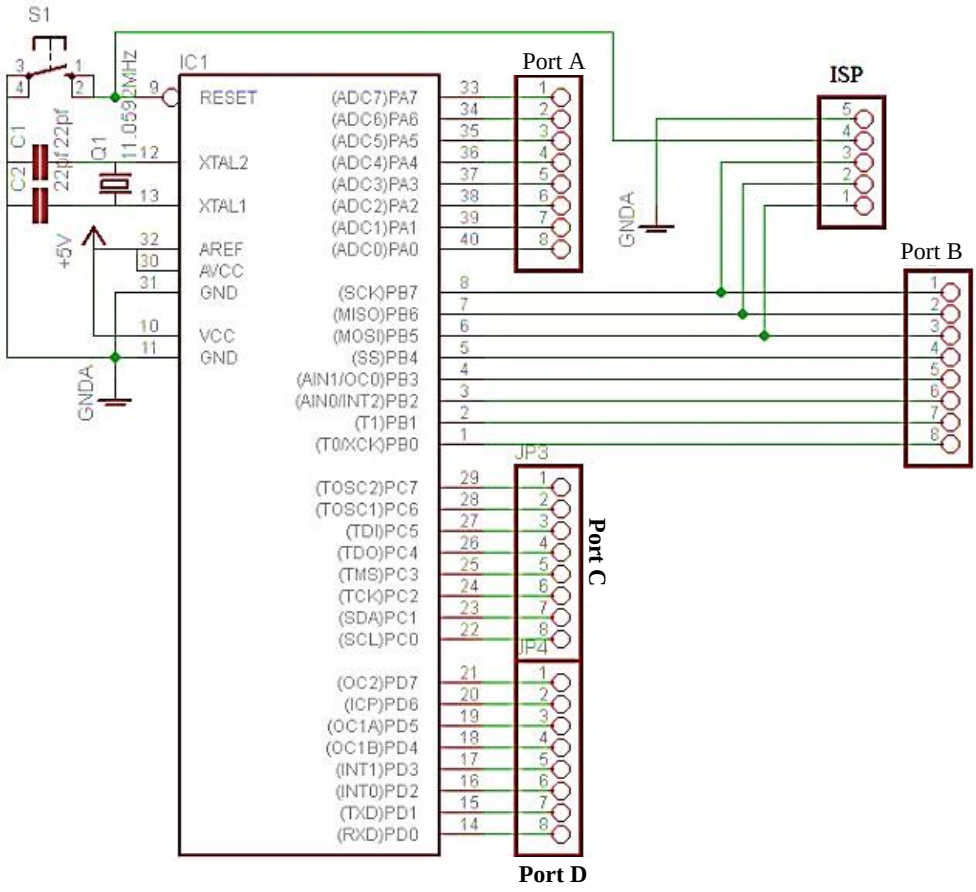
Nilai keluaran dari rangkaian *limit switch* pada Gambar 3.4 (a) berupa logika 1 dan 0. Apabila mekanisme penggerak penjepit berada di ujung, maka *limit switch* akan tertekan sehingga memberikan masukan logika 0 kepada mikrokontroler. Sedangkan bila *limit switch* tidak tertekan maka masukan yang diberikan kepada mikrokontroler berupa logika 1. Jumlah *limit switch* yang digunakan pada sistem ini adalah sebanyak 9 buah, output dari tiap-tiap *limit switch* dihubungkan ke Atmega8535 di pin A0 sampai pin A7 dan pin C3.

3.2.2 *Mikrokontroler Atmega8538*

Mikrokontroler yang digunakan sebagai pengontrol pergerakan sistem pengambilan objek adalah mikrokontroler Atmega8535. Mikrokontroler memberi perintah ke *Personal Computer* untuk mendeteksi posisi objek, kemudian setelah menerima data posisi dari *Personal Computer* maka mikrokontroler akan mengendalikan sistem pengambil objek untuk menggerakkan *gripper* ke posisi yang telah dikirimkan oleh *Personal Computer*. Kemudian setelah *gripper* berada

diposisi yang ditentukan maka mikrokontroler akan memerintahkan *gripper* untuk mengambil objek.

Sistem minimum yang digunakan untuk mendukung kinerja dari IC Atmega8535, terdiri dari rangkaian reset, rangkaian *clock* dan konfigurasi *ISP* yang digunakan untuk mengisikan program kedalam IC Atmega8535. Agar mempermudah proses pengkabelan, pin-pin I/O dihubungkan menggunakan *header*. Pada Gambar 3.5 menunjukkan sistem minimum Atmega8535.



