

**Optimalisasi Persediaan Multi Item Inventory Ketika Anggaran  
Terbatas Di CV. Jaya Abadi Furniture**

**TUGAS AKHIR**

**Diajukan sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata-1  
Jurusan Teknik Industri**



Oleh :

Nama : Benny Hady Saputra

No. Mahasiswa : 04 522 091

**JURUSAN TEKNIK INDUSTRI  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
2011**

**APLIKASI SENSOR ULTRASONIK  
UNTUK PENGATURAN PARKIR KENDARAAN**

**TUGAS AKHIR**

**Diajukan Untuk Salah Satu Syarat**

**Untuk Memperoleh Gelar Sarjana**

**Teknik Elektro**



Oleh:

**Nama : Akmad Fauzan**

**No.Mahasiswa : 00 524 096**

**FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2011**

**LEMBAR PENGESAHAN PEMBIMBING**

**APLIKASI SENSOR ULTRASONIK  
UNTUK PENGATURAN PARKIR KENDARAAN**

**TUGAS AKHIR**



Oleh:

Nama : Akhmad Fauzan

No Mahasiswa : 00 524 096

Yogyakarta, Juli 2011

Pembimbing I

A handwritten signature in black ink, consisting of a large, stylized 'T' and 'Y'.

Tito Yuwono, ST., M.sc.

Pembimbing II

A handwritten signature in black ink, consisting of a large, stylized 'D' and 'R'.

Dwi Ana Ratna Wati, ST., M.Eng.

**LEMBAR PENGESAHAN PENGUJI**

**APLIKASI SENSOR ULTRASONIK  
UNTUK PENGATURAN PARKIR KENDARAAN**

**TUGAS AKHIR**

Oleh:

Nama : Akhmad Fauzan

No Mahasiswa : 00 524 096

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat  
untuk Memperoleh Gelar Sarjana Teknik Elektro  
Fakultas Teknologi Industri Universitas Islam Indonesia.

Yogyakarta, Juli 2011

Tim penguji:

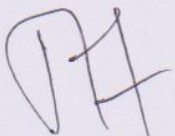
Tito Yuwono, ST., M.sc.  
Ketua

Wahyudi Budi Pramono, ST., M.Eng. : \_\_\_\_\_  
Anggota I

Ir. Hj. Budi Astuti, M.T.  
Anggota II

Mengetahui :

Ketua Jurusan Teknik Elektro  
Universitas Islam Indonesia

  
( Tito Yuwono, ST., M.sc. )

## **HALAMAN PERSEMBAHAN**

Alhamdulillahirobbil'alamin

Segenap puji dan syukur hamba haturkan kepada Allah SWT yang  
yak henti-hentinya memberikan nikmat dan rahmatnya, segala  
kesempatan yang telah diberikannya, semoga Allah SWT akan  
selalu menyertai setiap langkahku.

Ku Persembahkan tugas akhir ini kepada bapak-ibuku, kakak-  
kakakku, adik-adikku, teman dan sahabat seiman serta siapa saja  
yang membaca tugas akhir ini

## MOTTO

Sesungguhnya sembahyangku, ibadatku, hidupku dan matiku  
hanyalah untuk Allah, Tuhan semesta Alam (QS Al An'am : 162)

Sesungguhnya Sesudah Kesulitan itu Ada Kemudahan  
(QS. Al-Insyirah : 6)

Allah memberi kita cobaan  
Agar kita kuat



## KATA PENGANTAR

*Assalamualaikum Wr, Wb.*

Alhamdulillahirabbil‘alamiin. Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya, sholawat serta salam selalu tercurah kepada Nabi besar Muhammad SAW atas teladannya yang telah memberikan kekuatan, kesabaran dan taqwa kepada hamba-Nya dalam memperjuangkan hak dan kewajiban untuk menempuh suatu keinginan dan pengetahuan sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“APLIKASI SENSOR ULTRASONIK UNTUK PENGATURAN PARKIR KENDARAAN”** untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.

Penyelesaian penyusunan laporan Tugas Akhir ini tidak terlepas dari bantuan, saran, nasehat dan dukungan dari berbagai pihak. Untuk itu penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Ir. Gumbolo Hadi Susanto, M.Sc., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Tito Yuwono, ST., M.sc., selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia dan juga sebagai Dosen Pembimbing I yang telah meluangkan waktu serta memberikan pengarahan dan bimbingan sepenuhnya dalam penyusunan laporan tugas akhir ini.

3. Ibu Dwi Ana Ratna Wati ST., M.Eng., selaku Dosen Pembimbing II yang telah meluangkan waktu serta memberikan pengarahan dan bimbingan sepenuhnya dalam penyusunan laporan tugas akhir ini.
4. Seluruh Dosen Teknik Elektro, pegawai dan staf Universitas Islam Indonesia.
5. Bapak Muslich (*alm*) dan Ibu Siti latifah (*almh*) yang telah memberikan cinta kasih selama hidupnya.
6. Kakak-kakakku Farah Ellyana, Kartika Fitriani, M. Agus Prawoto, Ulya Abida, Arif Rachman, Muhammad Fuad, adikku Unun Fathimah dan Ratna Faizah yang selalu memberikan cinta yang tulus, kasih sayang, materi, motivasi, semangat dan do'a restunya.
7. Teman-teman Mitra Transport Kang Aziz, Joko, sunardi yang telah memberikan semangat dan dukungan untuk menyelesaikan Tugas Akhir ini.
8. Teman-teman Teknik Elektro angkatan 2000.
9. Semua pihak-pihak yang telah membantu terselesaikannya Tugas akhir ini.

Penulis menyadari bahwa dalam penulisan laporan Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu kritik dan saran yang membangun sangat penulis harapkan. Semoga laporan Tugas Akhir ini dapat berguna dan memberikan manfaat bagi pembaca.

***Wassalamu'alaikum Wr, Wb.***

Yogyakarta, Juli 2011

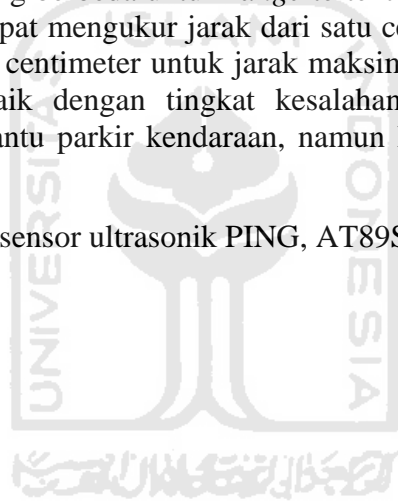
Penulis



## ABSTRAKSI

Jumlah volume kendaraan yang semakin meningkat kurang diimbangi dengan infra struktur yang memadai. Semakin berkurangnya lahan parkir merupakan salah satunya. Pengemudi kendaraan sering kali kesulitan untuk memarkirkan kendaraannya pada tempat yang sempit. Hal inilah yang menginspirasi penulis untuk membuat alat bantu parkir kendaraan dengan memanfaatkan sensor ultrasonik. Sistem alat ini terdiri dari sensor ultrasonik sebagai input, mikrokontroler sebagai pengendali utama, LCD, LED dan *buzzer* sebagai output. Sensor ultrasonik PING digunakan sebagai pengukur jarak dengan halangan yang ada kemudian diolah pada mikrokontroler AT89S52 yang kemudian jarak ditampilkan pada LCD dalam satuan centimeter dan status berupa jarak aman, hati-hati dan stop, indikator lain ditambahkan berupa nyala LED serta bunyi *buzzer* dengan *interval* yang berbeda untuk *range* tertentu. Dalam pengujian yang telah dilakukan alat ini dapat mengukur jarak dari satu centimeter sampai dengan dua ratus lima puluh lima centimeter untuk jarak maksimal Secara umum alat ini dapat bekerja dengan baik dengan tingkat kesalahan  $\pm 0,89$  % dan dapat digunakan sebagai alat bantu parkir kendaraan, namun kehati-hatian pengemudi tetap yang paling utama.

Kata kunci: sensor parkir, sensor ultrasonik PING, AT89S52, LCD, jarak



## DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PENGESAHAN PENGUJI	iii
PERSEMBAHAN	iv
HALAMAN MOTTO	v
KATA PENGANTAR	vi
ABSTRAKSI	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Sistematika Penulisan	3
BAB II STUDI PUSTAKA	5
2.1 Studi Pustaka	5
2.2 Gelombang Ultrasonik	6
2.3 Sensor Ultrasonik PING	7
2.3.1 Spesifikasi Sensor Ultrasonik PING	7
2.3.2 Prinsip Kerja Sensor Ultrasonik PING	8
2.4 Mikrokontroler AT89S52	10
2.4.1 Spesifikasi Mikrokontroler AT89S52	10
2.4.2 Konfigurasi Pin Mikrokontroler AT89S52	12
2.5 LCD ( <i>Liquid Crystal Display</i> )	14
2.5.1 Pin-Pin LCD M1632	17
2.6 Indikator Peringatan	18
2.6.1 LED ( <i>Light Emitting Diode</i> )	18

2.6.2 <i>Buzzer</i>	20
<b>BAB III PERANCANGAN SISTEM</b>	<b>23</b>
3.1 Perancangan Umum sistem	23
3.2 Perancangan Perangkat Keras	24
3.2.1 Sistem Minimum Mikrokontroler AT89S52	24
3.2.2 Rangkaian LCD ( <i>Liquid Crysta Display</i> )	25
3.2.3 Instalasi Sensor Ultrasonik PING	27
3.2.4 Pemasangan LED dan <i>Buzzer</i>	28
3.2.5 Catu Daya	29
3.3 Perancangan Perangkat Lunak	30
3.4 Pemograman	31
3.4.1 Inisialisasi LCD	31
3.4.2 Pemograman Sensor PING	32
3.4.3 Pemograman LED dan <i>Buzzer</i>	33
<b>BAB IV PENGUJIAN DAN ANALISIS</b>	<b>36</b>
4.1 Pengujian Alat	36
4.2 Analisis	39
<b>BAB V PENUTUP</b>	<b>40</b>
5.1 Kesimpulan	40
5.2 Saran	40
<b>DAFTAR PUSTAKA</b>	<b>42</b>
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

Gambar 2.1 PING <i>Ultrasonic Range Finder</i> -----	7
Gambar 2.2 Ilustrasi Cara Kerja Sensor Ultrasonik PING -----	9
Gambar 2.3 Timming Diagram Sensor Ultrasonik PING -----	9
Gambar 2.4 Blok Diagram Mikrokontroler AT89S52-----	11
Gambar 2.5 Konfigurasi Pin Mikrokontroler AT89S52-----	12
Gambar 2.6 LCD ( <i>Liquid Crystal Display</i> ) -----	15
Gambar 2.7 Simbol LED -----	20
Gambar 2.8 Simbol <i>Buzzer</i> -----	21
Gambar 3.1 Diagram Blok Sistem-----	23
Gambar 3.2 Skema Rangkaian Sistem Minimum Mikrokontroler AT89S52-----	25
Gambar 3.3 Skema Rangkaian LCD ( <i>Liquid Crystal Display</i> )-----	26
Gambar 3.4 Konfigurasi Pin Sensor Ultrasonik PING -----	27
Gambar 3.5 Skema Rangkaian LED -----	28
Gambar 3.6 Skema Rangkaian <i>Buzzer</i> -----	29
Gambar 3.7 Skema Rangkaian Catu Daya -----	29
Gambar 3.8 Flowchart Program -----	30

## DAFTAR TABEL

Tabel 2.1 Fungsi Alternatif <i>Port 1</i> -----	13
Tabel 2.2 Fungsi Alternatif <i>Port 3</i> -----	13
Tabel 2.3 Pola Pola Karakter Yang Tersimpan Dalam CGROM -----	17
Tabel 2.2 Fungsi Pin LCD M1632 -----	18
Tabel 4.1 Data Hasil Pengukuran Jarak -----	36
Tabel 4.2 Pengukuran LED -----	38



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Seiring dengan berjalannya waktu dan pertumbuhan ekonomi yang semakin baik, jumlah para pengguna kendaraan semakin meningkat. Meningkatnya volume kendaran itu kurang diimbangi dengan infrastruktur yang memadai, salah satunya adalah lahan parkir yang semakin bekurang atau sempit. Pengemudi mobil seringkali mengalami kesulitan untuk memarkir mobilnya di lokasi yang sempit. Tidak sedikit pengemudi yang menabrak tiang pembatas atau menabrak tembok yang ada di belakang kendaraanya ketika memundurkan mobilnya, Itu disebabkan karena si pengemudi kurang bisa melihat ataupun memperkirakan jarak antara mobilnya dengan halangan yang ada di belakang, selama ini kontrol pengemudi hanya mengandalkan kaca *spion* untuk melihat bagian belakang kendaraan. Hal inilah yang menginspirasi penulis untuk merancang alat bantu dalam memarkirkan kendaraan khususnya pada saat mundur yaitu “Aplikasi Sensor Ultrasonik Untuk Pengaturan Parkir Kendaraan”.

