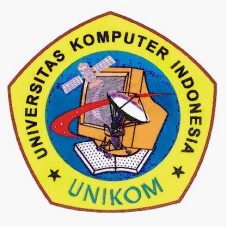
**TUGAS AKHIR**

# KENDALI KECEPATAN MOTOR DC BERDASARKAN PERUBAHAN JARAK MENGGUNAKAN METODE PWM BERBASIS MIKROKONTROLLER AT89C51

**Diajukan Sebagai Salah Satu Syarat dalam Menyelesaikan**

**Jejang Sarjana Strata Satu (S1)**

**Jurusan Teknik Elektro**




**Oleh :**

**TATA RISDIAN ROSDIANA**

**13103027**

**JURUSAN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS KOMPUTER INDONESIA**

**2009**

**LEMBAR PENGESAHAN**

# KENDALI KECEPATAN MOTOR DC BERDASARKAN PERUBAHAN JARAK MENGGUNAKAN METODE PWM BERBASIS MIKROKONTROLLER AT89C51

Telah disetujui dan disahkan sebagai salah satu syarat untuk mendapatkan Gelar Sarjana Strata Satu (S1) pada Jurusan Teknik Elektro
Universitas Komputer Indonesia

Disusun Oleh

**Tata Risdian Rosdiana**
**13103027**

Bandung, Februari 2009
Menyetujui,

**Pembimbing**

**Muhammad Aria, ST.**
**NIP : 4127.70.04.008**

Mengetahui
Ketua Jurusan Teknik Elektro

**Muhammad Aria, ST.**
**NIP : 4127.70.04.008**

**LEMBAR PENGESAHAN**

# KENDALI KECEPATAN MOTOR DC BERDASARKAN PERUBAHAN JARAK MENGGUNAKAN METODE PWM BERBASIS MIKROKONTROLLER AT89C51

Telah disetujui dan disahkan sebagai salah satu syarat untuk mendapatkan Gelar Sarjana Strata Satu (S1) pada Jurusan Teknik Elektro
Universitas Komputer Indonesia

Disusun Oleh
**Tata Risdian Rosdiana**
**13103027**

Bandung, Februari 2009
Menyetujui,

| **Penguji I** | **Penguji II** |
|---|---|
| **Tri Rahajoeningroem, M.T** | **Levy Olivia Nur, M.T** |
| **NIP. 4127.70.04.015** | **NIP. 4127.70.04.014** |

# ABSTRAK

## KENDALI KECEPATAN MOTOR DC BERDASARKAN PERUBAHAN JARAK MENGGUNAKAN METODE PWM BERBASIS MIKROKONTROLLER AT89C51

**Oleh**
**Tata Risdian Rosdiana**
**13103027**

Sistem kendali kecepatan motor DC berdasarkan perubahan jarak ini dikendalikan oleh sistem kendali yang berbasis mikrokontroler dengan menggunakan metode PWM dan sensor ultrasonik (PING). Mikrokontroler mengirim data ke sensor dan dari sensor dikirim lagi kemikrokontroler. Sinyal diterima oleh mikrokontroler akan mengeset lalu memasukan nilai PWM dari hasil data sensor.

Sistem pengereman motor DC mengacu pada sistem penurunan kecepatan motor dari kecepatan penuh sampai dengan berhenti. Masukan data dari hasil sensor maka akan diproses guna mendapatkan nilai *duty cycle* sinyal PWM. Dengan metode ini, kecepatan motor DC dapat dikurangi secara bertahap tergantung dari jarak sensor dengan objek penghalang. Sistem ini dirancang dengan ukuran relatif kecil, oleh karena itu sistem minimal mikrokontroler hanya mengandalkan ragam chip tunggal.

Hasil pengamatan menunjukkan bahwa desain sistem penurunan kecepatan bekerja dengan baik. Ketika kendaraan tidak menemukan adanya objek penghalang, motor berjalan dengan kecepatan penuh. Namun ketika terdapat objek penghalang, kecepatan akan diturunkan secara bertahap sampai kendaraan berhenti pada jarak dekat dengan objek penghalang.

***Kata kunci**: Metode PWM, Sensor Ultrasonik (PING), mikrokontroler AT89C51, driver motor.*

# *ABSTRACT*

## PURSUANT DC MOTOR SPEED REINS TO CHANGE DISTANCE THE USE PWM METHOD MIKROKONTROLER AT89C51 ON BASE

**By**

**Tata Risdian Rosdiana**

**13103027**

*Pursuant DC motor speed condect system to change distance this conduct by which conduct system mikrokontroler on base with use PWM method and (PING) censor ultrasonic. send in Mikrokontroler date to censor and from censor in send again to mikrokontroler. Accept in signal by mikrokontroler will keset then enter price PWM from result of date censor.*

*DC motor system braking refer system degradation motor speed from speed full of shall by stop. In date from result of censor hence will process to get point duty cycle PWM signal. With method this, speed DC motor can in lensen in phases depended from distance censor with the obstacle. System this design with measure relative small, by because that system minimum only mikrokontroler to rely on single chip tune.*

*Perception result refer is that system design degradation work speed with good. vehicle when don't discover the existence of obstacle, speed full with walk motor. When there are obstacle,speed will to degrade the phases shall vehicle stop distance near by with obstacle.*

***Key Word** : PWM method, (PING) Ultrasonic censor, Mikrokontroler AT89C51 , motor driver ,AT Command*

# KATA PENGANTAR

بسم الله الرحمن الرحيم

Assalamu'alaikum Wr. Wb,

Alhamdulillah! Dengan memanjatkan puji syukur kehadhlirat Allah SWT, atas rahmat dan hidayah-Nya serta perlindungan, pertolongan dan ridho-Nya kepada Penulis sehingga Penulis dapat menyelesaikan tugas akhir ini, serta tak terlupakan iringan salam dan sholawat bagi junjungan kami Nabi Besar Muhammad SAW.

Dengan pengerahan segenap usaha, akhirnya penulis dapat menyelesaikan tugas akhir dengan judul.

**Kendali Kecepatan Motor DC Berdasarkan Perubahan Jarak Menggunakan Metode PWM Berbasis Mikrokontroler AT89C51**

Pembuatan dan penyusunan tugas akhir ini diajukan sebagai syarat untuk menyelesaikan studi Strata1 di jurusan Teknik elektro, universitas komputer indonesia (UNIKOM).

Penulis berharap dapat memberikan yang terbaik bagi semuanya dan dapat menjadikan tugas akhir bermanfaat bagi pembaca. Sebagai manusia yang tidak luput dari kesalahan, maka Penulis mohon maaf apabila ada kekeliruan yang sengaja maupun yang tidak sengaja

Wassalamu'alaikum WR.Wb.

Bandung, Februari 2009

Penulis

# UCAPAN TERIMA KASIH

Dalam pelaksanaan dan pembuatan tugas akhir ini, saya sangat bersyukur dan berterima kasih kepada Allah SWT atas segala rahmat dan karunianya. Juga tak lupa Shalawat serta Salam saya haturkan kepada Junjungan kita Rosulullah Muhammad SAW.

Dengan keterbatasan pengetahuan yang ada, penulis tak akan dapat menyelesaikan tugas akhir ini tanpa peran serta pihak lain. Oleh karena itu ijinkanlah penulis untuk menyampaikan ucapan terima kasih yang sebesar-besarnya kepada.

1. Bapak dan Ibuku tercinta yang senantiasa mengiringi setiap langkahku dengan do'a dan kasihnya.
2. Bapak Ir. Eddy Suryanto Soegoto, M.Sc., Rektor Universitas Komputer Indonesia
3. Bapak Prof. Dr. Ir. Ukun Sastraprawira, M.Sc. selaku Dekan Fakultas Teknik Universitas Komputer Indonesia.
4. Bapak Muhammad Aria, ST selaku Ketua Jurusan Teknik Elektro Universitas Komputer Indonesia dan juga sebagai dosen pembimbing yang selalu memberikan bimbingan serta solusi-solusi dan motivasi untuk menyelesaikan tugas akhir ini.
5. Semua dosen dan Staff Teknik Elektro Universitas Komputer Indonesia.
6. Rekan-rekan Himpunan Mahasiswa Teknik elektro khususnya angkatan'03.

7. Rekan-rekan seperjuangan Ismail froshop, David, Ako dower, Agus hebat (thanxs bgt) yang senantiasa meluangkan waktunya. Terima kasih banyak atas kerjasamanya dengan semua ini kita lebih tau makna persahatan dan kebersamaan. “good luck for my n my only friends”
8. Rina Dahli@ yang aku cintai dan yang aku sayangi terima kasih atas do’a dan dukunganya.
9. Teteh Devi yang memberikan semagat, Papah Rina terima kasih atas dukungannya.
10. My Soul’Q yang selalu menemani kemanapun aku pergi.
11. Serta semua pihak yang telah membantu baik secara langsung ataupun tidak langsung yang tidak dapat penulis sebutkan satu persatu.

Di dalam penulisan buku ini, penulis telah berusaha semaksimal mungkin, walaupun demikian penulis menyadari bahwa buku ini masih jauh dari sempurna. Untuk itu penulis akan selalu menerima dengan tangan terbuka segala masukan yang ditujukan untuk menyempurnakan buku ini dan semoga dapat bermanfaat bagi mahasiswa UNIKOM yang akan melaksanakan Tugas Akhir.

Bandung, Februari 2009

Penulis

# DAFTAR ISI

Halaman

Lembar Pengesahan..... i

Abstrak..... iii

*Abstract*..... iv

Kata Pengantar ..... v

Ucapan Terima Kasih..... vi

Daftar Isi ……..... viii

Daftar Gambar..... xi

Daftar Tabel ……..... xiii

Daftar Persamaan..... xiv

**BAB I PENDAHULUAN.....** 1

1.1 Latar Belakang …..……..... 1

1.2 Tujuan .....……..... 2

1.3 Rumusan masalah .....……..... 2

1.4 Batasan masalah ……..... 2

1.5 Metodologi Penelitian…….... 3

1.6 Sistematika penulisan ……..... 3

**BAB II DASAR TEORI …….....** 5

2.1 Mikrokontroller AT89C51..... 5

2.1.1 Deskripsi Mikrokontroller AT89C51……..... 7

2.1.2 Struktur Memori ……..... 10

2.1.3 Special Function Registers (SFR)……... 12

2.2 Bahasa Assembly ……..... 15

2.2.1 Program sumber assembly..... 15

2.2.2 Bagian label..... 16

2.2.3 Bagian kode operasi..... 17

2.2.4 Bagian operand..... 18

2.2.5 Bagian komentar…… 19

2.3 Motor Dc..... 21

2.3.1 Prinsip Kerja Motor DC..... 21

2.3.2 Rangkaian *Ekuivalen* Motor DC …….. 23

2.4 Sensor Ultrasonik……..... 24

2.5 Pulse Width Modulation..... 26

2.5 .1 Modulasi lebar pulsa …… 27

**BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM..** 29

3.1 Perancangan *Hardware*..... 29

3.1.1 Rangkaian Penurun Tegangan 12 Volt menjadi 5 Volt..... 30

3.1.2 Input Referensi (penambahan atau pengurangan kecepatan motor)……. 30

3.1.3 Mikrokontroller AT89C51……..... 31

3.1.4 Driver Motor DC..... 33

3.1.5 Sensor Ultrasonik (PING)..... 34

3.1.6 Pengaturan Duty Cycle (%) 35

3.2 Perancangan dan Realisasi Perangkat Lunak..... 36

3.2.1 Spesifikasi Perangkat Lunak…… 37

3.2.2 Program Utama..... 37

**BAB IV PENGUJIAN ALAT DAN ANALISA.....** 39

4.1 Pengujian dan Analisa Perangkat Keras ……... 39

4.1.1 Pengujian dan Analisa Rangkaian catu daya..... 39

4.1.2 Pengujian dan Analisa Port Mikrokontroller..... 41

4.1.3 Pengujian dan Analisa Rangkaian Driver Motor. 42

4.1.3.1 Pengujian Rangkaian Driver Motor Tanpa Beban..... 42

4.1.3.2 Pengujian Rangkaian Driver Motor Dengan Beban..... 44

4.1.4 Pengujian dan Analisa Sensor Ultrasonik (PING)..... 45

4.1.5 Pengujian Sistem..... 50

**BAB V PENUTUP.....** 54

5.1 Kesimpulan.....……..... 54

5.2 Saran ……..... 55

**DAFTAR PUSTAKA**..... 56

**LAMPIRAN**……..... 57

# DAFTAR GAMBAR

**Gambar 2.1** *Konfigurasi pin mikrokontroller AT89C51*..... 6

**Gambar 2.2** *Blok diagram AT89C51*..... 7

**Gambar 2.3** *Struktur memori mikrokontroler AT89C51*..... 10

**Gambar 2.4** *Alamat RAM Internal dan Flash PEROM*..... 12

**Gambar 2.5** *Lokasi - lokasi Memori pada AT89C51*..... 14

**Gambar 2.6** *Rangkaian Oscilator pada AT89C51*..... 15

**Gambar 2.7** *Induksi gaya pada konduktor*..... 22

**Gambar 2.8** *Gaya pada konduktor*..... 23

**Gambar 2.9** *Rangkaian ekuivalen motor DC*..... 23

**Gambar 2.10** *Ilustrasi Cara Kerja Sensor Ultrasonik*..... 24

**Gambar 2.11** *Diagram waktu sensor*..... 25

**Gambar 2.12** *Sinyal PWM Dengan Variasi Duty Cycle*..... 27

**Gambar 2.13** *Hubungan Nilai Duty Cycle Dengan Tegangan Rara-rata*..... 28

**Gambar 3.1** *Blok diagram sistem*..... 29

**Gambar 3.2** *Skema rangkaian penurun tegangan 12 Volt menjadi 5 Volt*..... 30

**Gambar 3.3** *Rangkaian Mikrokontroller AT89C51*..... 31

**Gambar 3.4** *Driver Motor DC*..... 33

**Gambar 3.5** *Bentuk dari Sensor Ultrasonik (PING)*..... 35

**Gambar 3.6** *Proses pembentukan sinyal PWM*..... 36

**Gambar 3.7** *Diagram alir atau flowchart*..... 38

**Gambar 4.1** *Titik-Titik Pengukuran Tegangan*..... 40

**Gambar 4.2** *Pengujian Rangkaian Driver Tanpa Beban*..... 42

**Gambar 4.3** *Pengujian Rangkaian Driver Dengan Motor*..... 44

**Gambar 4.4** *Pengujian Rangkaian Sensor Ultrasonik (PING)*..... 47

**Gambar 4.5** *Duty Cycle Terhadap Jarak*..... 51

**Gambar 4.6** *Blok Diagram Pengujian Sistem*..... 52

# DAFTAR TABEL

**Tabel 2.1** *Keterangan fungsi pin - pin pada port 3*..... 9

**Tabel 2.2** *Mode berbagai pengalamatan program assembler*..... 20

**Tabel 3.1** *Kebenaran untuk satu kanal*..... 35

**Tabel 4.1** *Nilai Tegangan Tiap Titik*..... 40

**Tabel 4.2** *Hasil Pengujian Driver Tanpa Beban*..... 43

**Tabel 4.3** *Hasil Pengujian Driver Dengan Beban (Motor DC)*..... 45

**Tabel 4.4** *Hasil Pengujian Rangkaian Ultrasonik*..... 48

**Tabel 4.5** *Data Hasil Pengujian Sistem*..... 51

# DAFTAR PERSAMAAN

**Persamaan 2.1**..... 23

**Persamaan 2.2** ..... 24

**Persamaan 2.3** ..... 25

**Persamaan 2.4**..... 27

**Persamaan 2.5**..... 27

**Persamaan 2.6**..... 27

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang Masalah

Reduksi kecepatan dalam konsep pengereman dilakukan berdasarkan jarak kendaraan dengan sebuah benda atau penghalang didepanya. Jenis motor yang digunakan sebagai penggerak dalam kendaraan listrik adalah motor DC yang relatif mudah untuk dikendalikan. Salah satu cara pengendalian kecepatan motor DC yang sering digunakan adalah PWM. Sering kita jumpai keamanan berkendaraan terlihat diabaikan, bahwa dalam berkendaraan meski berhati-hati karena sering terjadi kecelakaan. Akibat jarak dari kendaraan yang ditumpangi sangat dekat dan susah dikendalikan.

