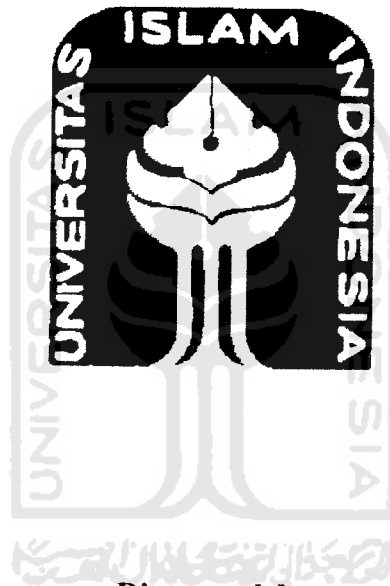


**TUGAS AKHIR**

**OTOMATIS ROBOT UNTUK MEMBUAT MENARA  
BERBASIS MIKROKONTROLER**

Diajukan Sebagai Syarat Untuk Memperoleh Gelar Sarjana Pada  
Jurusan Teknik Elektro Fakultas Teknologi Industri  
Universitas Islam Indonesia



**Disusun oleh:**

**Nama : Irfan Arfan**

**No. Mhs : 99 524 106**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2007**

**TUGAS AKHIR**

**OTOMATIS ROBOT UNTUK MEMBUAT MENARA  
BERBASIS MIKROKONTROLER**

Diajukan Sebagai Syarat Untuk Memperoleh Gelar Sarjana Pada  
Jurusan Teknik Elektro Fakultas Teknologi Industri  
Universitas Islam Indonesia



Disusun oleh:

Nama : Irfan Arfan

No. Mhs : 99 524 106

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2007**

**LEMBAR PENGESAHAN DOSEN PEMBIMBING**

**OTOMATIS ROBOT UNTUK MEMBUAT MENARA  
BERBASIS MIKROKONTROLER  
(AUTOMATIC ROBOT FOR MAKING TOWER BASED ON  
MICROCONTROLLER)**

**TUGAS AKHIR**




**di susun oleh:**

**Nama : Irfan Arfan**

**No. Mhs : 99 524 106**

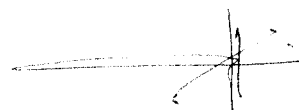
**Yogyakarta, Januari 2007**

**Pembimbing I**



**(Ir. Hj. Budi Astuti, MT.)**

**Pembimbing II**



**(Medilla Kusriyanto, ST.)**

**LEMBAR PENGESAHAN PENGUJI**

**OTOMATIS ROBOT UNTUK MEMBUAT MENARA  
BERBASIS MIKROKONTROLER**

**(AUTOMATIC ROBOT FOR MAKING TOWER BASED ON  
MICROCONTROLLER)**

**TUGAS AKHIR**

**disusun oleh :**

**Nama : Irfan Arfan**

**No.Mhs : 99 524 106**

**Telah Dipertahankan Di Depan Sidang Penguji Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Jurusan Teknik Elektro  
Fakultas Teknologi Industri Universitas Islam Indonesia  
Yogyakarta, 27 Januari 2007**

**Tim Penguji :**

**Ir. Hj. Budi Astuti, MT**  
**Ketua**

**Medilla Kusriyanto, ST**  
**Anggota I**

**Yusuf Aziz Amrullah, ST**  
**Anggota II**

**Mengetahui,**

**Ketua Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Universitas Islam Indonesia**



**Fitri Kuswono, ST., M.Sc )**

# Halaman Persembahan



*Kupersembahkan Tugas Akhir ini untuk:*

## **Allah SWT**

*..Yang tiada henti-hentinya mencurahkan rahmat dan kasih sayangNya kepada seluruh hamba-hambaNya. Yang telah menciptakan langit, bumi, serta segala isinya dan keindahannya..*

## **Ayahanda Sjamsul Bahri Siregar**

**&**

## **Ibunda Linda Kusuma**

*..Maafkan saya jika pernah sekalipun dari perbuatan dan sikap saya melukai dan tidak berkenan dihati Papa dan Mama..*

*..Untuk semua dukungan, kasih sayang, dan doa yang Papa Mama curahkan sejak aku kecil, semoga apa yang aku lakukan dapat membuat kebanggaan dihati Papa dan Mama..*

*I Love You, Mom!, I Love You, Dad!*

## **Abangku Irfing Ardiles Siregar, Adikku Irene Arisandy Siregar, Adikku Raka Atilla "Raja"**

*..Kehangatan, kebersamaan, serta keceriaan kita adalah yang terbaik..*

*I Love You All!*

## Motto



“Jadikanlah Sabar Dan Sholatmu Sebagai Penolongmu. Dan Sesungguhnya Yang Demikian Itu Sungguh Berat, Kecuali Bagi Orang-Orang Yang Khusyu”

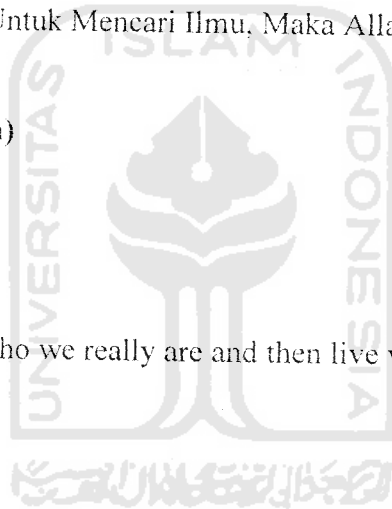
(QS. Al Baqorah : 45)

“Barang Siapa Menempuh Jalan Untuk Mencari Ilmu, Maka Allah Memudahkan Jalan Baginya Menuju surga”

(HR. Muslim dan Abu Hurairah)

“I think that somehow, we learn who we really are and then live with that decision”.

(Eleanor Roosevelt 1884 – 1962)



## KATA PENGANTAR



Assalamu'alaikum Wr Wb,

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT. atas segala rahmat dan karunia-Nya yang telah dilimpahkan kepada penulis, sehingga berhasil menyelesaikan penyusunan Tugas Akhir ini yang berjudul "**Otomatis Robot Untuk Membuat Menara Berbasis Mikrokontroler**" (*Automatic Robot For Making Tower Based On Microcontroller*). Tugas Akhir ini disusun untuk memenuhi persyaratan akhir guna meraih gelar Sarjana Teknik di jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia.

penulis menyadari bahwa masih banyak dibutuhkan penyempurnaan dalam Tugas Akhir ini. untuk itu penulis mengharapkan teman-teman, adik-adik penulis, serta pihak yang berkepentingan untuk dapat meneruskan apa yang sudah penulis lakukan hingga tahap ini.

Dalam kesempatan ini tidak lupa penulis ingin menyampaikan ucapan terima kasih sebanyak banyak-nya kepada :

1. Bapak Fathul Wahid, ST., M.Sc. selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Tito Yuwono, ST., M.Sc. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Ibu Ir. Hj. Budi Astuti, MT. selaku Dosen Pembimbing I Tugas Akhir.
4. Bapak Medilla Kusriyanto, ST. selaku Dosen Pembimbing II Tugas Akhir.

5. Segenap Dosen Teknik Elektro dan FTI yang pernah mengajar Penulis selama Penulis kuliah, atas ilmu dan pengetahuan yang diberikan.
6. Kedua orang tuaku tercinta yang tiada henti-hentinya mencurahkan kasih sayang, dukungan, serta doanya kepadaku. Ayahanda Sjamsul Bahri Siregar, dan Ibunda Linda Kusuma. "Terima kasih atas segalanya".
7. Abangku Irfing Ardiles Siregar, serta adik-adikku Irene Arisandy Siregar, dan Raka Atilla. "Terima kasih telah menyayangiku".
8. Prapti Astuti, S.IP. dan Dian Pratiwi, S.Sos., atas dukungan dan dorongannya yang tiada henti-hentinya.
9. Semua sahabat, teman-temanku, di jurusan Teknik Elektro, Universitas Islam Indonesia.
10. Seluruh teman-teman di RICCIA, Deni dan Mba Iin, Roni, Indra, Andi "Nesta", Papah, Eko "Sri", Gaban dan Desi, Lino, Tatto dan Nissa, Bepe, Ardi, Irpad, Aries, Tanto, Ari, Andi "Ujung". ("I 4 All n All 4 I").
11. Serta semua pihak yang telah membantu terselesaikannya penyusunan laporan Tugas Akhir ini yang tidak dapat disebutkan satu per satu.

Besar harapan agar laporan ini dapat bermanfaat kepada penulis pada khususnya dan pembaca pada umumnya.

*Wassalamualaikum. Wr. Wb.*

**Yogyakarta, Januari 2007**

**Irfan Arfan**



## DAFTAR ISI

<b>LEMBAR PENGESAHAN DOSEN PEMBIMBING .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN PENGUJI.....</b>	<b>ii</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>iii</b>
<b>MOTTO.....</b>	<b>iv</b>
<b>KATA PENGANTAR.....</b>	<b>v</b>
<b>DAFTAR ISI .....</b>	<b>vii</b>
<b>DAFTAR GAMBAR.....</b>	<b>x</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>ABSTRAKSI .....</b>	<b>xii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Perancangan.....	3
1.5 Manfaat Perancangan.....	3
1.6 Metode Perancangan.....	3
1.6.1 Sumber Data.....	3
1.6.2 Teknik Pengumpulan Data.....	3
1.7 Sistematika Penulisan .....	4
<b>BAB II LANDASAN TEORI .....</b>	<b>6</b>

2.1 Mikrokontroler.....	6
2.1.1 Mikrokontroler 89S52.....	7
2.1.1.1 Fungsi – Fungsi Pin 89S52 .....	8
2.1.1.2 Special Function Register .....	13
2.1.1.3 Perangkat Lunak AT89S52.....	17
2.2 LM 324.....	21
2.2.1 Op-Amp Sebagai Pembanding (komparator).....	21
2.3 Motor DC Magnet Permanen.....	22
2.4 IRED (Infrared Emitting Diode).....	23
2.5 Fototransistor .....	23
2.6 H - Bridge L293D .....	24
2.7 LM 317.....	25
<b>BAB III PERANCANGAN .....</b>	<b>26</b>
3.1 Perancangan Sistem Kerja .....	26
3.2 Perancangan Perangkat Keras.....	27
3.2.1 Perancangan Blok Sensor .....	28
3.2.2 Perancangan Blok Mikrokontroler.....	30
3.2.3 Perancangan Blok Driver Motor.....	31
3.3 Perancangan Perangkat Lunak.....	33
<b>BAB IV PENGUJIAN DAN ANALISA.....</b>	<b>35</b>
4.1 Pengujian Sensor Garis (Sensor Infra Merah) .....	35
4.1.1 Analisa Pengujian Sensor Garis (Sensor Infra Merah).....	37
4.1.2 Pengukuran Tegangan Keluaran Sensor Garis .....	39

4.2 Pengujian Sensor Pendeteksi Benda (Kit Sensor Infra Merah) .....	41
4.2.1 Analisa Pengujian Sensor Pendeteksi Benda .....	41
4.2.2 Pengukuran Tegangan Keluaran Sensor Pendeteksi Benda.....	42
4.3 Pengujian Lengan Robot.....	42
4.3.1 Analisa Pengujian Lengan Robot.....	43
<b>BAB V PENUTUP</b> .....	<b>45</b>
5.1 Kesimpulan .....	45
5.2 Saran .....	46
<b>DAFTAR PUSTAKA</b> .....	<b>47</b>
<b>LAMPIRAN</b> .....	<b>48</b>



## DAFTAR GAMBAR

Gambar 2.1 Kaki-kaki mikrokontroler 89S52.....	8
Gambar 2.2 Simbol penguat operasional .....	21
Gambar 2.3 Op-Amp sebagai pembanding tegangan.....	22
Gambar 2.4 IC L293D.....	24
Gambar 2.5 <i>Adjustable regulator</i> dengan menggunakan LM 317.....	25
Gambar 3.1 Diagram blok robot .....	27
Gambar 3.2 Rancangan sensor.....	28
Gambar 3.3 Susunan sensor.....	28
Gambar 3.4 Rangkaian sensor saris.....	29
Gambar 3.5 Rangkaian driver motor.....	32
Gambar 3.6 Diagram alir kerja robot.....	34
Gambar 4.1 Titik pengukuran tegangan keluaran sensor.....	40
Gambar L.1 Rangkaian robot.....	49

## DAFTAR TABEL

<b>Tabel 2.1</b> Fungsi pin port 1 .....	10
<b>Tabel 2.2</b> T2CON = Timer / Counter 2 Control Register .....	14
<b>Tabel 2.3</b> Fungsi bit pada register interupsi .....	14
<b>Tabel 2.4</b> Simbol assembler khusus .....	19
<b>Tabel 3.1</b> Tabel kebenaran .....	29
<b>Tabel 4.1</b> Percobaan sensor garis pada garis dengan lebar 0,8 cm .....	35
<b>Tabel 4.2</b> Percobaan sensor garis pada garis dengan lebar 1,5 cm .....	36
<b>Tabel 4.3</b> Percobaan sensor garis pada garis dengan lebar 1,9 cm .....	36
<b>Tabel 4.4</b> Percobaan sensor garis pada garis dengan lebar 2,5 cm .....	36
<b>Tabel 4.5</b> Percobaan sensor garis pada garis dengan lebar 3,5 cm .....	36
<b>Tabel 4.6</b> Percobaan sensor garis pada garis dengan lebar 5 cm .....	37
<b>Tabel 4.7</b> Percobaan sensor garis pada garis dengan lebar 6 cm .....	37
<b>Tabel 4.8</b> Pengukuran tegangan keluaran sensor garis titik1 .....	40
<b>Tabel 4.9</b> Pengukuran tegangan keluaran sensor garis titik2 .....	40
<b>Tabel 4.10</b> Hasil percobaan pendeteksian benda sensor pendeteksi benda .....	41
<b>Tabel 4.11</b> Pengukuran tegangan keluaran sensor pendeteksi benda .....	42
<b>Tabel 4.12</b> Hasil percobaan menyusun menara 1 .....	43
<b>Tabel 4.13</b> Hasil percobaan menyusun menara 2 .....	43
<b>Tabel 4.14</b> Hasil percobaan menyusun menara 3 .....	43

## ABSTRAK

Pada saat ini dunia mengalami perkembangan teknologi yang sangat cepat, salah satunya adalah teknologi dibidang elektronika, seiring dengan itu kebutuhan manusia akan alat dengan teknologi yang canggih untuk mempermudah manusia dalam melakukan pekerjaannya agar lebih praktis, aman, dan berkualitas pun semakin tinggi. Untuk memenuhi kebutuhan tersebut maka diperlukan teknologi yang tepat dalam perancangannya. salah satunya adalah perancangan otomatis robot untuk membuat menara. Hal ini akan sangat bermanfaat untuk membantu pekerjaan manusia dalam situasi dan kondisi medan yang kurang mendukung untuk manusia itu sendiri.