Dengan adanya alat ini si pengemudi dapat melihat dan memonitor berapa jarak sisi belakang mobilnya dengan halangan yang ada di belakangnya. Sekaligus dapat menjadi peringatan dini bagi si pengemudi ketika jarak sudah terlalu dekat agar resiko menabrak dapat dikurangi.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan maka dapat dibuat rumusan masalah penelitian ini sebagai berikut:

Bagaimana merancang suatu alat yang dapat membantu pengemudi untuk memonitor jarak kendaraan dengan halangan yang ada di belakangnya sekaligus menjadi peringatan dini bagi si pengemudi. Yaitu aplikasi sensor ultrasonik untuk pengaturan parkir kendaraan.

## 1.3 Batasan Masalah

Agar penelitian ini tidak menyimpang dari pokok permasalahan maka diperlukan batasan-batasan permasalahan yang dibahas, antara lain:

- a. Penggunaan sensor ultrasonik sebagai sensor.
- b. Penggunaan mikrokontroler sebagai pengendali utama.
- c. penggunaan LCD, LED dan *buzzer* sebagai media penampil data dan indikator peringatan.
- d. Prinsip kerja dari sensor parkir kendaraan.
- e. Analisis dan pengujian alat yang telah dibuat.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian tentang aplikasi sensor ultrasonik untuk pengaturan parkir kendaraan adalah untuk menciptakan sebuah rangkaian elektronik atau alat yang dapat mempermudah pengemudi kendaraan untuk memarkirkan kendaraannya pada tempat yang sempit dan implementasi dari teori-teori yang didapat di bangku kuliah.

## 1.5 Sistematika Penulisan

Adapun sistematika penulisan Tugas Akhir adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Bagian ini berisi tentang latar belakang pembuatan alat aplikasi sensor ultrasonik untuk pengaturan parkir kendaraan. Terdiri dari latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian dan sistematika penulisan.

### **BAB II STUDI PUSTAKA**

Studi pustaka memuat penelitian bidang sejenis yang telah dilakukan sebelumnya. Bagian ini juga berisi teori-teori pendukung mengenai gelombang ultrasonik, sensor ultrasonik, prinsip kerja sensor ultrasonik PING, mikrokontroler AT89S52 dan LCD sebagai media penampil data yang digunakan sebagai landasan pemecahan masalah.

### **BAB III PERANCANGAN SISTEM**

Bagian ini menjelaskan perancangan sistem yang digunakan, seperti perancangan perangkat keras yang meliputi sensor ultrasonik sebagai *input*, sistem minimum mikrokontroler sebagai pusat pengendali, LCD, LED dan buzzer sebagai *output* sistem. Sedangkan perancangan perangkat lunak meliputi pemograman mikrokontroler.

### **BAB IV PENGUJIAN, ANALISIS**

Bagian ini berisi tentang data hasil pengujian alat dan analisis dari alat yang telah dibuat serta hambatan-hambatan yang dialami selama pengambilan data.



## **BAB V PENUTUP**

Bagian ini berisi tentang kesimpulan-kesimpulan dari unjuk kerja sistem yang telah dibuat, terutama pada analisis kerja sistem. Bagian ini juga berisi saran-saran untuk pengembangan lebih lanjut yang masih bisa diwujudkan dari penelitian yang telah dilaksanakan.



## **BAB II**

### **STUDI PUSTAKA**

#### **2.1 Studi Pustaka**

Penelitian tentang pengukuran jarak menggunakan pemanfaatan gelombang ultrasonik pernah dilakukan Iksan Maulana dengan judul “Peningkatan Akurasi Pada Alat Ukur Jarak Digital”. Pada penelitian ini hanya pada tentang pengukuran jarak dengan memanfaatkan sensor ultrasonik yang kemudian ditampilkan pada display LCD. Pada prinsipnya, prinsip kerja dari alat yang ada adalah dengan menghitung perbedaan waktu antara saat gelombang ultrasonik dikirim oleh sensor dan gelombang ultrasonik yang dipantulkan oleh obyek yang kemudian diterima sensor, sehingga diperoleh perbandingan lurus antara sensor dengan obyek yang akan diukur. Selain sebagai pengukur jarak, sensor ultrasonik juga dimanfaatkan sebagai navigasi pada bidang robotika, pemantauan tinggi *level* air sungai dan juga pemanfaatan yang lain.

Atas dasar-dasar diatas maka dilakukan penelitian lanjutan dengan memanfaatkan sensor ultrasonik sebagai Alat pengukur jarak untuk pengaturan parkir kendaraan. Selain sebagai pengukur jarak dengan halangan di belakang kendaraan juga ditambahkan lampu indikator LED dan *buzzer* sebagai peringatan bagi pengemudi, dengan adanya bunyi *beep* dan indikator lampu LED diharapkan pengemudi agar lebih waspada dan berhati-hati.

## 2.2 Gelombang Ultrasonik

Gelombang ultrasonik adalah gelombang yang memiliki frekuensi di atas 20 Khz dan tidak dapat didengar oleh telinga manusia. Gelombang ultrasonik tergolong dalam gelombang mekanik longitudinal, sehingga secara fisik gelombang ultrasonik mempunyai sifat yang sama dengan gelombang suara biasa yang dapat didengar oleh telinga manusia. Karena termasuk jenis gelombang mekanik, maka gelombang ultrasonik hanya dapat merambat dan bergetar melalui suatu medium penghantar dan tidak dapat merambat melalui ruang hampa (vakum). Selain itu kecepatan rambat gelombang ultrasonik di ruang bebas sama dengan cepat rambat gelombang suara, yaitu kurang lebih 344 m/s.

Suatu gelombang ultrasonik yang merambat melalui suatu medium, getarannya merupakan getaran dari partikel-partikel medium yang dilaluinya. Gelombang ultrasonik akan dipantulkan kembali ke arah gelombang tersebut dipancarkan bila terjadi *diskontinuitas* pada medium dimana gelombang ultrasonik itu merambat. Dengan demikian sifat gelombang ultrasonik ini dapat digunakan untuk mengukur jarak dengan syarat selang waktu antara saat gelombang dikirim dan gelombang pantul diterima, jarak antara alat pengukur dan benda penghalang bisa dihitung.

Sedangkan Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara frekuensi tinggi lebih dari 20 Khz (pada umumnya menggunakan frekuensi 40 Khz) dimana sensor menghasilkan gelombang suara yang kemudian menangkapnya kembali dengan perbedaan waktu sebagai dasar pengindraannya.

### 2.3 Sensor Ultrasonik PING

PING *Ultrasonic Range Finder*, adalah modul pengukur jarak dengan ultrasonik buatan Parallax Inc. yang didesain khusus untuk teknologi robotika. Dengan ukurannya yang cukup kecil (2,1cm x 4,5cm), sensor ini dapat mengukur jarak antara 3 cm sampai 300 cm. Keluaran dari PING berupa pulsa yang lebarnya merepresentasikan jarak. Lebar pulsanya bervariasi dari 115 uS sampai 18,5 mS. Pada dasarnya PING terdiri dari sebuah chip pembangkit sinyal 40 KHz, sebuah *speaker* ultrasonik dan sebuah mikrofon ultrasonik. *Speaker* ultrasonik mengubah sinyal 40 KHz menjadi suara, sementara mikrofon ultrasonik berfungsi untuk mendeteksi pantulan suaranya. Pada modul PING terdapat 3 pin yang digunakan untuk jalur power supply (+5V), *ground* dan SIG (*signal*). Pin SIG merupakan input sekaligus output dan dapat langsung dihubungkan dengan mikrokontroler tanpa tambahan komponen apapun.



Gambar 2.1. PING *Ultrasonic Range Finder*

#### 2.3.1 Spesifikasi Sensor Ultrasonik PING

Sensor ultrasonik PING memiliki spesifikasi antara lain:

- power supply 5 VDC

- arus 30 mA ; 35 mA max
- Range 3 cm sampai 3 m
- Input Trigger - positif TTL pulsa, 2 uS min, 5 uS typ
- Echo Pulse positif TTL pulsa, 115 uS sampai 18,5 mS
- Frekuensi burst - 40 kHz untuk 200 uS
- Indicator LED untuk menunjukkan aktivitas sensor
- Delay sebelum pengukuran berikutnya - 200 uS
- Ukuran - 22 mm x 46 mm H W x 16 mm D (0,84 x 1,8 x 0,6 di dalam)

### 2.3.2 Prinsip Kerja Sensor Ultrasonik PING

PING mendeteksi objek dengan cara mengirimkan suara ultrasonik dan kemudian “mendengarkan” pantulan suara tersebut. Ping hanya akan mengirimkan suara ultrasonik ketika ada pulsa *trigger* dari mikrokontroler (Pulsa high selama 5uS). Suara ultrasonik dengan frekuensi sebesar 40KHz akan dipancarkan selama 200uS. Suara ini akan merambat di udara dengan kecepatan 344.424m/detik (atau 1cm setiap 29.034uS), mengenai objek untuk kemudian terpantul kembali ke Ping. Selama menunggu pantulan, sensor akan menghasilkan sebuah pulsa. Pulsa ini akan berhenti (*low*) ketika suara pantulan terdeteksi oleh sensor. Oleh karena itulah lebar pulsa tersebut dapat merepresentasikan jarak antara sensor dengan objek. Selanjutnya mikrokontroler cukup mengukur lebar pulsa tersebut dan mengkonversinya dalam bentuk jarak dengan perhitungan sebagai berikut :

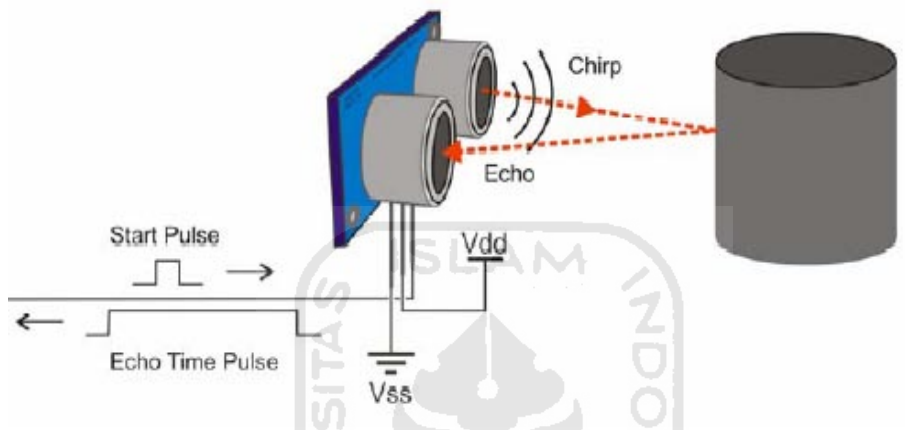
$$\text{Jarak} = (\text{Lebar Pulsa}/29.034\text{uS})/2 \quad (\text{dalam cm}) \dots\dots\dots (2.1)$$

atau

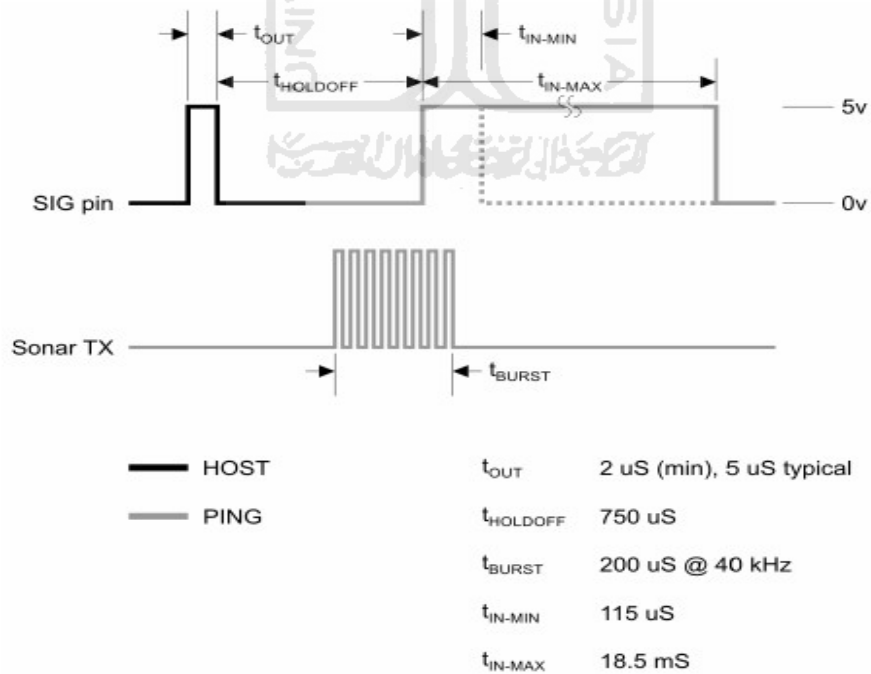
$$\text{Jarak} = (\text{Lebar Pulsa} \times 0.034442) / 2 \quad (\text{dalam cm}) \dots\dots\dots (2.2)$$

Karena  $1/29.034 = 0.34442$

Berikut ini adalah ilustrasi cara kerja dan *timing* sensor ultrasonik PING.