Mencegah terjadi kecelakaan dalam kendaaran maka harus dibuat suatu sistem pengereman otomatis dengan menggunakan jarak benda didepannya. Oleh karena itu penulis membuat keamanan pengereman otomatis pada kendaraan yaitu kendali kecepatan motor DC dengan berdasarkan perubahan jarak menggunakan metode PWM. Pengendalian yang dilakukan yaitu mengatur tegangan terminal yang diterapkan ke motor. Keuntungan pengendalian kecepatan motor DC dengan PWM adalah praktis dan ekonomis dalam penerapannya.

Dengan adanya kemajuan teknologi *hybrid* yang memungkinkan penggunaan penyaklaran PWM dengan kecepatan tinggi dapat mengendalikan dengan cara teratur karena metode ini sangat halus untuk pengaturan motor DC, maka pengendalian kecepatan dapat dilakukan dengan baik.

## 1.2 Tujuan Penulisan

Tujuan dari penulisan tugas akhir ini adalah.

1. Merancang metode alternatif kendaraan listrik otomatis penjaga jarak.
2. Merancang metode alternatif kendaraan listrik pengereman otomatis secara bertahap menggunakan metode PWM.
3. Merancang dan mengimplementasikan perangkat keras (*hardware)* dan perangkat lunak (*software*) berbasis mikrokontroller AT89C51 yang berfungsi untuk mengontrol kecepatan motor berdasarkan perubahan jarak.
4. Menganalisis alat berupa sensor ultrasonik PING dan pengaturan motor menggunakan metode PWM.

## 1.3 Rumusan Masalah

Tugas akhir ini menitikberatkan pada pengontrol perputaran motor dengan perubahan jarak menggunakan metode PWM agar mudah dalam melakukan pengoprasian penurunan kecepatan motor. Fungsi sensor ultrasonik PING ini sebagai sarana pengatur jarak yang dihubungkan dengan mikrokontroller. Mikrokontroler dihubungkan dengan driver motor DC.

## 1.4 Batasan Masalah

Pada tugas akhir ini permasalahan dibatasi sebagai berikut.

1. Menggunakan metode PWM untuk pengaturan motor DC.
2. Sensor yang digunakan sensor ultrasonik PING.
3. Tidak membahas motor DC secara *detail* dan perhitungan rpmnya.
4. Jarak minimum dari sensor ke benda sampai berhenti 19 cm.

## 1.5 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Studi literatur dan browsing mengenai PWM, motor DC, mikrokontroler AT89C51 dan sensor ultrasonik PING.
2. Merancang dan membuat sistem secara hardware dengan IC mikrokontroler AT89C51 dengan menggunakan bahasa *Assembler*.
3. Menguji, menganalisa dan membuat kesimpulan dari kendali kecepatan motor DC berdasarkan menggunakan metode PWM berbasis mikrokontroler AT89C51.

## 1.6 Sistematika Penulisan

Dalam penulisan tugas akhir ini, materi-materi pembahasan disusun sebagai berikut.

**Bab I Pendahuluan,** bab ini berisi latar belakang, tujuan, perumusan masalah, batasan masalah dan metodologi penelitian tugas akhir ini.

**Bab II Teori Pendukung,** pada bab ini penulis membahas tentang teori dasar yang berhubungan dengan topik sistem kendali kecepatan motor dengan perubahan jarak menggunakan metode PWM , yaitu : *Mikrokontroller AT89C51*, *Motor DC, driver motor, PWM (pulse Width Modulation), Sensor ultrasonik PING.*

**Bab III Perancangan dan Implementasi Sistem,** pada bab ini penulis membahas secara menyeluruh tentang perancangan perangkat serta menjelaskan fungsi dan prinsip kerja tiap blok diagram dari rangkaian.

**Bab IV Pengujian Alat dan Analisa**, pengujian perangkat dan analisa data pengamatan ini ditulis untuk membuktikan kebenaran dari teori-teori elektronika yang mendukung dalam merancang perangkat.

**Bab V Penutup**, pada bab akhir ini penulis memberikan kesimpulan dari seluruh pembahasan tugas akhir ini, serta saran dan kritik untuk pengembangan. Tugas Akhir ini di lengkapi pula dengan lampiran foto perangkat, gambar rangkaian lengkap, *listing program*, daftar komponen serta *data sheet*.

# BAB II

# DASAR TEORI

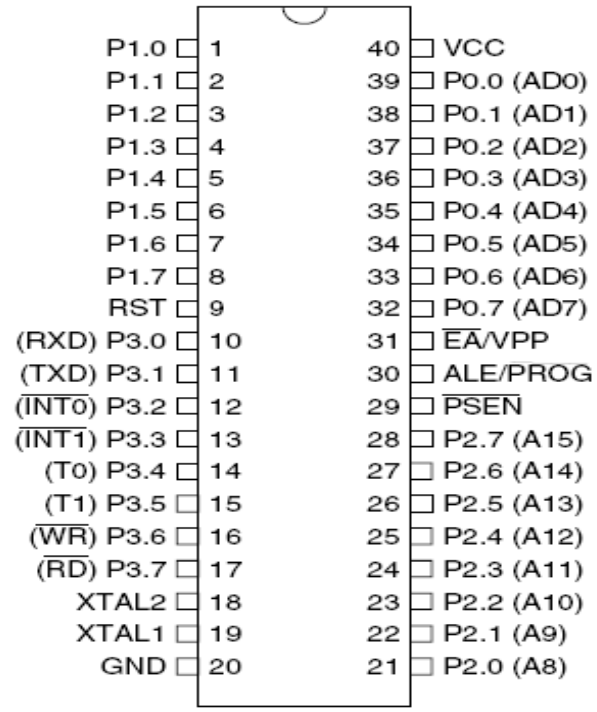
## 2.1 Mikrokontroller AT89C51

Mikrokontroler adalah suatu piranti yang digunakan untuk pengolahan data-data biner (*digital*) yang didalamnya merupakan gabungan dari rangkaian-rangkaian elektronik yang dikemas dalam bentuk suatu chip *Integrated Circuit (*IC). Mikrokontroler AT89C51 merupakan mikrokontroller dengan arsitektur MCS-51 seperti 8031 dengan memori Flash *Programmable and Erasable Read Only Memory (PEROM)*. Memori ini biasa digunakan untuk menyimpan intruksi (perintah) berstandar MCS-51 sehingga memungkinkan mikrokontroller ini untuk bekerja dalam *mode single chip opereation* (mode operasi keeping tunggal) yang tidak memerlukan *external memory* (memori luar) untuk menyimpan *source code* tersebut.

Mikrokontroler AT89C51 memiliki keistimewaan sebagai berikut.

1. Sebuah CPU 8 bit.
2. Osilator internal dan pewaktu.
3. RAM internal 128 byte.
4. Empat buah programmable port I/O, masing-masing terdiri atas 8 buah jalur I/O.
5. Dua buah *timer/counter* 16 bit.
6. Lima buah jalur *interupsi* (2 buah *interupsi eksternal* dan 3 buah *interupsi internal*).

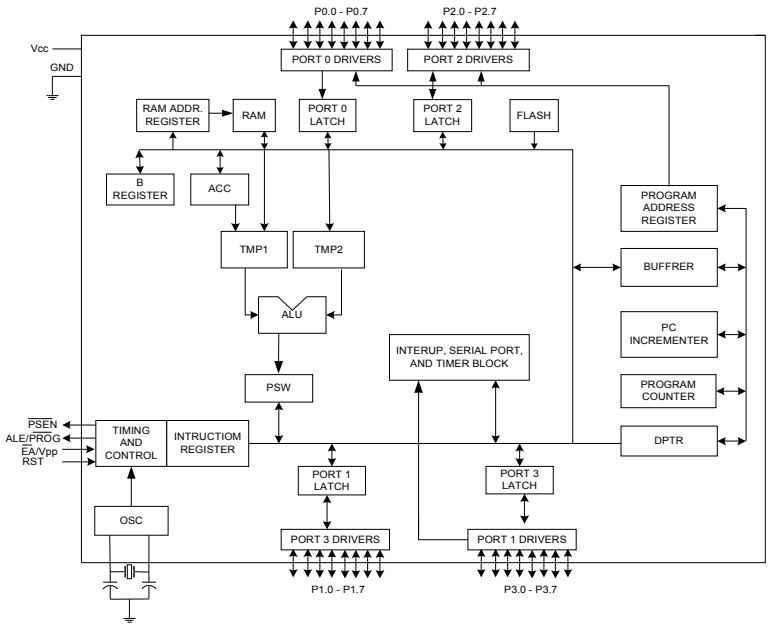
7. Sebuah port serial dengan *control serial Full Duplex UART*.
8. Kemampuan melaksanakan operasi perkalian, pembagian dan operasi Boolean.
9. Kecepatan pelaksanaan intruksi dari 4 MHz sampai 24 MHz.




**Gambar 2.1** *Konfigurasi Pin Mikrokontroller AT89C51*

Konfigurasi pin mikrokontroler ditunjukkan pada **Gambar 2.1** dan fungsi dari pin-pin tersebut dijelaskan pada paragraf di bawah.

### 2.1.1 Deskripsi Mikrokontroller AT89C51




**Gambar 2.2** *Blok Diagram AT89C51*

Diagram blok dari inti AT89C51 ditunjukkan pada **Gambar 2.2** dan fungsi dari masing masing bagian adalah sebagai berikut :

**1. Port 0**

Terdiri atas pin 32 sampai pin 39. Pin - pin ini dapat berfungsi sebagai I/O biasa, *low order multiplex address/data* ataupun menerima kode byte pada saat *Flash Programming* Pada fungsi sebagai I/O biasa *port* ini dapat memberikan *output sink* ke delapan buah TTL input atau dapat diubah sebagi input dengan memberikan logika 1 pada port tersebut. Pada fungsi sebagai *low order multiplex address/*data *port* ini akan mempunyai *internal pull up*. Pada saat *Flash* Programming diperlukan *external pull up* terutama pada saat *verifikasi* program.

**2. Port 1**

Port 1 berfungsi sebagai I/O biasa atau menerima *low order address bytes* selama pada saat *flash programming. Port* ini mempunyai internal *pull up* dan berfungsi sebagai *input* dengan memberikan logika 1 Sebagai *ouput port* ini dapat memberikan *output sink* keempat buah input TTL. *Port* ini terdiri dari pin 21 sampai pin 28.

**3. Port 2**

Port 2 berfungsi sebagai I/O biasa atau *high order address*, pada saat mengakses memori secara 16 bit (Movx @ Dptr). Pada saat mengakses memori secara 8 bit, (Mov @Rn) port ini akan mengeluarkan isi dari *P2 Special Function Register Port* ini mempunyai *internal pull up* dan berfungsi sebagai input dengan memberikan logika 1 Sebagai *ouput*, *port* ini dapat memberikan *output sink* ke empat buah input TTL.

**4. Port 3**

Sebagai I/O biasa *Port* 3 mempunyai sifat yang sama dengan *Port* 1 maupun *Port* 2. Sedangkan sebagai fungsi spesial *port-port* ini mempunyai keterangan yang terdapat pada **Tabel 2.1** dibawah ini :

**Tabel 2.1** *Keterangan Fungsi Pin-Pin pada Port 3*

| Port | Pin | Fungsi |
|---|---|---|
| P3.0 | RXD | Port Serial Input |
| P3.1 | TXD | Port Serial Output |
| P3.2 | INT0 | Port External Interrupt 0 |
| P3.3 | INT1 | Port External Interrupt 1 |
| P3.4 | T0 | Port External Timer 0 Input |
| P3.5 | T1 | Port External Timer 1 Input |
| P3.6 | WR | External Data Memory Write Strobe |
| P3.7 | RD | External Data Memory Write Strobe |

**5. Reset (RST)**

Pin 30 / Reset akan aktif dengan memberikan input high selama 2 *cycle*.

**6. XTAL 1**

Pin 19 merupakan Input untuk penguat *inverting* osilator dan input untuk rangkaian pengoperasian *internal clock*.

**7. XTAL 2**

Pin 18 adalah keluaran dari penguat *inverting osilator.*

**8. Pin 30**

Pin ini dapat berfungsi sebagai *Address Latch Enable* (ALE) yang me-*latch low byte address* pada saat mengakses memori eksternal. Sedangkan pada saat *Flash Programming* (PROG) berfungsi sebagai pulse input untuk operasi normal ALE akan mengeluarkan sinyal *clock* sebesar 1/16 frekuensi *oscillator* kecuali pada saat mengakses memori *eksternal* sinyal *clock* pada pin ini dapat pula didisable dengan men-set bit 0 dari *Special Function Register* di alamat 8EH ALE hanya akan aktif pada saat mengakses memori eksternal (MOVX & MOVC).

**9. Pin 29 (PSEN)**

Pin ini berfungsi pada saat mengeksekusi program yang terletak pada memori *eksternal*. PSEN akan aktif dua kali setiap *cycle.*

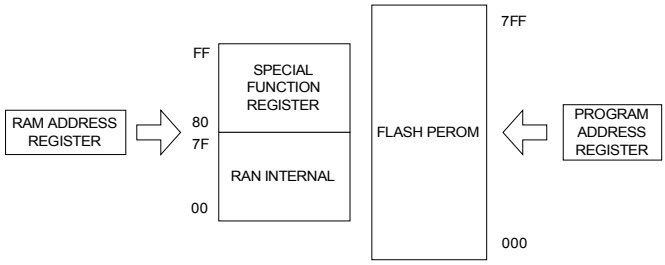
**10. Pin 31 (EA)**

Pada kondisi *low*, pin ini akan berfungsi sebagai EA yaitu mikrokontroller akan menjalankan program yang ada pada memori eksternal setelah sistem direset, jika berkondisi *high*, pin ini akan berfungsi untuk menjalankan program

yang ada pada memori internal. Pada saat *Flash* Programming pin akan mendapat tegangan 12 volt (VP).