Gambar 3.5 Rangakaian sistem minimum Atmega8535

Penggunaan pin-pin I/O ditunjukkan pada Tabel 3.1.

Tabel 3.1 Alokasi penggunaan pin pada Atmega8535

Nama Pin	No PIN	Tipe PIN	Fungsi
PIN A	0	I/O	*Switch* penanda sebelah kanan
	1	I/O	*Switch* penanda tengah
	2	I/O	*Switch* penanda sebelah kiri
	3	I/O	*Switch* penanda bagian belakang
	4	I/O	*Switch* penanda bagian depan
	5	I/O	*Switch* penanda bagian bawah
	6	I/O	*Switch* penanda bagian atas
	7	I/O	*Switch* penjepit
PIN D	0	I/O	Komunikasi serial dengan laptop
	1	I/O	Komunikasi serial dengan laptop
	2	I/O	Kendali motor maju mundur CW
	3	I/O	Kendali motor maju mundur CCW
	4	I/O	PWM motor geser kanan kiri
	5	I/O	PWM motor naik turun
	6	I/O	Kendali motor naik turun CW
	7	I/O	Kendali motor naik turun CCW
PIN C	0	I/O	Kendali motor geser kanan kiri CW
	1	I/O	Kendali motor geser kanan kiri CCW
	2	I/O	Kendali gerak capit
	3	I/O	*Switch* tengah atas bawah

3.2.3 *Sistem Pengambil Objek*

Sistem pengambil objek memiliki dua bagian, bagian pertama adalah bagian *gripper* dan bagian kedua adalah bagian yang menggerakkan *gripper*.

Bagian *gripper* menggunakan motor servo yang digunakan untuk menjepit objek. Untuk menggerakkan motor servo dibutuhkan 3 masukan, yaitu Vcc, GND dan pulsa dari mikrokontroler. Pulsa yang diterima oleh motor servo akan

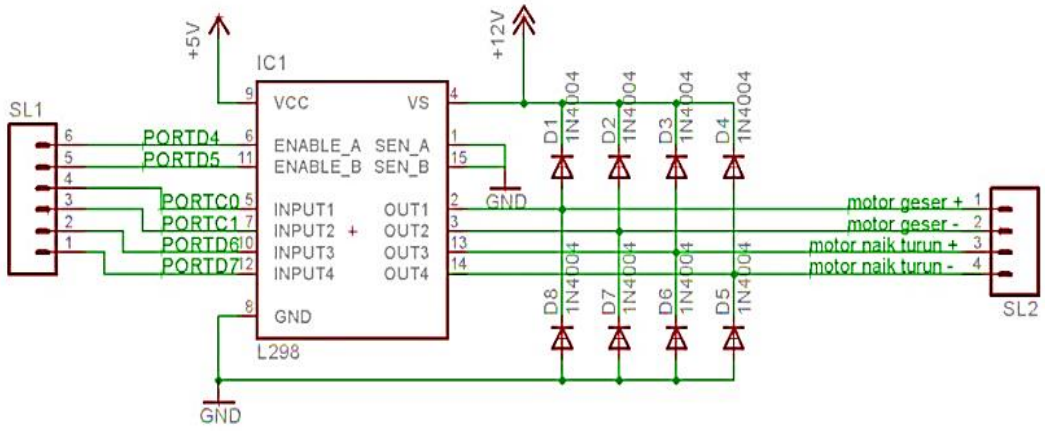
meggerakkan motor servo ke posisi yang diinginkan. Pada BASCOM-AVR perintah yang digunakan untuk mengirimkan pulsa adalah:

```
PULSEOUT Port, Pin, Duration
```

Keterangan:

- PULSEOUT : Perintah untuk mengeluarkan pulsa
- Port : Port yang digunakan untuk mengeluarkan pulsa
- Pin : Nomor pin yang digunakan untuk mengirimkan pulsa
- Duration : Lamanya waktu pengiriman pulsa

Untuk menggerakkan *gripper* dibutuhkan 3 buah motor DC. Motor DC ini digunakan untuk menggeser *gripper*, menaikkan dan menurunkan *gripper*, dan untuk maju serta mundur *gripper*. IC *driver motor* yang digunakan untuk menggerakkan motor DC adalah IC L298, berikut adalah rangkaian IC *driver* motor L298.



