

# Analisis gerak robot *line tracer* dengan program *artificial intelligence* menggunakan Microcontroller AT89S51

I Wayan Widhiada

Jurusan Teknik Mesin Universitas Udayana, Kampus Bukit Jimbaran Bali

---

## Abstrak

Penelitian ini membahas bagaimanakah analisis gerak robot *line tracer* yang menggunakan program AI dengan menggunakan bahasa pemrograman assembler. Perangkat yang digunakan adalah mikrokontroller tipe AT89S51, Meitan ASM 2005 Standart Edition version 1.00 sebagai development tools dan Meitan 2005 sebagai modul perangkat keras. Dari hasil penelitian dapat diketahui bahwa robot *line tracer* dengan AI dapat membedakan warna latar putih dengan garis yang berwarna hitam. Program AI mampu menyelaraskan kerja antara sensor dengan bagian rangkaian driver motor. Sedangkan robot yang tidak menggunakan AI tidak dapat bekerja dengan baik dan roda hanya dapat berputar satu arah.

**Kata kunci:** Robot *line tracer*, program Artificial Intelligence, MCU tipe AT89S51

## Abstract

The research discussed about the work system of The Line Tracer Robot movement analysis, with using the AI program and assembler programming language. The tools that used are microcontroller type AT89S51, Meitan ASM 2005 Standard Edition version 1.00 as development tools and Meitan 2005 as the hardware tools modul. From the result of research, it can be known that The Line Tracer Robot with AI program is able to show differences, among the white as a based colour and black as a line colour. The AI program is able to accelerate/coordinate the system of work, between the censor and the parts of driver motor combinations. The robot that did not use AI program can not run well, and the wheel just able to rotate in one direction.

**Keywords:** Line tracer robot, Artificial Intelligence programming, AT89S51 MCU type

---

## 1. Pendahuluan

Robotika adalah salah satu implementasi dari wacana teoritis yang selanjutnya akan terus berkembang sebagai bahan penelitian. Robot dapat sangat membantu pekerjaan manusia dalam banyak hal dan khususnya mencakup pekerjaan dengan tingkat ketelitian yang tinggi dan dengan demikian juga dapat mencegah kecelakaan pada tubuh manusia.

Robot dapat bekerja di darat, udara, air, sesuai dengan kebutuhan. Pada saat sekarang, sangat banyak Robot dipakai pada Industri besar dan kecil, karena kemampuannya dalam melakukan pekerjaan untuk menghasilkan sesuatu yang sukar dilakukan manusia dengan durasi waktu yang serupa dalam setiap pekerjaannya. Itulah sebabnya mengapa robot menjadi pilihan utama dalam mencakup beberapa pekerjaan tersebut.

Robot pada dasarnya terdiri dari beberapa bagian antara lain: otak (*brain*), umumnya disebut computer; aktuator (*actuator*) dan bagian mekanik (*mechanical part*) seperti motor, piston, roda (*wheels*), gigi (*gears*), pencengkram (*grippers*); Sensor, seperti: mata (*vision*), suara (*sound*), temperatur (*suhu*), pergerakan (*motion*), cahaya (*light*), sentuhan (*touch*), dll.

Dari beberapa bagian tersebut berfungsi untuk mendapatkan informasi berupa data, mengolah, dan melakukan control kerja pada setiap bagiannya. Robot *Line Tracer* adalah robot yang dapat berjalan secara otomatis mengikuti garis yang mempunyai

warna berbeda dengan warna latarnya (Seperti garis hitam dengan background putih). Robot *line tracer* dapat digunakan untuk pemindah barang ataupun pemadam kebakaran.

Ada 3 bagian yang harus dimiliki oleh *line tracer* robot :

1. Roda dan Motor
2. Sensor Cahaya
3. MCU (*microcontroller unit*)

Robot bekerja sesuai masukan yang berupa respon dari lingkungan dan bergerak sesuai perintah atau program yang tertanam dalam otak robot. Disini yang berperan sebagai otak adalah MCU (*microcontroller unit*) yang memiliki I/O (*input/output*) dan memory sebagai penyimpan perintah atau program. Robot *line tracer* yang ada selama ini dirakit tanpa AI.