### 2.1.2 Struktur Memori

Mikrokontroller MCS-51 memiliki pembagian ruang alamat untuk program dan data, sebagaimana ditunjukkan pada **Gambar 2.3**. Memori program hanya dapat dibaca tidak dapat ditulisi. Sedang memori data dapat ditulisi. Program yang berukuran lebih dari kapasitas EEPROM (4KB untuk AT89C51, dan 8 KB untuk AT89C52) disimpan di EEPROM *eksternal*. Sinyal yang membolehkan pembacaan dari memori program *eksternal* adalah dari pena *Program Store Enable* (PSEN).




**Gambar 2.3** *Struktur memori mikrokontroler AT89C51*

Mikrokontroler intel 98C51 memiliki struktur *memory* yang terdiri atas.

1. RAM *Internal*, memory sebesar 128 byte yang biasanya digunakan untuk menyimpan *variable* atau data yang bersifat sementara.
2. *Special Function Register* (Register Fungsi Khusus), memori yang berisi *register-register* yang mempunyai fungsi fungsi khusus yang disediakan oleh mikrokontroller tersebut, seperti timer, serial dan lain – lain.

3. Flash PEROM, memori yang digunakan untuk menyimpan *intruksi-intruksi* MCS51.

AT89C51 mempunyai sistem memori yang terpisah antara RAM *internal* dan *Flash* – PEROMnya. Seperti yang tampak pada **Gambar 2.4** RAM *internal* dialamati oleh RAM *Address register* (*Register* Alamat RAM) sedangkan Flash PEROM yang menyimpan perintah-perintah MCS-51 dialamati oleh *Program Address register* (Register Alamat Program). Dengan adanya struktur memori yang terpisah tersebut, walaupun RAM *internal* dan *Flash* PEROM, mempunyai alamat awal yang sama yaitu 00 namun secara fisiknya kedua memori tersebut tidak berhubungan.

RAM *internal* terdiri atas :

1. *Register Banks*

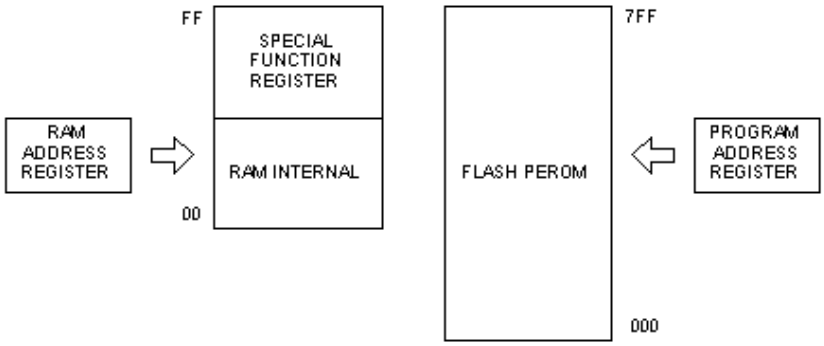
AT89C51 mempunyai 8 buah *register* yang terdiri dari $R_0$ hingga $R_7$. *Register-register* tersebut selalu terletak pada alamat 00H hingga 07H pada setiap kali di-*reset*. Posisi $R_0$ hingga $R_7$ dapat dipindah ke *Bank* 1 (08H hingga 0FH), *Banks* 2 (10H hingga 17H), dan *Bank* 3 (18H hingga 1FH) dengan mengatur bit RS0 dan RS1.

2. Bit *Addressable* RAM

RAM pada alamat 20 H hingga 2 FH dapat diakses secara pengalamatan bit sehingga hanya sebuah intruksi setiap bit dapat di-*set*, *clear* AND dan OR.

3. RAM keperluan umum

RAM pada alamat 30H hingga 7FH dapat diakses dengan pengalamatan langsung maupun tak langsung.




**Gambar 2.4** *Alamat RAM Internal dan Flash PEROM*

### 2.1.3 Special Function Registers (SFR)

*Special Function Registers* (SFR) yang dimiliki oleh AT89C51 sebanyak 21 SFR yang terletak pada alamat 80H hingga FFH. Beberapa dari SFR mampu dialamat dengan pengalamatan bit. Di bawah ini beberapa *register* pada SFR, yaitu:

1. *Accumulator*

Register ini terletak pada alamat E0H. *Accumulator* banyak digunakan untuk operasi aritmatika dan operasi logika. *Register* ini juga diperlukan pada proses pengambilan dan pengiriman data ke memori *eksternal*

2. *Port*

AT89C51 mempunyai 4 buah *port*, yaitu *port* 0, *port* 1, *port* 2 dan *port* 3 yang terletak pada alamat 80H, 90H, A0H dan B0H. Semua *port* ini dapat diakses dengan pengalamatan bit.

*3. Stack Pointer (SP)*

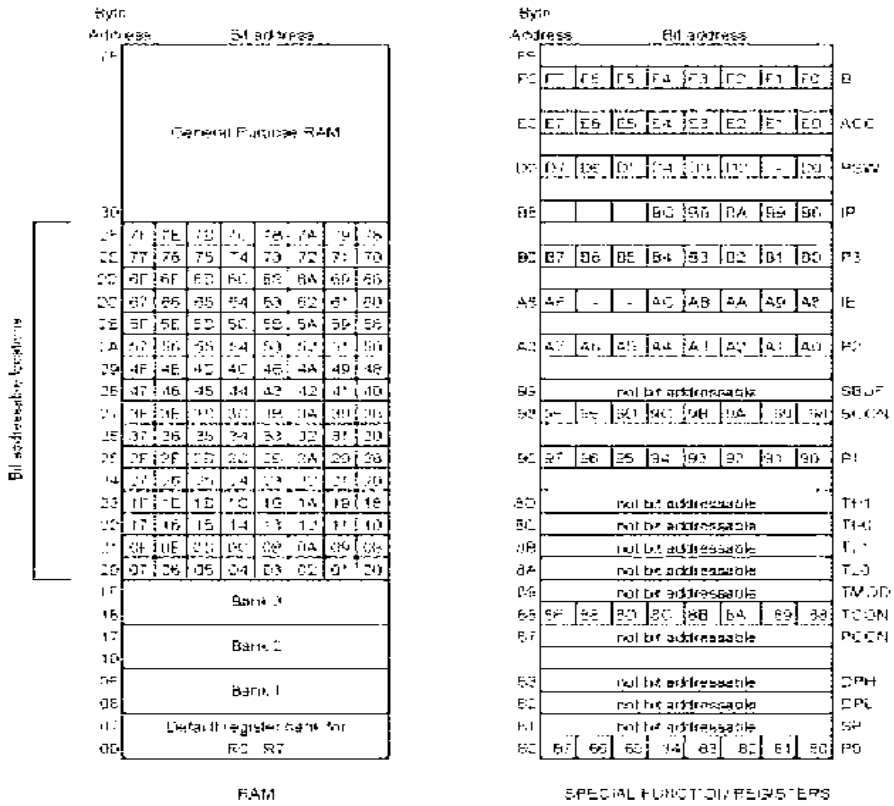
*Stack pointer* adalah suatu *register* yang menunjuk pada *stack*, nilai pada *stack pointer* akan bertambah jika data disimpan pada *stack* melalui perintah *PUSH*, *CALL* atau rutin interupsi dilaksanakan.

*4. Data pointer (DPTR)*

*Register* ini merupakan *register* 16 bit yang terdiri atas *register* DPL & DPH.

*5. Register Timer*

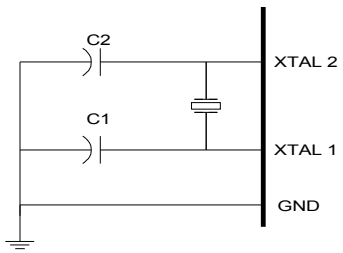
AT89C51 mempunyai dua buah 16 bit *Timer / Counter*, yaitu *Timer* 0 dan *Timer* 1.




**Gambar 2.5** *Lokasi - lokasi Memori pada AT89C51*

Program yang ada pada *Flash* PEROM akan dijalankan jika pada sistem di-reset, pin EA berlogika high sehingga mikrokontroller aktif berdasarkan program pada Flash PEROM.

Mikrokontroller AT89C51 memiliki *oscilator internal* bagi sumber *clock* CPU. Pada penggunaan *osilator internal* diperlukan kristal antara pin XTAL 1 dan pin XTAL 2, dan dua buah kapasitor yang diperlihatkan pada **Gambar 2.6** dibawah ini. Kristal yang digunakan memiliki frekuensi dari 6 sampai 12 MHZ, sedangkan untuk kapasitor dapat bernilai 27 pF sampai 33 pF.




**Gambar 2.6** *Rangkaian Oscilator pada AT89C51*

## 2.2 Bahasa *Assembler*

Secara fisik, kerja dari sebuah mikrokontroler dapat dijelaskan sebagai siklus pembacaan instruksi yang tersimpan di dalam memori. Mikrokontroller menentukan alamat dari memori program yang akan dibaca, dan melakukan proses baca data dimemori. Data yang dibaca diinterprestasikan sebagai instruksi. Alamat instruksi disimpan oleh mikrokontroller di *register*, yang dikenal sebagai program *counter*. Instruksi ini misalnya program aritmatika yang melibatkan 2 *register*. Sarana yang ada dalam program *assembler* sangat minim, tidak seperti dalam bahasa pemrograman tingkat atas (*high level language programming*) semuanya sudah siap pakai. Penulis program *assembler* harus menentukan

segalanya, menentukan letak program yang ditulisnya dalam memori-program, membuat data konstan dan tabel konstan dalam memori-program, membuat *variabel* yang dipakai kerja dalam memori-data dan lain sebagainya.

### 2.2.1 Program Sumber *Assembler*

Program - sumber assembler (*assembler source program*) merupakan kumpulan dari baris-baris perintah yang ditulis dengan program penyunting - teks (*text editor*) sederhana, misalnya program EDIT. COM dalam DOS, atau program NOTEPAD dalam Windows atau MIDE-51. Kumpulan baris - perintah tersebut biasanya disimpan ke dalam file dengan nama ekstensi *.ASM dan lain sebagainya, tergantung pada program *assembler* yang akan dipakai untuk mengolah program-sumber *assembler* tersebut.

Setiap baris-perintah merupakan sebuah perintah yang utuh, artinya sebuah perintah tidak mungkin dipecah menjadi lebih dari satu baris. Satu baris perintah bisa terdiri atas 4 bagian, bagian pertama dikenali sebagai label atau sering juga disebut sebagai simbol, bagian kedua dikenali sebagai kode operasi, bagian ketiga adalah *operand* dan bagian terakhir adalah komentar antara bagian-bagian tersebut dipisahkan dengan sebuah spasi atau *tabulator*.

### 2.2.2 Bagian label

Label dipakai untuk memberi nama pada sebuah baris-perintah, agar bisa mudah menyebitnya dalam penulisan program. Label bisa ditulis apa saja asalkan diawali dengan huruf, biasa panjangnya tidak lebih dari 16 huruf. Huruf-huruf

berikutnya boleh merupakan angka atau tanda titik dan tanda garis bawah. Kalau sebuah baris-perintah tidak memiliki bagian label, maka bagian ini boleh tidak ditulis namun spasi atau *tabulator* sebagai pemisah antara label dan bagian berikutnya mutlak tetap harus ditulis. Dalam sebuah program sumber bisa terdapat banyak sekali label, tapi tidak boleh ada label yang kembar.

Sering sebuah baris-perintah hanya terdiri dari bagian label saja, baris demikian itu memang tidak bisa dikatakan sebagai baris-perintah yang sesungguhnya, tapi hanya sekedar memberi nama pada baris bersangkutan. Bagian label sering disebut juga sebagai bagian simbol, hal ini terjadi kalau label tersebut tidak dipakai untuk menandai bagian program, melainkan dipakai untuk menandai bagian data.

### 2.2.3 Bagian kode operasi

Kode operasi (*operation code* atau sering disingkat sebagai *OpCode*) merupakan bagian perintah yang harus dikerjakan. Dalam hal ini dikenal dua macam kode operasi, yang pertama adalah kode-operasi untuk mengatur kerja mikroprosesor/mikrokontroller. Jenis kedua dipakai untuk mengatur kerja program *assembler*, sering dinamakan sebagai *assembler directive*. Kode-operasi ditulis dalam bentuk *mnemonic,* yakni bentuk singkatan-singkatan yang relatip mudah diingat, misalnya adalah MOV, ACALL, RET dan lain sebagainya. Kode-operasi ini ditentukan oleh pabrik pembuat mikroprosesor/mikrokontroler, dengan demikian setiap *prosesor* mempunyai kode-operasi yang berlainan.

Kode–operasi berbentuk *mnemonic* tidak dikenal mikroprosesor/ mikrokontroller, agar program yang ditulis dengan kode *mnemonic* bisa dipakai untuk mengendalikan prosesor, program semacam itu diterjemahkan menjadi program yang dibentuk dari kode-operasi *kode-biner*, yang dikenali oleh mikroprosesor/mikrokontroler. Tugas penerjemahan tersebut dilakukan oleh program yang dinamakan sebagai Program *Assembler*. Di luar kode-operasi yang ditentukan pabrik pembuat mikroprosesor/mikrokontroller, ada pula kode-operasi untuk mengatur kerja dari program *assembler,* misalnya dipakai untuk menentukan letak program dalam memori (ORG), dipakai untuk membentuk *variabel* (DS), membentuk tabel dan data konstan (DB, DW) dan lain sebagainya.

### 2.2.4 Bagian Operand

*Operand* merupakan pelengkap bagian kode operasi, namun tidak semua kode operasi memerlukan operand, dengan demikian bisa terjadi sebuah baris perintah hanya terdiri dari kode operasi tanpa *operand*. Sebaliknya ada pula kode operasi yang perlu lebih dari satu *operand*, dalam hal ini antara *operand* satu dengan yang lain dipisahkan dengan tanda koma.

Bentuk operand sangat bervariasi, bisa berupa kode-kode yang dipakai untuk menyatakan *Register* dalam *prosesor*, bisa berupa nomor-memori (alamat memori) yang dinyatakan dengan bilangan atau pun nama label, bisa berupa data yang siap di-operasi-kan. Semuanya disesuaikan dengan keperluan dari kode-operasi. Untuk membedakan *operand* yang berupa nomor-memori atau *operand*

yang berupa data yang siap di-operasi-kan, dipakai tanda-tanda khusus atau cara penulisan yang berlainan.