Gambar 2.2. Ilustrasi cara kerja sensor ultrasonik PING



Gambar 2.3. Timing diagram sensor ultrasonik PING

## 2.4 Mikrokontroler AT89S52

Mikrokontroler adalah sebuah *chip* yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program didalamnya. Mikrokontroler umumnya terdiri dari CPU (*Central Processing Unit*), memori, I/O tertentu dan unit pendukung seperti *Analog-to-Digital Converter* (ADC) yang sudah terintegrasi di dalamnya. Kelebihan utama dari mikrokontroler ialah tersedianya RAM dan peralatan I/O pendukung sehingga ukuran *board* mikrokontroler menjadi sangat ringkas.

Mikrokontroler AT89S52 merupakan mikrokomputer CMOS 8 bit dengan 8 KB *Flash PEROM* (*Programmable and Erasable Only Memory*) yang dapat dihapus dan ditulisi sebanyak 1000 kali. Mikrokontroler ini diproduksi dengan menggunakan teknologi *high density non-volatile memory* Atmel. Kombinasi CPU 8 bit serba guna dan *Flash PEROM*, menjadikan mikrokontroler AT89S52 menjadi *microcomputer* handal yang fleksibel.

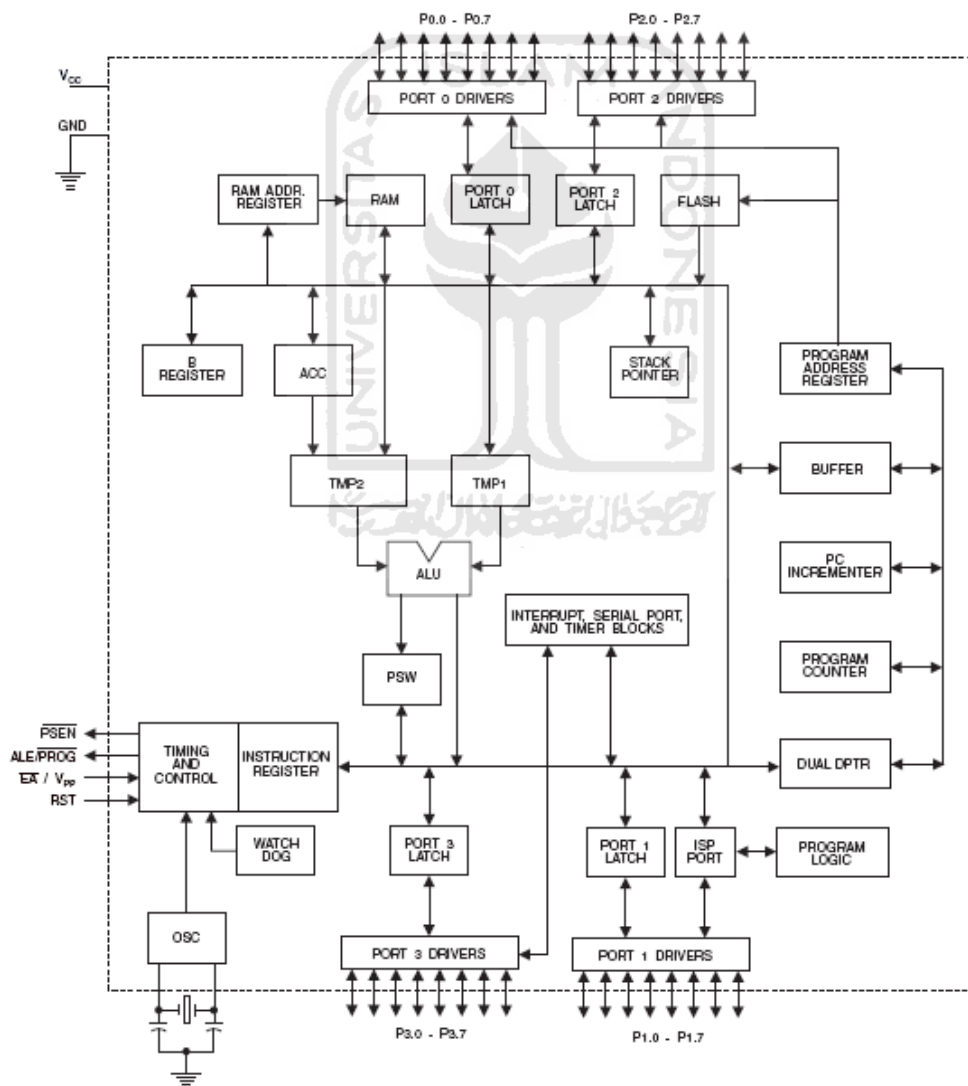
### 2.4.1 Spesifikasi Mikrokontroler AT89S52

Spesifikasi dan fasilitas yang dimiliki mikrokontroler AT89S52 adalah sebagai berikut:

- Kompatibel dengan keluarga mikrokontroler MCS51 sebelumnya.
- 8 K *bytes In System Programmable* (ISP) *flash* memori dengan kemampuan 1000 kali baca/tulis.
- Mode pemrograman ISP yang fleksibel (*Byte dan Page Mode*).
- tegangan kerja 4,0 sampai 5,5 V.
- Bekerja dengan rentang 0 sampai 33 MHz.

- Memiliki 256 x 8-bit RAM internal.
- Delapan sumber *interrupt*
- Empat buah *programmable port* I/O yang masing-masing terdiri dari delapan buah jalur I/O.
- Memiliki 3 *timer/counter* 16 bit
- Saluran *full duplex* serial UART

Berikut ini adalah blok diagram mikrokontroler AT89S52



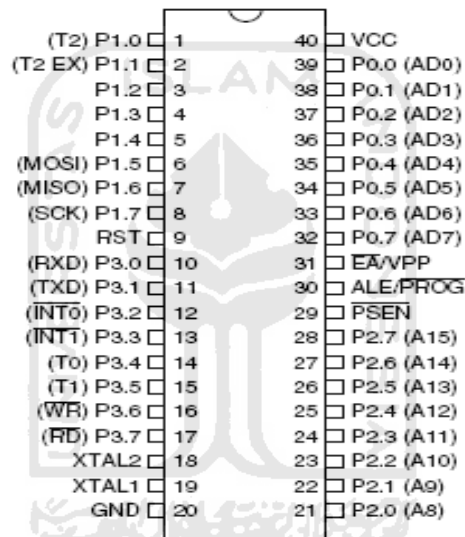
Gambar 2.4. Blok diagram mikrokontroler AT89S52



### 2.4.2 Konfigurasi Pin Mikrokontroler AT89S52

Mikrokontroler AT89S52 mempunyai 40 pin, dalam kemasan DIP (*dual in line package*). 32 pin digunakan untuk keperluan 4 buah port paralel. 1 port terdiri dari 8 pin yang dapat dihubungkan untuk *interfacing* ke paralel *device*, seperti ADC, sensor dan sebagainya, atau dapat juga digunakan secara sendiri setiap bit-nya untuk *interfacing single bit* seperti *switch*, LED dan lain lain.

Berikut adalah konfigurasi pin mikrokontroler AT89S52



Gambar 2.5. Konfigurasi pin mikrokontroler AT89S52

Adapun fungsi dari masing masing pin dapat dideskripsikan sebagai berikut:

1. VCC (pin 40) sebagai sumber tegangan +5v
2. GND (pin 20) dihubungkan ke *ground*
3. RST (pin 9) mengembalikan kondisi kerja mikrokontroler pada posisi awal.
4. *Port 0* (pin 32-38) merupakan I/O delapan bit jalur dua arah dengan internal *pull-up*

5. *Port 1* (pin 1-8) merupakan I/O delapan bit dua arah dengan resistor internal *pull-up*, yang juga mempunyai fungsi alternatif seperti ditunjukkan pada tabel berikut:

Tabel 2.1. Fungsi alternatif *port 1*

<b>Port</b>	<b>Fungsi alternatif</b>
P1.0	T2 ( <i>timer/counter 2 input eksternal</i> )
P1.1	T2EX ( <i>timer/counter 2 input eksternal</i> )
P1.5	MOSI ( <i>SPI Bus Master Output/Slave Input</i> )
P1.6	MISO ( <i>SPI Bus Master Input/Slave Output</i> )
P1.7	SCK ( <i>SPI Bus Serial Clock</i> )

6. *Port 2* (pin 21-28) merupakan I/O delapan bit dua arah yang dilengkapi internal *pull-up*
7. *Port 3* (pin 10-17) merupakan I/O delapan bit dua arah yang dilengkapi internal *pull-up*, yang juga mempunyai fungsi alternatif seperti yang ditunjukkan pada tabel berikut:

Tabel 2.2 fungsi alternatif *port 3*

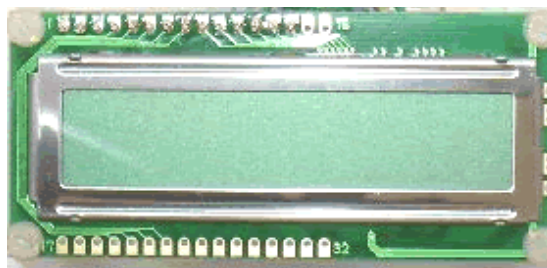
<b>Port</b>	<b>Fungsi Alternatif</b>
P3.0	RXD (Port komunikasi input serial)
P3.1	TXD (port komunikasi input)
P3.2	INT0 (Saluran interupsi eksternal 0)
P3.3	INT1 (Saluran interupsi eksternal 1)
P3.4	T0 (Input timer 0 eksternal)
P3.5	T1 (Input timer 1 eksternal)
P3.6	WR (Jalur menulis memori data eksternal)
P3.7	RD (Jalur membaca memori data eksternal)

8. ALE (pin 30) Adalah Address Latch Enable (ALE) berfungsi menahan sementara alamat *byte* rendah pada proses pengalamatan ke memori eksternal.
9. EA/Vpp (pin 31) Adalah *External Acces Enable* (EA) yang merupakan sinyal kontrol untuk pembacaan memori program. Apabila diset rendah (L) maka mikrokontroler akan melaksanakan seluruh instruksi dari memori program eksternal, sedangkan jika diset tinggi (H) maka mikrokontroler akan melaksanakan instruksi dari memori program internal. Pin ini juga berfungsi sebagai tegangan pemograman Vcc 12V selama proses pemograman.
10. PSEN (pin 29) Adalah *Program Store Enable* (PSEN ), yaitu masukan sinyal baca untuk memori program eksternal agar masuk ke dalam bus selama proses pemberian/pengambilan instruksi (*fetching*).
11. XTAL1 Adalah masukan ke rangkaian osilator internal. Sumber osilator eksternal atau *quartz crystal* dapat digunakan.
12. XTAL2 Adalah masukan ke rangkaian osilator internal, koneksi *quartz crystal* atau tidak dikoneksikan apabila digunakan eksternal osilator.