Gambar 3.6 Rangkaian driver motor menggunakan IC L298

Satu buah motor DC dikendalikan oleh tiga buah pin, satu buah pin *enable* pada IC L298 terhubung dengan pin keluaran PWM pada mikrokontroler Atmega8535, pin ini digunakan untuk mengatur kecepatan motor. Pin input1 dan pin input2 digunakan untuk mengatur arah putaran dari motor DC yang akan dikendalikan. Tabel 3.2 merupakan tabel kebenaran pada *IC L298*.

Tabel 3.2 Tabel kebenaran rangkaian driver motor

ENABLE A	INPUT 1	INPUT 2	Status Motor Geser
Low	X	X	Motor berhenti
High / Pulsa	Low	Low	Motor Berhenti
High / Pulsa	Low	High	Motor berputar CW
High / Pulsa	High	Low	Motor berputar CCW
High / Pulsa	High	High	Motor berhenti

ENABLE B	INPUT 3	INPUT 4	Status Motor Naik Turun
Low	X	X	Motor berhenti
High / Pulsa	Low	Low	Motor Berhenti
High / Pulsa	Low	High	Motor berputar CW
High / Pulsa	High	Low	Motor berputar CCW
High / Pulsa	High	High	Motor berhenti

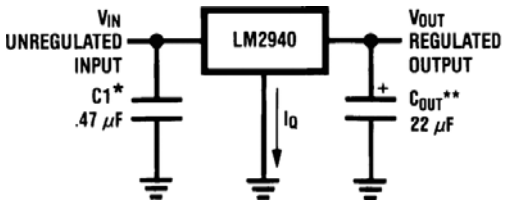
Satu buah IC L298 dapat menggerakkan 2 buah motor DC, karena terdapat 3 buah motor yang digunakan maka IC L298 yang digunakan sebanyak 2. IC L298 yang pertama digunakan untuk menggeser gripper dan menaik dan menurunkan gripper, sedangkan IC L298 yang dua digunakan untuk menggerakkan gripper kedepan dan kebelakang.

3.2.4 *Catu daya*

Catu daya digunakan sebagai sumber daya mikrokontroler, modul kamera, dan *actuator*. Sumber catu daya pada sistem ini dibagi menjadi dua, sumber yang pertama berasal dari baterai *Personal Computer*, sumber ini digunakan untuk mengaktifkan *Personal Computer* dan kamera. Sumber catu daya yang kedua berasal dari baterai *LiPo 2200 mAh* dengan tegangan 11.1 Vdc. Pada sumber catu daya kedua ini dibagi lagi menjadi dua bagian, bagian pertama digunakan untuk mengaktifkan mikrokontroler dan kendali logika pada *driver* motor, dan bagian kedua digunakan untuk mengaktifkan motor DC serta motor servo. Pembagian ini

bertujuan agar daya listrik yang dibutuhkan oleh mikrokontroler tidak terganggu oleh tarikan atau kebutuhan daya listrik untuk motor.

Mikrokontroler dan *logic driver* motor bekerja pada tegangan 5 Vdc, untuk mendapatkan tegangan sebesar 5 Vdc maka diperlukan sebuah rangkaian regulator yang dapat menurunkan tegangan dari 11.1 Vdc menjadi 5 Vdc. Regulator yang digunakan untuk menurunkan tegangan menjadi 5Vdc adalah LM2940. LM2940 dapat mengalirkan arus hingga ± 1 A.