Robot ini bekerja dengan mengikuti garis dan mengambil benda yang ada dilintasannya untuk disusun menjadi menara. Mikrokontroler berfungsi untuk mengendalikan seluruh modul-modul motor driver, sensor infra merah, dan melakukan perhitungan untuk mencari titik tengah terhadap suatu benda, serta menentukan gerak robot untuk maju, mundur, atau berhenti, berbelok dan berjalan lurus sesuai dengan jalur. Untuk mendeteksi benda yang berada di depan robot dilengkapi dengan kit sensor infra merah yang akan membaca jarak benda terhadap lengan.

Robot ini dapat berputar kembali mencari jalur semula apabila pergerakan robot telah mencapai ujung jalur. Kepresisian penyusunan menara sangat dipengaruhi oleh ketelitian sensor pendeteksian benda dalam membaca benda, sistem kerja kit sensor infra merah pendeteksi juga dipengaruhi oleh luas serta tingginya permukaan benda yang menghadap ke sensor. Lengan robot akan bekerja setelah kit sensor infrared mendeteksi benda pada jarak yang telah ditentukan. *Switch* tekan berfungsi sebagai pembatas gerakan lengan ke atas dan lengan kebawah, juga digunakan untuk mendeteksi benda untuk mengetahui benda telah terjepit secara baik atau belum.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Penggunaan sensor dalam teknologi sekarang ini sangat luas, karena sensor merupakan indera dari suatu alat untuk mengetahui keadaan lingkungannya. Teknologi sensor mengalami perkembangan pesat mulai dari dunia industri, pembangunan, penelitian, peralatan rumah tangga sampai mainan. Sensor mempunyai banyak jenis baik dari media yang dideteksi, hasil yang diperoleh, maupun kecepatan respon dari sensor. Salah satu sensor yang sering digunakan adalah sensor infrared, sensor infrared dapat digunakan dalam pembuatan robot pengikut garis sebagai pendeteksi garis dan pendeteksi benda. Robot pengikut garis adalah sebuah robot yang mampu berjalan sesuai dengan jalur yang telah ditentukan. Hal ini akan sangat bermanfaat untuk melakukan pekerjaan yang tidak mungkin dilakukan manusia karena keterbatasan ruang gerak, situasi dan kondisi yang dapat membahayakan manusia.

Sebuah robot dirancang untuk berjalan dengan mengetahui keadaan lingkungannya, dan mengambil benda satu-persatu yang ada di depannya untuk disusun menjadi sebuah menara. Untuk itu diperlukan indera atau sensor pada robot yang berfungsi mendeteksi adanya benda untuk membuat menara yang berada didepannya, dan merespon dengan baik terhadap pendeteksian tersebut.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas dapat dirumuskan permasalahan sebagai berikut :

“Bagaimana mewujudkan dan membuat sebuah robot yang berjalan mengikuti suatu garis atau jalur yang telah ditentukan dan dapat mengangkat benda dihadapannya dan kemudian disusun menjadi menara”.

## 1.3 Batasan Masalah

Dengan adanya batasan masalah, dapat lebih disederhanakan dan mengarahkan penelitian agar tidak menyimpang dari apa yang diteliti. Batasan masalah yang dibuat adalah:

1. Dalam penelitian difokuskan pada perancangan dan pembuatan otomatis robot untuk membuat menara berbasis mikrokontroler.
2. Bahan untuk menyusun menara setinggi tiga tumpukan berupa benda berbentuk kotak berukuran 4,5 x 4 x 5 cm. dan terbuat dari sterofom. dan berat hanya dibatasi maksimum 100gr dan diletakan pada jalur lurus yang akan dilalui robot.
3. Sistem pengendalian robot menggunakan rangkaian mikrokontroler AT89S52.
4. Sensor yang digunakan pada robot adalah sensor infra merah.



#### 1.4 Tujuan Perancangan

Tujuan pembuatan tugas akhir adalah merancang dan merealisasikan robot yang bisa berjalan mengikuti garis dan dapat menyusun benda yang dipasang dilintasan jalur robot menjadi menara.

#### 1.5 Manfaat Perancangan

Manfaat alat ini :

1. Dapat terwujudnya sebuah robot yang mampu berjalan sesuai dengan jalur yang ditentukan dan dapat menyusun benda yang terdapat pada jalur menjadi sebuah menara.
2. Membantu meringankan pekerjaan manusia.

#### 1.6 Metode Perancangan

##### 1.6.1 Sumber Data

Data diperoleh dari studi pustaka yang meliputi literatur, buku, artikel dan tutorial yang tersedia pada *website* di internet.

##### 1.6.2 Teknik Pengumpulan Data

Teknik pengumpulan data meliputi :

##### 1. Studi Literatur

Literatur diperoleh dari *data sheet* komponen, buku, artikel serta tutorial yang tersedia di website internet.

## 2. Perancangan dan pembuatan

Merancang dan membuat otomatis robot untuk membuat menara berbasis mikrokontroler.

## 3. Pengujian

Setelah alat jadi akan dilakukan pengujian otomatis robot untuk membuat menara berbasis mikrokontroler sesuai dengan batasan masalah yang ada.

## 4. Analisa

Otomatis robot untuk membuat menara berbasis mikrokontroler yang sudah diuji akan dianalisa sesuai dengan dasar teori yang ada.

Pengumpulan data ini digunakan untuk mendapatkan informasi-informasi yang berkaitan dengan proses penyusunan tugas akhir sehingga dapat digunakan sebagai acuan dalam proses perancangan.

## 1.7 Sistematika Penulisan

Sistematika penulisan tugas akhir ini adalah sebagai berikut :

### BAB I Pendahuluan

Bab ini memuat latar belakang masalah, rumusan masalah, batasan masalah, tujuan perancangan, manfaat perancangan, metode perancangan dan sistematika penulisan.

## BAB II Landasan Teori

Bab ini memuat teori-teori yang berhubungan dengan perancangan.

## BAB III Perancangan

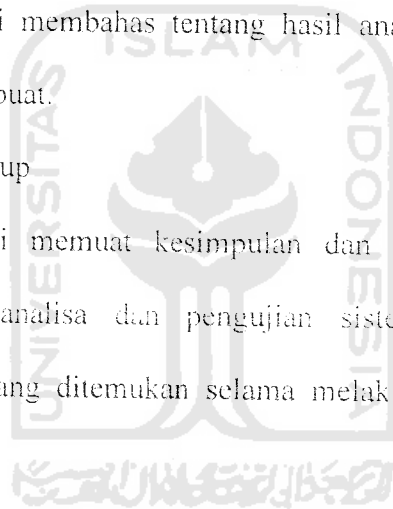
Bagian ini menjelaskan metode-metode perancangan yang digunakan.

## BAB IV Analisis dan Pembahasan

Bab ini membahas tentang hasil analisa dan pengujian dari sistem yang dibuat.

## BAB V Penutup

Bab ini memuat kesimpulan dan saran-saran dari proses perancangan, analisa dan pengujian sistem, serta keterbatasan-keterbatasan yang ditemukan selama melakukan perancangan tugas akhir.



## BAB II

### LANDASAN TEORI

#### 2.1 Mikrokontroler

Pengetahuan tentang perbedaan antara mikroprosesor dan mikrokontroler perlu diketahui. mikroprosesor adalah bagian dari CPU (*Central Processing Unit*) pada suatu komputer, tanpa memori, masukan, keluaran, dan antarmuka yang dibutuhkan oleh sebuah sistem yang lengkap. Sebagai contoh 8088 dan 80X86 adalah jenis-jenis mikroprosesor yang banyak dikenal. Untuk dapat bekerja mikroprosesor membutuhkan perangkat pendukung seperti RAM, ROM, dan masukan-keluaran (I/O) yang secara fisik terpisah dari mikroprocessor. Selain itu mikroprosesor dalam sebuah komputer mampu digunakan untuk menangani berbagai aplikasi untuk bermacam keperluan, baik untuk pengolahan data, multimedia ataupun pengendalian.

Apabila sebuah mikroprosesor dikombinasikan dengan masukan-keluaran dan memori akan dihasilkan sebuah mikrokomputer. Pada kenyataannya mengkombinasikan CPU dengan memori dan masukan-keluaran dapat juga dilakukan dengan *level chip*, yang menghasilkan *chip* tunggal mikrokomputer yang mampu berdiri sendiri dalam menjalankan aplikasi. Untuk membedakannya dengan mikrokomputer selanjutnya disebut mikrokontroler. Selain itu mikrokontroler hanya digunakan untuk satu macam aplikasi yang programnya sudah dimasukkan kedalamnya.

### 2.1.1 Mikrokontroler 89S52

AT89S52 merupakan 8 bit mikrokontroler CMOS berdaya rendah dengan 8k bytes Flash Programmable yang mempunyai performa tinggi. Instruksi dan pin-out pada AT89S52 kompatibel dengan standart industri 80C51 dan 80C52. On-chip Flash membuat memory program dapat diprogram ulang. Dengan mengkombinasikan CPU 8 bit dengan flash pada chip monolitik, 89S52 merupakan chip yang handal dengan tingkat fleksibilitas yang tinggi dan harga murah pada banyak aplikasi kontrol. Selain itu 89S52 memiliki kemampuan untuk diprogram secara serial langsung dari komputer tanpa menggunakan rangkaian khusus.

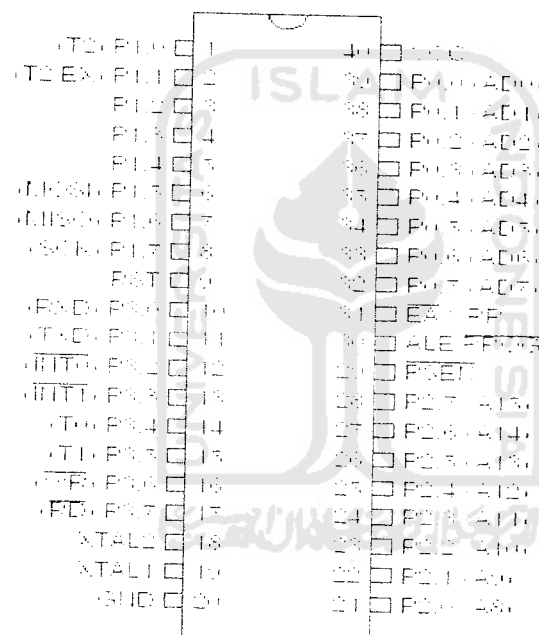
Mikrokontroler 89S52 memiliki spesifikasi standar sebagai berikut:

- Kompatibel dengan standart industri MCS-51
- 8K Bytes Flash Programmable
- Daya tahan : 1000 kali penulisan dan penghapusan
- 256 x 8 bit internal RAM
- Tiga buah *timer/counter* 16 bit.
- Serial port yang dapat diprogram.
- Delapan sumber interupsi dengan dua level prioritas.
- *On-chip* osilator.
- Empat port terprogram masukan-keluaran (32 jalur I/O terprogram)
- Dapat diprogram secara serial.

Dengan keistimewaan di atas, pembuatan alat dengan menggunakan 89S52 menjadi lebih sederhana dan tidak memerlukan IC pendukung yang lebih banyak.

### 2.1.1.1 Fungsi – Fungsi Pin 89S52

Pada Gambar 2.1 berikut merupakan bentuk fisik mikrokontroler 89S52 dan fungsi masing-masing kakinya.



**Gambar 2.1** Kaki-kaki mikrokontroler 89S52.

Penjelasan masing-masing kaki 89S52 adalah sebagai berikut:

**a) Vcc**

Vcc digunakan sebagai sumber tegangan.

**b) GND**

GND digunakan bersama pin Vcc sebagai sumber tegangan.

### c) Port 0

Port 0 merupakan port keluaran / masukan (I/O) bertipe open drain bi-directional. Sebagai port keluaran, masing-masing kaki dapat menyerap arus delapan masukan TTL (sekitar 3,8 mA). Pada saat '1' dituliskan ke kaki-kaki port 0 ini, maka kaki-kaki port 0 dapat digunakan sebagai masukan-masukan berimpedansi tinggi. Port 0 juga dapat dikonfigurasi sebagai bus alamat/data bagian rendah (low byte) selama proses pengaksesan memory data dan program eksternal. Jika digunakan dalam mode ini Port0 memiliki pull up internal. Port 0 juga menerima kode-kode yang dikirimkan kepadanya selama proses pemrograman dan mengeluarkan kode-kode selama proses verifikasi program yang telah tersimpan dalam flash. Dalam hal ini dibutuhkan pull up eksternal selama verifikasi program.

### d) Port 1

Port 1 merupakan port I/O dwi-arah yang dilengkapi dengan pull up internal. Penyangga keluaran port 1 mampu memberikan/menyerap arus masukan TTL (sekitar 1,6 mA). Jika '1' dituliskan ke kaki-kaki port 1, maka masing-masing kaki akan di-pulled high dengan pull up internal sehingga dapat digunakan sebagai masukan. Sebagai masukan, jika kaki-kaki port 1 dihubungkan ke ground, maka masing-masing kaki akan memberikan arus karena pulled high secara internal. Port 1 juga menerima alamat bagian rendah (low byte) selama pemrograman dan verifikasi flash. Adapun fungsi – fungsi lain dari Port 1 dapat dilihat pada Tabel 2.1.