Di samping itu *operand* bisa berupa persamaan matematis sederhana atau persamaan *Boolean*, dalam hal semacam ini program *assembler* akan menghitung nilai dari persamaan-persamaan dalam *operand*, selanjutnya merubah hasil perhitungan tersebut ke kode *biner* yang dimengerti oleh prosesor. Jadi perhitungan di dalam *operand* dilakukan oleh program *assembler* bukan oleh *prosesor.*

### 2.2.5 Bagian komentar

Bagian komentar merupakan catatan-catatan penulis program, bagian ini meskipun tidak mutlak diperlukan tapi sangat membantu masalah dokumentasi. Membaca komentar-komentar pada setiap baris-perintah, dengan mudah bisa dimengerti maksud tujuan baris bersangkutan, hal ini sangat membantu orang lain yang membaca program. Pemisah bagian komentar dengan bagian sebelumnya adalah tanda spasi atau *tabulator*, meskipun demikian huruf pertama dari komentar sering-sering berupa tanda titik-koma, merupakan tanda pemisah khusus untuk komentar.

Untuk keperluan dokumentasi yang intensip, sering-sering sebuah baris yang merupakan komentar saja, dalam hal ini huruf pertama dari baris bersangkutan adalah tanda titik-koma. AT89C51 memiliki sekumpulan instruksi yang sangat lengkap.

Instruksi MOV untuk *byte* dikelompokan sesuai dengan mode pengalamatan (*addressing modes*). Mode pengalamatan menjelaskan bagaimana *operand* dioperasikan. Berikut penjelasan dari berbagai mode pengalamatan. Bentuk program *assembler* yang umum ialah sebagai berikut :

**Tabel 2.2** *Mode berbagai Pengalamatan Program Assembler.*

| **Label / Simbol** | **Opcode** | **Operand** | **Komentar** |
|---|---|---|---|
| | Org | 0H | |
| Start:<br><br>Kiri:<br><br><br><br><br><br>Delay:<br>Del1:<br>Del2: | Mov<br>Mov<br>Mov<br>Call<br>RL<br>DEC<br>CJNE<br>Sjmp<br>mov<br>mov<br>djnz<br>djnz<br>ret<br>end | A, #11111110b<br>R0, #7<br>P0, A<br>Delay<br>A<br>R0<br>R0, #0, Kiri<br>Start<br>R1, #255<br>R2, #255<br>R2, del2<br>R1, del1 | ; Isi Akumulator<br>; Isi R0 dengan 7<br>; Copy A ke P0<br>; Panggil Delay |

Isi memori ialah bilangan heksadesimal yang dikenal oleh mikrokontroller kita, yang merupakan representasi dari bahasa assembler yang telah kita buat. Mnemonic atau *opcode* ialah kode yang akan melakukan aksi terhadap operand .

Operand ialah data yang diproses oleh *opcode*. Sebuah *opcode* bisa membutuhkan 1, 2 atau lebih operand, kadang juga tidak perlu *operand*. Sedangkan komentar dapat kita berikan dengan menggunakan tanda titik koma (;). Berikut contoh jumlah *operand* yang berbeda beda dalam suatu *assembler*.

```
CJNE R5,#22H, aksi ;dibutuhkan 3 buah operand
MOVX @DPTR, A ;dibutuhkan 2 buah operand
RL A ;1 buah operand
NOP ; tidak memerlukan operand
```

Program yang telah selesai kita buat dapat disimpan dengan ekstension asm. Lalu kita dapat membuat program objek dengan *ekstension* HEX dengan menggunakan *compiler* MIDE-51, yang dijelaskan sebagai berikut:

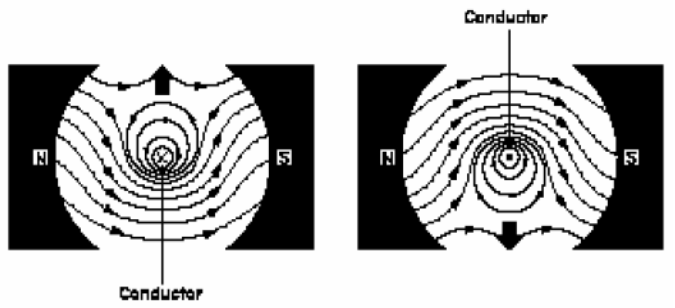
## 2.3 Motor DC

Pada prinsipnya mesin listrik dapat berlaku sabagai motor maupun sebagai generator, perbedaanya hanya terletak pada konversi energinya. *Generator* adalah suatu mesin listrik yang dapat mengubah energi mekanik menjadi energi lisrik, sebaliknya pada motor mengubah energi listrik menjadi energi mekanik. Maka dengan membalik *generator* arus searah, di mana sekarang tegangan *Vt* menjadi sumber dan tegangan jangkar *Ea* merupakan ggl, mesin arus searah ini akan berlaku sebagai motor.

### 2.3.1 Prinsip Kerja Motor DC

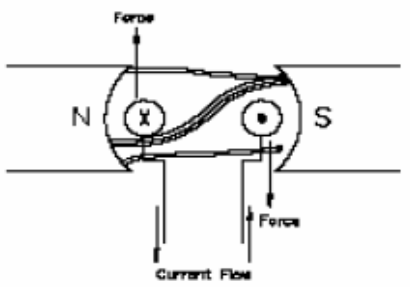
Ada dua kondisi yang dibutuhkan untuk menghasilkan gaya pada *konduktor*: Pertama *konduktor* harus mengalirkan arus, kemudian kedua *Konduktor* harus berada di dalam medan magnet. Ketika dua kondisi ini ada, maka *konduktor* akan menghasilkan gaya karena menggerakkan *konduktor* sesuai dengan arah medan magnet. Prinsip ini merupakan dasar operasi motor DC.

Setiap *konduktor* yang mengalirkan arus memiliki aliran medan magnet disekitarnya. Jika *konduktor* berarus ditempatkan pada medan magnet, akan terinduksi gaya yang menggerakkan *konduktor*. Hal ini digambarkan pada **Gambar 2.7**. Tanda silang (x) menunjukka arah arus adalah masuk, sementara tanda titik (.) menunjukkan arah arus keluar. Di atas *konduktor* sebelah kiri, medan magnet yang dihasilkan *konduktor* berlawanan dengan medan magnet utama. Sementara di bagian bawah, medan magnet *konduktor* searah dengan medan magnet utama. Maka medan di bawah *konduktor* lebih kuat dari medan di atas sehingga gaya terinduksi pada *konduktor* ke arah atas. Hal sebaliknya terjadi pada *konduktor* kanan, di mana gaya induksi yang terjadi adalah ke arah bawah.




**Gambar 2.7** *Induksi Gaya pada Konduktor*

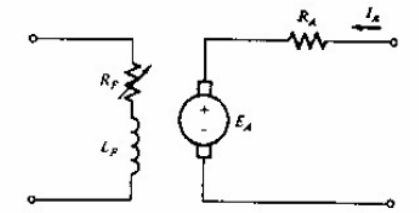
Pada motor DC, konduktor dibentuk sebagai *loop* sehingga dua bagian konduktor berada pada medan magnet pada saat yang bersamaan. Kombinasi efek dari kedua *konduktor* dan medan magnet utama mengahasilkan gaya pada masing-masing bagian *konduktor*. Ketika *konduktor* ditempatkan pada rotor (bagian yang bergerak) gaya yang dihasilkan akan menyebabkan rotor berputar sebagaimana digambarkan pada **Gambar 2.8** berikut:




**Gambar 2.8** *Gaya pada Konduktor*

### 2.3.2 Rangkaian *Ekuivalen* Motor DC

Rangkaian *ekuivalen* motor DC adalah sama dengan rangkaian *ekuivalen* generator DC kecuali pada arah aliran arusnya. Pada motor, arus mengalir masuk ke rangkaian jangkar. Rangkaian *ekuivalen* motor DC diperlihatkan pada **Gambar 2.9**.




**Gambar 2.9** *Rangkaian Ekuivalen Motor DC*

Ketika motor berputar maka ia juga bertindak sebagai generator. *Konduktor* yang bergerak memotong garis medan magnet akan menginduksi GGL pada *konduktor* yang berlawanan dengan sumber. GGL induksi ini disebut sebagai tegangan jangkar (Ea). Besar tegangan induksi Ea dirumuskan sebagai:

$$E = K\,n\mathrm{F}a \quad \text{.....(2.1)}$$

Dimana

K = konstanta

n = kecepatan putar jangkar

F = flux medan

a = jangkar

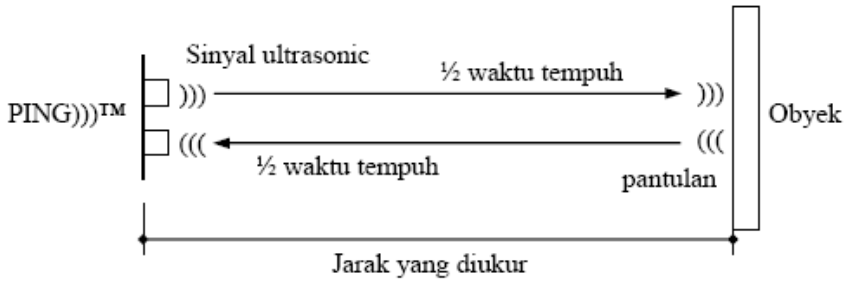
Sementara torsi induksi merupakan gaya yang mengasilkan dan mempertahankan rotasi. Besar torsi induksi T *ind* pada mesin diberikan oleh persamaan:

$$T_{ind\ a} = kFI \quad \text{.....(2.2)}$$

dengan Ia adalah arus jangkar.

## 2.4 Sensor Ultrasonik

Sensor ultrasonik merupakan sensor yang dapat mengukur jarak, pengukuran dilakukan dengan mengirimkan gelombang ultrasonik dengan *frekuensi* 40 Khz dan kecepatan 344 m/s kemudian akan menerima pantulan, lalu mengirim logic.



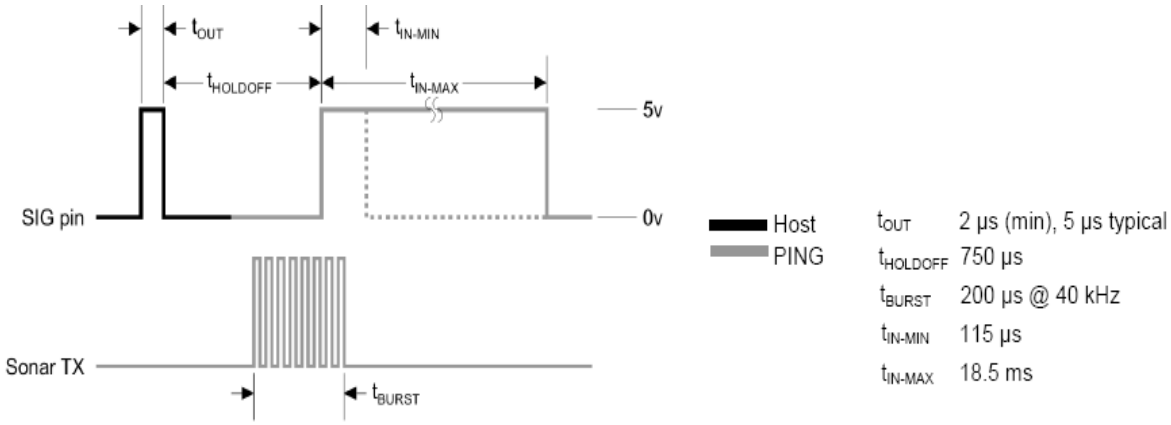

**Gambar 2.10** *Ilustrasi Cara Kerja Sensor Ultrasonik*

Ilustrasi cara kerja sensor ultrasonik ditunjukkan pada **Gambar 2.10**
Sesuai rumus fisika

$$S = \frac{V}{2} x \ \text{t} \quad \text{.....(2.3)}$$

dengan $S$ = *Jarak yang terukur (meter)*
$V$ = *Kecepatan rambat suara (m/s)*
$t$ = *waktu tempuh (s)*

Namun waktu yang dihitung adalah waktu pergi dan waktu datang sehingga jarak yang ditempuh adalah dua kali. Dari gambar diatas PING tersebut hanya memiliki tiga buah kaki, VCC, GROUND, dan SIGNAL. SIGNAL ini berfungsi sebagai pengatur serta penerima sensor yang akan ditangkap oleh PIR. Mengirim sinyal kasih tegangan kotak 0 1 0 menerima sinyal pantul maka ping akan jadi 0 untuk lebih jelasnya dapat dilihat **Gambar 2.11** berikut ini.




**Gambar 2.11** *Diagram Waktu Sensor*

Dari grafik diatas kita dapat melihat prinsip kerjanya. Bisa dilihat gambar grafik SIGNAL *Host* (mcs) yang sinyal lebih tebal.

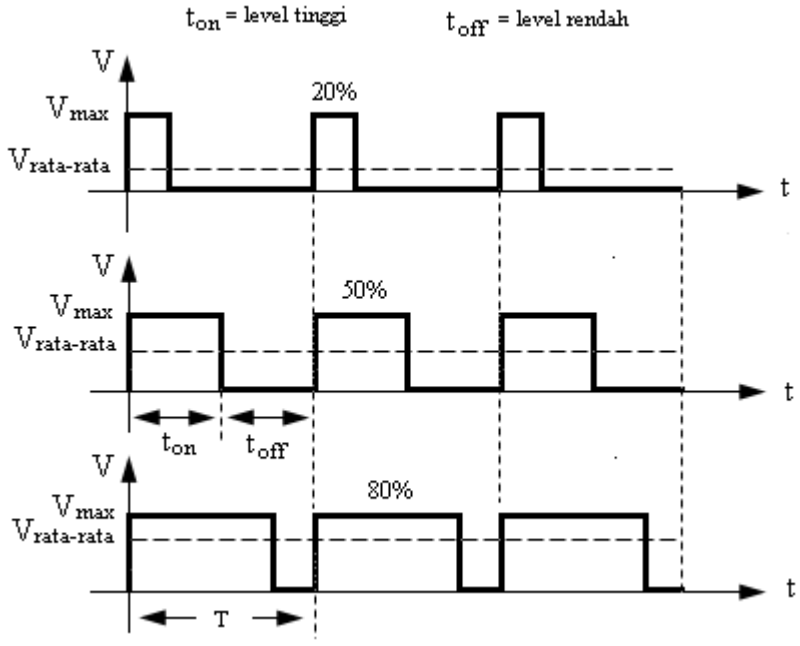
## 2.5 Pulse Width Modulation

Metode *Pulse Width Modulation* (PWM) adalah metode yang cukup efektif untuk mengendalikan kecepatan motor DC. PWM ini bekerja dengan membuat gelombang persegi yang memiliki *frekuensi* tetap, rasio (*duty cycle*) pulsa tinggi terhadap pulsa rendah yang telah ditentukan, yang bisa diatur dari 0% hingga 100%. Rasio pulsa tinggi terhadap pulsa rendah digunakan untuk mengatur penyambungan (*switch*) *suplay* tegangan terminal motor DC. Semakin besar lebar pulsa tinggi dari pada pulsa rendah berarti penyambungan suplay tegangan terminal motor mendapat porsi waktu yang lebih lama, menyebabkan kecepatan motor semakin besar.