Seperti seorang operator di dalam sebuah mesin, program menentukan fungsi dan kerja dari bagian mekanik (*mechanical part*) pada sebuah robot. Tetapi beberapa robot *line tracer* yang telah ada berjalan kurang baik, hal inilah yang menjadi alasan penulis untuk merancang program sebagai sebuah AI (*Artificial Intelligence*) dari sebuah robot *line tracer* yang menggunakan sensor infra merah.

Tujuan penelitian ini adalah:

1. Meningkatkan pemrograman AI (*Artificial Intelligence*) atau kecerdasan buatan robot agar robot berjalan selaras.

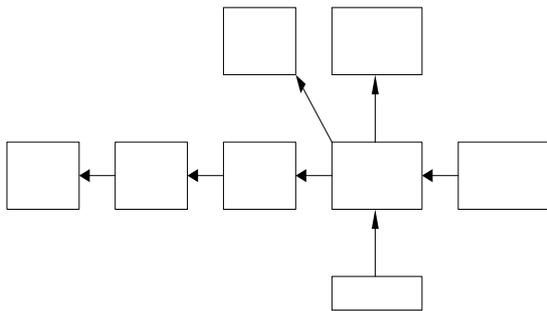
2. Mengetahui kinerja dan respon robot terhadap lingkungan dengan AI (*Artificial Intelligence*) yang telah tertanam di dalam MCU (*microcontroller unit*).
3. Mengetahui fungsi *port-ing* pada MCU (*microcontroller unit*) tipe AT89S51.

## 2. Dasar Teori

### 2.1. Robot

Robot adalah rangkaian peralatan mekanikal dan elektronika yang dirangkai bersama – sama yang bekerja dan beroperasi sesuai dengan instruksi atau program. Robot yang selama ini kita kenal adalah sebuah mesin berbentuk manusia yang dapat berbicara dan berjalan layaknya manusia. Robot tersebut adalah salah satu dari jenis robot berdasarkan bentuknya yaitu kategori Android. Robot jenis ini berbentuk seperti kendaraan yang dilengkapi dengan roda dan bergerak seperti sebuah mobil. Perbedaan dengan mobil adalah kemampuan

Automatic Robot bergerak berdasarkan perintah-perintah yang telah diprogramkan sebelumnya atau berdasarkan masukan dari sensor-sensornya. Robot jenis ini bergerak berdasarkan perintah-perintah yang dikirimkan secara manual baik melalui remote control, PC atau *joystick*.



**Gambar 1. Skema bagian-bagian robot**

Gambar 1 di atas menunjukkan bagian-bagian robot secara garis besar. Tidak seluruh bagian ada pada setiap robot, hal ini dibedakan berdasarkan fungsinya saja. Contohnya, sistem kendali hanya digunakan pada robot yang kategori *teleoperated* saja.

### 2.2. Perancangan program pada AT89S51

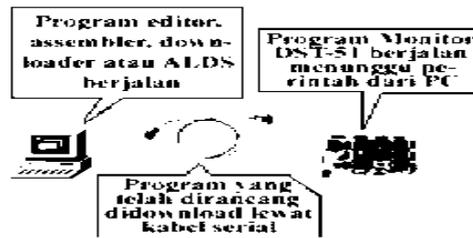
Untuk merancang suatu program pada AT89S51 maka proses dilakukan dalam tahap-tahap berikut.

#### I. Tahap Pengembangan

Tahap ini adalah tahap di mana pembuat rancangan sistem elektronik masih melakukan penulisan program terhadap program rancangannya. Perancangan program akan menghasilkan program rancangan yang baik dan efektif apabila dilakukan dengan tahap demi tahap dan penuh kepastian. Untuk itu digunakan sebuah Development System di mana perancangan program dapat dilakukan secara online di perangkat keras.