**Tabel 2.1** Fungsi pin port 1.

Port pin	Fungsi Lain
P1.0	T2 (Masukan timer / counter 2)
P1.1	T2EX (Pengendalian arah dan pemicuan timer / counter 2)
P1.5	MOSI yang digunakan untuk pemrograman serial.
P1.6	MISO yang digunakan untuk pemrograman serial.
P1.7	SCK yang digunakan untuk pemrograman serial.

**e) Port 2**

Port 2 merupakan port I/O dwi arah dengan dilengkapi pull up internal. Penyangga keluaran port 2 mampu memberikan/menyerap arus masukan TTL (sekitar 1,6 mA). Jika '1' dituliskan ke kaki-kaki port 2, maka masing-masing kaki akan di-pulled high dengan pullup internal sehingga dapat digunakan sebagai masukan. Sebagai masukan, jika kaki-kaki port dihubungkan ke ground, maka masing-masing kaki akan memberikan arus karena di-pulled high secara internal. Port 2 akan memberikan byte alamat bagian tinggi (high byte) selama pengambilan instruksi dari program eksternal dan selama pengaksesan memory data eksternal yang menggunakan perintah dengan alamat 16-bit (misalnya `MOVX @DPTR`). Dalam aplikasi ini, jika ingin mengirimkan '1' maka digunakan pull up internal yang sudah disediakan. Selama pengaksesan memori data eksternal yang menggunakan perintah dengan alamat 8-bit (misalnya : `MOVX @Ri`), port 2 akan



mengirimkan isi dari SFR P2. Port 2 juga menerima alamat bagian tinggi selama pemrograman dan verifikasi flash

**f) Port 3**

Port 3 merupakan port I/O dwi arah dengan dilengkapi pull up internal. Penyangga keluaran port 2 mampu memberikan/menyerap arus masukan TTL (sekitar 1,6 mA). Jika '1' dituliskan ke kaki-kaki port 3, maka masing-masing kaki akan di-pulled high dengan pull up internal sehingga dapat digunakan sebagai masukan. Sebagai masukan, jika kaki-kaki port dihubungkan ke ground, maka masing-masing kaki akan memberikan arus karena di-pulled high secara internal. Port 3.

**g) RST**

Masukan reset. Kondisi '1' selama 2 siklus mesin selama osillator bekerja akan mereset mikrokontroler yang bersangkutan.

**h) ALE/PROG**

Keluaran ALE atau *addres Latch Enable* menghasilkan pulsa-pulsa untuk mengunci byte rendah (low byte) alamat selama mengakses memory eksternal. Kaki ini juga berfungsi sebagai masukan pulsa program atau PROG selama pemrograman flash. Pada operasi normal, ALE akan berpulsa dengan laju 1/6 dari frekuensi kristal dan dapat digunakan sebagai pewaktu (timing) atau clocking rangkaian eksternal. Catatan, ada satu pulsa yang dilompati selama pengaksesan memory data eksternal. Jika dikehendaki, operasi ALE bisa dimatikan dengan cara mengatur bit 0 dari SFR lokasi 8Eh. Jika isinya '1',

ALE hanya akan aktif selama dijumpai instruksi MOVX atau MOVC. Selain itu, kaki ini akan secara lemah di-pulled high. Mematikan ALE tidak akan ada efeknya jika mikrokontroler mengeksekusi program secara eksternal

**i) PSEN(aktif rendah)**

Program Store Enable merupakan sinyal baca untuk memori program eksternal. Saat mikrokontroler keluarga MCS-51 menjalankan program dari eksternal, PSEN akan diaktifkan dua kali per siklus mesin, kecuali dua aktivitas PSEN dilompati (diabaikan) saat mengakses memori data eksternal.

**j) EA (aktif rendah) / Vpp**

External Access Enable, EA harus selalu dihubungkan ke ground, jika mikrokontroler akan mengeksekusi program dari memori eksternal lokasi 0000h hingga FFFFh. Selain dari itu, EA harus dihubungkan ke Vcc agar mikrokontroler mengakses program secara internal. Kaki ini juga berfungsi menerima tegangan 12 volt (Vpp) selama pemrograman flash.

**k) XTAL 1**

Masukan ke penguat pembalik osillator dan masukan ke clock internal rangkaian.

**l) XTAL 2**

XTAL 2 adalah keluaran dari penguat pembalik osillator.

### 2.1.1.2 Special Function Register

Merupakan daerah dalam memori yang mengatur fungsi khusus dari mikrokontroler 89S52. Dalam memetakan SFR, tidak semua alamat ditempati. Alamat yang tidak ditempati tersebut tidak dapat digunakan dan dianggap sebagai cacat. Penulisan ke dalam atau pembacaan dari register-register cacat tersebut akan menghasilkan nilai-nilai atau kelakuan yang tidak sah. Lebih dianjurkan untuk tidak membaca atau menulis pada alamat SFR yang belum dialokasikan pada sebuah SFR. Apabila tetap dilakukan maka akan menyulut pada kelakuan yang tak terdefinisi dan kemungkinan akan menyebabkan program yang dibuat tidak kompatibel dengan mikrokontroler yang menggunakan SFR untuk beberapa tujuan lain. Adapun SFR yang digunakan dalam alat ini adalah sebagai berikut.

#### a) PSW (Program Status Word)

PSW berisi bit – bit status yang berkaitan dengan kondisi CPU saat itu. Status yang tersimpan dalam PSW meliputi carry bit, dua bit pemilih bank register, *overflow flag*, dan sebuah bit paritas dan dua bit status yang dapat didefinisikan sendiri oleh pemakai.

Carry bit, selain sebagai status bit sisa pada operasi aritmatika, juga bisa digunakan sebagai “akumulator” untuk beberapa operasi Boolean.

#### b) Timer 0 dan Timer 1

Pengaturan timer 0 dan timer 1 terletak pada register yang sama, yaitu TCON dan TMOD yang keduanya dapat diset secara terpisah.

### c) Timer 2 register

T2CON dan T2MOD berisi bit-bit kontrol dan status untuk timer 2. Pasangan register (RCAP2H, RCAP2L) adalah capture/reload register untuk timer 2 pada 16 bit capture mode atau 16 bit auto-reload mode.

### d) Register Interupsi

Setiap sumber interupsi dapat diaktifkan atau dilumpuhkan secara individual dengan mengatur satu bit di SFR yang bernama IE (Interrupt Enable), dan dapat diprogram menjadi satu atau dua tingkat prioritas dengan mengatur IP (Interrupt Priority) seperti yang dijelaskan pada tabel 2.2 dan tabel 2.3.

Alamat T2CON = 0C8h

Reset value = 0000 0000h

**Tabel 2.2** T2CON = Timer / Counter 2 Control Register.

TF2	EXF2	RCLK	TLCK	EXEN2	TR2	C/T2	CP/RL2
7	6	5	4	3	2	1	0

**Tabel 2.3** Fungsi bit pada register interupsi

Simbol	Fungsi
TF2	TF2 akan diset apabila didapati timer 2 overflow dan harus dihapus dengan software. TF2 tidak akan diset apabila RCLK atau TCLK bernilai '1'

EXF2	<p>EXF2 diset saat salah satu dari capture mode atau reload mode disebabkan oleh transisi negatif dari T2EX dan EXEN2 = 1. Ketika interupsi timer 2 diaktifkan, EXF2 = 1 menyebabkan CPU menuju vector interupsi. EXF2 harus dihapus dengan software. EXF2 tidak menyebabkan interupsi pencacah naik / turun (DCEN=1).</p>
RCLK	<p>Receive Clock Enable. Ketika diset akan menyebabkan port serial menggunakan pulsa overflow dari timer 2 untuk diterima pada mode serial 1 dan 3. RCLK=0 menyebabkan overflow dari timer 2 digunakan untuk mengirimkan clock</p>
TCLK	<p>Transmit Clock Enable. Saat diset akan menyebabkan serial port menggunakan pulsa overflow dari timer 2 untuk mengirimkan clock pada mode serial 1 dan 3. TCLK=0 menyebabkan overflow dari timer2 digunakan untuk mengirim clock.</p>
EXEN2	<p>Timer 2 External Enable. Saat diset akan mengijinkan capture atau reload untuk menjadi keluaran dari transisi negatif pada T2EX jika timer 2 tidak digunakan untuk clock serial port. EXEN2=0 menyebabkan timer 2 mengabaikan semua kejadian pada T2EX.</p>
TR2	<p>Untuk memulai dan mengakhiri timer 2. TR2=1 menyebabkan timer 2 mulai.</p>

C/T2	Pemilih timer atau counter pada timer 2. C/T2=0 untuk fungsi sebagai timer. C/T2=1 untuk counter kejadian luar (dipicu dengan transisi tinggi ke rendah).
CP/RL2	Pemilih Capture/Reload. CP/RL2=1 menyebabkan capture untuk terjadi pada transisi negatif pada T2EX jika EXEN2=1. CP/RL2=0 menyebabkan pengisian secara otomatis terjadi ketika overflow timer 2 atau transisi negatif terjadi pada T2EX saat EXEN2=1. Ketika RCLK ataupun TCLK=1, bit ini akan diabaikan dan timer akan dipaksa untuk mengisi secara otomatis pada overflow timer 2

89S52 mempunyai 256 byte RAM di dalamnya. 128 upper byte berada di alamat paralel pada SFR. Artinya bahwa 128 upper byte tersebut mempunyai alamat yang sama dengan SFR tetapi secara fisik terpisah dari SFR.

Ketika sebuah instruksi mengakses sebuah lokasi internal di atas 7FH, alamat mode digunakan untuk menetapkan instruksi meskipun CPU mengakses 128 upper byte dari RAM atau ruang SFR. Instruksi yang menggunakan pengalamatan langsung mengakses ruang SFR.

Contoh di bawah ini merupakan instruksi pengalamatan langsung pada lokasi alamat 0A0H (Port 2).

```
Mov 0A0H, #data
```

Instruksi pengalamatan tidak langsung digunakan untuk mengakses 128 upper byte dari RAM. Sebagai contoh untuk mengakses data pada suatu alamat adalah sebagai berikut.

```
Mov @R0, #data
```

### 2.1.1.3 Perangkat Lunak AT89S52

Mikrokontroler keluarga MCS-51 memiliki 256 perangkat instruksi. Seluruh instruksi dapat dikelompokkan ke dalam 4 bagian yang meliputi instruksi 1 byte sampai 4 byte. Sedangkan kecepatan proses instruksi juga bervariasi dari 1 cycle sampai dengan 4 cycle, yang setiap cycle-nya diselesaikan dengan 12 detak oscillator. Adapun instruksi – instruksi tersebut digolongkan menjadi beberapa jenis instruksi, antara lain.

#### a) Instruksi Transfer Data

Instruksi ini memindahkan data antara register-register, memori-memori, register-memori, antar muka register, dan antar muka memori.

#### b) Instruksi Aritmatika

Instruksi ini melaksanakan operasi aritmatika yang meliputi penjumlahan, pengurangan, penambahan satu (increment), pengurangan satu (decrement), perkalian, dan pembagian.

#### c) Instruksi Logika dan Manipulasi Bit

Melaksanakan operasi logika AND, OR, XOR, perbandingan, pergeseran, dan komplemen data.

#### d) Instruksi Percabangan

Instruksi ini mengubah urutan normal pelaksanaan suatu program. Dengan instruksi ini program yang dilaksanakan akan mencabang ke suatu alamat tertentu. Instruksi percabangan dibedakan atas percabangan bersyarat dan percabangan tanpa syarat.

#### e) Instruksi Stack, I/O, dan Kontrol

Instruksi ini mengatur stack, pembacaan atau penulisan port I/O, serta pengontrolan-pengontrolan.

Sebuah mikrokontroler tidak akan bekerja apabila tidak diberikan program kepadanya. Program tersebut memberitahukan kepada mikrokontroler apa yang harus dilakukan. Sebuah mikrokontroler yang telah bekerja dengan baik pada suatu program, tidak akan bekerja sesuai dengan program itu lagi jika program tersebut telah diganti.

Instruksi-instruksi perangkat lunak berbeda untuk masing-masing jenis mikrokontroler. Instruksi-instruksi tersebut hanya akan dapat dipahami oleh mikrokontroler yang bersangkutan. Sebagai contoh mikrokontroler buatan Intel berbeda dengan buatan Motorola. Bentuk umum semua instruksi dalam keluarga 8052 dapat ditulis sebagai berikut:

*[Label:] Mnemonic [Operand] [,Operand] [:Komentar]*

Jumlah *operand* tergantung pada tipe *mnemonic*. Semua *operand* dapat dibagi menjadi lima kelompok sebagai berikut:



### e.a) Simbol assembler khusus

Assembler MCS-51 telah menyediakan beberapa simbol untuk menunjukkan register tertentu sebagai *operand*. Adapun simbol – simbol tersebut dijelaskan pada tabel 2.4.

**Tabel 2.4** Simbol assembler khusus

Simbol Khusus	Arti
A	Akumulator
R0-R7	Register serba guna
DPTR	Penunjuk data 16 bit
PC	Pencacah program 16 bit
C	Tanda adanya carry
AB	Akumulator/register B

### e.b) Pengalamatan tidak langsung

Operand pengalamatan tak langsung menunjuk ke sebuah register yang berisi lokasi alamat memori yang akan digunakan dalam operasi. Lokasi yang nyata tergantung pada isi register saat instruksi dijalankan. Untuk menjalankan pengalamatan tak langsung digunakan simbol  $\tilde{}$ .

Contoh:

ADD A,  $\tilde{R0}$  : Tambahkan isi RAM yang lokasinya ditunjukkan R0

DEC @,R1 ; Kurangi isi RAM yang alamatnya ditunjuk R1

MOVX @DPTR,A ; Pindahkan isi A yang lokasinya diunjuk DPTR

### e.c) Pengalamatan langsung

Pengalamatan langsung dilakukan dengan memberi nilai ke suatu register secara langsung. Untuk melaksanakan hal tersebut digunakan simbol #.