### 2.5.1 Modulasi lebar pulsa (*Pulse Width Modulation*)

Modulasi lebar pulsa (*Pulse Width Modulation*) merupakan suatu teknik untuk mengendalikan tegangan input rangkaian *analog* dengan memanfaatkan sinyal *output* digital (Pulsa-pulsa tegangan digital). Pengaturan tegangan output dapat dilakukan dengan mengubah-ubah nilai *duty cycle*, dimana *duty cycle* merupakan *prosentase* lebar pulsa pada kondisi aktif (level tinggi) per *periode*.

Besar tegangan yang dihasilkan merupakan tegangan rata-rata, hasil perkalian nilai *duty cycle* dengan tegangan maksimum sumber. **Gambar 2.12** menunjukkan contoh pulsa PWM dengan 3 variasi nilai *duty cycle* yaitu 20%, 50% dan 80%.




**Gambar 2.12** *Sinyal PWM dengan Variasi Duty Cycle*

Tegangan rata-rata yang diperoleh dari pulsa PWM seperti gambar di atas dapat dihitung dengan persamaan di bawah ini :

$$Vrata-rata = \frac{Ton}{T} . V_{maks} \quad \text{.....(2.4)}$$

Karena

$$dc = \frac{Ton}{T}.100\% \quad \text{.....(2.5)}$$

Maka :

$$V_{rata-rata} = dc \cdot V_{maks} \quad \text{.....(2.6)}$$

Dengan :

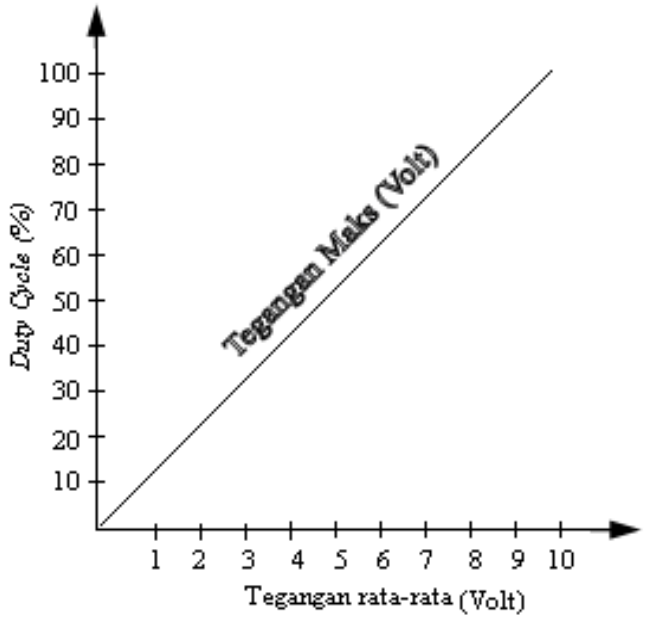
$V$ = tegangan

$T$ = Perioda

$t_{on}$ = waktu aktif (detik)

$dc$ = duty cycle ( % )

dari persamaan **(2.6)** diketahui bahwa tegangan *maksimum linier* terhadap perbandingan nilai *duty cycle* dan tegangan rata – rata, grafik perbandingan nilai *duty cycle* dan tegangan rata – rata ditunjukkan pada **Gambar 2.13** di bawah ini :



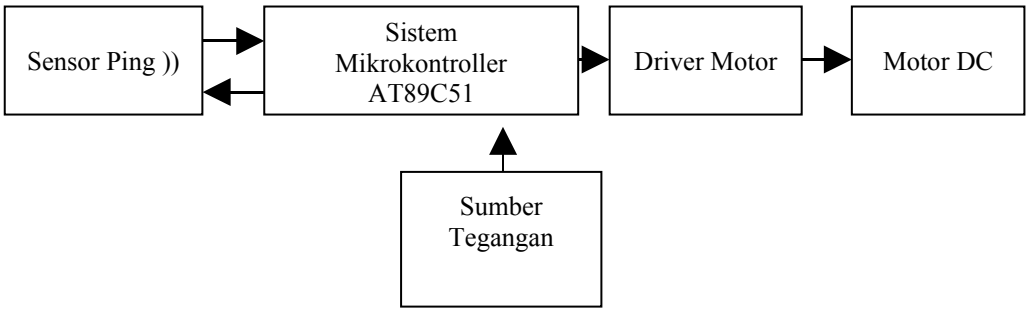

**Gambar 2.13** *Hubungan Nilai Duty Cycle dengan Tegangan Rata-rata*

Jika pulsa PWM digunakan untuk mengaktifkan rangkaian *driver* motor DC, perubahan lebar pulsa PWM menyebabkan terjadinya perubahan tegangan input motor DC. Karena perubahan motor DC dapat dikendalikan dengan cara mengatur nilai *duty cycle* (lebar pulsa PWM). *Frekuensi* yang digunakan pada pulsa PWM merupakan *frekuensi* tinggi konstan dengan nilai 20 Khz atau lebih.

# BAB III

## PERANCANGAN DAN IMPLEMENTASI SISTEM

Perancangan sistem adalah tahap untuk merancang suatu sistem dengan menggabungkan masing-masing blok sehingga tercipta suatu sistem yang lengkap. Perancangan sistem ini terdiri atas 2 (dua) bagian, yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*). **Gambar 3.1** di bawah menunjukkan blok diagram dari sistem yang dirancang.



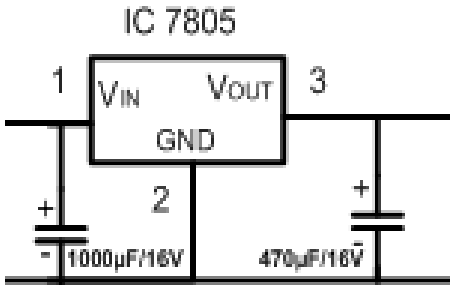

**Gambar 3.1** *Blok Diagram Sistem*

### 3.1 Perancangan *Hardware*

Perancangan *hardware* ini adalah tahap merancang dan membuat rangkaian dari masing-masing blok diagram. Kemudian dari setiap blok tersebut digabungkan menjadi satu, sehingga menjadi suatu kesatuan yang utuh.

### 3.1.1 Rangkaian Penurun Tegangan 12 Volt menjadi 5 Volt

Rangkaian ini berfungsi sebagai penurun tegangan 12 Volt menjadi 5 Volt yang merupakan sumber tegangan (VCC) yang dibutuhkan rangkaian mikrokontrller, motor DC, Driver motor DC dan sensor Ultrasonik PING.




**Gambar 3.2** *Skema Rangkaian Penurun Tegangan 12 Volt menjadi 5 Volt*

Prinsip kerja rangkaian penurun tegangan 12 Volt menjadi 5 Volt, adalah.

1. Tegangan dari suplay 12 Votl merupakan masukan bagi rangkaian ini.
2. IC AN7805 merupakan komponen yang karakteristiknya mampu mengubah nilai masukan sebesar 12 Volt menjadi 5 Volt.
3. Nilai tegangan keluaran 5 Volt tersebut dihubungkan pada tegangan masukan (Vcc) Rangkaian mikrokontroller, *driver* motor, motor DC dan sensor ultrasonik PING.

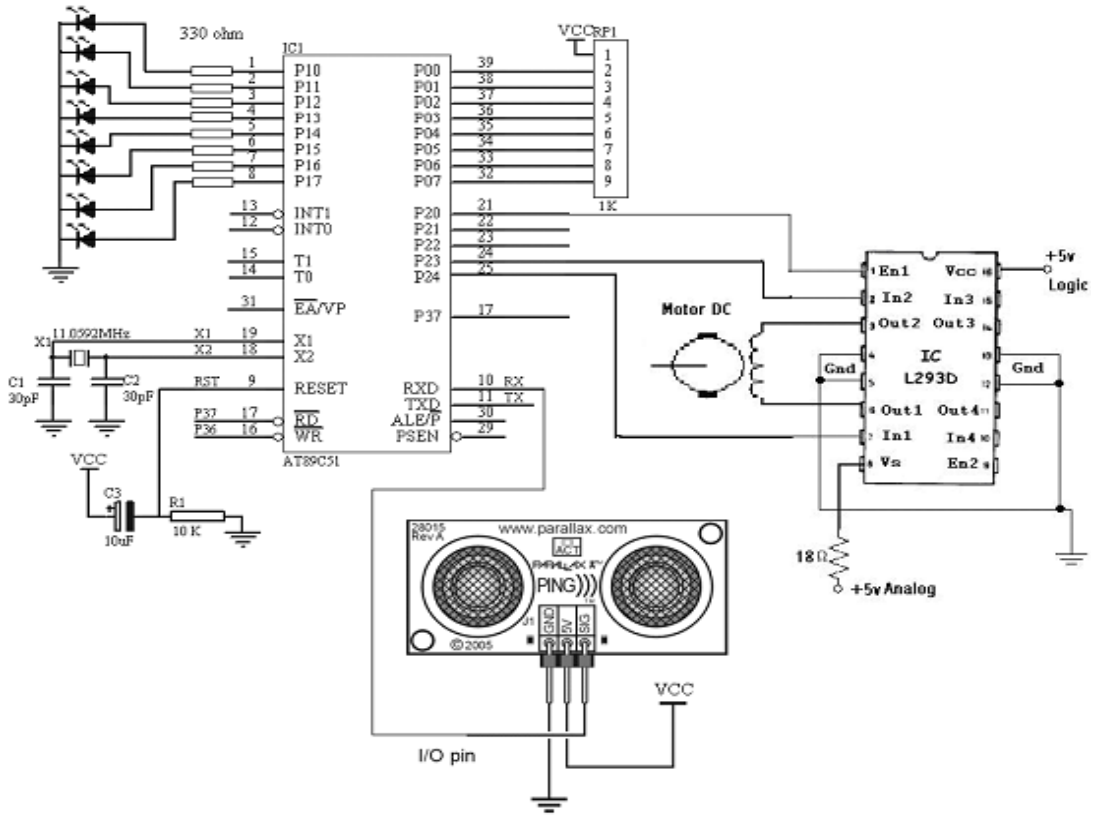
### 3.1.2 Input Referensi (penambahan atau pengurangan kecepatan motor)

Penambahan atau pengurangan kecepatan motor dirancang serta direalisasikan dengan memanfaatkan tegangan VCC 5 Volt, yang dioprasikan dengan *algoritma* yang telah ditentukan pada perangkat lunak mikrokontroler dengan keluaran P2.0 pengatur kecepatan motor.

### 3.1.3 Mikrokontroller AT89C51

Rangkaian ini merupakan rangkaian pengendali dimana digunakan IC mikrokontroller AT89C51. Mikrokontroler ini mempunyai 4 buah port yang difungsikan untuk berbagai hal, seperti terlihat pada **Gambar 3.3** Pin-pin yang digunakan adalah sebagai berikut.

1. Port 0 masing masing digunakan untuk pin control.
2. Port 1 masing - masing digunakan untuk LED *Indikator Channel.*
3. Port 2 pin 0, 4, 5 digunakan untuk mengaktifkan *driver* dari motor DC.
4. Port 3.0 digunakan untuk mengirimkan sinyal I/O SIGNAL pada PING.




**Gambar 3.3** *Rangkaian Mikrokontroller AT89C51*

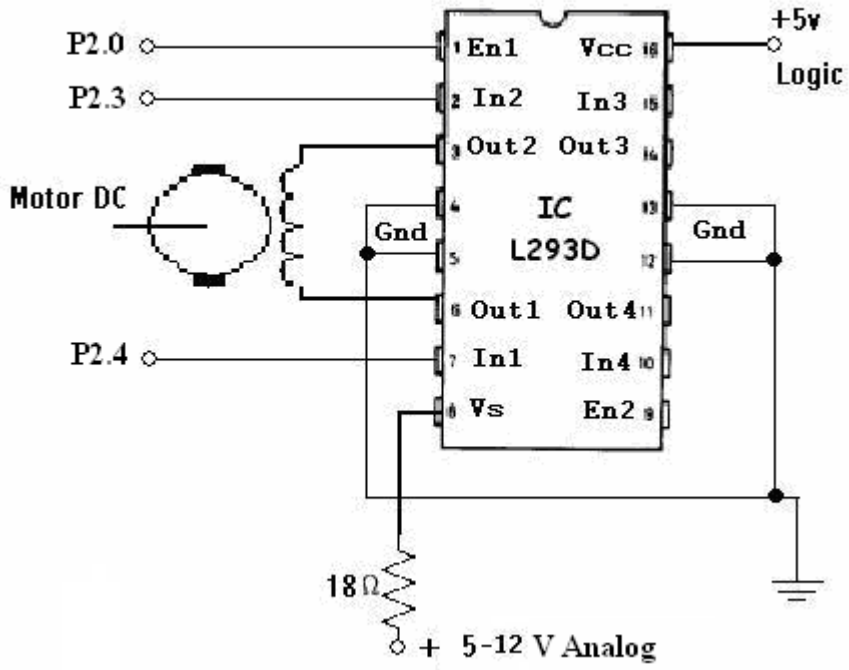
Agar mikrokontroler dapat bekerja maka dibutuhkan suatu rangkaian *osilator* sebagai sumber *clock*. Dalam hal ini digunakan *osilator internal* yang sudah ada dalam mikrokontroler AT89C51. Untuk menggunakannya dilakukan dengan menghubungkan sebuah *resonator kristal* diantara kaki-kaki XTAL1 dan XTAL2. Kristal yang digunakan adalah 11.0592 MHz.

C1 dan C2 digunakan untuk menjaga kestabilan *clock* dan merupakan saran atau rekomendasi dari pabrik pembuat ATMEL. Prinsip kerja dari mikrokontroler ini sesuai dengan program yang dibuat. Penjelasan program dijelaskan bersama dengan diagram alir atau *flowchart* yang dibuat pada bab ini dan listing programnya dapat dilihat pada lampiran.

Agar mikrokontroler dapat mengeksekusi program dari awal program (alamat 00H) maka mikrokontroler perlu direset secara otomatis saat catu daya pertama kali dihidupkan dimana untuk reset otomatis ini dilakukan oleh C3 dan R1 (*Power* On *Reset*). Prinsip kerja dari reset otomatis ini adalah proses pengisian dan pengosongan C3 dimana pin reset membutuhkan logika *high*. Pada saat catu daya dihidupkan maka C3 mulai diisi sementara pada pin *reset* belum ada tegangan. Setelah C3 penuh maka tegangan dari C3 akan menyulut pin *reset high* sehingga terjadi *reset*. Pada saat catu daya dimatikan maka akan berlangsung pengosongan C3 melalui R1 sehingga saat catu daya dihidupkan kembali maka akan terjadi lagi proses pengisian sehingga terjadi *reset* kembali.

### 3.1.4 Driver Motor DC

Driver motor DC ini, merupakan rangkaian penguat untuk pergerakan motor dengan penyaklaran kecepatan tinggi sampai dengan 5 KHz. Rangkaian *driver* hanya terdiri dari sebuah IC yaitu L293D yang praktis dan umum digunakan untuk motor berukuran kecil. Arus maksimal yang mampu dihasilkan oleh L293D sebesar 600 mA. Penguat ini mampu melakukan fungsi pensaklaran kecepatan tinggi sampai dengan 5 KHz. Kaki masukan 1 merupakan jalur sinyal PWM sebagai pensaklaran tegangan 1 motor 12 V. Masukan 2 merupakan jalur 0 V. Masing-masing jalur menghasilkan tegangan keluaran yang akan diterapkan ke kutub-kutub motor DC ditunjukkan **Gambar 3.4** berikut.