Pada tahap ini terdapat dua buah mode kerja, yaitu Mode Monitor di mana download program dan uji coba jalannya program dapat dilakukan dengan

mudah, dan Mode Stand Alone di mana program rancangan diuji coba pada Development System secara stand alone (tidak terhubung dengan PC).

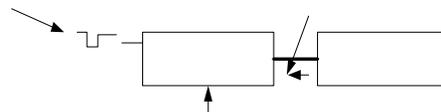


**Gambar 2. Program rancangan didownload ke Development System**

Mode Monitor adalah mode di mana program rancangan hanya akan bekerja saat menerima perintah dari PC. Namun pada aplikasinya sebuah program rancangan harus dapat langsung bekerja saat sistem elektronik mendapatkan sumber daya (Power ON Reset). Untuk itu program harus disimpan di sebuah memori yang tidak akan kehilangan data walaupun tidak mendapat sumber daya.

UV EPROM (*Ultraviolet Erasable Programmable Read Only Memory*) adalah memori yang tidak akan kehilangan data walaupun sumber daya dihilangkan dan seringkali dijumpai di perangkat – perangkat elektronik. Namun UV EPROM memerlukan proses yang rumit dalam mengisi maupun menghapus data - datanya.

EEPROM (*Electrical Erasable Programmable Read Only Memory*) adalah pilihan yang tepat dimana memori tersebut dapat diisi ulang dan dihapus dengan mudah seperti pada RAM (*Random Access Memory*) namun tidak akan kehilangan data walaupun sumber daya dimatikan. Development System DST-51 telah menggunakan memori jenis ini sebesar 8 Kb. Agar Development System bekerja pada kondisi Stand Alone, maka sinyal reset harus dibangkitkan dari kondisi masuknya Sumber Daya ke sistem tersebut sehingga 8951 akan langsung mengeksekusi program yang tersimpan dalam EEPROM saat power supply diaktifkan.



**Gambar 3. Proses Eksekusi Program di Memori Eksternal dengan Power On Reset Rutin Siap Pakai (Built In Routine)**

Rutin Siap Pakai atau Built In Routine adalah potongan-potongan program berupa fungsi-fungsi yang telah disediakan oleh Development System di mana pembuat program hanya perlu melakukan setting pada beberapa register (bila diperlukan) dan memanggilnya. Rutin-rutin ini biasanya tersimpan pada Flash PEROM AT8951 yang tersedia pada Development System. Untuk mengetahui isi rutin-

rutin ini dapat dilihat pada paket dari Development System.

II. Tahap Prototype

Tahap ini adalah tahap di mana program rancangan telah berjalan dengan baik pada Mode Stand Alone. Namun sifat sistem elektronik masih berupa prototype atau sebuah sistem contoh yang masih harus diuji coba kelayakannya. Dalam tahap ini, masih terjadi kemungkinan di mana sistem elektronik tersebut masih harus diubah baik rancangan program maupun konfigurasi perangkat kerasnya. Agar perubahan tetap dapat dilakukan dengan mudah, maka Development System DST-51 juga mempunyai fasilitas untuk beroperasi pada Mode Prototype.

Sebuah prototype memang harus dirancang sangat mendekati sistem akhir yaitu sebuah system yang ringkas mungkin. AT89S51 mempunyai memori Flash PEROM sebesar 4 Kb sebagai media penyimpanan program, oleh karena itu apabila program rancangan yang telah selesai dalam memori EEPROM memang cukup untuk dimasukkan ke Flash PEROM maka program tersebut sebaiknya dipindah ke Flash PEROM sehingga pada tahap akhir, EEPROM dapat dihilangkan.

III. Tahap Akhir

Tahap ini adalah tahap di mana sistem elektronik harus dibuat ringkas mungkin dengan menghilangkan komponen-komponen yang hanya dibutuhkan pada saat tahap pengembangan sama seperti PPI 8255, antar muka RS232 dan EEPROM.

3. Metode Perancangan

3.1. Diagram alir perancangan

Proses perancangan dilakukan sesuai diagram alir perancangan yang ditunjukkan dalam Gambar 4.