## 2.5 LCD ( *liquid crystal Display* )

Banyak sekali fungsi dari LCD dalam perancangan suatu sistem yang menggunakan mikrokontroler. LCD berfungsi untuk menampilkan suatu nilai dari sensor, menampilkan teks atau menampilkan menu pada aplikasi mikrokontroler. LCD M1632 merupakan modul LCD dengan tampilan 16 x 2 baris dengan

konsumsi daya yang rendah. LCD ini terdiri dari 2 bagian, yang pertama merupakan panel LCD sebagai penampil informasi dalam bentuk karakter huruf/angka. Bagian kedua merupakan sebuah sistem yang dibentuk dengan mikrokontroler yang ditempatkan dibalik panel LCD, berfungsi mengatur tampilan informasi serta mengatur komunikasi dengan mikrokontroler yang memakai tampilan LCD tersebut. Untuk berhubungan dengan mikrokontroler pemakai, modul LCD M1632 mempunyai 8 jalur data ( DB0 sampai DB7 ) yang dipakai untuk menyalurkan kode ASCII (*American Standard Code for Information Interchange*) maupun perintah pengatur kerjanya M1632. Selain itu juga terdapat E, R/W dan RS seperti layaknya komponen yang yang kompatibel dengan mikroprosesor. R/W digunakan untuk pemilihan mode *read* R/W=1 atau *write* R/W=0. Sedangkan untuk RS ( *Register Select* ) digunakan untuk membedakan jenis data yang dikirim ke M1632, kalau RS=1 data yang dikirim adalah perintah untuk mengatur kerja M1632 dan sebaliknya apabila RS=0 maka data yang dikirim adalah kode ASCII yang ditampilkan atau *register* data.



Gambar 2.6. LCD (*Liquid Crystal Display*) M1632

Pada modul LCD M1632 terdapat Mikrokontroler HD44780 yang berfungsi sebagai pengendali LCD, mikrokontroler ini mempunyai tiga jenis memori yaitu DDRAM (*Display Data Random Access Memory*), CGRAM

(*Character Generator Random Access Memory*) dan CGROM (*Character Generator Read Only Memory*).

**a. DDRAM**

DDRAM adalah merupakan memori tempat karakter yang ditampilkan berada. Misalkan untuk karakter 'A' atau 41H yang ditulis pada alamat 00, maka karakter tersebut akan tampil pada baris pertama dan kolom pertama dari LCD. Apabila karakter tersebut ditulis di alamat 40, maka karakter tersebut akan tampil pada baris kedua kolom pertama dari LCD.

**b. CGRAM**

CGRAM adalah merupakan memori untuk menggambarkan pola sebuah karakter di mana bentuk dari karakter dapat diubah-ubah sesuai keinginan. Namun memori ini akan hilang saat power supply tidak aktif, sehingga pola karakter akan hilang.

**c. CGROM**

CGROM merupakan memori untuk menggambarkan pola sebuah karakter di mana pola tersebut sudah ditentukan secara permanen dari HD44780 sehingga pengguna tidak dapat mengubah lagi. Namun karena ROM bersifat permanen, maka pola karakter tersebut tidak akan hilang walaupun power supply tidak aktif. Tabel berikut ini adalah pola-pola karakter yang tersimpan dalam lokasi-lokasi tertentu dalam CGROM. Sebagai contoh Pada saat HD44780 akan menampilkan data 41H yang tersimpan pada DDRAM, maka HD44780 akan mengambil data di alamat 41H (0100 0001) yang ada pada CGROM yaitu pola karakter A.

Tabel 2.3. Pola-pola karakter yang tersimpan dalam CGROM

Lower 4 bits \ Upper 4 bits	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	a	P	\	P		-	9	3	0	P
xxxx0001	(2)	!	1	A	Q	a	q	a	7	7	4	ä	q
xxxx0010	(3)	"	2	B	R	b	r	r	イ	ウ	×	β	θ
xxxx0011	(4)	#	3	C	S	c	s	」	ウ	7	E	ε	∞
xxxx0100	(5)	\$	4	D	T	d	t	.	エ	ト	ト	μ	Ω
xxxx0101	(6)	%	5	E	U	e	u	.	ア	ナ	1	σ	Ü
xxxx0110	(7)	&	6	F	V	f	v	ウ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	'	7	G	W	g	w	ア	7	ヌ	ラ	q	π
xxxx1000	(1)	<	8	H	X	h	x	イ	フ	ネ	リ	フ	×
xxxx1001	(2)	>	9	I	Y	i	y	ウ	ク	ル		°	4
xxxx1010	(3)	*	:	J	Z	j	z	エ	コ	ン	レ	i	7
xxxx1011	(4)	+	;	K	[	k	[	ク	サ	ヒ	ロ	×	7
xxxx1100	(5)	,	<	L	¥	l	l	ト	シ	フ	ワ	φ	7
xxxx1101	(6)	-	=	M	]	m	]	ユ	ズ	ハ	ン	±	÷
xxxx1110	(7)	.	>	N	^	n	^	ヨ	セ	ホ	ハ	ñ	
xxxx1111	(8)	/	?	O	_	o	_	ウ	リ	マ	°	ö	■

2.5.1 Pin-Pin LCD M1632

LCD karakter M1632 memiliki 16 kaki dengan fungsi masing-masing.

Berikut ini adalah tabel fungsi dari masing – masing kaki LCD M1632.

Tabel 2.4. Fungsi pin LCD M1632

No	Nama	Keterangan
1	VSS	Tegangan negatif catu daya / <i>ground</i> (0V)
2	VDD	Tegangan positif catu daya
3	V0	Tegangan Kontras LCD
4	RS	Register Select, 0 = Register Perintah, 1 = Register Data
5	R/W	1 = Read, 0 = Write
6	E	Enable, logika 1 setiap kali pengiriman atau pembacaan data
7	D0	Data Bus 0
8	D1	Data Bus 1
9	D2	Data Bus 2
10	D3	Data Bus 3
11	D4	Data Bus 4
12	D5	Data Bus 5
13	D6	Data Bus 6
14	D7	Data Bus 7
15	ANODA	Tegangan positif <i>backlight</i>
16	KATODA	Tegangan negatif <i>backlight</i>

## 2.6 Indikator Peringatan

### 2.6.1 LED (*Light Emitting Diode*)

LED merupakan singkatan dari *Light Emitting Diode*. Dari sisi penggolongan, LED merupakan komponen aktif bipolar semikonduktor, karena

itu hanya mampu mengalirkan arus dalam satu arah saja. Untuk menyalakan LED, cukup dengan mengalirkan arus dari anoda ke katoda (*forward biass*) dengan beda potensial minimum berkisar antara 1,5 hingga 2 volt dan arusnya berkisar di 20mA. Perlu diperhatikan juga bahwa LED juga memiliki tegangan nyala maksimum, jika tegangan tersebut terlewati maka LED akan rusak. Di Pasaran umumnya LED dikemas berkaki dua (*katoda* dan *anoda*) dengan bermacam-macam warna nyala. Untuk membedakan kedua kaki tersebut, kaki *anoda* biasanya dibuat lebih panjang daripada katoda.

LED banyak digunakan untuk indikator dan transmisi sinyal atau bahkan untuk penerangan.

### **Antarmuka LED**

LED dapat menyala pada arus searah (DC) maupun arus bolak-balik (AC), yang membedakan adalah kontinuitas. Pada arus DC LED menyala secara kontinyu. Sedangkan pada arus AC, LED akan menyala secara tidak kontinyu (nyala-padam secara periodik), menyala pada setengah gelombang pertama dan padam pada setengah gelombang berikutnya, hal ini terjadi secara periodik pada frekwensi senilai dengan frekwensi AC yang diterapkan. Hal ini terjadi karena LED hanya mengalirkan arus satu arah saja, sebagai akibatnya LED hanya akan menyala pada fasa dimana LED mendapatkan *forward biass* (hanya setengah gelombang). Mata manusia terkadang terlalu lambat untuk merespon aktifitas nyala-padam tersebut, pada frekwensi tertentu (biasanya 85 Hz atau lebih) LED akan terlihat tetap menyala meskipun faktanya berkedip-kedip.



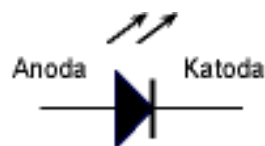
Pada umumnya rangkaian digital menggunakan tegangan operasi 5 s.d 12 volt DC. Karena LED memiliki tegangan maksimum dan tegangan minimum maka arus dan tegangan LED harus diatur sedemikian rupa sehingga berada dalam wilayah yang dapat diterima oleh LED. Tugas ini umumnya dapat diimplementasikan dengan pemasangan *resistor* dan LED secara seri. Prinsipnya adalah bagaimana memilih nilai resistor supaya LED dapat menyala pada tegangan diatas level minimum dan dibawah level maksimum pada tingkatan kecerahan yang dapat diterima. Nilai resistor dapat dihitung dengan rumus:

$$R = \frac{v_s - v_{led}}{I_{led}} \dots \dots \dots (2.3)$$

Dimana:

- R = Resistor
- $v_s$  = tegangan sumber
- $v_{led}$  = tegangan LED
- $I_{led}$  = arus LED

Pada aplikasinya nilai resistor tidaklah se-kritis teorinya, penyimpangan beberapa puluh ohm masih dapat diterima.

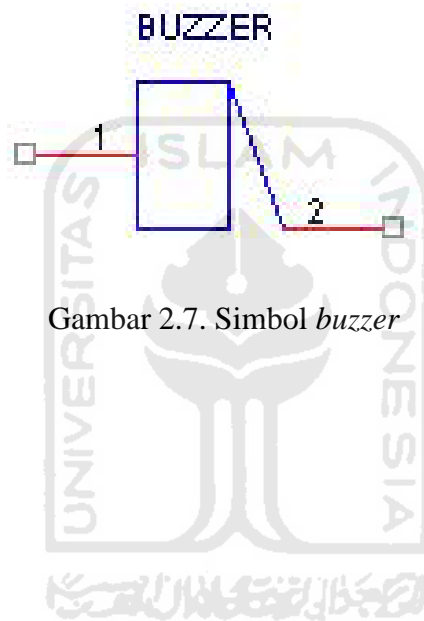


Gambar 2.7. simbol LED

### 2.6.2 Buzzer

*Buzzer* adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan loud speaker, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus

sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator bahwa sedang adanya proses atau terjadi suatu kesalahan pada sebuah alat (alarm).



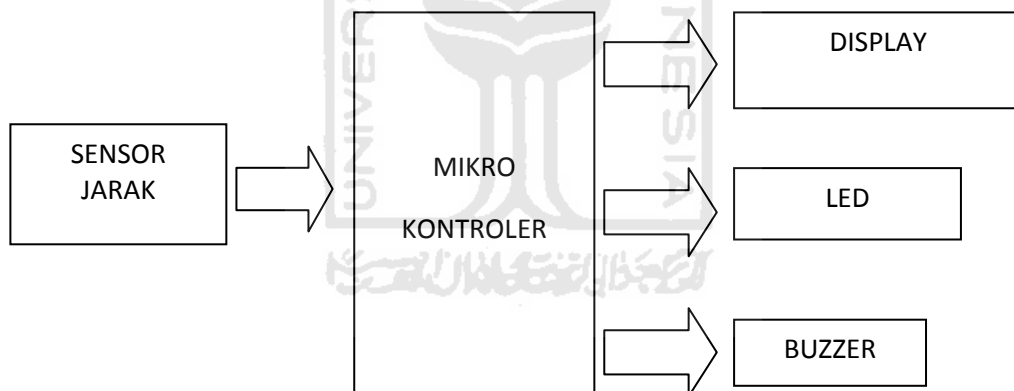
Gambar 2.7. Simbol *buzzer*

## BAB III

### PERANCANGAN SISTEM

#### 3.1 Perancangan Umum sistem

Secara garis besar alat yang akan dibuat terdiri dari tiga bagian yaitu bagian sensor, bagian pengontrol atau pengendali dan bagian penampil & indikator. Berikut ini merupakan diagram blok dari alat “Aplikasi Sensor Ultrasonik Untuk Pengaturan Parkir Kendaraan” yang terdiri dari sensor ultrasonik PING, Mikrokontroler AT89S52, display LCD karakter 16x2, dan indikator peringatan berupa LED dan *buzzer*.