**Gambar 3.4** *Driver Motor DC*

Masing-masing jalur menghasilkan tegangan keluaran yang akan diterapkan ke kutub-kutub motor DC dengan berpedoman pada tabel kebenaran L293D ditunjukkan pada **Tabel 3.1** sebagai berikut.

**Tabel 3.1** *Kebenaran untuk satu kanal.*

| Masukan | Enable | Keluaran |
|---|---|---|
| H | H | H |
| L | H | L |
| H | L | Z |
| L | L | Z |

Z = Impedansi keluaran tinggi

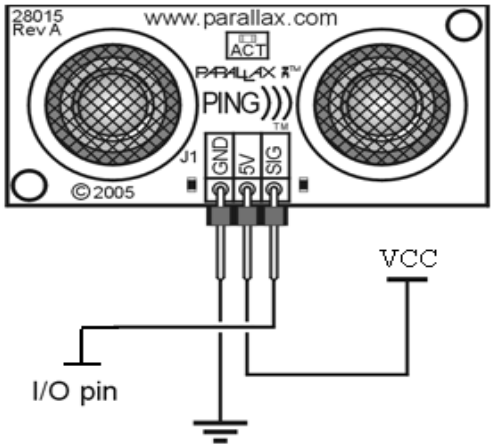
### 3.1.5 Sensor Ultrasonik (PING)

Sensor PING digunakan untuk mengetahui jarak berdasarkan *data sheet*, modul PING ini dapat digunakan untuk mengukur jarak benda sejauh 3 cm sampai 300 cm. Sensor PING mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 kHz) selama tBURST (200 µs) kemudian mendeteksi pantulannya.

Sensor PING memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroler pengendali (pulsa *trigger* dengan tOUT min. 2 µs). Gelombang ultrasonik ini melalui udara dengan kecepatan 344 m/detik, mengenai obyek dan memantul kembali ke sensor.

PING mengeluarkan pulsa *output high* pada pin SIG setelah memancarkan gelombang ultrasonik dan setelah gelombang pantulan terdeteksi PING akan

membuat *output low* pada pin SIG. Lebar pulsa *High* (tIN) akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk 2x jarak ukur dengan obyek. Maka jarak yang diukur adalah [(tIN s x 344 m/s) ÷ 2] meter. Untuk simulasi agar sensor ultrasonic (PING) bekerja maka dibuat *listing* program sederhana. Bisa dilihat bentuk fisik sensor ultrasonik (PING) yang ditunjukkan pada **Gambar 3.5** sebagai berikut.




**Gambar 3.5** *Bentuk dari Sensor Ultrasonik (PING)*

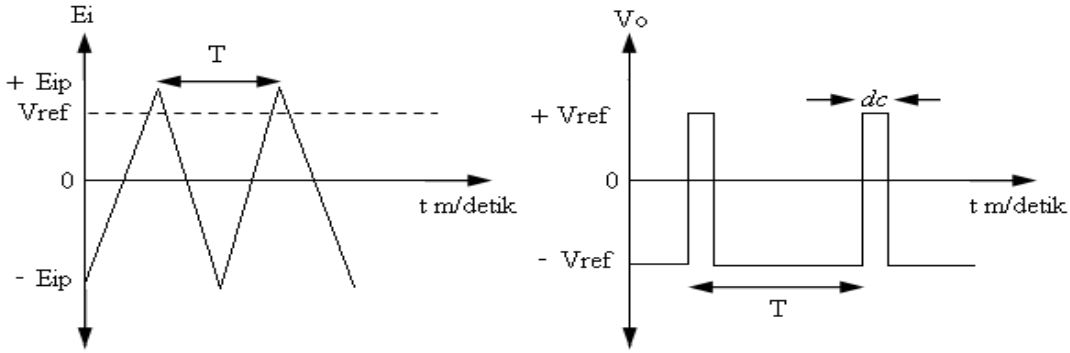
### 3.1.6 Pengaturan Duty Cycle (%)

Metode pengaturan kecepatan motor ini memberikan kemudahan pengendali yang berbasis digital (misal : mikrokontroller), karena tidak banyak memerlukan rangkaian kendali, dan kecepatan yang dihasilkan tidak mengurangi besar torsi.

Sebagai contoh Ei merupakan gelombang pembawa segitiga yang amplitude dan frekuensinya tetap seperti dalam **Gambar 3.6**. karena frekuensinya tetap, *periode* T dari gelombang itu juga tetap (T=1/f). Ei diberikan kemasukan

(+) dan mempunyai harga puncak Eip. Vref diberikan ke masukan (-) dan dianggap sebagai isyarat datanya, Vo akan menunjukkan *duty cycle* dari hasil persamaan (**3.1)** sebagai berikut :

$$dc = \frac{(Eip - Eref) \cdot 100\%}{2Eip} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.1)$$




(a) (b)

(a) Gelombang segitiga dengan Vref
(b) Pulsa PWM

**Gambar 3.6** *Proses Pembentukan Sinyal PWM*

## 3.2 Perancangan dan Realisasi Perangkat Lunak

Perangkat lunak adalah sekumpulan intruksi-intruksi yang digunakan sebagai sistem operasi yang mengontrol perangkat keras dalam memberikan input dan output data serta pertukaran informasi. Jadi perangkat lunak direalisasikan untuk mendukung kerja perangkat keras dalam melakukan tugasnya. Perangkat lunak yang dipakai dalam sistem ini memiliki batasan berdasarkan kriteria perangkat keras yang telah ditentukan maupun teori yang dipakai.

### 3.2.1 Spesifikasi Perangkat Lunak

Spesifikasi dari perangkat lunak yang akan direalisasikan adalah sebagai berikut.

1. Perangkat lunak digunakan untuk memberi intruksi pada unit kontrol penerimaan sinyal pantul sensor ultrasonik (PING).
2. Perangkat lunak memberikan intruksi pada unit kontrol untuk mengatur kecepatan motor DC dengan PWM.
3. Pemrosesan *algoritma* ini dilakukan dengan menggunakan intruksi-intruksi mikrokontroller MCS – 51.

### 3.2.2 Program Utama

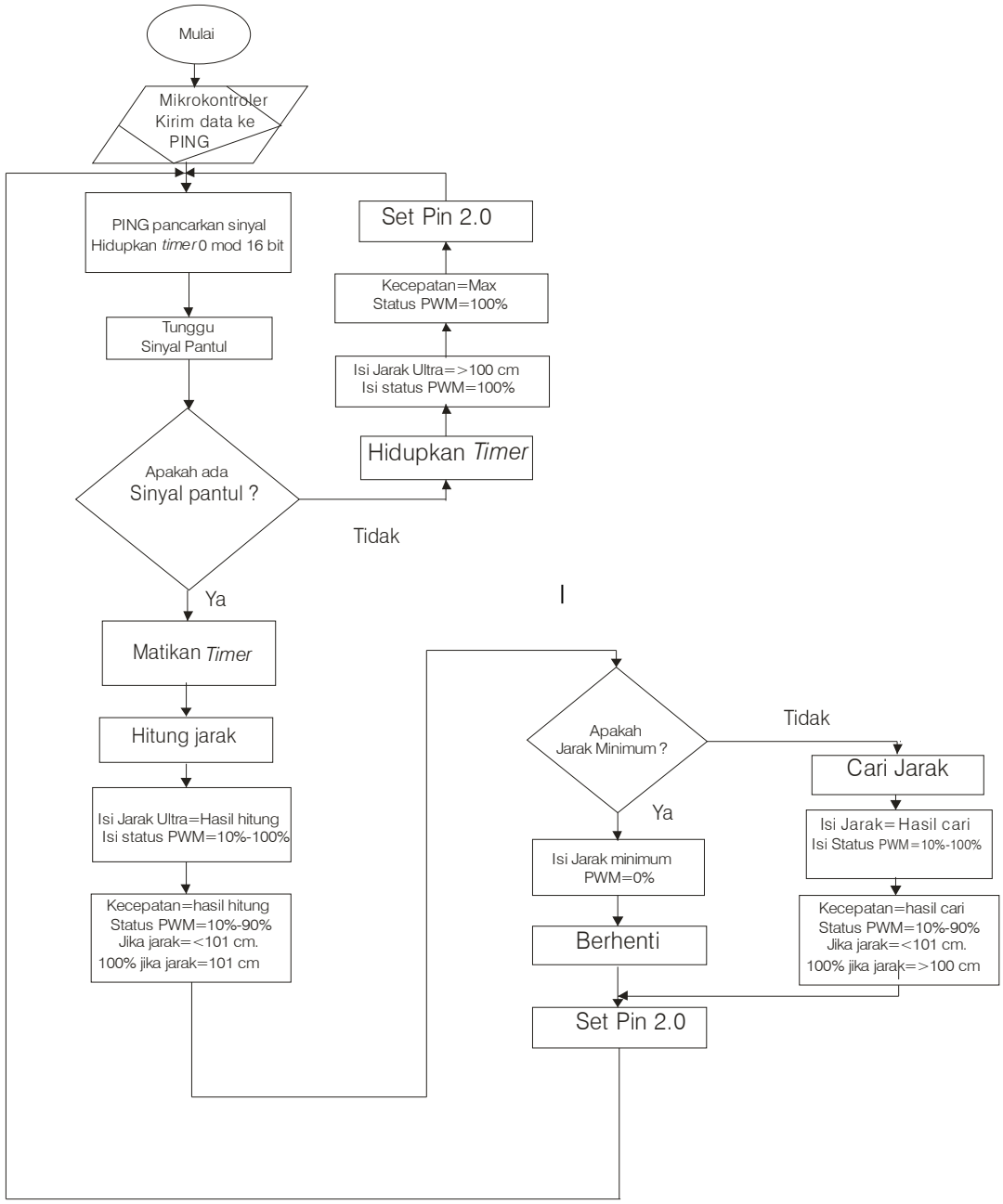
Perangkat lunak menggunakan mikrokontroller AT89C51 yang termasuk dalam keluarga MCS-51 dilakukan dengan cara membuat program dengan bahasa *assembler* yang merupakan suatu bahasa mesin yang dapat dimengerti oleh manusia. Setiap pernyataan dari bahasa *assembler* menggambarkan pernyataan suatu bahasa mesin.

Perangkat lunak yang digunakan memiliki batasan dan kriteria berdasarkan perangkat kerasnya masing-masing. Begitupun perangkat lunak yang digunakan pada sistem ini.

1. Perangkat lunak yang digunakan menggunakan bahasa *assembler* MCS-51.
2. File-file pada program dibuat dengan *ektension* *.*asm* pada program aplikasi *notepad* atau *teks editor.*

3. Perangkat lunak yang dibuat dengan bahasa *assembler* kemudian di-*compile* ke dalam bahasa mesin (*Heksa* atau *Biner*) agar dapat dibaca oleh mikrokontroller dengan menggunakan program aplikasi *crose assembler* ASM51.

Bisa dilihat **Gambar 3.7** menunjukkan diagram alir (*flow chart*) sistem.




**Gambar 3.7** *Diagram Alir atau Flowchart*

# BAB IV

# PENGUJIAN ALAT DAN ANALISA

Setelah perencanaan dan pembuatan sistem, maka langkah selanjutnya yaitu melakukan pengujian alat dan analisa terhadap perangkat keras yang telah

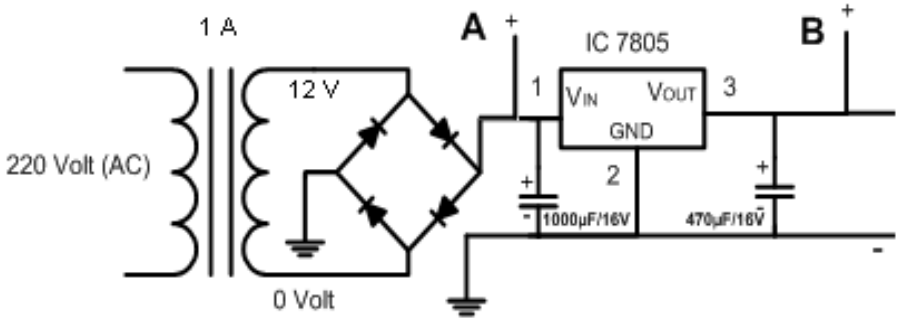
dibuat dan perangkat lunak yang telah dimasukan dalam mikrokontroler AT89C51. pengujian perangkat keras terdiri dari. pengujian rangkaian catu daya, pengujian port Mikrokontroler, pengujian rangkaian driver motor DC dan pengujian sensor ping.

## 4.1 Pengujian dan Analisa Perangkat Keras

Pengujian perangkat keras dilakukan dengan cara pengecekan dan pengukuran jalur rangkaian serta menguji komponen penunjang secara keseluruhan. Pengujian ini dilakukan untuk mengetahui peralatan yang ada pada perangkat keras yang dibuat (baik buruknya kondisi alat dan kinerjanya).

### 4.1.1 Pengujian dan Analisa Rangkaian Catu Daya

Pengujian catu daya dilakukan dengan menghubungkan *output* dari rangkaian catu daya dengan *Voltmeter*, Pengukuran dilakukan dengan menggunakan *Voltmeter* digital untuk memudahkan pembacaan hasil. Hasil pengukuran dapat dilihat pada **Tabel 4.1** sesuai dengan titik-titik rangkaian catu daya pada **Gambar 4.1**




**Gambar 4.1** *Titik-Titik Pengukuran Tegangan*

**Tabel 4.1** *Nilai Tegangan Tiap Titik*

| Titik | Keluaran | Teori | Nilai Pengukuran |
|---|---|---|---|
| **A** | DC | 12 Volt | 11.99 Volt |
| **B** | DC | 5 Volt | 4.99 Volt |

Dari hasil pengukuran catu daya seperti pada **Tabel 4.1**, dapat dianalisa tingkat kesetabilan IC dengan perhitungan.

$$\%Error = \frac{(Teori - Pengukuran)}{Teori} x100\%$$

$$\%Error = \frac{(12 - 11.99)}{12} x100\%$$

$$= 0.083\%$$

$$\%Error = \frac{(5 - 4.99)}{5} x100\%$$

$$= 0.2\%$$

Sangat kecilnya tingkat *error* dari hasil pengukuran dan perhitungan yang dilakukan pada rangkaian *power supply*, maka rangkaian sudah bekerja dengan baik seperti karakteristik IC *regulator* yang dipakai. Untuk IC7805 tegangan keluarnya antara 4,8 – 5,2 Volt.

### 4.1.2 Pengujian dan Analisa Port Mikrokontroler

Pengujian *port* Mikrokontroler dimaksudkan untuk mengecek apakah data yang dimasukkan *(input)* dan dikeluarkan *(output)* Mikrokontroler sesuai dengan deskripsi kerja sistem.