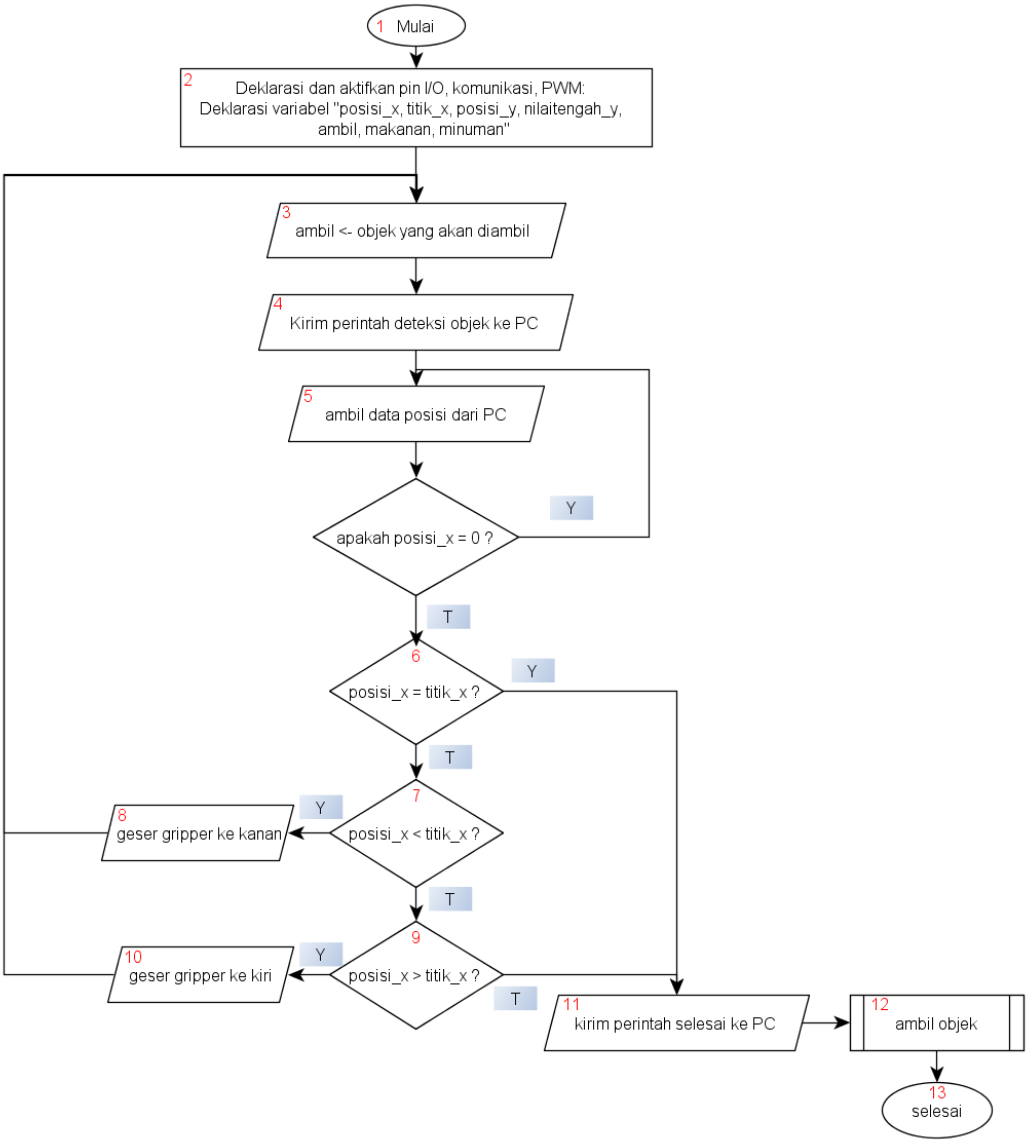
Gambar 3.7 Rangkaian regulator LM2940

3.3 Perancangan Algoritma

Perancangan algoritma meliputi algoritma yang digunakan untuk mencocokkan gambar dengan menggunakan titik-titik ciri yang telah dihasilkan dengan mengguanakan metode *SIFT* dan algoritma yang digunakan pada mikrokontroler.

3.3.1 *Algoritma pada Mikrokontroler*

Berikut ini merupakan algoritma pada mikrokontroler, algoritma ini digunakan untuk memberi perintah kepada *Personal Computer* untuk mendeteksi posisi objek, memposisikan *gripper* dan mengambil objek.