Gambar 3.1. Diagram blok sistem

Prinsip kerja sistem yang akan dibuat dalam penelitian adalah sebagai berikut : Pertama sensor jarak yang berupa sensor ultrasonik akan mengeluarkan gelombang ultrasonik kurang lebih 40 Khz, apabila gelombang tersebut mengenai suatu penghalang / benda maka akan dipantulkan yang kemudian diterimanya kembali. Lama waktu dari pengiriman gelombang ultrasonik sampai diterimanya

kembali inilah yang akan menjadi data masukan untuk diproses pada mikrokontroler, selanjutnya mikrokontroler memproses data dari sensor untuk dapat ditampilkan pada *display* (LCD) berupa jarak dalam satuan centimeter, disamping ditampilkan pada *display*, mikrokontroler juga mengolah data untuk diteruskan pada indikator peringatan berupa nyala lampu LED yang berbeda untuk jarak tertentu dan bunyi *beep* pada *buzzer* dengan interval yang semakin rapat apabila jarak dengan penghalang semakin dekat.

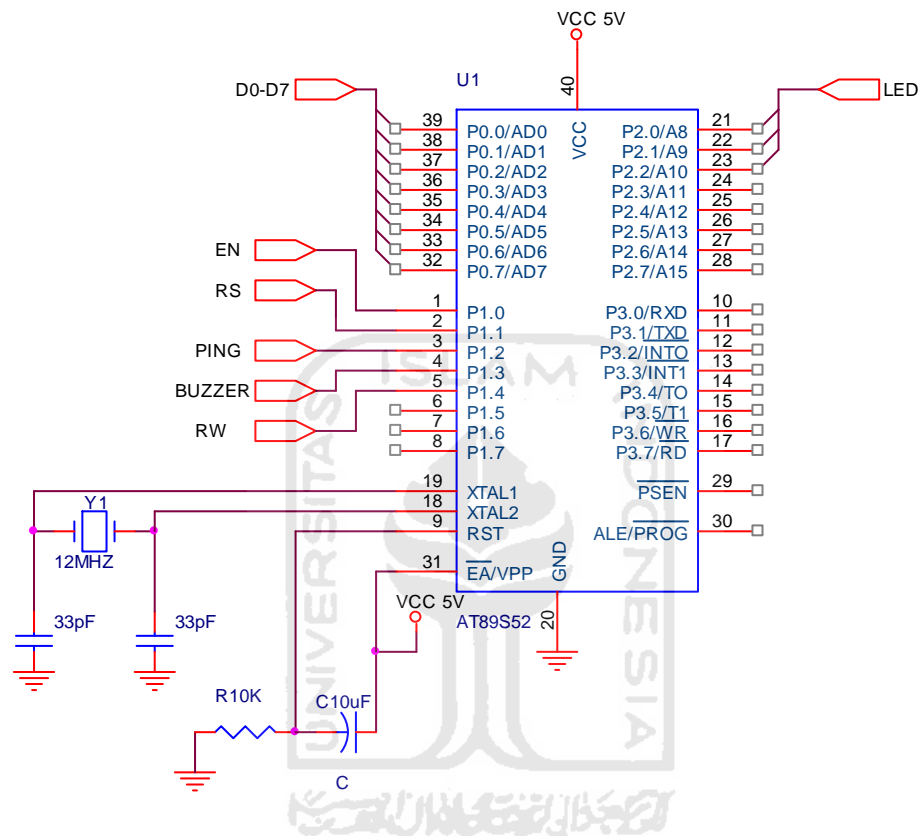
### **3.2 Perancangan Perangkat Keras**

pada sub bab ini akan dijelaskan tentang bagian dari perangkat keras yang digunakan dalam perancangan sistem ini. Perangkat keras ini merupakan penjabaran dari diagram blok yang saling berhubungan satu sama lain.

#### **3.2.1 Sistem minimum mikrokontroler AT89S52**

Mikrokontroler AT89S52 mempunyai empat buah port yang dapat digunakan sebagai masukan dan keluaran. Sebelum menggunakan IC Mikrokontroler AT89S52 ini langkah yang harus dipersiapkan adalah membuat rangkaian sistem minimum AT89S52. Setelah mendapatkan sebuah rangkaian sistem minimum yang lengkap, sistem minimum AT89S52 ini akan dioperasikan sebagai unit pengontrol rangkaian secara keseluruhan. Rangkaian ini hanya terdiri atas *single chip* mikrokontroler, sebuah osilator dan dua buah kapasitor yang berfungsi untuk menstabilkan frekuensi. Mikrokontroler ini memiliki osilator *on-chip* yang dapat digunakan sebagai sumber detak (*clock*) ke CPU. Untuk mengaktifkannya harus dipasang sebuah *resonator* (kristal) diantara kaki-kaki X1 dan X2 pada mikrokontroler dan dua buah kapasitor yang dihubungkan ke

ground. Dibutuhkan juga sebuah resistor dan sebuah kapasitor untuk mereset sebelum mikrokontroler menjalankan suatu program. Berikut ini merupakan skema rangkaian sistem minimum mikrokontroler AT89S52.

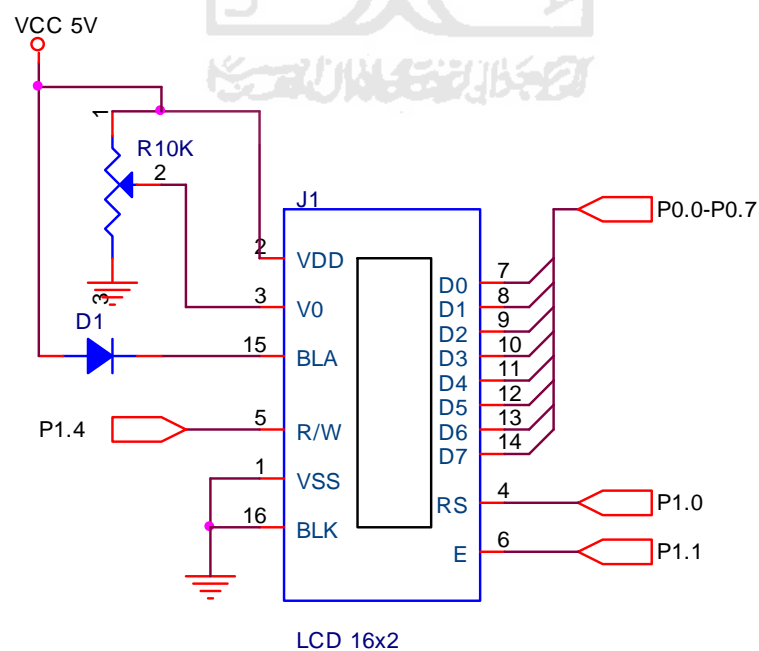


Gambar 3.2. Skema rangkaian sistem minimum mikrokontroler AT89S52

### 3.2.2 Rangkaian LCD (*Liquid Crystal Display*)

LCD (*Liquid Crystal Display*) adalah salah satu jenis tampilan yang dapat digunakan untuk menampilkan karakter angka, huruf, dan simbol-simbol lainnya. Hal ini karena modul ini menggunakan titik yang berbentuk matriks untuk menampilkan suatu karakter, sehingga dengan LCD dapat ditampilkan lebih banyak bentuk karakter. Pada perancangan ini digunakan modul LCD 16x2 M1632 yang dapat menampung 16 karakter dalam tiap baris.

Pada modul LCD sudah dilengkapi dengan mikrokontroler khusus untuk mengendalikan LCD. Untuk menghubungkan dengan mikrokontroler telah di persiapkan pin-pin pada modul LCD yang secara *kompatibel* dapat langsung dihubungkan dengan *port-port* Mikrokontroler tanpa membutuhkan IC perantara lainnya sehingga antar muka komponen menjadi sederhana. Proses transfer data tampilan diatur oleh Mikrokontroler AT89S52. Agar LCD dapat bekerja seperti yang diinginkan perlu adanya setingan awal untuk menjalankan LCD yaitu dengan menghubungkan pin 5 (R/W) ke port P1.4 yang akan digunakan sebagai mode *read* atau *write*. Disini mode *read* (R/W=1) hanya dipakai untuk membaca *busy flag*. Pin 2 (VDD) terhubung dengan catu daya 5 volt, pin 1 (VSS) terhubung dengan *ground* sedangkan pin 3 (V0) terhubung dengan sebuah resistor *variable* untuk mengatur kontras LCD. Skema rangkaian untuk LCD ditunjukkan pada gambar 3.3.



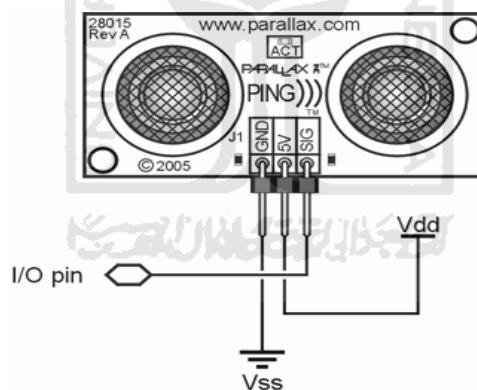
Gambar 3.3. Skema Rangkaian LCD (*Liquid Crystal Display*)

### 3.2.3 Instalasi Sensor Ultrasonik PING

Sensor ultrasonik memiliki prinsip kerja mendeteksi objek dengan cara mengirimkan suara ultrasonik dan kemudian “mendengarkan” pantulan suara tersebut sebagai dasar pengindranya.

Dalam perancangan alat ini digunakan PING buatan Parallax.Inc, karena sensor tersebut sudah dipabrikasi dan dikemas dengan baik, sehingga dapat mengurangi interferensi sinyal yang dipancarkan dan diterima. Selain itu sensor tersebut juga cukup mudah didapatkan dan mudah untuk disambungkan dengan mikrokontroler, karena hanya menggunakan 1 pin input/output (pin SIG) sebagai pemicu dan penerima informasi sinyal pantulan yang dihubungkan ke mikrokontroler.

Berikut merupakan konfigurasi pin sensor ultrasonik PING



Gambar 3.4. Konfigurasi pin sensor ultrasonik PING

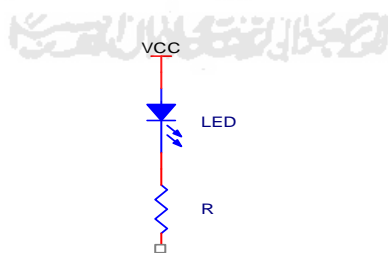
Gambar blok rangkaian pemancar dan penerima gelombang ultrasonik tidak dibahas secara *detail*, karena rangkaian tersebut sudah merupakan suatu kesatuan dari hasil pabrikasi. Konfigurasi pin sensor ping seperti terlihat pada gambar 3.4. Sensor ini memiliki 3 pin, yang masing-masingnya dihubungkan ke Ground, Vcc (5V) dan pin ketiga merupakan pin I/O (SIG) dihubungkan ke port P1.2 mikrokontroler.

### 3.2.4 Pemasangan LED Dan Buzzer

#### a. LED

LED merupakan komponen aktif *bipolar* semikonduktor, karena itu hanya mampu mengalirkan arus dalam satu arah saja. Untuk menyalakan LED, cukup dengan mengalirkan arus dari anoda ke katoda (*forward biass*) dengan beda potensial minimum berkisar antara 1,5 hingga 2 volt dan arusnya berkisar di 20mA. Dalam perancangan ini, LED difungsikan sebagai indikator peringatan dengan nyala yang berbeda untuk *range* jarak tertentu.