Contoh:

MOV A,#01H ; Isi akumulator dengan bilangan 01H

MOV DPTR,#1900H ; Isi register DPTR dengan bilangan 1900H

### e.d) Pengalamatan bit

Pengalamatan bit adalah penunjukan alamat lokasi bit baik dalam RAM internal maupun bit-bit SFR. Untuk melakukan pengalamatan bit digunakan simbol titik (.).

Contoh:

SETB TR1 ; Set TR1 (timer 1 on)

SETB 88H.6 ; Set bit 6 pada lokasi 88H (timer 1 on)

### e.e) Pengalamatan kode

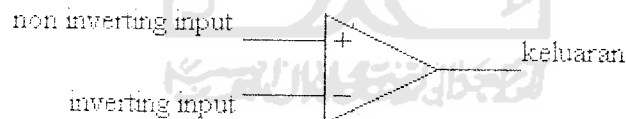
Ada tiga macam instruksi yang dibutuhkan dalam pengalamatan kode, yaitu lompatan relatif, lompatan dalam kotak dan lompatan panjang.

## 2.2 LM 324

LM 324 adalah sebuah komponen elektronika yang berfungsi sebagai op-amp, op-amp merupakan suatu penguat gandengan langsung dan penguatan tinggi yang dilengkapi dengan umpan balik untuk mengendalikan karakteristik tanggapannya secara menyeluruh. Pada penguat operasional terdapat semua keuntungan yang dimiliki suatu untai terpadu monolitik, yaitu ukurannya kecil, keandalan tinggi, hemat biaya dan stabil terhadap perubahan suhu.

Penguat operasional dikatakan ideal apabila mempunyai karakteristik sbb:

- Impedansi masukan tinggi mendekati tak terhingga.
- Impedansi keluaran rendah mendekati 0 ohm.
- Penguatan tegangan adalah tak terhingga.
- Keluaran dapat berubah secara seketika.



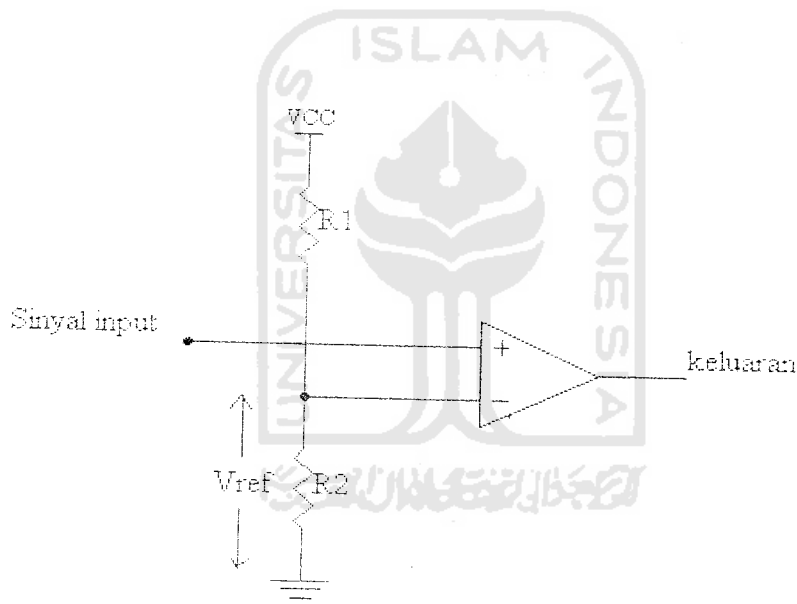
Gambar 2.2 Simbol penguat operasional

### 2.2.1 Op-Amp Sebagai Pembanding (komparator)

Komparator adalah suatu rangkaian yang digunakan untuk mengetahui sinyal mana yang lebih besar dari dua buah sinyal masukan, dengan kata lain apakah suatu sinyal masukan melebihi atau kurang dari nilai acuan yang ditetapkan sebelumnya.

Komparator dapat difungsikan sebagai pembentuk gelombang yang sangat tidak linier, sebab bentuk gelombang keluaran menjadi tidak seperti bentuk gelombang masukannya.

Op-amp yang ideal adalah memiliki penguatan yang tak terhingga, sehingga ketika *level* tegangan masukan lebih besar dari tegangan referensi maka selisih tersebut akan dikuatkan. Walaupun penguatan op-amp ini adalah tak terhingga, tetapi tidak dapat menghasilkan tegangan keluaran yang tak terhingga juga, tetapi maksimal sebesar catu daya.



Gambar 2.3 Op-Amp sebagai pembanding tegangan

### 2.3 Motor DC Magnet Permanen

Motor DC magnet permanen mempunyai prinsip kerja yang sama dengan motor DC *shunt*, seri maupun kompon, perbedaannya terletak pada sumber medan

(field). Pada motor DC jenis lain, medan berasal dari kumparan medan yang mendapat tegangan masukan, tetapi untuk motor DC magnet permanen, sumber medan berasal dari magnet permanennya.

#### 2.4 IRED (*Infrared Emitting Diode*)

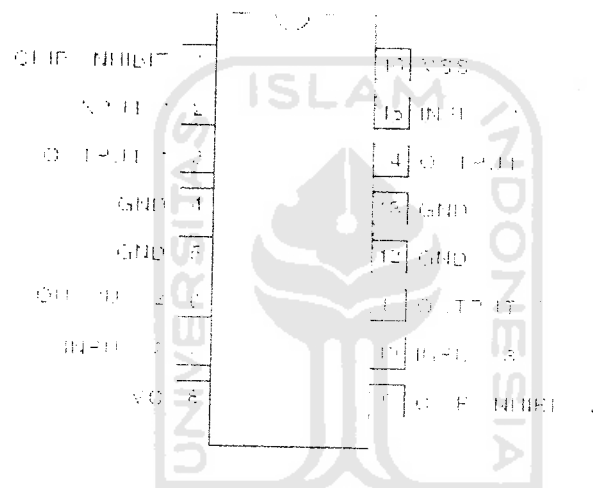
IRED adalah jenis dioda yang dapat memancarkan cahaya infra merah. Cahaya infra merah ini tidak dapat dilihat dengan mata. Implementasi dari infra merah ini biasanya sebagai perantara komunikasi yang berada pada ruangan yang relatif kecil dan tertutup. Pancaran cahaya infra merah tidak dapat menembus benda-benda padat, tetapi apabila cahaya infra merah mengenai suatu benda padat maka akan memantul. Penerapan infra merah paling banyak digunakan pada *remote control* alat-alat elektronik seperti televisi, *air conditioner*, radio tape dan lain sebagainya.

#### 2.5 Fototransistor

Fototransistor adalah sebuah jenis transistor yang cara kerjanya berhubungan dengan cahaya. Pada umumnya cahaya yang diperlukan untuk mengaktifkan fototransistor adalah cahaya infra merah, fototransistor jika dilihat dari cara kerjanya hampir sama dengan sebuah saklar cahaya, apabila sebuah fototransistor terkena sinar infra merah maka kaki kolektor-*emitter* akan tersambung dan berfungsi sebagai saklar yang terhubung singkat. Akan tetapi apabila fototransistor tidak terkena cahaya infra merah atau hanya terkena cahaya biasa maka kaki kolektor-*emitter* tidak terhubung.

## 2.6 H - Bridge L293D

*H - bridge* L293D adalah sebuah *driver push-pull* yang berbentuk segi empat yang mampu menghasilkan keluaran arus 600mA tiap kanal secara terus menerus. Tiap kanal dikontrol oleh sebuah TTL *compatible* unit dengan input *logic* dan tiap pasang (satu buah *hi-bridge*) dilengkapi dengan sebuah masukan (*inhibit*) yang akan mengontrol empat buah transistor yang ada di dalam *IC*.



Gambar 2.4 IC L293D

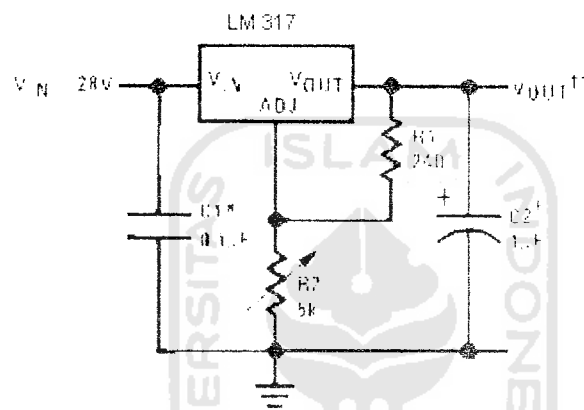
IC jenis *H - Bridge* dengan tipe L. 293D banyak digunakan untuk mengendalikan motor yang mempunyai arus yang kecil. Keunggulan dari *IC* ini diantaranya adalah:

1. Arus keluaran tiap *channel* 600mA.
2. Mudah dalam perancangan.
3. Tahan terhadap *noise*.
4. Murah harganya.

## 2.7 LM 317

LM 317 adalah *regulator* bertegangan positif yang mampu mensuplai lebih - kurang 1,5 A dengan tegangan keluaran yang dapat diatur antara 1,2 V – 37 V.

IC ini mudah digunakan dengan menggunakan dua resistor eksternal untuk mengatur keluaran tegangannya, dan toleransi keluarannya adalah sebesar 1%.



Gambar 2.5 *Adjustable regulator* dengan menggunakan LM 317

## **BAB III**

### **PERANCANGAN**

#### **3.1 Perancangan Sistem Kerja**

Perancangan sistem kerja otomatis robot untuk membuat menara berbasis mikrokontroler diperlihatkan dengan diagram alir pada gambar 3.6.

mikrokontroler berfungsi sebagai pengendali dan komputasi untuk menentukan posisi, mengukur jarak dan menahan agar robot tidak keluar dari lintasan. Setiap saat robot akan membaca lintasan yang berupa garis gelap untuk mempertahankan robot berada pada posisi lintasan garis.

Untuk mendeteksi benda yang menjadi bahan untuk menyusun menara yang berada di depan robot, maka robot dilengkapi dengan kit sensor infrared yang akan membaca jarak benda terhadap robot,

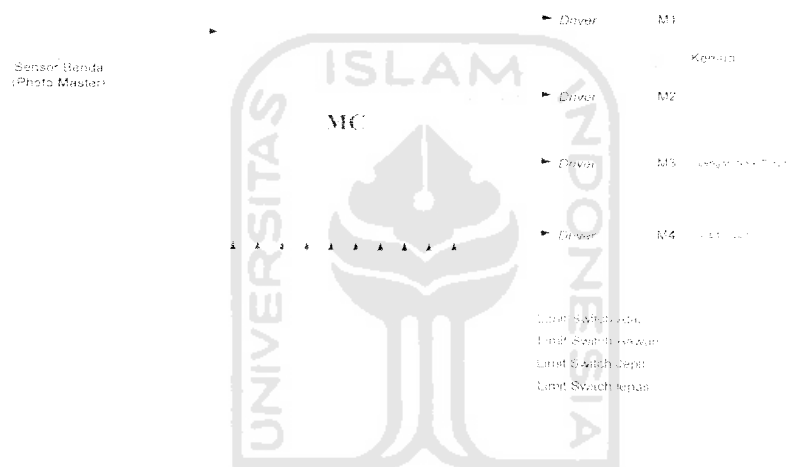
Sistem kemudi robot digerakan oleh dua buah motor DC yang bekerja seirama bila robot berjalan lurus dan akan bekerja secara tidak sinkron ketika robot akan berbelok atau berputar. Untuk menjaga agar robot bergerak mengikuti lintasan yang telah ditentukan dalam hal ini garis yang berwarna gelap, pada robot dipasang sensor infra merah yang berfungsi sebagai pemandu robot untuk mengikuti lintasan garis.

Lengan robot dikendalikan oleh mikrokontroler melalui masukan dari kit sensor infrared yang mendeteksi benda di depan robot dan mencari titik tengahnya agar lengan robot dapat bergerak dan mengambil benda tersebut tanpa terhalang oleh



benda itu sendiri, dan untuk membatasi pergerakan lengan ketika mengangkat, menurunkan, menjepit, dan melepaskan benda digunakan limit switch.

Robot akan berjalan ke tempat yang telah ditentukan setelah menjepit benda yang akan digunakan untuk membuat menara, dan menyusun benda tersebut. robot kembali kejalurnya untuk mengambil benda yang lainnya untuk disusun menjadi menara ditempat yang telah ditentukan.