Gambar 3.8 Diagram alir pada mikrokontroler Atmega8535

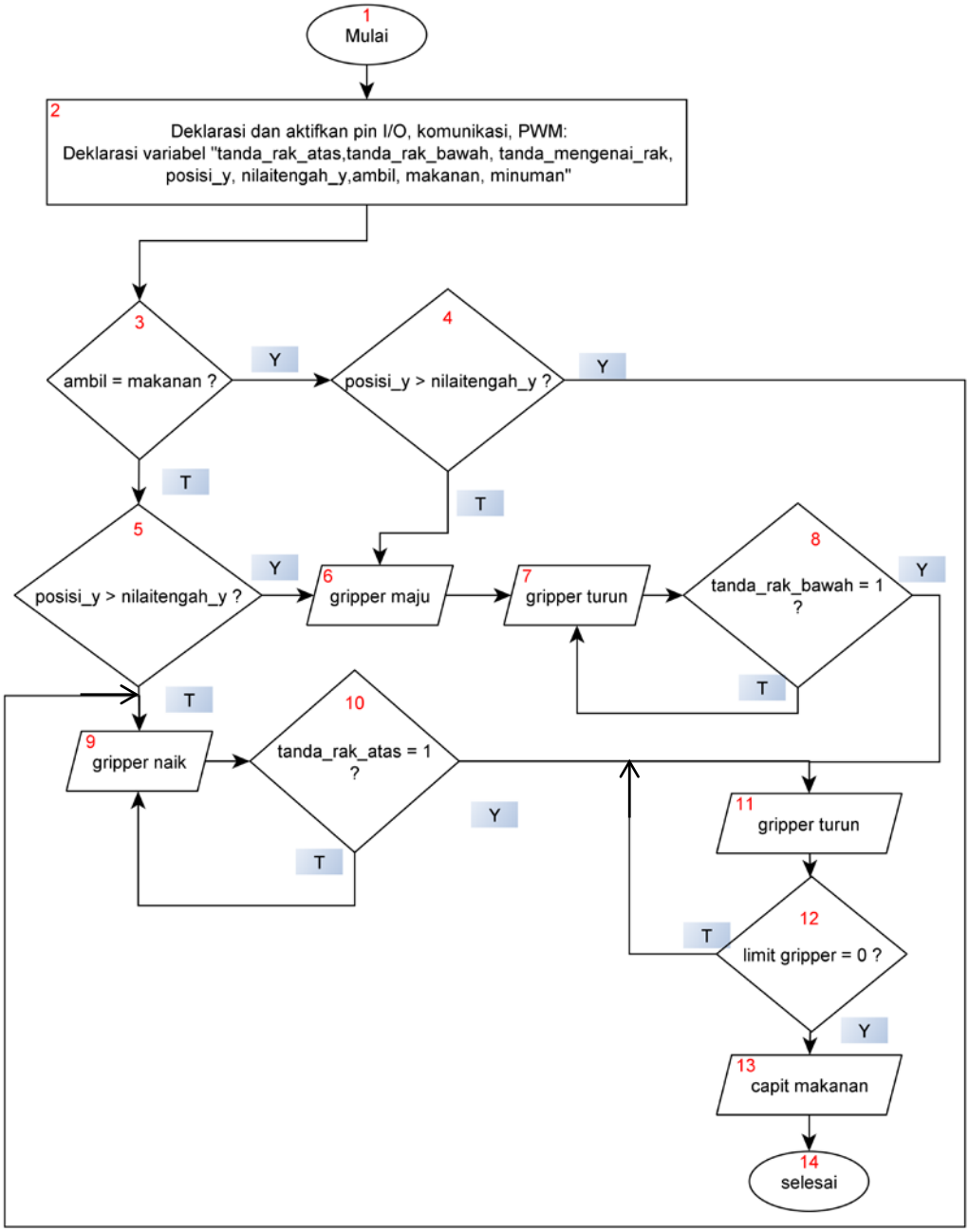
Pada Gambar 3.8 menunjukkan diagram alir pada mikrokontroler Atmega8535, berikut merupakan penjelasan dari diagram alir tersebut.

Tabel 3.3 Keterangan diagram alir dari Gambar 3.8

No	Keterangan
1	Mulai program
2	Deklarasi dan aktifkan pin I/O, komunikasi, PWM. Deklarasi variabel "posisi_x, titik_x, posisi_y, nilaitengah_y, ambil, makanan, minuman".

3	Masukkan jenis objek yang akan diambil (piring atau minuman kemasan) ke variabel ambil.
4	Kirim perintah ke *Personal Computer* untuk memulai mendeteksi posisi objek yang akan diambil
5	Ambil data posisi dari *Personal Computer*
6	Cek nilai posisi x objek terhadap nilai titik tengah kamera. Jika iya (nilai posisi x objek sama dengan nilai titik tengah kamera) maka loncat ke instruksi no 11, sedangkan jika tidak (nilainya tidak sama) maka loncat ke instruksi no 7.
7	Cek nilai posisi x objek terhadap nilai titik tengah kamera. Jika iya (nilai posisi x objek lebih kecil dari nilai titik tengah kamera) maka loncat ke instruksi no 8, sedangkan jika tidak (nilainya lebih besar) maka loncat ke instruksi no 9.
8	Berikan perintah ke motor DC untuk menggerakkan posisi *gripper* ke kanan.
9	Cek nilai posisi x objek terhadap nilai titik tengah kamera. Jika iya (nilai posisi x objek lebih besar dari nilai titik tengah kamera) maka loncat ke instruksi no 10, sedangkan jika tidak maka loncat ke instruksi no 11.
10	Berikan perintah ke motor DC untuk menggerakkan posisi *gripper* ke kiri.
11	Kirim perintah selesai pendeteksian ke *Personal Computer*
12	Prosedur ambil objek
13	Selesai