Pada umumnya rangkaian digital menggunakan tegangan operasi 5 s.d 12 volt DC. Karena LED memiliki tegangan maksimum dan tegangan minimum maka arus dan tegangan LED harus diatur sedemikian rupa sehingga berada dalam wilayah yang dapat diterima oleh LED. Tugas ini umumnya dapat diimplementasikan dengan pemasangan *resistor* dan LED secara seri. Pada perancangan LED dihubungkan dengan mikrokontroler pada port P2.0-P.2.2



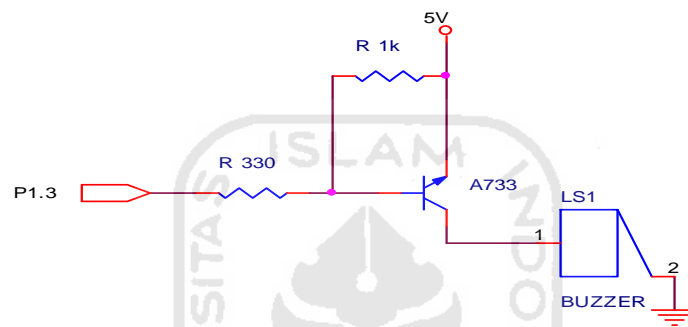
Gambar 3.5. Skema rangkaian LED

#### b. Buzzer

*Buzzer* pada perancangan ini juga digunakan sebagai indikator peringatan. Rangkaian *buzzer* terdiri dari sebuah transistor dan resistor. Pada perancangan ini *buzzer* dikendalikan oleh port P1.3 mikrokontroler, transistor berfungsi sebagai *driver* dikarenakan *buzzer* membutuhkan arus yang cukup besar untuk



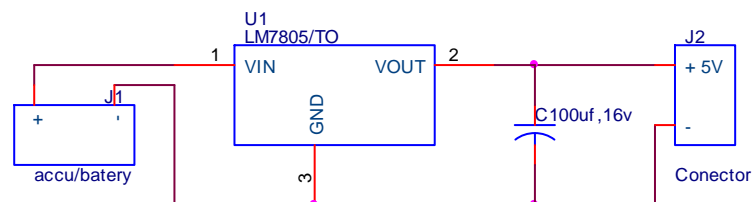
mengaktifkannya. Selain itu *buzzer* juga dimanfaatkan untuk mengira ngira jarak yang diukur. ini fungsinya jika mobil akan diparkir maka pengemudi tidak harus melihat display. Tetapi bisa sambil menengok keluar/kebelakang.dengan acuan suara *buzzer*. Dalam sistem ini *buzzer* diseting aktif dengan interval tertentu. Jika jarak makin dekat maka intervalnya semakin cepat. Dan jika jarak makin jauh maka interval makin lambat.



Gambar 3.6. Skema rangkaian *buzzer*

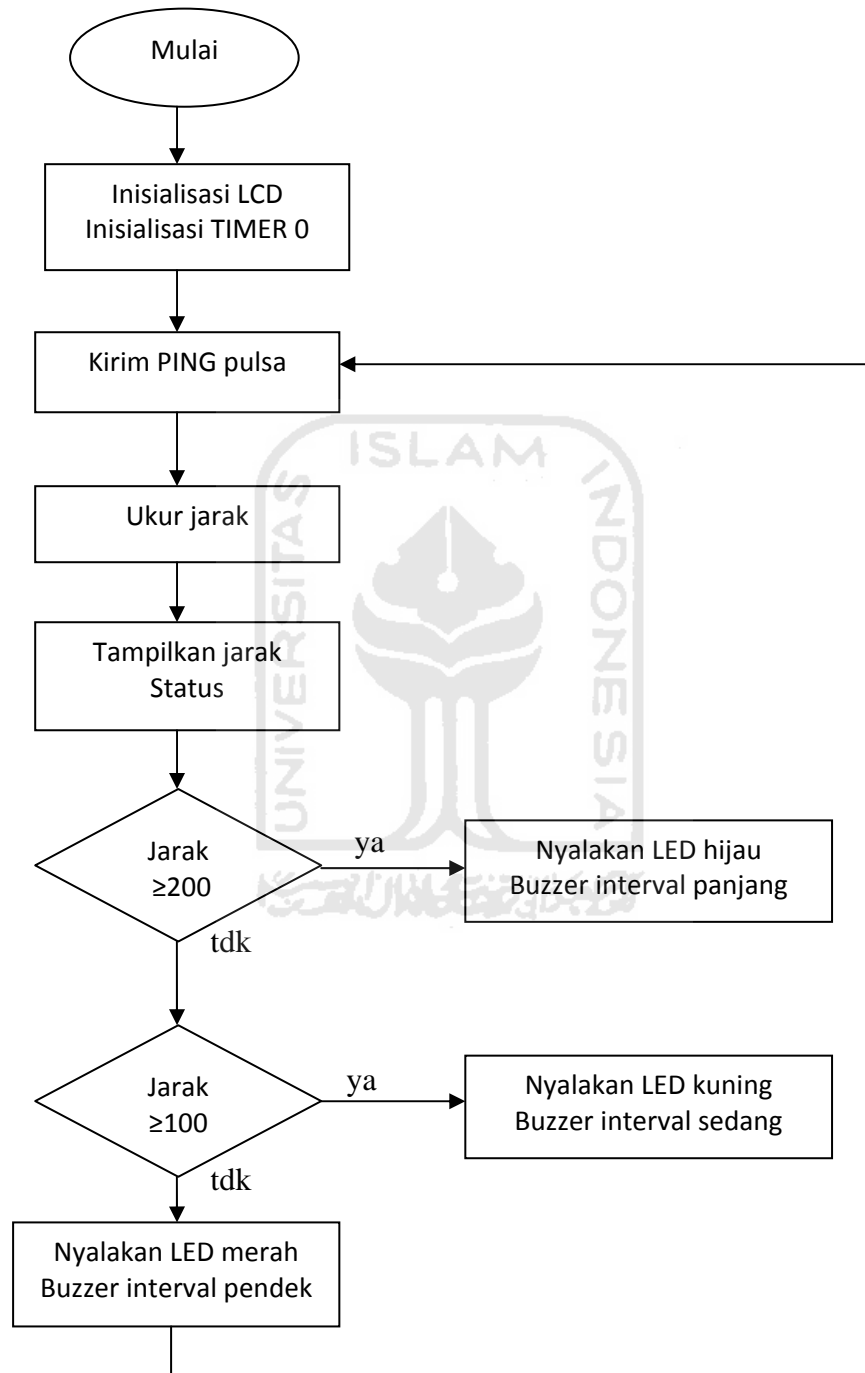
### 3.2.5 Catu Daya

Catu daya berfungsi untuk menghidupkan keseluruhan sistem. Karena aplikasinya digunakan pada kendaraan sumber tegangan bisa langsung dihubungkan dengan *accu* dengan menurunkan tegangan menjadi 5 volt DC. untuk itu diperlukan sebuah IC regulator LM7805 untuk menurunkan tegangan dan sebuah kapasitor untuk menstabilkan tegangan.



Gambar 3.7. Skema rangkaian catu daya

### 3.3 Perancangan Perangkat Lunak



Gambar 3.8. Flowchart program

Perancangan perangkat lunak (*software*) diperlukan agar sistem yang direncanakan dapat bekerja dengan baik. Sebelum membuat program, pertama kali yang harus diketahui adalah algoritma atau cara kerja dari alat tersebut. Biasanya diawali dengan pembuatan *flowchart* untuk memandu program yang akan ditulis, seperti yang ditunjukkan pada gambar 3.7., selanjutnya penulisan program pada editor. Editor yang digunakan disini adalah *notepad*. Jika program di *notepad* sudah jadi maka dilanjutkan dengan proses kompilasi. Disini *compiler* yang digunakan adalah asm 51. Jadi program assembly yang sudah selesai disimpan dalam ekstensi \*.asm. dalam *project* ini file diberi nama parkir1.asm. setelah itu file parkir1.asm di kompilasi menggunakan asm51 dengan cara file tersebut di drag ke file asm51. Setelah dikompilasi maka file parkir1.asm akan degenerate oleh *compiler* menghasilkan file parkir1.lst dan file parkir1.obj. file obj inilah yang kemudian di generate lagi menjadi file hex menggunakan file oh. Setelah kode hexa terbentuk maka file ini siap di upload ke mikrokontroller untuk kemudian dijalankan oleh mikro sesuai dengan algoritma dari pemrograman yang diinginkan.

### **3.4 Pemograman**

Pemograman yang dijelaskan disini adalah cara-cara untuk membuat progam secara modular.

#### **3.4.1 Inisialisasi LCD**

Inisialisasi LCD adalah pengiriman kode kode sesuai dengan datasheet dari pabrikan pembuat LCD tersebut. Untuk kode yang dikirimkan bisa dilihat dari kode sumber yang terlampir pada halaman lampiran. Untuk inisialisasinya

yaitu dengan cara pemanggilan sub rutin LCD tersebut. Dalam hal ini contohnya “*acall lcd\_init*” dan rutin ini juga bisa digunakan untuk semua merk LCD apapun, asalkan typenya sama. Misalnya dalam hal ini typenya adalah 16 kolom x 2 baris (16x2). Pada alat ini konfigurasi LCD menggunakan format pengiriman 8 bit. Meskipun dalam beberapa kasus bisa menggunakan format 4 bit. Setelah di inisialisasi maka LCD siap menerima data dalam bentuk *byte*. Data *byte* yang dikirim disini adalah format ASCII.

Cara menuliskan data ke LCD yaitu sebagai berikut:

```
Mov      lcd_position,#line2
```

```
Mov      a,#'1'
```

```
Acall    byte_display
```

Contoh diatas adalah cara penulisan angka 1 di baris 2. Jadi posisi cursor/ tempat yang akan ditulis ditunjuk oleh *variable* LCD position kemudian data dimasukan ke akumulator harus dalam format ASCII kemudian baru dipanggil sub rutin *byte\_display*.

### 3.4.2 Pemograman Sensor PING

Setelah pemrograman LCD selesai kemudian dilanjutkan dengan pemrograman sensor PING. Untuk mengukur jarak yaitu dengan cara mengetahui dari cara kerja alat tersebut. Ini bisa didapatkan dari datasheet sensor PING yang dikeluarkan oleh *parallax*. Dari datasheet tersebut maka algoritmanya didapatkan, kemudian diprogram sesuai algoritma. Dalam pengukuran jarak disini menggunakan *interup* dari timer 0 mode 2 *autoreload*. Nilai yang diisikan untuk isi ulangannya yaitu langsung nilai untuk setiap kenaikan 1 centi meternya. Karena

cara kerja PING dengan prinsip pantulan, maka nilai autoreload menjadi 2 kali nilai per centi meternya. Dalam perhitungan dari datasheet ditentukan bahwa untuk setiap 1cm cepat rambat gelombang ultrasonik membutuhkan waktu 29 uS maka untuk 1 pengukuran dibutuhkan 2 kali jarak aslinya. Maka didapat nilai waktunya adalah 58 uS. Nilai inilah yang kemudian diisikan kedalam TH0 dimana setiap 58 uS akan terjadi interupsi yang akan menambah 1 (*increment* jarak). Nilai yang ada dalam *variable* jarak ini sudah mewakili jarak sesungguhnya dalam satuan centimeter, dikarenakan kenaikan satuannya setiap 58uS. Setelah pemrograman ping selesai kemudian diuji cobakan ke alat yang dalam hal ini hanya menampilkan perhitungan jarak saja.

### 3.4.3 Pemograman LED dan *Buzzer*

Pemrograman indikator LED yaitu dengan cara membandingkan data jarak terukur dengan konstanta 200 dan 100. Jika jarak yang terukur diatas 200 maka hanya dinyalakan led hijau. Dalam hal ini statusnya adalah “AMAN”. Kumudian jika nilainya kurang dari 200 dan diatas 100 maka hanya dinyalakan led kuning sebagai tanda dari status “HATI-HATI” dan jika tidak masuk dalam kedua klausul diatas maka ditetapkan nilainya pasti kurang dari 100, maka akan dinyalakan hanya led merah yang artinya “STOP”. Setelah pemrograman LED ini selesai maka dilanjutkan dengan pemrograman *buzzer*. Disini *buzzer* diprogram dengan interval sebanding dengan jarak yang terukur. Jika jaraknya jauh maka intervalnya lama, dan jika jaraknya semakin dekat intervalnya juga semakin cepat. Untuk pengaturan interval, dimana jarak yang terukur akan digunakan sebagai faktor kali dalam intervalnya. Dalam hal ini pengaktifan *buffer* menggunakan *interup* timer 1

mode 1 (16 bit). Isi register timer di set 3922. Jika jarak terukur adalah 255cm maka interval dari buzzer adalah  $255 \times 3922 \text{ uS} = 1000110 \text{ uS}$  atau dalam pembulatan 1 detik. Jadi jika jarak yg terukur adalah 255 cm maka interval dari bunyi *buzzer* adalah 1 detik.