3.1.1. Pengujian komponen

Pengujian komponen dilakukan dengan menggunakan multi tester. Tujuan dari pengujian ini adalah memastikan bahwa setiap komponen yang akan dipasang dalam kondisi baik atau hidup sebelum komponen dirangkai pada papan PCB.

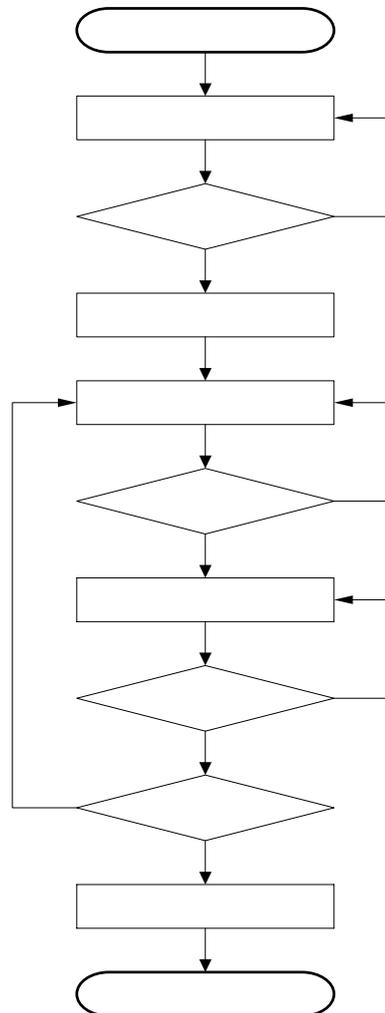
3.1.1. Pemasangan komponen

Setelah skema rangkaian telah dirancang dan digambar dengan software designer pcb (ewb, protel, dll.), hasil design digambar pada papan pcb dan selanjutnya dapat dilakukan pelunturan cetakan. Papan pcb yang telah tercetak dilakukan pemeriksaan secara visual memastikan bahwa cetakan sesuai dengan gambar design, selanjutnya dapat dilakukan pembuatan lubang untuk penempatan kaki – kaki komponen. Langkah selanjutnya adalah pemasangan komponen dengan memperhatikan pada saat penyambungan dengan solder agar tidak terlalu panas untuk menghindari kerusakan komponen.

3.1.3. Pemrograman AT89S51

Pemrograman AT89S51 adalah prosedur mengisi flash memory dengan program yang berupa

kode – kode mesin, kode – kode tersebut memerintahkan MCU untuk melakukan suatu pekerjaan tertentu. Guna mempermudah pemrograman, diciptakan penterjemah atau dalam istilah software disebut compiler, yang mengkonversi bahasa yang lebih tinggi, yang lebih dimengerti oleh manusia.



Gambar 4. Diagram alir perancangan robot dengan AI

3.1.4. Pemasangan rangkaian pada bagian mekanik

Setelah dilakukan proses downloading atau pengisian program pada MCU, rangkaian dapat dipasang pada bagian mekanik atau penggerak. Pada saat pemasangan dipastikan bahwa setiap sambungan pada rangkaian tepat dan tidak kendur. Sebelum dilakukan pengujian dan baterai terpasang pastikan arus terbagi pada setiap komponen dengan menggunakan multi tester.

3.1.5. Pengujian pada lintasan

Pengujian dilakukan dengan lintasan dengan perbedaan warna yang mencolok (putih dengan garis hitam) sebagai panduan sensor. Pengujian ini dilakukan untuk melihat apakah robot dapat merespon terhadap lingkungan. Pada sensor juga perlu dilakukan penyesuaian jarak antara sensor dengan sensor dan sensor dengan garis, hal ini

dikarenakan efektivitas sensor bekerja dengan baik pada jarak tertentu.

3.1.6. Respon AI

Dengan melakukan pengujian pada lintasan kita dapat melihat bagaimana kinerja AI terhadap lingkungan. Hal ini ditunjukkan dengan kerja robot dengan lintasan, bila kurang sesuai maka diperlukan penyesuaian lagi pada tahap pemrograman.