Pada Gambar 3.9 merupakan diagram alir yang digunakan untuk mengambil objek.



Gambar 3.9 Diagram alir prosedur ambil objek

Berikut merupakan penjelasan diagram alir dari prosedur ambil objek pada Gambar 3.9.

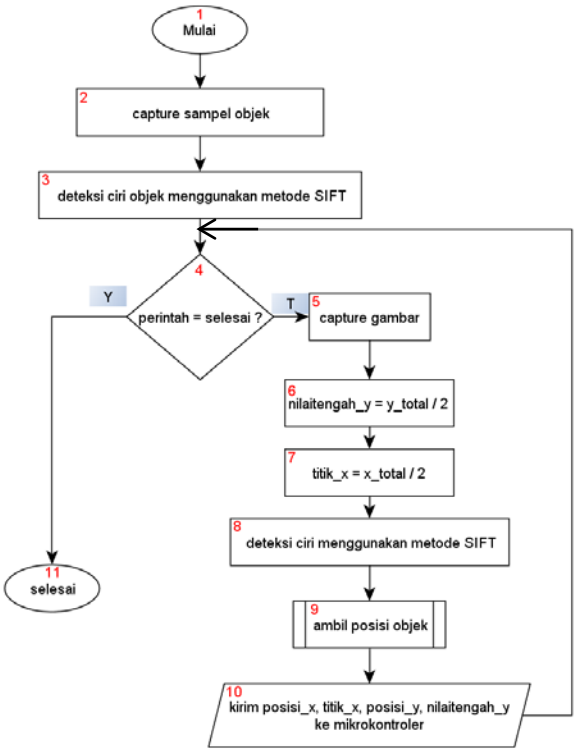
Tabel 3.4 Keterangan diagram alir dari Gambar 3.9

No	Keterangan
1	Mulai program
2	Deklarasi dan aktifkan pin I/O, komunikasi, PWM. Deklarasi variabel "tanda_rak_atas,tanda_rak_bawah, tanda_mengenai_rak, posisi_y, nilaitengah_y,ambil, makanan, minuman"
3	Cek objek yang diambil, apabila objek yang harus diambil berupa makanan maka loncat ke instruksi no 4, sedangkan bila bukan makanan maka loncat ke instruksi no 5.
4	Cek nilai posisi y objek berdasarkan nilai tengah kamera, jika iya (nilai posisi y objek lebih besar nilai titik tengah kamera) maka loncat ke instruksi no 9, sedangkan jika tidak (nilainya tidak lebih besar) maka loncat ke instruksi no 6.
5	Cek nilai posisi y objek berdasarkan nilai tengah kamera, jika iya (nilai posisi y objek lebih besar nilai titik tengah kamera) maka loncat ke instruksi no 6, sedangkan jika tidak (nilainya tidak lebih besar) maka loncat ke instruksi no 9.
6	Berikan perintah ke motor DC untuk menggerakkan posisi *gripper* ke depan.
7	Berikan perintah ke motor DC untuk menggerakkan posisi *gripper* ke bawah.
8	Cek nilai *limit switch* bawah. Jika iya (*limit switch* berlogika 0) maka loncat ke instruksi no 11, sedangkan jika tidak maka loncat ke instruksi no 7.
9	Berikan perintah ke motor DC untuk menggerakkan posisi *gripper* ke atas.
10	Cek nilai *limit switch* atas. Jika iya (*limit switch* berlogika 0) maka loncat ke instruksi no 11, sedangkan jika tidak maka loncat ke

	instruksi no 9.
11	Berikan perintah ke motor DC untuk menggerakkan posisi *gripper* ke bawah.
12	Cek nilai *limit switch gripper*. Jika iya (*limit switch* berlogika 0) maka loncat ke instruksi no 13, sedangkan jika tidak maka loncat ke instruksi no 11.
13	Ambil objek dengan cara dicapit
14	Selesai

3.3.2 *Algoritma pada Personal Computer*

Algoritma pada Personal Computer merupakan algoritma yang digunakan dalam proses pendeteksian objek. Setelah Personal Computer menerima perintah untuk melakukan pendeteksian objek dari mikrokontroler, Personal Computer akan mengambil gambar dari kamera dan kemudian mencocokkan gambar tersebut dengan objek yang telah di deskripsikan sebelumnya.