## BAB IV

### PENGUJIAN DAN ANALISIS

Untuk mengetahui apakah hasil perancangan alat yang telah dibuat dapat bekerja sesuai dengan yang yang diinginkan maka dilakukan pengujian. Pada bab ini dijelaskan tentang pegujian kerja sistem alat sensor ultrasonik untuk pengaturan parkir kendaraan dan analisa data yang diperoleh.

#### 4.1. Pengujian Alat

Pengujian alat disini merupakan pengujian terhadap kinerja alat secara keseluruhan. Pengujian terhadap sensor jarak dilakukan dengan cara membandingkan antara jarak sesungguhnya yang diukur dengan meteran manual dengan jarak yang tertampil pada alat, dalam hal ini dengan mendekatkan dan menjauhkan alat dengan penghalang. Berikut ini adalah data hasil pengukuran jarak dengan interval per sepuluh centimeter.

Tabel 4.1. Data hasil pengukuran jarak.

No	JARAK REAL (meteran)	JARAK TERTAMPIL (Alat)	NYALA LED	SELISIH	ERROR %
1	10 cm	9 cm	M	1 cm	10 %
2	20 cm	19 cm	M	1 cm	5 %
3	30 cm	30 cm	M	0 cm	0 %
4	40 cm	39 cm	M	1 cm	2,5 %
5	50 cm	49 cm	M	1 cm	2 %
6	60 cm	60 cm	M	0 cm	0 %
7	70 cm	69 cm	M	1 cm	1,43 %

Lanjutan Tabel 4.1. Data hasil pengukuran jarak.

No	JARAK REAL (Meteran)	JARAK TERTAMPIL (Alat)	NYALA LED	SELISIH	ERROR %
8	80 cm	80 cm	M	0 cm	0 %
9	90 cm	90 cm	M	0 cm	0 %
10	100 cm	100 cm	K	0 cm	0 %
11	110 cm	109 cm	K	1 cm	0,91 %
12	120 cm	120 cm	K	0 cm	0 %
13	130 cm	130 cm	K	0 cm	0 %
14	140 cm	140 cm	K	0 cm	0 %
15	150 cm	150 cm	K	0 cm	0 %
16	160 cm	160 cm	K	0 cm	0 %
17	170 cm	170 cm	K	0 cm	0 %
18	180 cm	180 cm	K	0 cm	0 %
19	190 cm	190 cm	K	0 cm	0 %
20	200 cm	201 cm	H	1 cm	0,5 %
21	210 cm	210 cm	H	0 cm	0 %
22	220 cm	220 cm	H	0 cm	0 %
23	230 cm	230 cm	H	0 cm	0 %
24	240 cm	240 cm	H	0 cm	0 %
25	250 cm	250 cm	H	0 cm	0 %

Keterangan : M = Merah

K = Kuning

H = Hijau

Untuk nilai persentase error adalah :

$$\% \text{ error} = \frac{\text{jarak real} - \text{jarak tertampil}}{\text{jarak real}} \times 100 \%$$



$$\begin{aligned} \text{rata - rata error} &= \frac{\text{total error}}{\text{banyaknya data}} \\ &= \frac{22,34}{25} \% = 0,89 \% \end{aligned}$$

Pengujian indikator LED dilakukan dengan mengukur tegangan pada kaki katoda yang terhubung dengan mikrokontroler pada *port* P2.0-P2.2 pada jarak 0 sampai dengan 255 centimeter.

Tabel 4.2. Pengukuran LED

Jarak (Cm)	LED	P2.0	P2.1	P2.2
≥200	Hijau	5 volt	5 volt	0 volt
100-199	Kuning	5 volt	0 volt	5 volt
0-99	Merah	0 volt	5 volt	5 volt

Dari tabel diatas dapat diketahui bahwa led menyala pada saat katoda yang terhubung ke port mikrokontroler *low* (0 volt) dan mati pada saat high (5 volt) Karena led akan menyala jika ada arus yang mengalir dari anoda ke katoda, dan disini anoda terhubung dengan catu daya 5 volt.

Pengujian *buzzer* didapatkan hasil bahwa *buzzer* akan terus berbunyi dengan interval sesuai dengan jarak yang terukur. Jika jarak makin jauh maka intervalnya makin lama dan jarak semakin dekat intervalnya semakin pendek atau dengan kata lain bunyi *beep* akan semakin rapat. Interval *buzzer* berbanding dengan jarak yang terukur, tetapi suara dari *buzzer* tidak dipakai untuk mengitung jarak yang nyata dan hanya sebagai indikator saja untuk memperkirakan jarak yang terukur

## 4.2 Analisis

Dari hasil pengujian sensor jarak yang didapat bahwa *error* terjadi hanya selisih 1 cm dan itu hasilnya sama untuk setiap pengukuran, artinya *error* tidak ikut bertambah besar ketika alat digunakan untuk mengukur jarak yang lebih besar. Pada umumnya kesalahan terjadi akibat kondisi lingkungan objek yang diukur. Satu hal yang perlu diperhatikan bahwa sensor PING tidak bisa mengukur obyek yang permukaannya menyerap suara seperti busa atau peredam lainnya sehingga pantulan tidak diterima sempurna. Karena alat ini aplikasinya dipakai dalam pengukuran jarak yang fungsinya mengetahui jarak dari kendaraan terhadap benda dibelakang, maka toleransi 1 cm sudah dianggap cukup mengingat alat tersebut sudah dilengkapi *buzzer* dan lampu led, dimana jarak dibawah 100 cm LED sudah menyala merah sebagai indikasi untuk segera menghentikan kendaraan.

Untuk indikator LED nyala lampu sudah sesuai dengan jarak yang diinginkan yaitu jarak aman diwakili oleh warna hijau, jarak hati-hati diwakili oleh warna kuning dan jarak stop diwakili oleh warna merah. Secara garis besar nilai jarak yang digunakan untuk menyalakan led bisa disesuaikan sesuai kebutuhan.

Indikator *buzzer* sangat membantu pengemudi terutama ketika mengarahkan mobil dimana ia harus melihat kaca spion atau menengok ke sekeliling sehingga pengemudi tidak perlu melihat display dari alat ukur tersebut. Dengan adanya interval panjang dan pendek maka pengemudi bisa memperkirakan jarak mobil terhadap rintangan yang ada.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan pada pengujian dan analisa terhadap perancangan *hardware* aplikasi sensor ultrasonik untuk pengaturan parkir kendaraan maka dapat ditarik kesimpulan:

1. Nilai yang didapat dari pengukuran jarak ada error yang besarnya maksimal 1 cm. Hal ini wajar mengingat nilai pengali yang digunakan untuk mengukur jarak, dalam hal ini nilai timer yang fungsinya untuk mengukur lebar pulsa sudah ditentukan dalam satuan centimeter.
2. Modul sensor ultrasonik PING bekerja berdasarkan prinsip pantulan untuk dasar penginderanya, hasil pengukuran akan kurang maksimal bila mengenai bidang pantul yang dapat menyerap suara atau sudut yang tajam.
3. Secara umum alat ini sudah bisa diaplikasikan dalam kendaraan sebagai alat bantu untuk memarkir kendaraan, dengan tingkat kesalahan  $\pm 0,89 \%$ . Tetapi tidak bisa dijadikan sebagai jaminan dalam keamanan memarkir kendaraan, kehati-hatian pengemudi tetap hal yang paling utama.

#### **5.2 Saran**

Untuk pengembangan lebih lanjut yang lebih baik tentang alat bantu untuk pengaturan parkir kendaraan Perlu adanya uji coba penambahan indikator misalnya kamera yang dipasang bersama sensor sehingga lebih memudahkan pengemudi. perlu juga dilakukan uji coba dengan memasang sensor pada

beberapa titik tidak hanya pada sisi belakang kendaraan tetapi juga bisa di depan dan juga di sisi samping kendaraan. Semakin banyak sensor akan semakin mewakili kondisi disekitar kendaraan, yang mungkin juga bisa diaplikasikan untuk alarm keamanan dengan pendeteksian *moving object*.



```

rw_lcd      bit    p1.4
rs          bit    p1.1
en         bit    p1.0
buzer      bit    p1.3
update     bit    20h.0

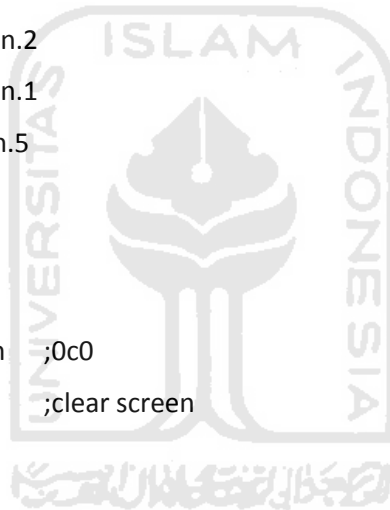
t2con      equ    0c8h
tl2        equ    0cch
th2        equ    0cdh
tf2        bit    t2con.7
tr2        bit    t2con.2
ct_2       bit    t2con.1
et5        bit    0a8h.5

;constan
line1      equ    80h
line2      equ    0c0h ;0c0
blank     equ    01h ;clear screen

count_t    equ    25h
reg_lo     equ    26h
reg_hi     equ    27h
count_bit  equ    28h
sum_of_dat equ    29h

lcd_io     equ    2ah ;reg temp for lcd data
lcd_position equ    2bh ;reg for position display
sum_of_char equ    2ch ;sum of data will be displayed
portlcd   equ    p0

```



```
jarak      equ    2dh
counter    equ    2eh
interval   equ    2fh
```

```
red        bit    p2.0
yellow     bit    p2.1
green      bit    p2.2
```

```
sig        bit    p1.2
```

```
org    00h
mov    sp,#80h
jmp    start
```

```
org    0bh
jmp    int_tmr0
```

```
org    1bh
push   acc
push   psw
push   count_t
clr    tr1
mov    a,interval
clr    c
subb   a,#10
jc     beepNquit
dec    interval
jmp    outsj
```

```
beepnquit:
```



```

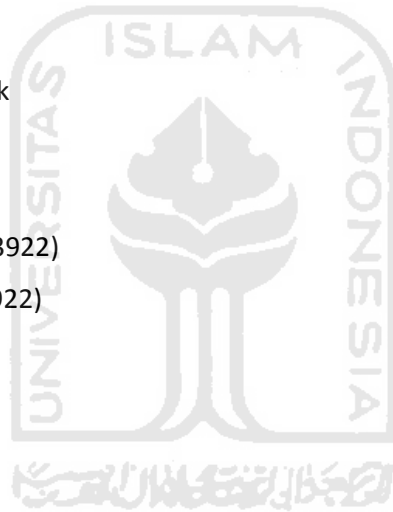
        clr    buzer
        clr    update
        push  6
        push  7
        mov   r6,#100
lpp:
        mov   r7,#0
        djnz  r7,$
        djnz  r6,lpp
        setb  buzer
        pop   7
        pop   6
        mov   interval,jarak

outsj:
        mov   th1,#high (-3922)
        mov   tl1,#low (-3922)
        pop   count_t
        pop   psw
        pop   acc
        setb  tr1
        reti

int_tmr0:
        clr   tr0
        mov   a,jarak
        cjne a,#255,lanjut
        setb  tr0
        reti

lanjut:
        inc   jarak
        setb  tr0