4. Pembahasan

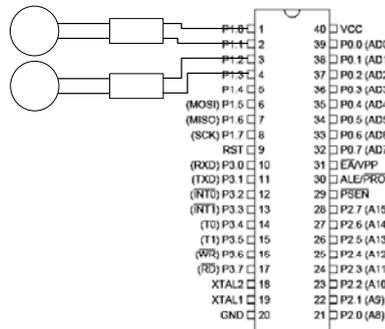
4.1. Program Assembler

Pada MCU harus diberikan diberikan program untuk melaksanakan perintah. Program terdiri dari susunan intruksi – intruksi, seperti halnya kata – kata yang menyusun perintah. Simbol – simbol instruksi seperti CPL, MOV dan seterusnya. Instruksi - instruksi tersebut disebut mnemonic, yang kodenya disebut opcode (operation code) dan biasa diikuti atau tidak oleh salah satu atau dua operand. Opcode menentukan operasi yang akan dilakukan, sedang operand menentukan variable mana yang akan terlibat dalam operasi tersebut, data apa yang akan digunakan atau instruksi apa yang akan digunakan, atau instruksi apa yang akan dilakukan berikutnya. Susunan perintah mnemonic membentuk suatu perintah dalam bahasa assembler (Assembly Language).

Sebenarnya bahasa assembler hanya untuk mempermudah penulisan program bagi pemakai, setiap mnemonic harus diterjemahkan kedalam bahasa mesin (mesin code) untuk dapat dilaksanakan oleh MCU. Perangkat lunak yang digunakan adalah Meitan ASM 2005 Standart Edition version 1.00 sebagai development tools dan Meitan 2005 sebagai modul perangkat keras.

4.2. Pengaturan fungsi port MCU

Dengan perangkat lunak Meitan ASM 2005 Standart Edition version 1.00 sebagai development tools sebelum melakukan pemrograman. User terlebih dahulu harus fungsi – fungsi port yang dipakai pada MCU untuk mengetahui dan menjalankan fungsi kerja dari komponen atau perangkat yang terhubung dengan MCU.



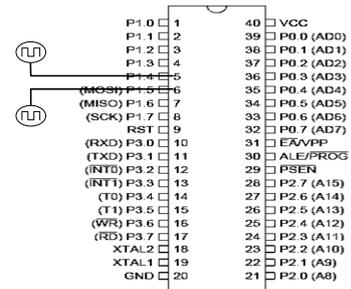
Gambar 5. Penggunaan port MCU untuk motor driver

Pada port 1.0 dan port 1.1 digunakan untuk servo motor bagian kanan, sedangkan untuk port 1.2 dan port 1.3 digunakan untuk servo motor kiri.

Sedangkan untuk mengatur fungsi keluaran pada port MCU akan diatur pada tabel dibawah ini.

Tabel 1. Pengaturan fungsi port menurut kerja motor

Perangkat	Port	Maju	Kanan	Kiri	Mundur
Driver Mtr Kanan	1.0	1	0	1	0
	1.1	0	1	0	1
Driver Mtr Kiri	1.2	1	1	0	0
	1.3	0	0	1	1



Gambar 6. Penggunaan port MCU untuk sensor penerima

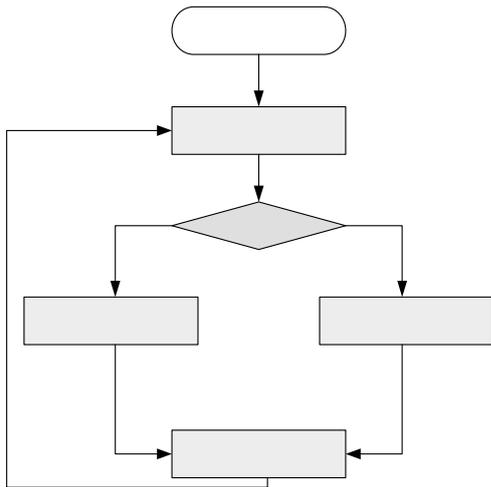
Pada MCU port yang digunakan untuk sensor penerima hanya bersifat menerima respon dari sensor yang berupa tegangan. Respon yang masuk kedalam MCU menunjukkan bahwa adanya perbedaan intensitas cahaya yang masuk kedalam sensor. Selanjutnya pada MCU dengan adanya respon masuk akan mengatur gerak roda sesuai posisi letak roda dan sensor seperti pada tabel dibawah ini.