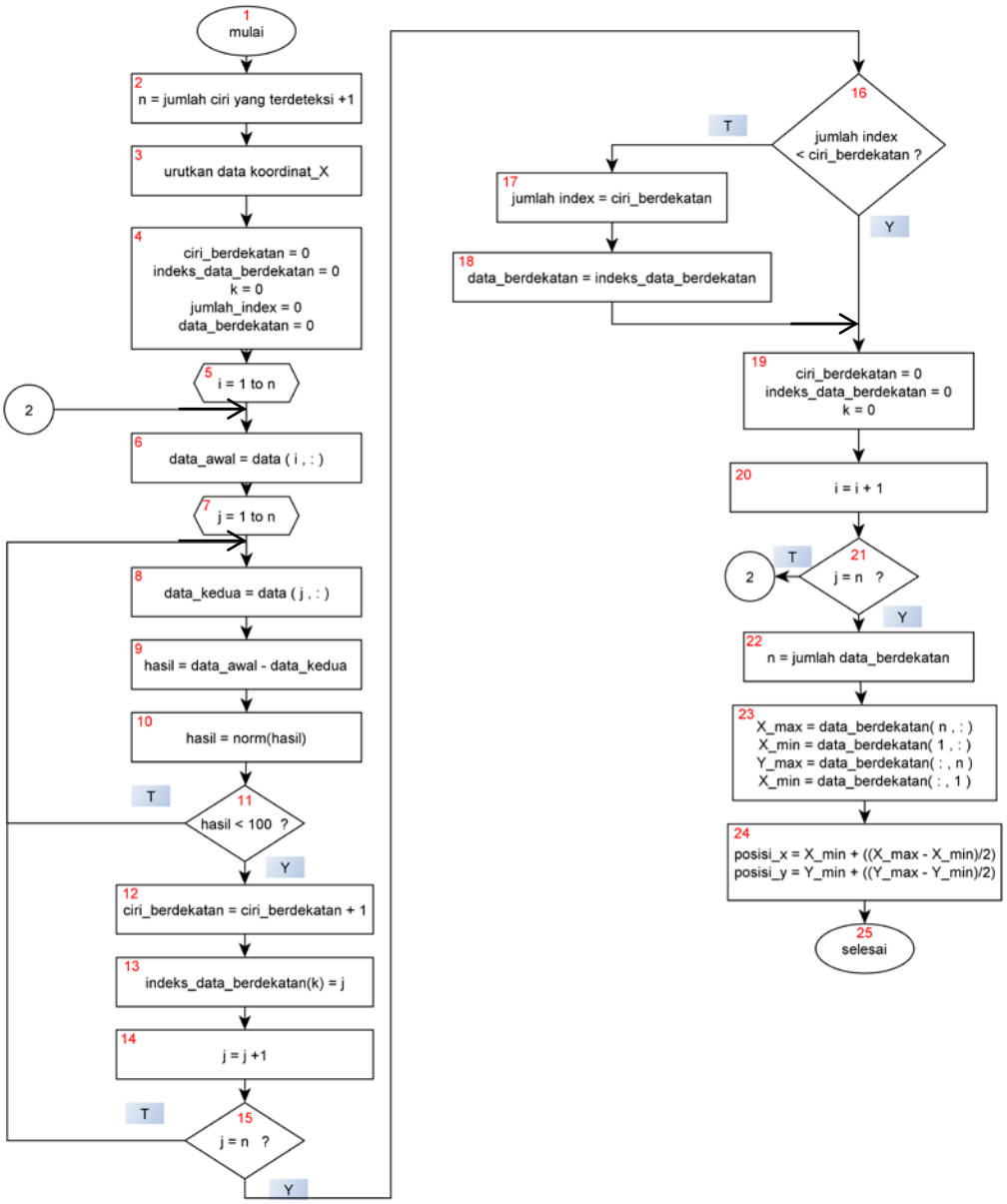
Gambar 3.10 Diagram alir pendeteksian objek di Personal Computer

Berikut merupakan penjelasan diagram alir dari Gambar 3.10.