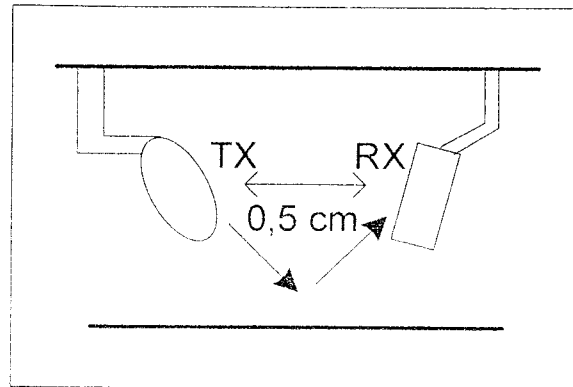
Gambar 3.1 Diagram blok robot

### 3.2 Perancangan Perangkat Keras

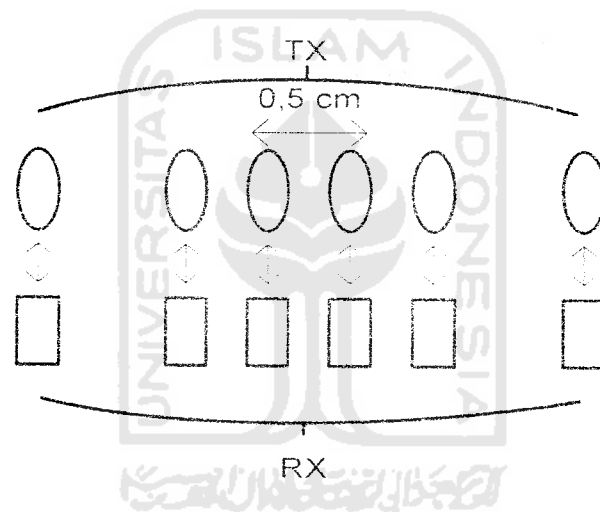
Berdasarkan diagram blok robot diatas, maka perancangan perangkat keras otomatis robot pembuat menara dapat dibagi menjadi tiga bagian, yaitu :

1. Perancangan blok sensor.
2. Perancangan blok mikrokontroler.
3. Perancangan blok driver motor.

### 3.2.1 Perancangan Blok Sensor



Gambar 3.2 Rancangan sensor

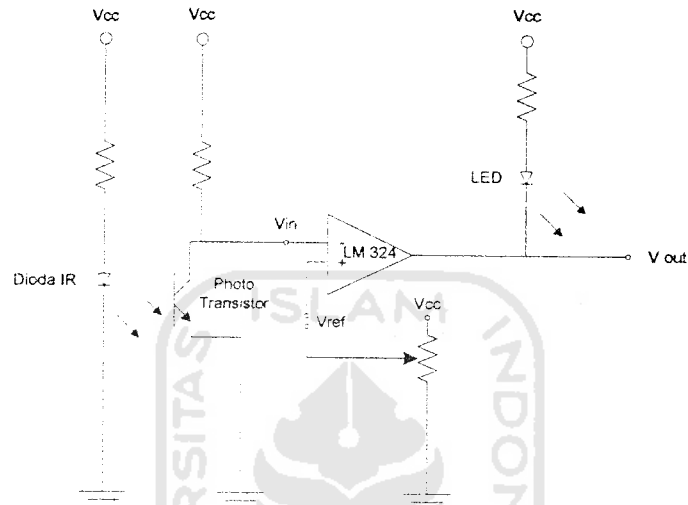


Gambar 3.3 Susunan sensor

Sensor garis yang digunakan pada robot ini berjumlah enam buah. Tiap satu buah sensor terdiri dari sebuah IRED dan fototransistor. Pembuatan rangkaian sensor ini dibuat dengan pola memanjang. Tiap satu buah sensor, IRED dan fototransistor dipasang secara sejajar.

Posisi TX (pemancar) dan RX (penerima) harus benar benar sejajar agar pantulan sinar infra merah dari TX bisa langsung mengenai sensor RX, sehingga pada

bagian ini perlu diatur jarak antara RX dan TX. Ketepatan dalam penempatan TX dan RX ini merupakan salah satu faktor yang mempengaruhi sensitivitas sensor dalam membaca jalur.



Gambar 3.4 Rangkaian sensor garis

Tabel 3.1 Tabel kebenaran

Kondisi Input	Out
$V_{in} > V_{ref}$	Low
$V_{in} < V_{ref}$	High

Pada saat robot berjalan maka sensor pendeteksi jalur akan mendeteksi dengan menggunakan IRED (*Infra Red Emitting Diode*) dan fototransistor. Jika sinar infra merah mengenai bidang yang dapat memantulkan cahaya misalkan warna putih maka cahaya akan dipantulkan dan akan diterima oleh fototransistor. Pada saat fototransistor terkena cahaya infra merah maka kaki *colector* dan *emitter* akan

terhubung seperti sebuah saklar yang tertutup. Hal ini akan menyebabkan tegangan keluaran fototransistor mendekati nol dan lebih kecil dari tegangan referensi yang telah ditentukan. Keluaran fototransistor dihubungkan ke *input inverting* sedangkan tegangan referensi dihubungkan ke *input non inverting*, maka keluaran dari komparator akan berlogika *high*. Apabila sinar infra merah mengenai bidang yang kurang dapat memantulkan cahaya misalkan warna hitam maka sinar infra merah tidak akan dipantulkan, sehingga fototransistor tidak akan menerima cahaya infra merah, maka kaki *colector* dan *emitter* akan terhubung buka. Hal ini menyebabkan tegangan keluaran fototransistor lebih besar dari tegangan referensi yang telah ditentukan dan keluaran komparator akan berlogika *low*.

### 3.2.2 Perancangan Blok Mikrokontroler

Mikrokontroler yang digunakan pada robot ini menggunakan mikrokontroler ATMEGA 89S52 dengan susunan pin seperti yang terlihat digambar 2.1 dan penggunaan pin tersebut pada otomatis robot pembuat menara adalah sebagai berikut:

- a. Pin 1 – 8 adalah bagian dari port 1. Port 1 digunakan untuk memasukan data yang dikeluarkan dari sensor garis kemikrokontroler.
- b. Pin 9 adalah pin reset, yang digunakan untuk mereset mikrokontroler.
- c. Pin 10 – 17 adalah dari port 3. Port 3 digunakan untuk memasukan data yang dikeluarkan dari limit switch dan sensor benda kemikrokontroler.
- d. Pin 18 – 19 adalah pin XTAL 2 (pin 18) dan pin XTAL 1 (pin 19), yang digunakan untuk crystal 128 Mhz sebagai clock mikrokontroler.

- e. Pin 20 adalah pin ground.
- f. Pin 21 – 24 adalah bagian dari port 2, pin ini tidak digunakan.
- g. Pin 25 – 28 adalah bagian dari port 2, dan pin 32 – pin 33 adalah bagian dari port 0, pin – pin ini digunakan untuk driver motor lengan robot (motor pengangkat lengan, dan motor penjepit benda).
- h. Pin 29 adalah pin PSEN.
- i. Pin 30 adalah pin ALE.
- j. Pin 31 adalah pin EA/VPP.
- k. Pin 34 – pin 39 adalah bagian dari port 0, pin ini digunakan sebagai driver motor penggerak robot.
- l. Pin 40 adalah pin VCC.

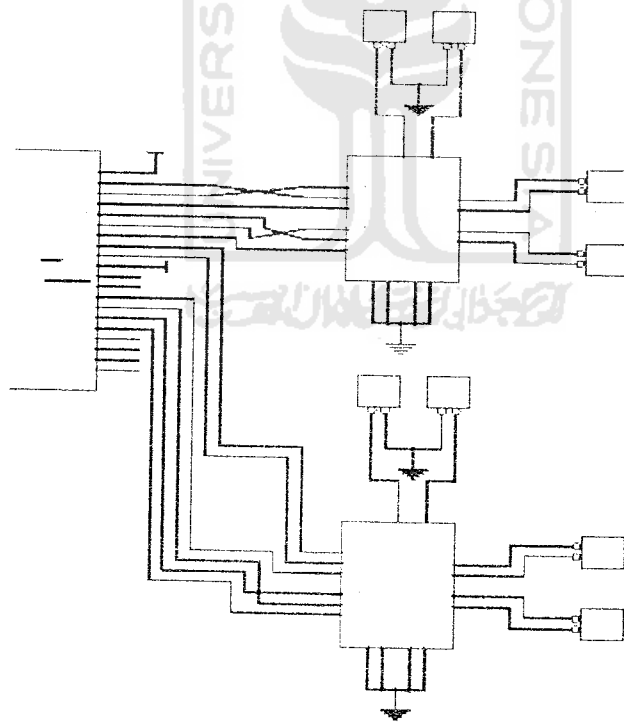
### 3.2.3 Perancangan Blok Driver Motor

Pada robot pembuat menara ini digunakan dua driver motor, yaitu driver motor penggerak robot, dan driver motor lengan robot. Pada setiap driver motor digunakan dua motor DC, pada driver motor penggerak robot, motor digunakan untuk menggerakkan robot maju, mundur, belok kanan, belok kiri, dan berputar. Sedangkan pada driver motor lengan robot, motor digunakan untuk mengangkat lengan robot, menurunkan lengan robot, menjepit benda, dan melepas benda.

Rangkaian driver motor penggerak pada robot ini menggunakan IC L293 dengan susunan pin seperti yang terlihat pada Gambar 2.4 dan penggunaan pin tersebut dibagi menjadi empat bagian, yaitu :

- a. Pin 1 dan 9 adalah pin enable, yang digunakan untuk mengaktifkan logika yang diberikan oleh mikrokontroler pada pin 2, pin 7, dan pin 10, pin 15. Pin 1 untuk enable masukan logika pada pin 2 dan 7, Pin 9 untuk enable masukan logika pada pin 10, dan 15.
- b. Pin 4 dan 5 serta pin 12 dan 13 adalah pin ground.
- c. Pin 3 dan 6 serta pin 11 dan 14 adalah pin output, output dari pin ini digunakan sebagai masukan untuk motor.

Pin 8 dan 16 adalah pin untuk tegangan VCC dan VDD, VCC diberikan pada pin 8 dan VDD diberikan pada pin 16.



Gambar 3.5 Rangkaian driver motor

### 3.3 Perancangan Perangkat Lunak

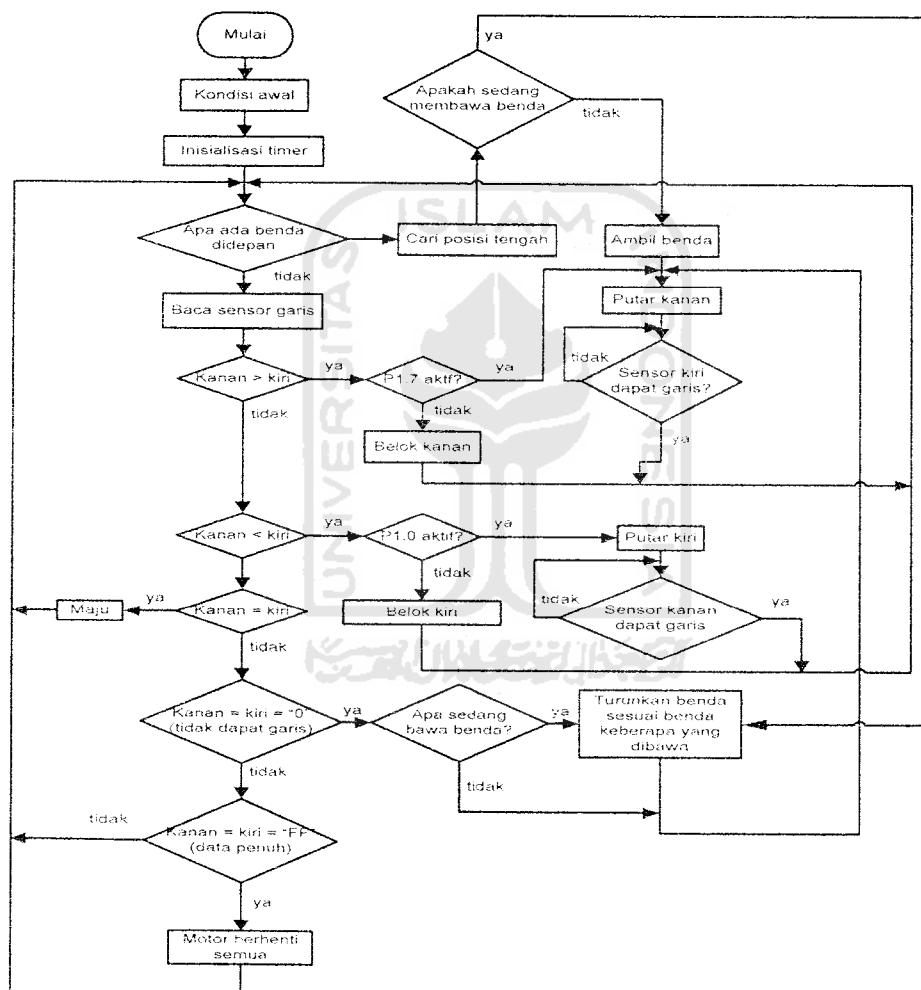
Program untuk berjalan yang dibuat pada robot ini menggunakan sistem perbandingan, mikrokontroler akan selalu mendeteksi jalur garis yang dideteksi oleh sensor infra merah yang berjumlah delapan buah, posisi sensor berurutan mulai dari sensor 0 sampai sensor 7, hasil dari pendeteksian sensor inilah yang akan membuat robot berjalan lurus jika data masukan dari sensor melalui bit 0 sampai bit 7 port 1 ke mikrokontroler seimbang, dan berbelok jika data tidak seimbang, atau berputar jika sensor 0 atau sensor 7 mengenai garis.

Robot akan berhenti berjalan ketika kit sensor infra merah mendeteksi benda. kit sensor infra merah akan memberi masukan data pada mikrokontroler melalui port 3, dan memerintahkan motor lengan dan motor penjepit berputar untuk mengambil benda tersebut.

Pergerakan lengan robot dan penjepit dibatasi oleh limit switch, yang memberi masukan data pada mikrokontroler jika lengan sudah turun sampai posisi benda, atau lengan sudah kembali keposisi atas (posisi awal lengan, dan posisi lengan ketika robot sedang bergerak membawa benda), atau penjepit sudah menjepit benda, atau penjepit sudah melepas benda, melalui port 3.

Setelah robot mendapatkan benda, maka mikrokontroler akan membaca rutin program selanjutnya agar robot berputar dan berjalan sampai akhir garis untuk meletakkan benda, atau menumpuk benda tersebut jika benda tersebut adalah benda kedua atau ketiga.

Setelah robot meletakkan benda maka mikrokontroler akan kembali membaca rutin program awal dan mengulangi lagi pergerakan robot dari semula yaitu membaca garis, dan mencari benda untuk disusun sampai benda yang ditumpuk menjadi tiga tumpukan.



Gambar 3.6 Diagram alir kerja robot



## BAB IV

### PENGUJIAN DAN ANALISA

Pada bab ini akan dijelaskan mengenai pengujian dan analisa terhadap sensor garis robot (sensor infra merah), sensor pendeteksi benda robot (kit sensor infra merah), dan lengan robot.

Pengujian dan analisa dilakukan dengan pengamatan kerja robot dan pengukuran tegangan keluaran dari sensor – sensor. Pengujian dan analisa hasil pengujian berfungsi untuk membandingkan antara perancangan dengan hasil pengujian untuk mengetahui unjuk kerja dari alat.

#### 4.1 Pengujian Sensor Garis (Sensor Infra Merah)

Pengujian dilakukan dengan cara menjalankan robot pada jalur lurus sepanjang 60 cm dan berbelok patah dengan sudut  $135^{\circ}$ . Jalur yang dilintasi berupa garis berwarna hitam dengan tujuh jenis lebar garis, 0,8 cm, 1,5 cm, 1,9 cm, 2,5cm, 3,5 cm, 5 cm, dan 6 cm.