Tabel 2. Pengaturan gerak roda sesuai respon sensor

Sensor Kanan	Sensor Kiri	Gerak Roda	Roda Kanan Port 1.0/1.1	Roda Kiri Port 1.2/1.3
0	0	Maju	1	1
			0	0
1	0	Belok Kanan	0	1
			1	0
0	1	Belok Kiri	1	0
			0	1
1	1	Mundur	0	0
			1	1

4.3. Pemrograman MCU

Setelah dilakukan pengaturan fungsi kerja pada port MCU, maka selanjutnya dilakukan pemrograman. Untuk mempermudah pemrograman dibuatlah diagram alur kerja dengan memasukkan fungsi port MCU seperti digambarkan pada diagram di bawah ini.



**Gambar 7. Diagram alir kerja AI terhadap respon sensor**

Sehingga semua rangkaian perintah program tersebut menjadi: Tidak

```

;-----
; // Judul      : Program AI
; // Program    : Meitan ASM 2005 Standart Edition version 1.00
; // Keterangan : PROGRAM LINE TRACER
; // Date       : 16-07-2007
;-----
Org      0h
Start:   Clr    P1.4    ;nolkan P1.4
         Clr    P1.5    ;nolkan P1.5
         Mov    A,P1.4  ;pindahkan P1.4 ke accumulator A
         Mov    P1.1,A  ;pindahkan accumulator A ke P1.1
         Mov    P1.0,R0 ;pindahkan register R0 ke P1.0
;-----
Sub Routine R0:
         Cpl    A      ;complement accumulator A
         Mov    R0,A   ;isi register R0 dari A
         Ret                    ;kembali dari sub routine
;-----
End Sub Routine R0
         Mov    B,P1.5  ;pindahkan P1.5 ke accumulator B
         Mov    P1.3,B  ;pindahkan acculator B ke P1.3
         Mov    P1.2,R1 ;pindahkan register R1 ke P1.2
;-----
Sub Routine R1:
         Cpl    B      ;complement accumulator B
         Mov    R1,B   ;isi register R1 dari B
         Ret                    ;kembali dari sub routine
;-----
End Sub Routine R1
         Jmp    Start  ;kembali ke start
End
  
```

**4.4. Penulisan program**

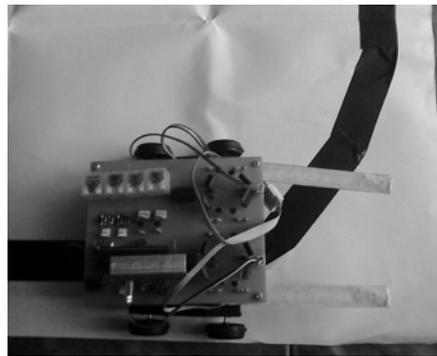
Setelah program ditulis dan disimpan dalam sebuah file, maka selanjutnya program dilakukan download melalui development system kedalam MCU. Setelah proses downloading selesai, file yang telah masuk kedalam MCU dapat kita periksa apakah file tersebut telah masuk kedalam MCU dengan cara membaca melalui tools yang tersedia pada software. Hasil yang terbaca pada software adalah berupa file yang telah dirubah menjadi sebuah hexa oleh software.

**4.5. Pengujian robot line tracer dengan AI**

Pengujian dilakukan dengan memberikan dua warna latar berbeda (hitam dan putih) pada sensor,

dengan memberikan dua warna latar ini kita dapat melihat apakah sensor, program AI dan mekanisme motor servo dapat bekerja dengan baik atau tidak. Pada saat pengujian tanpa litaran ini menunjukkan semua system dapat bekerja dengan baik, yaitu dapat dilihat dengan saat diberi warna latar putih motor servo berputar maju dan pada saat diberi warna hitam motor servo berputar kearah sebaliknya. Tetapi respon servo motor sangat lambat, ini terlihat pada saat diberikan warna latar yang berbeda servo motor tidak dapat langsung berhenti, bahkan pada saat servo motor akan berputar kearah sebaliknya terdapat selang waktu untuk berputar.