Untuk simulasi awal pengecekan *I/O* menggunakan simulasi nyala LED dengan menggunakan program sederhana menyalakan LED di *port* 1. Berikut ini merupakan listing program menyalakan LED di port 1.

```
---------------------------------------------------------------------------------
                Org 00H                     ; awal program
Mulai:          Mov a, #00000001b           ; isi a dengan 1
                Mov R0, #8                  ; jumlah pergeseran
Loop:           Mov p1, a                   ; keluarkan isi a ke p1
                Acall delay                 ; pangil delay
                RL,a                        ; isi a diputar kekiri
                DJNZ R0, loop               ; apakah sudah 8x
                Sjmp mulai                  ; jika ya kembali ke mulai
Delay:          Mov r3, #20
                Mov TMOD, a 1H
Loopt:          Mov TH0, #hightn(-50000)
                Mov TLO, #low(-50000)
                SETB TR0
                JNB TF0, #
                CLR TRO
                CLR TFO
Djnz             R3,loopt
Ret
End
---------------------------------------------------------------------------------
```

Berdasarkan program di atas maka tampilan yang didapatkan pada nyala LED yaitu pada saat program pertama kali dijalankan maka LED akan menyala yaitu LED yang dihubungkan dengan P1.0 sampai dengan P1.7 kemudian setelah selang waktu yang telah ditentukan pada *delay* maka nyala LED akan berjalan dari awal hingga akhir kemudian diulang lagi mulai awal hingga akhir begitu seterusnya.

### 4.1.3 Pengujian dan Analisa Rangkaian Driver Motor

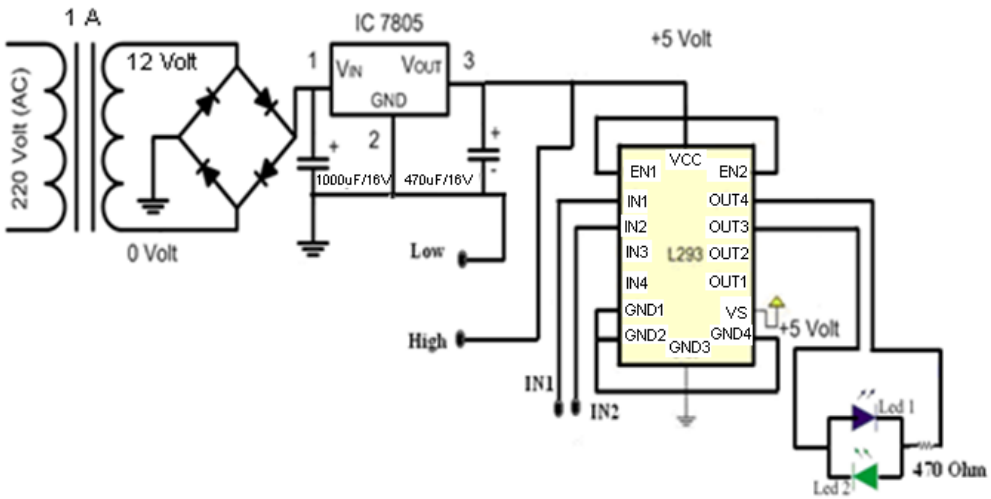
Setelah melakukan pengujian catu daya yang telah diterangkan di atas, disini akan menerangkan pengujian dan analisa rangkaian *driver* motor dilakukan

dengan dua pengujian yaitu pengujian rangkaian tanpa beban dan pengujian rangkaian dengan beban.

### 4.1.3.1 Pengujian Rangkaian Driver Motor Tanpa Beban

Pengujian rangkaian *driver* tanpa beban dilakukan untuk mengetahui apakah rangkaian *driver* yang dibuat sudah dapat bekerja dengan baik dan sesuai dengan yang diharapkan atau belum, seperti terlihat pada **Gambar 4.2**. Peralatan yang digunakan.

- Rangkaian *driver* dihubungkan dengan LED.
- *Power Supply*




**Gambar 4.2** *Pengujian Rangkaian Driver Tanpa Beban*

Langkah-langkah pengujian rangkaian *driver* adalah sebagai berikut.

1. Memastikan bahwa semua jalur telah tersambung dengan benar dan semua komponennya telah terpasang dengan sempurna.
2. Menyiapkan *power supply* dengan *output* tegangan 5 Volt sebagai pengganti *Output* dari Mikrokontroler AT89C51.

Ketika power supply yang dihubungkan dengan rangkaian driver memiliki keluaran 5 Volt maka LED yang merupakan *indikator* dari rangkaian akan menyala. Tetapi beban dengan menggunakan LED sebagai *indikator* keluaran dari driver motor masih mati karena belum mempunyai inputan (IN1 dan IN2). Untuk inputan maka diperlukan tegangan 5 Volt (*High*) dan 0 Volt (*Low*), dengan adanya inputan maka data hasil percobaan dapat dilihat pada **Tabel 4.2**

**Tabel 4.2** *Hasil Pengujian Driver Tanpa Beban*

| **NO** | ***INPUT*** | | ***OUTPUT*** | |
|---|---|---|---|---|
| | **IN1** | **IN2** | **LED 1** | **LED 2** |
| 1 | High | Low | ON | OFF |
| 2 | Low | High | OFF | ON |
| 3 | High | High | OFF | OFF |
| 4 | Low | Low | OFF | OFF |

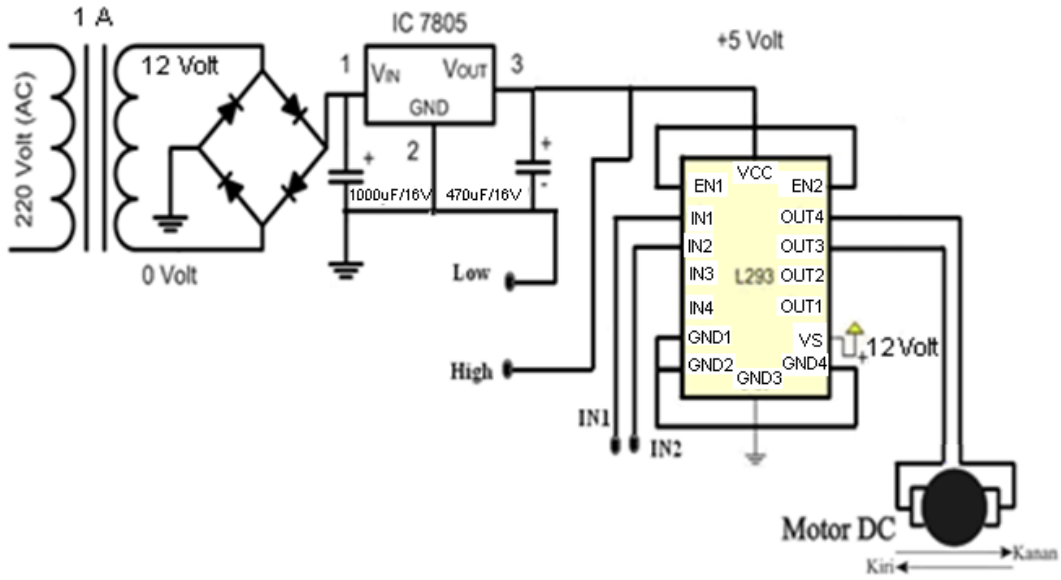
Berdasarkan tabel diatas maka dapat dianalisa bahwa ketika diberi *input High*, *Low* maka LED 1 ON dan led 2 OFF. Dan sebaliknya jika diberi *input* Low, *High* maka LED 2 ON dan LED 1 OFF. Tetapi jika inputan diberi nilai yang sama, maka LED 1 dan LED 2 OFF.

### 4.1.3.2 Pengujian Rangkaian Driver Motor Dengan Beban

Pengujian rangkaian *driver* dengan beban untuk mengetahui apakah rangkaian *driver* yang telah terhubung dengan beban dapat bekerja dengan baik atau belum, seperti terlihat pada **Gambar 4.3**. Peralatan yang digunakan.

1. Rangkaian *driver* yang telah dihubungkan dengan motor DC.

2. *Power Supply.*




**Gambar 4.3** *Pengujian Rangkaian Driver Dengan Motor*

Setelah semua langkah-langkah pada pengujian *driver* tanpa beban selesai dilaksanakan, selanjutnya dilakukan langkah-langkah sebagai berikut.

1. Memastikan bahwa semua jalur telah tersambung dengan benar dan semua komponennya telah terpasang dengan sempurna.
2. Menyiapkan *power supply* dengan *output* tegangan 5 Volt sebagai pengganti *output* dari Mikrokontroler AT89C51 dan 12 Volt untuk motor.

1. Menghubungkan beban (motor DC) dengan rangkaian *driver*.

**Tabel 4.3** *Hasil Pengujian Driver Dengan Beban (Motor DC)*

| No | *Input* | *Output* (Arah Putar Motor DC) |
|---|---|---|
| 1 | High, Low (Mundur) | Kiri |
| 2 | Low, High (maju) | Kanan |
| 3 | High, High | Off |
| 4 | Low, Low | Off |

Berdasarkan **Tabel 4.3** di atas maka dapat dianalisa bahwa ketika diberi *input High* dan *Low* maka arah putar motor DC akan bergerak ke kanan (maju). Dan sebaliknya jika diberi *input High* dan *Low* maka arah putar motor DC akan bergerak ke kiri (mundur), Tetapi jika inputan diberi nilai yang sama, maka motor DC OFF.

### 4.1.4 Pengujian dan Analisa Sensor Ultrasonik (PING)

Pengendalian meliputi pemancaran sinyal dan pengambilan data pewaktu setelah dideteksi adanya sinyal pantul. Pemancaran dilakukan dengan perintah SETB atau CLR kaki 10 mikrokontroler atau P.30/REX. Pada waktu pemancaran, mikrokontroler mulai melakukan perhitungan pewaktu 0 dengan mode 16 bit sampai diterima informasi adanya sinyal pantul. Apabila diterima pantulan, maka pewaktu dihentikan dan dicatat isinya kemudian dilakukan perhitungan jarak.

Berikut ini listing program untuk sensor ultrasonik agar dapat mendeteksi jarak objek dan memancarkan gelombang ultrasonik.

---

```
PING    EQU P1.0          ;GUNAKAN SATU PORT UNTUK KE PIN SIGNAL
JARAK1  EQU 21H
JARAK2  EQU 20H

ORG 00H


mulai:
SETB    PING              ; AKTIFKAN SIG KASIH 1
```

```
MOV     R6,#4           ; DELAY 6 uS
DJNZ    R6,$
CLR     PING            ; MATIKAN SIGNAL KASIH LOW
SETB    PING            ; SET P1.0 KASIH HIGH UNTUK INPUT
MOV     R6,#0FFH        ; DELAY BRUST 512 US
DJNZ    R6,$            ; 1 DJNZ = 2 CYCLE = 2US
JNB     PING,$          ; SELAMA SIG BELOM HIGH REPEAT
SETB    TR0             ; SIGNAL HIGH SINYAL DIKIRIM
SETB    TR0             ; TIMER AKTIFKAN
JB      PING,$          ; SELAMA BELUM ADA PANTULAN BIARKAN
CLR     TR0             ; SIGNAL LOW PANTULAN DITERIMA. MATIKAN TIMER
CLR     PING            ; MATIKAN PING
MOV     JARAK1,TL0      ; PINDAHKAN ISI TIMER LOW
MOV     JARAK2,TH0
MOV             TH0,#00h
MOV     TL0,#00h

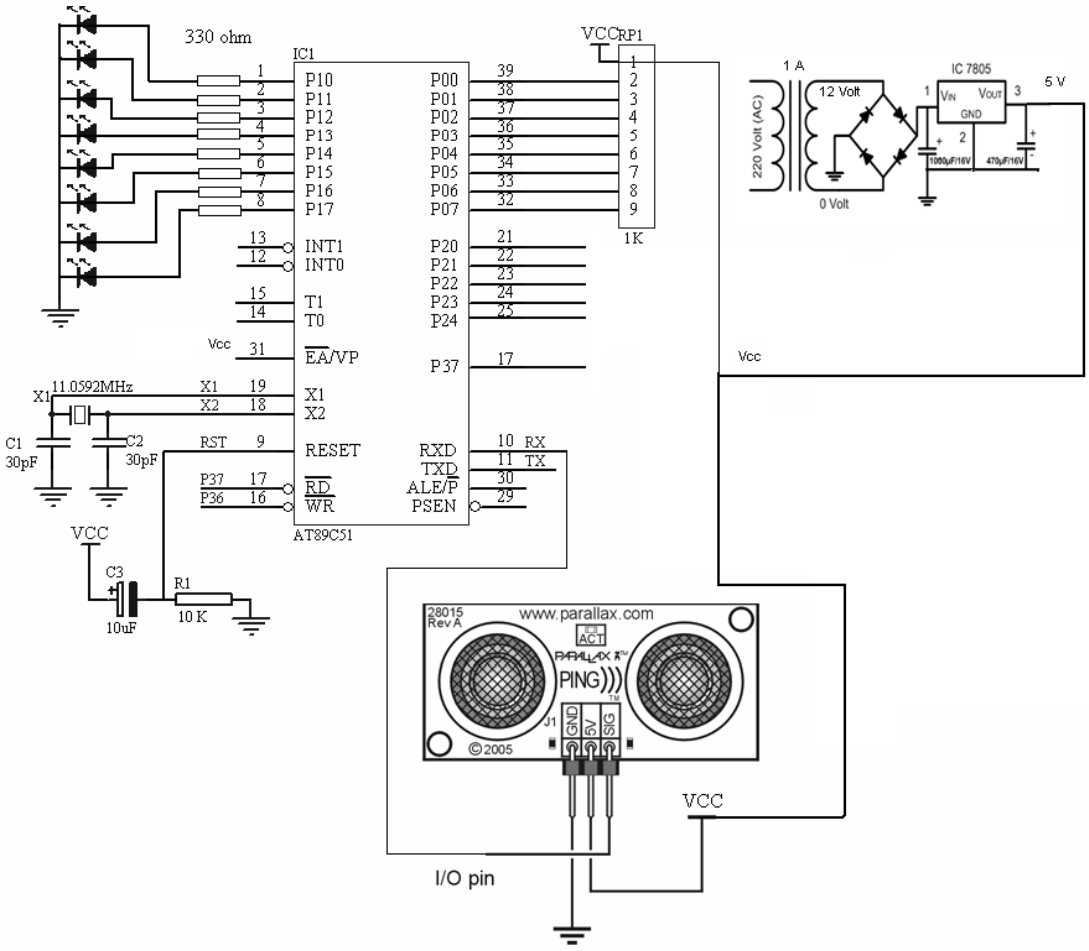
MOV             R1,21h
MOV             R2,20h
MOV             P2,R1
MOV             P3,R2
MOV     JARAK1,#0
MOV     JARAK2,#0
;jb             p1.2,$
sjmp mulai
END
```

---

Berdasarkan program diatas bahwa sensor ultrasonik (PING) menggunakan satu I/O. Aktifkan PING kasih *delay* 1 *brust* (PING siap-siap *send signal*). Setelah PING siap mengirim sinyal maka aktifkan *timer*, maka SIGNAL akan *high*. Lalu tunggu sinyal pantul yang dipancarkan, setelah menerima pantulan lalu matikan timer maka SIGNAL dari PING ini akan *low*.