```



```
reti ;otomatis th0 mengisi t10
```

```
;-----
```

```
;lcd initialized
```

```
;-----
```

```
lcd_init:
```

```
clr rw_lcd
```

```
clr rs
```

```
call delay20ms
```

```
mov lcd_io,#03Fh ;busy flag can't be check before this
```

```
call run_inst
```

```
call delay5ms
```

```
call run_inst
```

```
call delay100us
```

```
call run_inst
```

```
mov lcd_io,#0ch
```

```
call write_inst
```

```
mov lcd_io,#06h
```

```
call write_inst
```

```
ret
```



```
;-----
```

```
;lcd mode and write
```

```
;-----
```

```
check_bsf:
```

```
push acc
```

```
clr rs
```

```
setb rw_lcd ;read mode for bsf
```

```
bsfloop:
```

```
setb en
```

```
nop
```



```
nop
nop
mov a,portlcd
clr en
jb acc.7,bsfloop
pop acc
clr rw_lcd
ret
```

write\_inst:

```
call check_bsf
clr rs
mov portlcd,lcd_io
setb en
nop
nop
nop
clr en
ret
```

write\_data:

```
call check_bsf
setb rs
mov portlcd,lcd_io
setb en
nop
nop
nop
clr en
call check_bsf
ret
```

rd\_lcd\_data:



```

    mov    p0,#0ffh
    setb   rw_lcd
    setb   rs
    setb   en
    nop
    nop
    nop
    mov    a,p0
    clr    en
    clr    rw_lcd
    call   delay_r7
    ret

;-----
;clr all screen (return home)
;-----
clrscr:
    ;-----<clear all screen
    mov    lcd_io,#blank
    call   run_inst
    call   run_inst
    ret

run_inst:
    clr    rs
    mov    portlcd,lcd_io
    setb   en
    nop
    nop
    nop
    clr    en
    mov    count_t,#1
    mov    reg_lo,#low (-12750)
    mov    reg_hi,#high (-12750)
    call   delay_var

```



```

ret

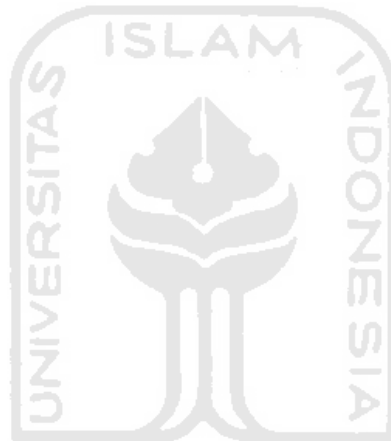
erase_line1:
    mov    lcd_position,#line1
    jmp    erase

erase_line2:
    mov    lcd_position,#line2
erase:
    call   align_position
    push  b
    mov   b,#16
    mov   lcd_io,#' '
ers_nxt:
    call  write_data
    djnz b,ers_nxt
    pop  b
    ret

;-----
;display data in prog memory=>dptr
;-----
display_buffer:
    call  align_position
dspnxt:
    mov  a,@r1
    add  a,#30h
    call char_display
    inc  r1
    djnz sum_of_char,dspnxt
    ret

dptr_display:

```



UNIVERSITAS ISLAM INDONESIA

```

        call    align_position
nxt_char:
        clr     a
        movc   a,@a+dptr
        mov    lcd_io,a
        call   write_data
        inc    dptr
        djnz   sum_of_char,nxt_char
        ret

```

```

;-----

```

```

;set position for display

```

```

;-----

```

```

align_position:

```

```

        mov    lcd_io,lcd_position
        call   write_inst
        call   write_inst
        ret

```

```

;-----

```

```

;display single byte in acc

```

```

;-----

```

```

byte_display:

```

```

        call   align_position
        mov    lcd_io,a
        call   write_data
        inc    lcd_position
        ret

```

```

;-----

```

```

;manual char display

```

```

;-----

```

```

char_display:

```

```

        mov    lcd_io,a
        call   write_data
        ret

```



```
;-----  
;delay rutin for general purpose
```

```
;-----
```

```
delay100us:
```

```
    mov    r7,#100  
    djnz  r7,$  
    ret
```

```
delay20ms:
```

```
    mov    reg_lo,#low (-20000)  
    mov    reg_hi,#high (-20000)  
    mov    count_t,#1  
    call   delay_var  
    ret
```

```
delay100ms:
```

```
    mov    reg_lo,#low (-20000)  
    mov    reg_hi,#high (-20000)  
    mov    count_t,#5  
    call   delay_var  
    ret
```

```
delay5ms:
```

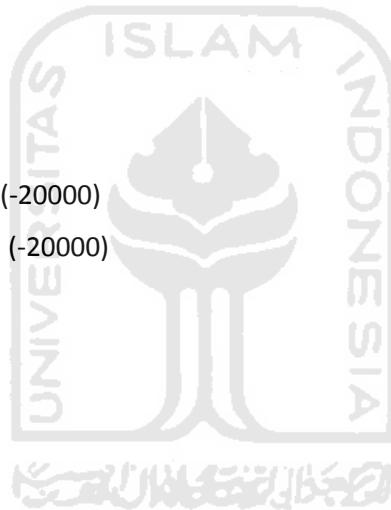
```
    mov    reg_lo,#low (-5000)  
    mov    reg_hi,#high (-5000)  
    mov    count_t,#1  
    call   delay_var  
    ret
```

```
delay_r7:
```

```
    mov    r7,#0ffh  
    djnz  r7,$  
    ret
```

```
delay05s:
```

```
    mov    reg_lo,#low (-50000)
```



```
    mov    reg_hi,#high (-50000)
    mov    count_t,#10
    call   delay_var
    ret
```

delay1s:

```
    mov    reg_lo,#low (-50000)
    mov    reg_hi,#high (-50000)
    mov    count_t,#20
    call   delay_var
    ret
```

delay\_kypd:

```
    mov    reg_lo,#low (-15000)
    mov    reg_hi,#high (-15000)
    mov    count_t,#20
    call   delay_var
    ret
```

fast\_delay:

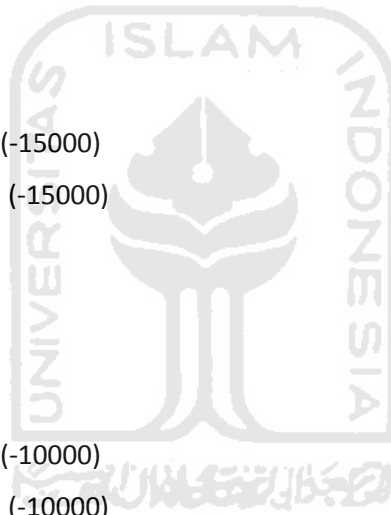
```
    mov    reg_lo,#low (-10000)
    mov    reg_hi,#high (-10000)
    mov    count_t,#20
    call   delay_var
    ret
```

delay\_var:

```
    mov    t2con,#0
    clr    tf2
    clr    tf2
```

lgNx:

```
    mov    t12,reg_lo
    mov    th2,reg_hi
```



```
setb tr2
jnb tf2,$
clr tr2
clr tf2
djnz count_t,IgNx
ret
```

timer\_config:

```
mov tmod,#12h
clr tr0
clr tf0
setb ea
setb et0
mov tl0,#-52 ;nilai satu detik crystal 12mhz
mov th0,#-52 ;awal 56,55,53
mov th1,#high (-3922)
mov tl1,#low (-3922)
clr tr1
clr tf1
setb et1
ret
```

delay200us:

```
mov r7,#200
djnz r7,$
ret
```

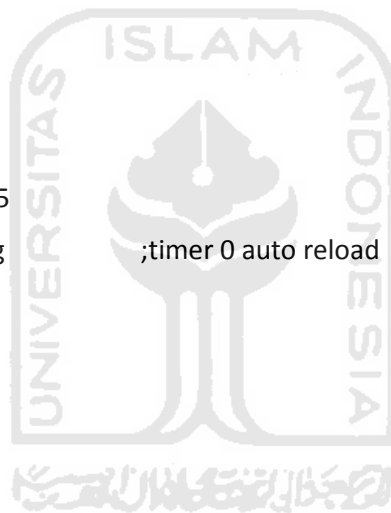
kirim\_sig\_pulse:

```
setb sig
nop
nop
nop
```

```
    nop
    nop
    clr    sig
    ret
```

START:

```
    clr    sig
    setb   update
    setb   green
    setb   yellow
    setb   red
    mov    jarak,#0
    mov    interval,#255
    call   timer_config ;timer 0 auto reload
    call   lcd_init
    setb   buzzer
    call   clrscr
    call   D_template
```



loop:

```
    clr    sig
    acall  delay200us
    clr    tr1
    acall  kirim_sig_pulse ;_|^|_
    acall  delay200us
    setb   sig
    mov    r7,#200
    djnz  r7,$

    jnb   sig,$
```



```
setb tr0 ;awal pengukuran
jb sig,$
clr tr0 ;stop pengukuran
clr sig
```

```
setb tr1
mov a,jarak
acall byte2asc
mov lcd_position,#line1+10
```

```
mov a,r3
add a,#30h
acall

mov a,r2
add a,#30h
acall byte_display

mov a,r1
add a,#30h
acall byte_display
```



```
jb update,skip_bel
mov interval,jarak
setb update
```

skip\_bel:

```
acall delay100ms
acall delay100ms
acall delay100ms
```

```
clr c
mov a,jarak
subb a,#200
```

```

jnc    jarak_aman

clr    c
mov    a,jarak
subb   a,#100
jnc    carefull

stop:
jnb    red,quit1
setb   green
setb   yellow
clr    red
call   d_stop
quit1:
jmp    end1

jarak_aman:
jnb    green,end1
clr    green
setb   yellow
setb   red
call   d_aman

end1:
mov    jarak,#0
jmp    loop

carefull:
jnb    yellow,quit3
setb   green
clr    yellow

```



```

        setb    red
        call   d_hati_hati
quit3:
        jmp    end1

;inp    :acc
;out    :r1,r2,r3    (1,2,3)lsb to msb
;-----
byte2asc:
        mov    r1,#0
        mov    r2,#0
        mov    r3,#0
        mov    r0,#1
        jz     convert_finish
loop_convert:
        push  acc
        mov   b,#10
        clr   c
        subb  a,b
        pop   acc
        jc   convert_finish
        div  ab
        mov  @r0,b
        inc  r0
        jmp  loop_convert
convert_finish:
        mov  @r0,a
        ret

template:
        db    ' JARAK :  cm '
        db    '  status  ' ;aman,hati hati,stop ;300,200,100

```



D\_template:

```
    mov    dptr,#template
    mov    sum_of_char,#16
    mov    lcd_position,#line1
    call   dptr_display
    mov    dptr,#template+16
    mov    sum_of_char,#16
    mov    lcd_position,#line2
    call   dptr_display
    ret
```

; animasi1:

```
    ; clr    buzzer
    ; acall  delay100ms
    ; acall  delay100ms
    ; acall  delay100ms
    ; setb  buzzer
    ; ret
```

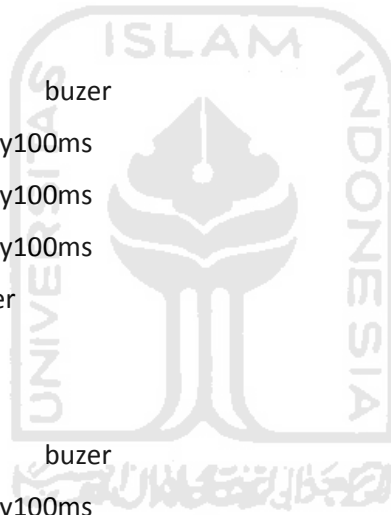
; animasi2:

```
    ; clr    buzzer
    ; acall  delay100ms
    ; acall  delay100ms
    ; setb  buzzer
    ; ret
```

; animasi3:

```
    ; clr    buzzer
    ; acall  delay100ms
    ; SETB  buzzer
    ; ret
```

\_1:



```
        db    ' AMAN  '
```

d\_aman:

```
        mov   dptr,#_1
        mov   sum_of_char,#16
        mov   lcd_position,#line2
        call  dptr_display
        call  delay100ms
        RET
```

d\_hati\_hati:

```
        mov   dptr,#_2
        mov   sum_of_char,#16
        mov   lcd_position,#line2
        call  dptr_display
        call  delay100ms
        RET
```

\_2:

```
        db    ' HATI HATI '
```

D\_STOP:

```
        mov   dptr,#_3
        mov   sum_of_char,#16
        mov   lcd_position,#line2
        call  dptr_display
        call  delay100ms
        RET
```

\_3: DB ' STOP '

```
        end
```