Tabel 3.5 Keterangan diagram alir dari Gambar 3.10

No	Keterangan
1	Mulai program
2	*Capture* objek yang dijadikan *sampel*
3	Melakukan pendeteksian ciri menggunakan metode *SIFT* dari objek yang telah di *capture.*
4	Cek perintah dari mikrokontroler Atmega8535, jika iya (perintahnya sama dengan selesai) maka loncat ke instruksi no 11, sedangkan jika tidak maka loncat ke instruksi no 5.
5	*Capture* objek yang harus di deteksi keberadaan objek yang sama dengan *sampel* objek.
6	Masukkan nilai tengah y ke variabel “nilaitengah_y”
7	Masukkan nilai tengah x ke variabel “titik_x”
8	Melakukan pendeteksian ciri menggunakan metode *SIFT* dari objek yang telah di *capture.*
9	Prosedur penandaan posisi pada objek yang sesuai dengan *sampel*
10	Kirim data posisi_x, titik_x, posisi_y, nilaitengah_y ke mikrokontroler Atmega8535
11	Selesai

Pada Gambar 3.11 merupakan diagram alir prosedur ambil posisi objek yang digunakan untuk menandai ciri yang sesuai dengan *sampel*.



Gambar 3.11 Diagram alir prosedur ambil posisi objek

Berikut ini merupakan penjelasan diagram alir dari Gambar 3.11

Tabel 3.6 Keterangan diagram alir dari Gambar 3.11

No	Keterangan
1	Mulai program
2	Masukkan data jumlah yang telah terdeteksi ke variabel n

3	Urutkan nilai posisi x, untuk urutan nilai posisi y akan mengikuti perubahan urutan nilai posisi x
4	Berikan nilai 0 untuk variabel-variabel “ciri_berdekatan, indeks_data_berdekatan, k, jumlah_index, data_berdekatan.
5	Masukkan nilai 1 ke variabel i untuk proses perulangan-1
6	Masukkan data x ke i ke variabel “data_awal”
7	Masukkan nilai 1 ke variabel j untuk proses perulangan-2
8	Masukkan data x ke j ke variabel “data_kedua”
9	Masukkan hasil pengurangan “data_awal – data_kedua” ke variabel “hasil”
10	Normalisasi variabel “hasil”
11	Cek nilai normalisasi hasil, jika iya (nilainya lebih kecil dari 100) maka kembali ke instruksi no 8, sedangkan jika tidak maka loncat ke instruksi no 12
12	Tambah nilai variabel “ciri_berdekatan” dengan nilai 1
13	Masukkan nilai j ke variabel “indeks_data_berdekatan” dengan indeks ke “k”
14	Tambah nilai variabel “j” dengan nilai 1
15	Cek nilai j, jika iya (nilai j sama dengan nilai n) maka loncat ke instruksi no 16, sedangkan jika tidak sama maka kembali ke instruksi no 8.
16	Bandingkan nilai variabel “jumlah_index” dengan nilai variabel “ciri_berdekatan”, jika iya (nilai jumlah_indeks lebih kecil dari nilai ciri_berdekatan) maka loncat ke instruksi no 19, sedangkan jika tidak maka loncat ke instruksi no 17.
17	Pindahkan nilai variabel “ciri_berdekatan” ke variabel “jumlah indeks”
18	Pindahkan nilai variabel “indeks_data_berdekatan” ke variabel “data_berdekatan”
19	Isi nilai variabel “ciri_berdekatan, indeks_data_berdekatan, k” dengan nilai 0
20	Tambah nilai variabel “i” dengan nilai 1

21	Cek nilai j, jika iya (nilai j sama dengan nilai n) maka loncat ke instruksi no 22, sedangkan jika tidak sama maka kembali ke instruksi no 6.
22	Masukkan nilai jumlah data berdekatan ke variabel n
23	X_max = data_berdekatan(n , :) X_min = data_berdekatan(1 , :) Y_max = data_berdekatan(: , n) X_min = data_berdekatan(: , 1)
24	posisi_x = X_min + ((X_max - X_min)/2) posisi_y = Y_min + ((Y_max - Y_min)/2)
25	Selesai