Hasil Percobaan:

##### a. Lebar garis 0,8 cm

**Tabel 4.1** Percobaan sensor garis pada garis dengan lebar 0,8 cm

Jalur Lurus	Belok Patah $135^{\circ}$
Gagal	Gagal

Keterangan : Garis terlalu kecil dan pendeteksian garis oleh sensor menjadi tidak seimbang sehingga robot hanya bergerak berputar.

**b. Lebar garis 1,5 cm**

**Tabel 4.2** Percobaan sensor garis pada garis dengan lebar 1,5 cm

Jalur Lurus	Belok Patah 135 <sup>o</sup>
Berhasil	Berhasil

Keterangan : Sensor dapat bekerja dengan baik.

**c. Lebar garis 1,9 cm**

**Tabel 4.3** Percobaan sensor garis pada garis dengan lebar 1,9 cm

Jalur Lurus	Belok Patah 135 <sup>o</sup>
Berhasil	Berhasil

Keterangan : Sensor dapat bekerja dengan baik.

**d. Lebar garis 2,5 cm**

**Tabel 4.4** Percobaan sensor garis pada garis dengan lebar 2,5 cm

Jalur Lurus	Belok Patah 135 <sup>o</sup>
Berhasil	Berhasil

Keterangan : Sensor dapat bekerja dengan baik.

**e. Lebar garis 3,5 cm**

**Tabel 4.5** Percobaan sensor garis pada garis dengan lebar 3,5 cm

Jalur Lurus	Belok Patah 135 <sup>o</sup>
Berhasil	Berhasil

Keterangan : Sensor dapat bekerja dengan baik.

#### f. Lebar garis 5 cm

**Tabel 4.6** Percobaan sensor garis pada garis dengan lebar 5 cm

Jalur Lurus	Belok Patah 135 <sup>o</sup>
Tidak Stabil	Tidak Stabil

Keterangan : Garis terlalu besar dan pendeteksian garis oleh sensor mengalami kesalahan baca, sehingga robot berjalan tidak stabil.

#### g. Lebar garis 6 cm

**Tabel 4.7** Percobaan sensor garis pada garis dengan lebar 6 cm

Jalur Lurus	Belok Patah 135 <sup>o</sup>
Gagal	Gagal

Keterangan : Garis terlalu besar sehingga menutupi semua sensor.

#### 4.1.1 Analisa Pengujian Sensor Garis (Sensor Infra Merah)

Sensor infra merah akan mendeteksi garis hitam di atas medan berwarna putih atau warna – warna terang. Sinyal infra merah berupa cahaya tidak tampak yang dikeluarkan oleh pemancar (TX) akan dipantulkan ke penerima (RX) jika mengenai medan berwarna putih atau berwarna terang, dan jika sinyal infra merah mengenai garis atau medan yang berwarna gelap maka sinyal tersebut tidak terpantul.

Pada sensor garis terdiri dari enam buah sensor yang tersusun sejajar, pada setiap sensor terdiri dari satu buah pemancar (TX) dan satu buah penerima (RX), enam buah sensor ini terhubung ke mikrokontroler port 1 bit 0, bit 1, bit 3, bit 4, bit 5, dan bit 7 setelah melewati komparator,

Data yang diterima mikrokontroler dari sensor garis akan digunakan untuk input IC L 293 melalui port 0 bit 0, 1, 2, 3, 4, 5. Output L 293 itu sendiri akan digunakan untuk menggerakkan motor DC 1 (kanan) dan motor DC 2 (kiri), motor DC ini digunakan untuk menggerakkan robot maju, mundur, belok kiri, belok kanan, dan berputar.

- - Port 0 bit 0 untuk motor kanan maju.
- Port 0 bit 1 untuk *enable* motor kanan.
- Port 0 bit 2 untuk motor kanan mundur.
- - Port 0 bit 3 untuk *enable* motor kiri.
- Port 0 bit 4 untuk motor kiri maju.
- Port 0 bit 5 untuk motor kiri mundur.

mikrokontroler akan membandingkan data yang dikeluarkan oleh sensor ke bit – bit di port satu. Jika sensor yang terhubung pada bit 0 dan bit 1 yaitu sensor 0 dan sensor 1 mendeteksi garis maka robot akan berbelok ke arah kiri untuk menyetabilkan posisi robot agar garis selalu terdeteksi oleh sensor yang terhubung ke bit 0 dan bit 4 yaitu sensor 0 dan sensor 4 yang mengakibatkan robot berjalan maju dan lurus kembali. begitu juga jika garis dideteksi oleh sensor 4 dan 5 yang terhubung ke bit 4 dan bit 5 maka robot akan berbelok ke arah kanan untuk menyetabilkan posisi robot agar garis

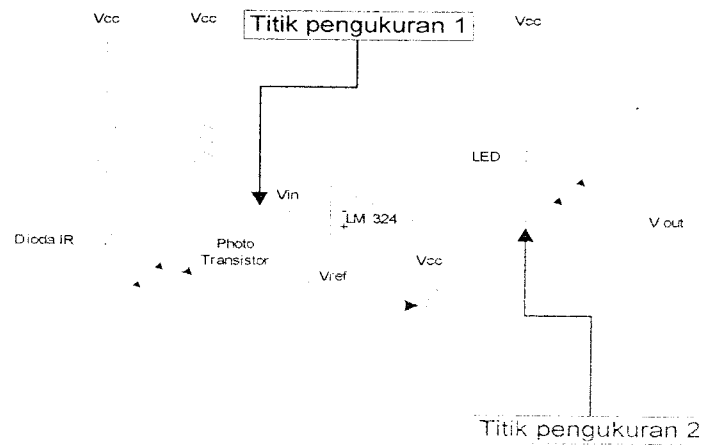
terdeteksi kembali pada sensor 0 dan 4 yang mengakibatkan robot berjalan maju dan lurus kembali.

Jika sensor yang terhubung pada bit 3 yaitu sensor 3 mendeteksi garis maka robot akan bergerak maju beberapa saat lalu bergerak berputar kearah kiri sampai sensor 0 mendeteksi garis dan kembali menyetabilkan posisi untuk bergerak maju dan lurus, begitu juga sebaliknya jika sensor yang terhubung pada bit 7 yaitu sensor 7 mendeteksi garis maka robot akan bergerak maju beberapa saat lalu bergerak berputar kearah kanan sampai sensor 4 mendeteksi garis dan kembali menyetabilkan posisi untuk bergerak maju dan lurus.

Jika robot bergerak hingga ke ujung jalur serta kondisi sensor 3 atau sensor 7 tidak mendeteksi garis dan terus bergerak hingga mengakibatkan keseluruhan sensor garis pada robot tidak mendeteksi garis, maka robot akan bergerak berputar kearah kanan hingga sensor 4 mendeteksi garis dan kembali menyetabilkan posisi untuk bergerak maju dan lurus.

#### **4.1.2 Pengukuran Tegangan Keluaran Sensor Garis**

Pengukuran tegangan keluaran dari sensor garis dilakukan pada dua titik seperti yang ditunjukkan pada gambar 4.1.



**Gambar 4.1** Titik pengukuran tegangan keluaran sensor

Hasil pengukuran tegangan keluaran dari sensor garis pada dua titik tersebut adalah sebagai berikut :

a. Pengukuran titik 1

**Tabel 4.8** Pengukuran tegangan keluaran sensor garis titik1

Pengukuran Tegangan pada titik 1						
sensor	3	1	0	4	5	7
Mendapat Garis	3,4 V	3,14 V	2,65 V	3,29 V	3,02 V	3,34 V
Tidak Mendapat Garis	0,41 V	0,24 V	0,25V	0,27 V	0,29 V	0,3 V

b. Pengukuran titik 2

**Tabel 4.9** Pengukuran tegangan keluaran sensor garis titik2

Pengukuran Tegangan pada titik 2						
Sensor	3	1	0	4	5	7
Mendapat Garis	2,76 V	2,86 V	2,84 V	2,83 V	2,82 V	2,79 V
Tidak Mendapat Garis	3,9 V	3,94 V	3,9V	3,92 V	3,94 V	3,92 V

## 4.2 Pengujian Sensor Pendeteksi Benda (Kit Sensor Infra Merah)

Pengujian dilakukan sebanyak sepuluh kali pendeteksian dengan cara menjalankan robot pada jalur yang telah diletakkan benda berwarna putih (sterofom) berbentuk kotak berukuran 4,5 x 4 x 5 cm pada lintasan jalur lurus sepanjang 60 cm. Jalur yang digunakan berupa garis berwarna hitam dengan lebar garis sebesar 1,9 cm.

Hasil Percobaan :

**Tabel 4.10** Hasil percobaan pendeteksian benda sensor pendeteksi benda

Benda Ke	Hasil Pendeteksian	Jarak Benda Terdeteksi
1	Terdeteksi	12 cm
2	Terdeteksi	13 cm
3	Terdeteksi	13,5 cm
4	Terdeteksi	11,5 cm
5	Terdeteksi	13 cm
6	Terdeteksi	12,5 cm
7	Terdeteksi	13,5 cm
8	Terdeteksi	13 cm
9	Terdeteksi	13,5 cm
10	Terdeteksi	13 cm

### 4.2.1 Analisa Pengujian Sensor Pendeteksi Benda

Sensor pendeteksi benda pada robot ini merupakan sensor infra merah yang mampu mendeteksi benda berwarna putih atau berwarna cerah. Sensor ini memberi masukan data ke mikrokontroler melalui port 3 bit 2. Jarak pendeteksian benda dapat diatur dengan menset gainnya sesuai dengan kebutuhan, tetapi pada pengujian robot ini sensor diset *gain*nya agar dapat

mendeteksi benda pada jarak 13 cm, hal ini dilakukan untuk menyesuaikan jarak benda dengan lengan robot yang akan turun untuk mengambil benda yang dideteksi oleh sensor pendeteksi benda.

Ketika sensor pendeteksi benda mendeteksi benda yang sudah ditumpuk setinggi dua tumpukan atau tiga tumpukan sensor sudah dapat mendeteksi benda dari jarak 15 cm, hal ini disebabkan oleh lebih luas dan lebih tingginya daerah permukaan benda yang menghadap kesensor.

#### 4.2.2 Pengukuran Tegangan Keluaran Sensor Pendeteksi Benda

Hasil pengukuran tegangan keluaran sensor pendeteksi benda ditunjukkan pada table 4.11.

Tabel 4.11 Pengukuran tegangan keluaran sensor pendeteksi benda

Tegangan Out Put Sensor Pendeteksi Benda	
Saat Mendeteksi Benda	Saat Tidak Mendeteksi Benda
5 V	0,03 V

#### 4.3 Pengujian Lengan Robot

Pengujian dilakukan dengan cara menjalankan robot untuk mengambil benda sebanyak 9 kali dan menyusunnya menjadi menara sebanyak tiga buah, setiap menaranya akan terdiri dari tiga tumpukan benda, benda yang akan diambil oleh lengan robot berukuran 4,5 x 4 x 5 cm dengan berat 25 gram



Hasil percobaan :

a. Percobaan menyusun menara 1

**Tabel 4.12** Hasil percobaan menyusun menara 1

Benda ke	Hasil Pengambilan	Hasil Penyusunan
1	Berhasil	Tersusun
2	Berhasil	Tidak Presisi
3	Berhasil	Tidak Presisi

b. Percobaan menyusun menara 2

**Tabel 4.13** Hasil percobaan menyusun menara 2

Benda ke	Hasil Pengambilan	Hasil Penyusunan
4	Berhasil	Tersusun
5	Berhasil	Tidak Presisi
6	Berhasil	Tidak Presisi

c. Percobaan menyusun menara 3

**Tabel 4.14** Hasil percobaan menyusun menara 3

Benda ke	Hasil Pengambilan	Hasil Penyusunan
7	Berhasil	Tersusun
8	Berhasil	Tidak Presisi
9	Berhasil	Gagal

#### 4.3.1 Analisa Pengujian Lengan Robot

Mikrokontroler akan membaca data yang dikeluarkan oleh sensor pendeteksi benda melalui port 3 bit 2. jika sensor pendeteksi benda mendeteksi bahwa ada benda didepan robot maka robot akan menghentikan gerak. jika robot sedang membawa benda maka robot akan menurunkan benda tersebut sesuai dengan benda yang keberapa, dan jika robot tidak dalam posisi

membawa benda maka robot akan menjepit dan mengangkat benda tersebut untuk diletakkan diposisi yang telah ditentukan.

Untuk menjepit dan melepas benda serta naik turun lengan robot digerakkan oleh dua motor DC, yaitu motor DC 3 dan motor DC 4, motor DC 3 digunakan untuk menaikkan dan menurunkan lengan, motor DC 4 digunakan untuk menjepit dan melepas benda. Kedua motor DC ini digerakkan oleh driver motor yaitu IC L293, sedangkan L293 itu sendiri mendapatkan masukan dari mikrokontroler dari :

- - Port 0 bit 6 untuk motor lengan naik (motor DC 3).
- Port 0 bit 7 untuk *enable* motor DC 3.
- Port 2 bit 7 untuk motor lengan turun (motor DC 3).
  
- - Port 2 bit 4 untuk motor lengan melepas benda (motor DC 4).
- Port 2 bit 5 untuk motor lengan menjepit benda (motor DC 4).
- Port 2 bit 6 untuk *enable* motor DC 4.