Pada saat pengujian di atas lintasan, robot cenderung keluar lintasan. Hal ini dikarenakan pada saat robot bergerak dan sensor mendeteksi warna yang berbeda servo motor tidak dapat berhenti dan berputar kearah sebaliknya dengan spontan. Sehingga pada saat robot akan mulai berhenti dan berputar kearah sebaliknya posisi sensor sudah melewati garis lintasan seperti yang ditunjukkan pada gambar dibawah ini.



**Gambar 8. Robot keluar lintasan saat pengujian**

Saat pengaturan jarak sensor dengan objek latar jarak terkadang sensitivitas sensor berubah – ubah. Pada saat pengujian juga ditemukan disturbance yang terkadang menyebabkan sensor tidak dapat merespon kembali masukan. Sehingga perlu dilakukan reset pada MCU untuk memulai ulang kerja AI.

Pada saat servo motor pada saat satu sisi mulai berhenti sisi yang satu sisi lainnya berputar semakin cepat, dan ini mengakibatkan servo motor yang mulai berputar kearah sebaliknya kehilangan beberapa persen dari arus yang masuk sehingga berputar lebih lambat. Hal ini dikarenakan pembagian arus pada saat transisi arah putaran cenderung lebih besar pada satu sisi yang terus berputar.

**5. Kesimpulan dan Saran**

Berdasarkan pada pembuatan, pemrograman dan pengujian maka dapat disimpulkan sebagai berikut:

1. Improving program dilakukan dengan tidak memberikan notasi pemrograman pengalamatan langsung (*direct addressing*) sehingga respon dapat disimpan pada accumulator yang selanjutnya dilakukan notasi rotasi dan penghapusan dumping.

2. Robot line tracer sudah dapat membedakan respon warna yang ditunjukkan dengan berubahnya arah putaran servo motor saat diberi rangsang warna yang berbeda pada sensor.
3. Fungsi penggunaan fungsi port (porting) adalah untuk mengendalikan dan menerima dari perangkat yang terpasang pada MCU selain perangkat primer/utamanya.

Dengan ditemukannya beberapa kendala mulai dari proses pembuatan sampai pengujian, maka penulis memberikan beberapa saran antara lain:

1. Sensor Phototransistor dengan tipe ST8LR2 perlu digantikan dengan tipe lain yang lebih peka dan lebih aman dari disturbance (gangguan).
2. Pada rangkaian komponen MCU perlu ditambahkan rangkaian reset agar mempermudah mereset saat terjadi disturbance tanpa memutus catu daya dari MCU.
3. Untuk mengatasi pencabangan arus yang tidak seimbang pada setiap rangkaian driver, MCU dan servo motor perlu dipasang catu daya tersendiri.

#### Daftar Pustaka

- [1] Rich, Elaine, 1983, *Artificial Intellegence*, Mcgraw-hill Book Company, Singapore.
- [2] Sudarma, Ir. Made, 2002, *Seri Mata Kuliah Digital Design*, Teknik Elektro Universitas Udayana, Bali.
- [3] Sudjadi, 2005, *Teori Dan Aplikasi Mikrokontroller*, Graha ilmu, Yogyakarta.
- [4] Widyatmo, Arianto; Haryono Eduard; Fendy, 1994, *Belajar Mikroprosesor – Mikrokontroler Melalui Komputer PC*, Elex Media Komputindo, Jakarta.
- [5] Endra Pitowarno, 2006, *Robotika-Disain, Kontrol dan Kecerdasan Buatan*, Yogyakarta.
- [6] Katsuhiko Ogata, 1997, *Modern Control Engineering*, University of Minnesota prentice-Hall International.
- [7] www. ALLDATASHEET.com (Online, diakses: Mei 2007).