Pengujian sensor utrasonik (PING) bertujuan apakah cara kerja dari sensor PING ini bekerja dengan baik atau belum. Peralatan yang digunakan.

1. Rangkaian sistem minimum mikrokontroler yang telah dihubungkan dengan sensor ultrasonik (PING).
2. *Power supply*.




**Gambar 4.4** *Pengujian Rangkaian Sensor Ultrasonik (PING)*

Setelah semua langkah-langkah pada pengujian rangkaian sensor ultrasonik (PING) selesai dilaksanakan yang ditunjukkan pada **Gambar 4.4**, selanjutnya dilakukan langkah-langkah sebagai berikut.

1. Memastikan bahwa semua jalur telah tersambung dengan benar dan semua komponennya telah terpasang dengan sempurna.
2. Menyiapkan *power supply* dengan *output* tegangan 5 Volt .
3. Menghubungkan sensor ultrasonik (PING) ke rangkaian mikrokontroler.

**Tabel 4.4** *Hasil Pengujian Rangkaian Ultrasonik*

| No | *Input* (jarak) | *Output* P.10-P.17 (*low*) |
|---|---|---|
| 1 | 10 cm | 00000010 |
| 2 | 20 cm | 00000100 |
| 3 | 30 cm | 00000110 |
| 4 | 40 cm | 00001000 |
| 5 | 50 cm | 00001011 |
| 6 | 60 cm | 00001101 |
| 7 | 70 cm | 00001111 |
| 8 | 80 cm | 00010000 |
| 9 | 90 cm | 00010011 |
| 10 | 100 cm | 00010101 |
| 11 | 110 cm | 00010111 |
| 12 | 120 cm | 00011001 |
| 13 | 130 cm | 00011011 |
| 14 | 140 cm | 00011101 |
| 15 | 150 cm | 00011111 |

Berdasarkan **Tabel 4.4** di atas maka dapat dianalisa bahwa ketika menerima pantulan dari objek dengan jarak yang berbeda maka hasil output dari mikrokontroler P.10 sampai P.17 akan berubah-ubah sesuai dengan jarak yang telah ditentukan.

Dari hasil pengujian pada **Tabel 4.4**, dapat dianalisa juga dengan perhitungan di mikrokontroler dengan persamaan:

$$S = \frac{V}{2} x \ t$$

$$t = \frac{\textit{Jarak yang terukur (cm)}}{\textit{Kecepatan rambat suara (cm/s)}}$$

$$t = \frac{10 \ cm}{34400 \ cm/s^{-1}}$$

$$= 0{,}029069 \ s = 0{,}0145345 \ s$$

$$t = \frac{150\ cm}{34400\ cm/s^{-1}}$$

$$= 0{,}436046\ s = 0{,}218023\ s$$

Nilai 58 merupakan *konstanta* untuk jarak 1 cm. Nilai ini didapatkan berdasarkan nilai *konstanta* perambatan suara 344 m/s. Vs = 344 m/s = 34400 $cm/s^{-1}$

$$t = \frac{1\ cm}{34400\ cm/s^{-1}}$$

$$= 29{,}0697\ \mu s$$

Dalam dua arah kirim dan pantul maka didapatkan :

$2t = 58{,}1395\mu s \approx 58\mu s$

Dengan *clock* mikrokontroler 1 µs maka didapatkan nilai *referensi* waktu per centimeter,

yaitu : $t = \frac{58\ \mu s}{1\ \mu s} = 58\ \mu s$

*dengan* $S$ = Jarak yang terukur (meter)
$V$ = Kecepatan rambat suara (m/s)
$t$ = waktu tempuh (s)

dari perhitungan di atas bahwa *konstanta* untuk 1 cm adalah 58 yang didapatkan berdasarkan perambatan suara 344 m/s. waktu yang dibutuhkan dalam

1 arah kirim adalah 29,0697, maka jika waktu yang dibutuhkan 2 arah kirim dan pantul adalah 58 *µs.* Jadi dengan *clock* mikrokontroler 1 *µs* maka didapatkan nilai *referensi* waktu percentimeter yaitu *58 µs.*

### 4.1.5 Pengujian Sistem

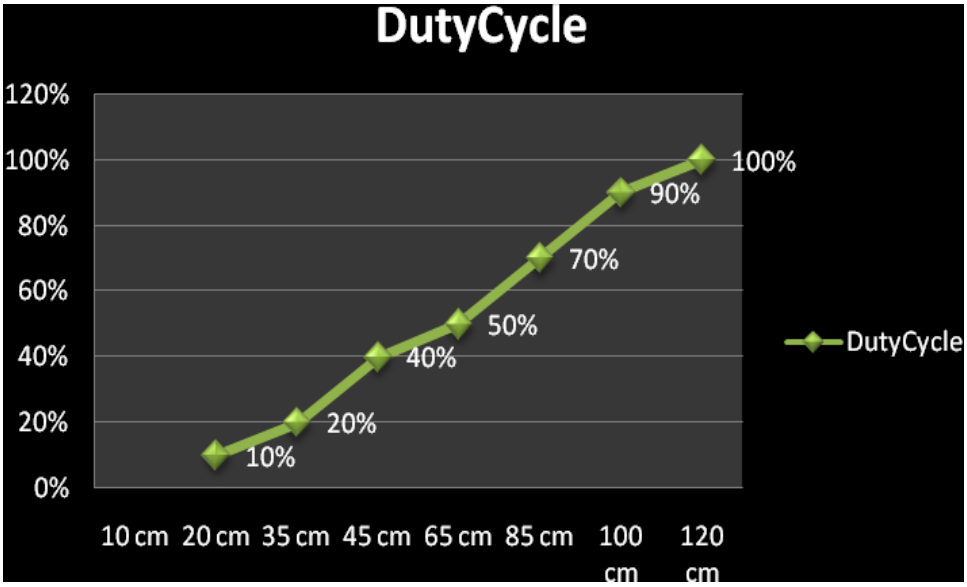
Pengujian dan analisa sistem dilakukan setelah pengecekan *hardware* secara per blok, pengujian ini mencakup pengujian sistem pada keseluruhan hardware yang telah dibuat. Pada **Gambar 4.5** di bawah dengan pengujian ini akan didapat hasil secara nyata dalam miniatur pengaturan motor DC berdasarkan perubahan jarak menggunakan metode PWM. Dapat dilihat pada **Tabel 4.5** data hasil pengujian sistem yang dilakukan dengan cara melihat langsung proses berjalannya alat atau sistem.

**Tabel 4.5** *Data Hasil Pengujian Sistem*

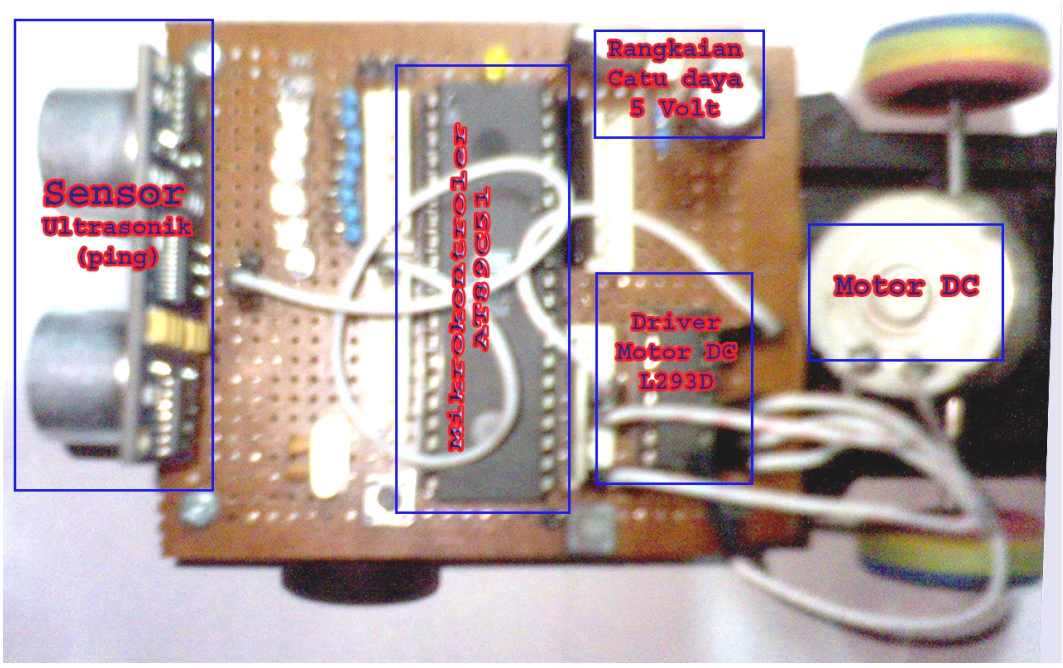
| No | *Dutycycle* | Jarak | waktu per cm |
|---|---|---|---|
| 1 | 0% | 5 - 19 cm | *0,029-0,1104 ms* |
| 2 | 10% | 20 cm | *1,160 ms* |
| 3 | 20% | 35 cm | *2,034 ms* |
| 4 | 40% | 45 cm | *2,616 ms* |
| 5 | 50% | 65cm | *3,778 ms* |
| 6 | 70% | 85 cm | *4,940 ms* |
| 7 | 90% | 100 cm | *5,800 ms* |
| 8 | 100% | >100 cm | *> 5,800 ms* |

Pada **Tabel 4.5** bisa dilihat bahwa semakin dekat jarak dari objek maka kecepatan motor semakin menurun sampai berhenti dalam jarak 19 cm. Jika jarak

dari benda lebih dari 100 cm atau 1 meter maka kecepatan motor maximum (*dc*=100%). **Gambar 4.5** menunjukan grafik kecepatan terhadap jarak bahwa semakin dekat jarak maka kecepatan menurun dan semakin jauh kecepatan akan naik.




**Gambar 4.5** *PWM Terhadap Jarak*



**Gambar 4.6** *Blok Diagram Pengujian Sistem*

**Gambar 4.6** menujukkan diagram pengujian sistem dan pengujian sistem dilakukan dengan proses sebagai berikut.

1. Pengukuran jarak dilakukan dengan variasi bahan objek penghalang, yaitu kayu, besi dan kertas (kardus). Didapatkan hasil yang akurat dan presisi yang ditandai dengan kesamaan antara hasil pengukuran dengan jarak sesungguhnya mulai 10 cm sampai 150 cm.
2. Di dalam mikrokontroler sendiri terdapat program dimana pengiriman sinyal dan penerimaan sinyal dari sensor ultrasonik (PING) akan terus berulang hingga mencapai tujuan sampai berhenti.
3. Sistem akan menjaga jarak supaya tidak menabrak objek, dengan jarak yang telah ditentukan yaitu 19 cm. jika kurang dari 19 cm maka sistem ini akan menyesuaikan sampai 19 cm.
4. Kecepatan motor dilakukan dengan kondisi catu daya 12 V jarak dibuat konstan 0 cm karena motor dikondisikan berputar di tempat.
5. Apabila tidak terdapat halangan atau jarak penghalang di atas 120 cm, maka motor berputar dengan kecepatan penuh. Namun apabila dideteksi jarak penghalang di bawah 120 cm atau secara bertahap mendekati penghalang maka proses pengereman/reduksi kecepatan mulai dilakukan.
6. Selama penghalang masih ada maka pengereman akan terus berjalan hingga pada jarak 19 cm motor berhenti dan menunggu penghalang dipindahkan.
7. Ketika jarak dari sensor berubah-ubah maka kecepatan motor juga akan berubah-ubah sesuai pengaturan PWM dan jarak yang telah ditentukan.

# BAB V

# PENUTUP

## 5.1 KESIMPULAN

Setelah melakukan proses pengujian dan analisa terhadap sistem yang telah dibuat dapat diambil kesimpulan bahwa.

1. Pada sistem kendali kecepatan motor DC berdasarkan perubahan jarak menggunakan metode PWM diperlukan sebuah sensor ultrasonik (PING), *power supply* 5 volt, Modul Mikrokontroler (AT89C51 dan *Driver* Motor DC) untuk mengatur pergerakan motor DC.
2. Hasil pengukuran jarak menunjukkan hasil ketelitian dari sensor adalah 5 cm.

3. Kecepatan motor DC dapat direduksi secara bertahap dengan menggunakan metode PWM mulai dari 10% sampai 100% dengan perubahan jarak yang diterima oleh sensor ditunjukan oleh indikator LED dari mikrokontroler.
4. Posisi benda/objek yang akan diukur jaraknya harus tegak lurus terhadap garis pandang sensor, jika tidak maka akan terjadi pemantulan gelombang ultrasonik yang tidak sempurna dan menyebabkan kesalahan pengukuran.
5. Kerugian modul PING menggunakan satu pin sebagai *input* dan *output*. Oleh karena itu, hal yang harus diperhatikan dalam menggunakan modul ini adalah penentuan saat yang tepat untuk mengatur *port* mikrokontroler sebagai *input* setelah mengeluarkan pulsa *trigger* ke PING. Bila terlalu cepat atau terlalu lambat mengatur *port* 3.0 sebagai *input* (diberi logika *high*) akan menyebabkan kesalahan pengukuran jarak, hasil pengukuran menjadi lebih jauh atau lebih dekat dari jarak sesungguhnya.
6. Keuntungan menggunakan modul PING ini yaitu pada sisi *hardware*. Modul PING tidak membutuhkan komponen tambahan, dan memiliki *output* digital, serta hanya memerlukan satu pin I/O sehingga menghemat pin mikrokontroler.

## 5.2 SARAN

Pada sistem kendali kecepatan motor DC berdasarkan perubahan jarak menggunakan metode PWM ini memiliki banyak kekurangan, maka perlu pengembangan lebih lanjut pada waktu yang akan datang. Adapun saran-saran untuk proyek akhir ini adalah.

1. Menggunakan tampilan pada LCD, baik jarak yang terukur maupun kecepatan pada motor. Agar pembuatan alat lebih sempurna.
2. Menggunakan kendaraan yang asli/sesungguhnya sehingga dapat langsung diaplikasikan ke masyarakat.

## DAFTAR PUSTAKA


1. http:// www.parallax.com, “Modul sensor PING)))
2. http://www.atmel.com, ”8 bit microcontroller with 4K bytes In System Programmable Flash AT89S51”, Desember 2006
3. Setio Budiyanto, 2006, *Implementasisistem kendali kecepatan motor DC 12 dengan metode PWM berbasis mikrokontroler pada sepeda listrik*,Tugas Akhir Institut Teknologi Nasional, Bandung (tidak diterbitkan).
4. http:// www.unisonic.com
5. http://www.datasheetcatalog.com
6. http://www.acroname.com

7. Afgianto Eko Putra (2003), Belajar Mikrokontroler AT89C51/52/55 Teori dan Aplikasi, GAVA MEDIA, Edisi Kedua

# LAMPIRAN