## BAB V

### PENUTUP

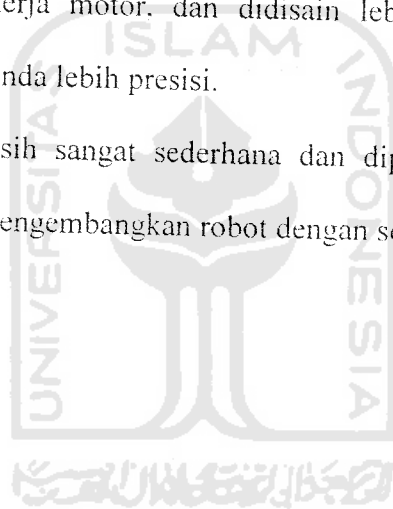
#### 5.1 Kesimpulan

Berdasarkan hasil dari perancangan dan pengujian dapat disimpulkan bahwa:

1. Sensitifitas dari setiap sensor garis (sensor infra merah) berbeda – beda.
2. Saat menurunkan benda kedua dan ketiga ketika robot sedang menyusun menara, Putaran *reverse* digunakan pada motor lengan robot untuk menanggulangi putaran sisa motor.
3. Lengan robot akan bekerja setelah robot mendeteksi titik tengah benda.
4. Kit sensor infra merah pendeteksi benda dapat mendeteksi benda lebih jauh ketika benda yang dideteksi telah mencapai dua tumpukan, hal ini disebabkan oleh lebih luas dan lebih tingginya permukaan benda yang menghadap ke sensor.
5. Garis yang terlalu kecil lebarnya akan menyebabkan salah pembacaan pada sensor garis.
6. Kurang pasnya posisi penjepit saat menjepit benda akan mempengaruhi kepresisian susunan menara.

## 5.2 Saran

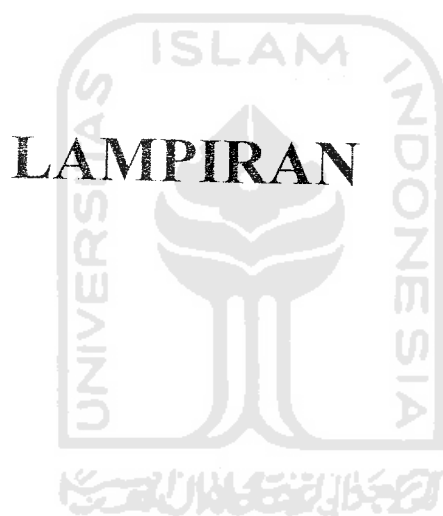
1. Fototransistor dan IRED harus diseting dengan baik karena akan mempengaruhi kerja dari sensor.
2. Sebaiknya menggunakan medan yang bersih untuk dilalui robot pengikut garis karena akan mempengaruhi kualitas pembacaan sensor garis.
3. Lengan robot sebaiknya dibuat dari bahan yang ringan agar tidak membebani kerja motor, dan didisain lebih baik lagi agar dapat meletakkan benda lebih presisi.
4. Robot ini masih sangat sederhana dan diperlukan penelitian lebih lanjut untuk mengembangkan robot dengan sensor infra red.

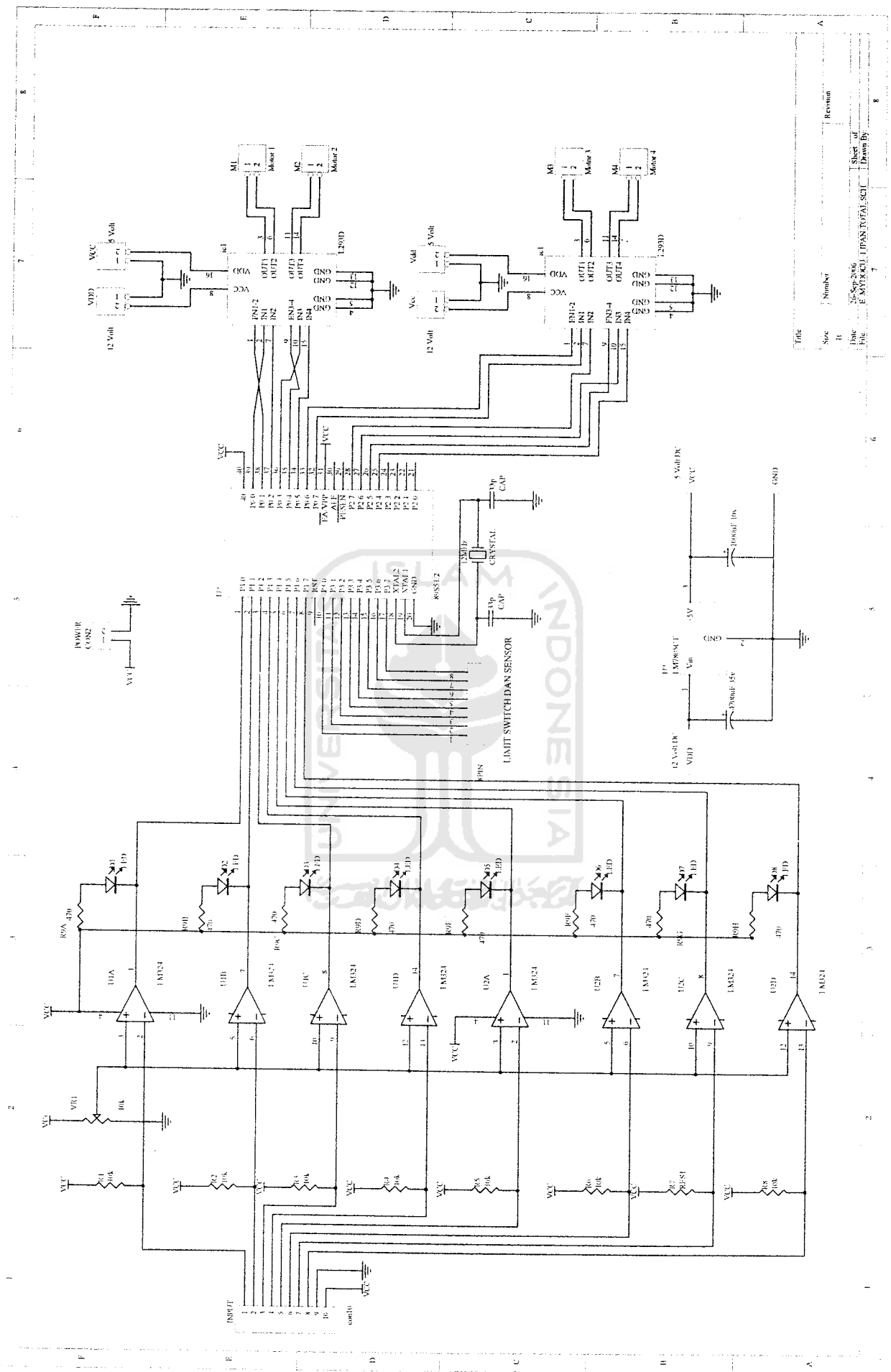


**DAFTAR PUSTAKA**

1. Agfianto Eko Putra, *Belajar Mikrokontroler AT89C51/52/55 Teori dan Aplikasi Edisi 2*, Gava Media, Yogyakarta, 2004
2. Malik, Moh. Ibnu dan Anistardi, *Bereksperimen dengan Mikrokontroler 8031*, Elex Media Komputindo, Jakarta, 1997
3. National Semi Conductor, *Analog Data Book National Operational Amplifiers Data Book*, National Semi Conductor
4. *Atmel 89S52 Technical Data*, Atmel Corporation 2003
5. *L 293D Technical Data*, SGS-Thomson Micro Electronics 1994







File	Number	Revision
H		
Date	Size	Sheet of
20-Sep-2006		8
File	Drawn By	
E:\MUTIKACU\TUPAKS\TUPAKS1.SCH		

Gambar L1.Rangkaian Robot.

```

PROGRAM.ASM
M01      Data      080H ; PORT 0 (UNTUK MOTOR dc 0 DAN 1) (RODA KANAN)
  (KAN KIRI)
  <anan_maju      Bit      80H.0
  <anan_E         Bit      80H.1 ; Enable M0 (Kanan)
  <anan_mundur    Bit      80H.2

  Kiri_E         Bit      80H.3 ; Enable M1 (Kiri)
  Kiri_maju      Bit      80H.4
  Kiri_mundur    Bit      80H.5

  Lengan_naik    Bit      80H.6
  Lengan_E       Bit      80H.7 ; Enable M2
  Lengan_turun   Bit      0A0H.7

sensor      Data      0B0H ;PORT 3 (UNTUK MASUKAN LIMIT SWITCH DARI
MEKANIK)

M23        Data      0A0H ; PORT 2 (UNTUK MOTOR STEPPER 2 DAN 3)

jepit_E     Bit      0A0H.6 ; Enable M3
jepit       Bit      0A0H.5
lepas       Bit      0A0H.4

Mx_0        Bit      0A0H.3
Mx_1        Bit      0A0H.2
Mx_2        Bit      0A0H.1
Mx_3        Bit      0A0H.0

IR          Data      090H ;PORT 1 (MENDETEKSI LINE)

TR2         Bit      0CAH
T2MOD       Data      0C9H
;T2CON      Data      0C8H
TL2         Data      0CCH
TH2         Data      0CDH
TF2         Bit      0C8H.7
RCAP2L     Data      0CAH
RCAP2H     Data      0CBH

IR_KN       Data      20H  ; INFFRA RED SENSOR GARIS KANAN
ir_kn_0     bit      20h.0
ir_kn_1     bit      20h.1
;ir_kn_x    bit      20h.2
ir_kn_2     bit      20h.3

IR_KR       Data      21H  ; INFFRA RED SENSOR GARIS KIRI
ir_kr_0     bit      21h.0
ir_kr_1     bit      21h.1
;ir_kr_x    bit      21h.2
ir_kr_2     bit      21h.3

reg_sensor  data      22h  ; data limit switch yang aktif
benda       bit      22h.2 ; ada benda (set = ada benda, clr = tdk ada)
benda)
jpt_buka    bit      22H.6 ; limit switch untuk penjepit lepas
semua
lengan_bawah bit      22H.5 ; limit switch untuk lengan posisi di bawah
Jpt_tutup   bit      22H.4 ; limit switch untuk penjepit klo
udah maksimal
lengan_atas bit      22H.7 ; limit switch untuk lengan posisi atas

IR_reg      data      23h
reg_1       bit      24h.0 ; set = timer0 menjadi counter up dan

```



PROGRAM.ASM

```

; berhenti dari setb tr0 luar,
; berhenti apabila R5 sudah 0,
mundur
reg_2          bit    24h.1    ; set = bawa benda, clr = tidak bawa benda
kanan_lebih_besar bit    24h.2    ; aktif 1 bila IR_Kn > IR_Kr
kiri_lebih_besar bit    24h.3    ; aktif 1 bila IR_Kn < IR_Kr
Data_sama      bit    24h.4    ; aktif 1 bila IR_Kn = IR_Kr
Data_nol       bit    24h.5    ; aktif 1 bila IR_Kn = IR_Kr = 0
Data_penuh     bit    24h.6    ; aktif 1 bila sensor garis menyala semua
CTRL           data    24h
jumlah        data    25h      ; jumlah tumpukan yang ada
count         data    26h      ; data cacahan dari R5 (TR0)
sample_data    data    27h      ; jumlah sampling data IR
ctr_kanan      data    28h      ; counter lebar dihitung dari kanan
ctr_kiri       data    29h      ; counter lebar dihitung dari kiri
reg_tr1_a      data    2Ah
reg_tr1_b      data    2Bh

sementara      data    2Ch
Smtr_Kn_mj     bit    2Ch.0
Smtr_Kn_E      bit    2Ch.1    ; Enable M0 (Kanan)
Smtr_Kn_mdr    bit    2Ch.2
Smtr_Kr_E      bit    2Ch.3    ; Enable M1 (Kiri)
Smtr_Kr_mj     bit    2Ch.4
Smtr_Kr_mdr    bit    2Ch.5

setting        data    2Dh      ; untuk setting timer 1

org 0000h
LJmp START

org 0003H ; vektor interup dr P3.2
nop
reti

;=====
ORG 000BH ; vector interup timer 0
jmp Interup_tr0
;=====
ORG 001BH ; vector interup timer 1
jmp Interup_tr1
;=====
ORG 002BH
nop
RetI
;=====
Interup_tr0:
Inc R4 ; digunakan unt membentuk signal pulsa pada
kontrol "enable" motor kanan dan kiri
Cjne R4,#09h,tr0_a ; selain itu digunakan untuk
menghitung naik/turun pergerakan motor
clr kanan_e
clr kiri_e
tr0_a:
Cjne R4,#2fh,T0_0
setb kanan_e
setb kiri_e

jnb reg_1, T0_1
Inc R5
mov R4,#00h

```

PROGRAM.ASM

```

T0_1:          ljmp T0_0
              mov R4,#00h
              dec R5
              cjne R5,#00H,T0_0
              clr Tr0
              lcall motor_semua_mati
T0_0:          RetI
;=====
Interup_tr1:  xch a,setting          ; untuk counter down dengan setting
cacahan diregister "setting"
              dec A
              jnz t1_end
              clr tr1
t1_end:       xch a,setting
              reti
;=====
tunda:        mov r3,#0ffh          ;4AB5 ; untuk memberikan tunda
waktu
tunda_C:      mov R2,#0fh
tunda_B:      mov r1,#19h
tunda_A:      djnz r1, tunda_A
              djnz r2, tunda_B
              djnz r3, tunda_c
              ret
;=====
tunda1:       mov r3,#03h          ;465 ; untuk memberikan tunda
waktu sesaat untuk keperluan reverse motor
              ljmp tunda_c
;=====
Turun:        lcall motor_semua_mati ; untuk menggerakkan lengan ke bawah
dengan batasan limit switch yang ada
              setb lengan_turun
              clr lengan_naik
              setb lengan_e
              jb reg_2,turun3
turun1:       lcall cari
turun2:       jnb lengan_bawah,turun1
              clr lengan_turun
              setb lengan_naik
              lcall tunda1
              jmp turun_end
turun3:       lcall cari
              jb lengan_bawah,turun2
              jb tr1,turun3
              jmp turun2
turun_end:    lcall motor_semua_mati
              clr tr1
              reti
;=====
Naik:         lcall motor_semua_mati ; untuk menggerakkan lengan keatas
denga batasan limit switch
              setb lengan_naik
              setb lengan_e
naik1:        lcall cari
              jnb lengan_atas,naik1
              setb lengan_turun
              clr lengan_naik
              lcall tunda1
              lcall tunda1
              clr lengan_e

```

## PROGRAM.ASM

```

                                lcall motor_semua_mati
                                ret
;=====
Mtr_Jepit:                      setb jepit                ; untuk menjepit benda
hingga limit switch aktif
                                clr lepas
                                setb jepit_e
jepit_1:                        lcall cari
                                jnb jpt_tutup,jepit_1
                                lcall tunda
                                lcall tunda
                                clr jepit_e
                                lcall motor_semua_mati
                                ret
;=====
Mtr_lepas:                      setb lepas                ; untuk membuka penjepit hingga limit switch
buka aktif
                                clr jepit
                                setb jepit_e
lepas_1:                        lcall cari
                                jnb jpt_buka,lepas_1
                                clr jepit_e
                                lcall motor_semua_mati
                                ret
;=====
Blk_Kn:                         mov M01,#00h                ; belok kanan, tetapi apabila
ir_kn_2 aktif maka putar kanan
                                jnb ir_kn_2,blk_kn_0
                                jnb ir_kn_x,blk_kn_0
;
                                lcall maju
                                lcall tunda
                                lcall tunda
                                lcall tunda
                                lcall tunda
                                lcall tunda
                                jmp putar_kanan
blk_kn_0:                       setb Kiri_maju
                                clr kiri_mundur
                                setb Kiri_E
                                ret
;=====
Blk_kr:                         mov M01,#00h                ; belok kiri, tetapi apabila ir_kr_2
aktif, maka putar kiri
                                jnb ir_kr_2,blk_kr_0
                                jnb ir_kr_x,blk_kr_0
;
                                lcall maju
                                lcall tunda
                                lcall tunda
                                lcall tunda
                                lcall tunda
                                lcall tunda
                                ljmp putar_kiri
blk_kr_0:                       setb Kanan_maju
                                clr kanan_mundur
                                setb Kanan_E
                                ret
;=====
putar_180:                      lcall putar_knan        ; putar kanan hingga didapat ir_kr_0
= aktif
P180:                          lcall cari
                                mov A,IR_kn

```

PROGRAM.ASM

```

        cjne A,#0fh,P180
        mov A,IR_kr
        cjne A,#0fh,P180
        ljmp zz1
;=====
putar_knan:      setb Kanan_mundur           ; putar kanan hingga dapat
instruksi selanjutnya
                clr kanan_maju
                setb Kiri_maju
                clr kiri_mundur
                setb Kiri_E
                setb Kanan_E
                ret
;=====
Putar_kanan:    lcall putar_knan           ; putar kanan hingga didapat
ir_kr_0 = aktif
zz1:           lcall cari
;             jnb ir_kn_2,zz2
;             mov c,reg_1
;             mov f0,c
;             setb tr0
zz2:           jnb ir_kn_0,zz1
;             LCALL MOTOR_SEMUA_MATI
;             clr tr0
;             mov C,f0
;             mov reg_1,C
;             ret
;=====
putar_kri:      setb Kanan_maju           ; putar kiri hingga didapat
instruksi selanjutnya
                clr kanan_mundur
                setb Kiri_mundur
                clr kiri_maju
                setb Kiri_E
                setb Kanan_E
                ret
;=====
putar_kiri:     lcall putar_kri           ; putar kiri hingga didapat ir_kn_0
= aktif
yy1:           lcall cari
;             jnb ir_kr_2,yy2
;             mov c,reg_1
;             mov f0,c
;             setb tr0
yy2:           jnb ir_kn_0,yy1
;             LCALL MOTOR_SEMUA_MATI
;             clr tr0
;             mov C,f0
;             mov reg_1,C
;             ret
;=====
maju:          setb Kanan_maju           ; maju hingga di dapat instruksi
selanjutnya
                clr kanan_mundur
                setb Kiri_maju
                clr kiri_mundur
                setb Kanan_E
                setb Kiri_E
                ret
;=====
mundur:        setb Kanan_mundur         ; mundur hingga didapat
instruksi selanjutnya
                clr kanan_maju

```

PROGRAM.ASM

```

setb Kiri_mundur
clr kiri_maju
setb Kanan_E
setb Kiri_E
ret
;=====
cari:      mov A,#0ffh
           mov sensor,#0ffh
           mov R0,#00h

cari_1:    mov IR,#0FFh
           anl A,IR
           mov reg_sensor,sensor
           inc R0
           cjne R0,#01h,cari_1
           cpl A
           Anl A,#0BBh
           mov IR_reg,A      ; mencari data garis dan menyimpan hasilnya
pada register IR_KN dan IR_KR
limit switch      ; selain itu untuk membaca kondisi dari

           anl a,#0Bh
           Mov IR_Kr,A

           Mov A,IR_reg
           anl A,#0B0h
           Swap A
           Mov IR_Kn,A

           ret
;=====
Banding_data:  clr kanan_lebih_besar
               clr kiri_lebih_besar
               clr data_sama
               clr data_nol
               clr data_penuh

Bd_0:          mov A,IR_reg
               jnz Bd_1
               setb data_nol
               ljmp Bd_end

Bd_1:          mov a, Ir_kn
               clr c
               subb a,Ir_kr
               jnc Bd_2
               setb kiri_lebih_besar
               ljmp Bd_end

Bd_2:          mov a,Ir_kr
               clr c
               subb a,Ir_kn
               jnc Bd_3
               setb kanan_lebih_besar
               ljmp Bd_end

Bd_3:          mov a,ir_reg
               cjne a, #0BBh, Bd_4
               setb data_penuh
               ljmp Bd_end

Bd_4:          mov a,Ir_kr

```

PROGRAM.ASM

```

        clr c
        subb a, Ir_kn
        jnz Bd_end
        setb data_sama

Bd_end:        ret

;=====
motor_semua_mati:    mov M01,#00h    ; motor semua pada kondisi mati
                    mov M23,#00h
                    ret
;=====
berhenti_cepat:    mov sementara, M01
                   cpl smtr_kn_mj
                   cpl smtr_kn_mdr
                   cpl smtr_kr_mj
                   cpl smtr_kr_mdr
                   mov M01,sementara
                   lcall tunda1
                   lcall tunda1
                   mov M01,#00h
                   ret
;=====
cari_posisi_tengah:    lcall motor_semua_mati ; mencari posisi tengah dari
benda yang dideteksi dengan
                    setb reg_1
                    setb tr0
                    lcall putar_knan ; menghitung lebar benda dn
membagi dua
                    setb tr0

Cpt_1:    mov reg_sensor,sensor
           jnb benda,Cpt_1
           clr tr0
           lcall motor_semua_mati

           mov R5,#00h
           mov R4,#00h
           setb reg_1
           setb tr0
           lcall putar_kri
           setb tr0

Cpt_2:    mov reg_sensor,sensor
           jnb benda,Cpt_2

Cpt_3:    mov reg_sensor,sensor
           jnb benda,Cpt_3
           clr tr0
           lcall motor_semua_mati
           mov ctr_kanan,R5

           mov r5,#00h
           mov R4,#00h
           setb tr0
           lcall putar_knan
           setb tr0

Cpt_4:    mov reg_sensor,sensor
           jnb benda,Cpt_4

```

PROGRAM.ASM

```

cpt_5:          mov reg_sensor,sensor
                jb benda,Cpt_5
                lcall motor_semua_mati
                clr tr0
                mov ctr_kiri,R5

                mov a,ctr_kanan
                mov b,#2h
                div AB
                mov r5,A
                mov R4,#00h
                clr reg_1
                setb tr0
                lcall putar_kri
                setb tr0

Cpt_6:          mov a,r5
                jnz Cpt_6
                clr tr0
                lcall motor_semua_mati
                mov r4,#00h
                ret

;=====
cari_titik_awal:
dideteksi          mov r5,#00h          ; mencari titik awal saat benda

                setb reg_1
                setb tr0
                lcall mundur

titik_awal_1:     mov reg_sensor,sensor
                jb benda,titik_awal_1
                clr tr0
                mov count,R5
                lcall motor_semua_mati
                mov R4,#00h
                ret

;=====
Cari_posisi:     lcall motor_semua_mati ; mencari posisi
                lcall cari_posisi_tengah
                jnb reg_2,CPx2
                ljmp benda_turun

Cpx2:            lcall mtr_tepas
                lcall turun
                lcall mtr_jepit
                lcall tunda
                lcall naik
                setb reg_2
                lcall putar_knan
                lcall tunda
                lcall tunda
                lcall tunda
                lcall putar_kanan
                ret

;=====
Cari_posisi1:   lcall cari_posisi_tengah          ; mencari posisi yang tepat
dengan mengkombinasikan
dan cari posisi tengah lcall cari_titik_awal          ; procedure cari titik awal

                clr reg_1
                mov r5,Count+20
                mov r4,#00h

```

PROGRAM.ASM

```

                                lcall maju
                                setb tr0
CP:                               mov reg_sensor,sensor
                                cjne r5,#00h,cp1

                                lcall cari_arah_benda
                                jmp cari_posisi

;                               lcall mundur
;                               mov r5,#0ffh
;                               clr tr0
;                               ret

Cp1:                              jnb benda,Cp
                                clr tr0
                                lcall cari_posisi_tengah
                                jnb reg_2,CP2
                                ljmp benda_turun
Cp2:                              lcall mtr_lepas
                                lcall turun
                                lcall mtr_jepit
                                lcall tunda
                                lcall naik
                                setb reg_2
                                lcall putar_knan
                                lcall tunda
                                lcall tunda
                                lcall tunda
                                lcall putar_kanan
                                ret

;=====
Benda_turun:                      mov a,jumlah ; procedure lengan turun saat
mengambil benda                  ; atau menurunkan benda (
                                cjne A,#00h,benda_turun_1
benda pertama s/d ke 3)          ;
                                clr reg_2
                                lcall turun
                                lcall mtr_lepas
                                lcall naik
                                inc jumlah
                                ljmp benda_turun_4

Benda_turun_1:                   mov A, jumlah
                                cjne A,#01h,benda_turun_2
                                mov setting,#43H
                                setb tr1
                                lcall turun
                                lcall mtr_lepas
                                lcall naik
                                inc jumlah
                                ljmp benda_turun_4

Benda_turun_2:                   mov A,Jumlah
                                cjne A,#02h,benda_turun_3

                                mov r5,#55h
                                clr reg_1
                                setb tr0
                                lcall maju
                                jb tr0,$
                                lcall cari_posisi_tengah

```



PROGRAM.ASM

```

mov setting,#3DH
setb tr1
lcall turun
lcall mtr_lepas
lcall naik
inc jumlah
ljmp benda_turun_4

Benda_turun_3:    mov r5,#0B0H
                  clr reg_1
                  setb tr0
                  lcall mundur
                  jnb Tr0,$
                  lcall motor_semua_mati
                  mov jumlah,#00h
                  ljmp benda_turun

Benda_turun_4:    clr reg_2
                  lcall kondisi_awal
                  lcall putar_knan
                  lcall tunda
                  lcall tunda
                  lcall tunda
                  lcall putar_kanan
                  ljmp cek_zero
                  ret
;=====
kondisi_awal:    call naik
                  call Mtr_lepas
                  ret
;=====
Cari_arah_benda:    setb reg_1
                  setb tr0
                  lcall maju
                  lcall tunda
                  Mov setting, #20
                  lcall putar_knan
                  setb tr1

Cab_0:           mov reg_sensor,sensor
                  jnb tr1,Cab_1
                  jnb benda, Cab_0
                  lcall motor_semua_mati
                  clr tr1
                  ret

Cab_1:           lcall motor_semua_mati
                  mov setting,#40

                  lcall putar_kri
                  setb tr1

Cab_2:           mov reg_sensor,sensor
                  jnb tr1,Cab_end
                  jnb tr1,Cab_2

Cab_end:         lcall motor_semua_mati
                  clr tr0
                  ret
;=====
START:           lcall tunda
                  lcall tunda
                  call kondisi_awal
                  Mov IE,#10001010B

```

```

                                PROGRAM.ASM
MOV TMOD,#0010010B
MOV T2MOD,#00000000B
MOV TL0,#00H
MOV TH0,#00H
MOV TL1,#00H
MOV TH1,#00H
MOV TL2,#00H
MOV TH2,#00H
MOV RCAP2L,#00H
MOV RCAP2H,#00H

MOV R4,#00 ; REGISTEER LOOPING UNTUK TIMER 0
MOV R5,#00 ; REGISTEER LOOPING UNTUK TIMER 1
;
;
MOV R2,#00 ; REGISTEER LOOPING UNTUK TIMER 2
MOV R3,#00 ;

lcall motor_semua_mati
MOV sensor,#0FFh
MOV jumlah,#00h
MOV CTRL,#00h

cek_zero:                MOV reg_sensor,sensor
                        MOV ir_reg,0FFh
                        JNB benda,Cz_1
                        lcall berhenti_cepat
                        lcall cari_posisi
                        ljmp cek_zero

Cz_1:                   lcall cari
                        lcall banding_data

Cz_1a:                  jnb kanan_lebih_besar,Cz_2
                        lcall blk_kn
                        ljmp Cz_end

Cz_2:                   jnb kiri_lebih_besar,Cz_3
                        lcall blk_kr
                        ljmp Cz_end

Cz_3:                   jnb data_sama,Cz_4
                        lcall maju
                        ljmp Cz_end

Cz_4:                   jnb data_nol,Cz_5
                        lcall berhenti_cepat
                        jnb reg_2,Cz_4a
                        MOV a,jumlah
                        jnz Cz_4b

                        lcall benda_turun
                        jmp Cz_end
Cz_4b:;                 MOV r5,#088h
;                       CLR reg_1
;                       SETB tr0
;                       lcall maju
;                       SETB tr0
Cz_4c:                 lcall cari
                        jnb tr0,Cz_4a
                        jnb benda, Cz_4c
;                       lcall tunda
;                       lcall tunda
;                       lcall tunda

```

```

                                PROGRAM.ASM
                                clr tr0
                                lcall berhenti_cepat
                                clr tr0
                                jmp cek_zero
;
;
;
;
Cz_4a:                ;        clr tr0
                                lcall putar_kanan

                                jmp Cz_end

Cz_5:                 ;        jnb data_penuh,Cz_end
                                lcall motor_semua_mati

Cz_end:               ;        jmp cek_zero

;=====
END

```

