

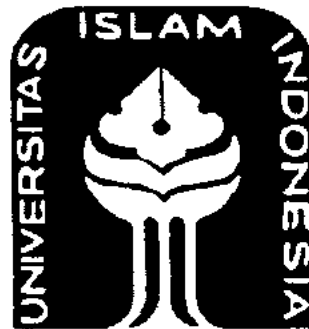
**PENINGKATAN AKURASI PADA ALAT UKUR JARAK  
DIGITAL**

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana

Pada Jurusan Teknik Elektro Fakultas Teknologi Industri

Universitas Islam Indonesia



**Disusun Oleh :**

**Nama : Iksan Maulana**

**No Mahasiswa : 04524049**

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

**2010**

**LEMBAR PENGESAHAN PEMBIMBING**  
**PENINGKATAN AKURASI PADA ALAT UKUR JARAK DIGITAL**

**TUGAS AKHIR**



Disusun oleh :

Nama : Iksan Maulana

No.Mahasiswa : 04 524 049

Yogyakarta, juni 2010

Pembimbing I

**Wahyudi Budi Pramono, ST.M.Eng.**

Pembimbing II

**Medilla Kusriyanto, ST. M.Eng**

**LEMBAR PENGESAHAN PENGUJI**  
**PENINGKATAN AKURASI PADA ALAT UKUR JARAK DIGITAL**  
**TUGAS AKHIR**

Oleh :

Nama : Iksan Maulana  
No. Mahasiswa : 04 524 049

**Telab Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat Untuk**  
**Memperoleh Gelar Sarjana Teknik Elektro**  
**Fakultas Teknologi Industri Universitas Islam Indonesia**  
Yogyakarta, Juli 2009

Tim Penguji,

Wahyudi Budi Pramono, ST.M.Eng.

Ketua



Medilla Kusriyanto, ST.M.Eng.

Anggota I



Dwi Ana Ratnawati, ST.M.Eng

Anggota II



Mengetahui,

Ketua Jurusan Teknik Elektro

Universitas Islam Indonesia



Tan Surosono, ST., Msc.

## HALAMAN PERSEMBAHAN

*Segala Puji bagi Allah SWT, atas Rahmat, Ridho dan Karunia-Nya*

*Atas kekuatan dan cahaya terang padaku*

*Semua cobaan dan kesabaran yang Aku syukuri dari-Nya*

*Segala sesuatu dalam karya yang sederhana ini dapat terselesaikan*

*dengan selalu beriman dan bertaqwa kepada-Nya*

*terima kasih,*

*Bapak dan Ibu.....*

*Atas segala doa, nasehat, cinta dan kasihnya, pengorbanan, dan perhatiannya*

*Yang telah diberikan takkan terbalas oleh apapun dan sampai kapan pun*

*Atas didikan, pengalaman, kepribadian dari yang kalian ajarkan...*

*Aku bisa kuat dan berusaha untuk maju sampai sekarang.....*

*Kakak-kakakku.....*

*Atas dukungan, perhatian dan doa-doanya*

*Kalian yang selalu menjadi sumber inspirasi untuk menuju yang lebih baik*

*Seluruh keluarga besarku*

*Atas segala harapan, doa dan dukungan yang diberikan dalam setiap langkahku*

*Sahabat-sahabat dan orang-orang terdekat*

*Atas segala pengalaman, ketulusan, bantuan, dukungan, kebersamaan dan keceriaan....*

## MOTTO

*“Dan (ingatlah juga), tatkala Tuhanmu memaklumkan ; “Sesungguhnya jika kamu bersyukur, pasti kami akan menambah (nikmat) kepadamu, dan jika kamu mengingkari (nikmat-Ku), Maka Sesungguhnya azab-Ku sangat pedih”*

*(QS. Ibrahim : 7)*

*“Sesungguhnya orang yang dirongganya tidak terdapat Al-Qur'an adalah seperti rumah yang rusak”*

*(HR, Ahmad, Tarmidzi dan Ad-Darmini)*

*“Tiadanya keyakinanlah yang membuat orang takut menghadapi tantangan; dan saya percaya pada diri saya sendiri.”*

*(Muhammad Ali)*

*“Manusia tidak merancang untuk gagal, mereka gagal untuk merancang.”*

*(William J. Siegel)*

## KATA PENGANTAR



*assalamu'alaikum warahmatullahi wabarakatuh*

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan Rahmat dan Hidayah-Nya sehingga penulis dapat menyelesaikan Tugas akhir yang berjudul “ **Peningkatan Akurasi Pada Alat Ukur Jarak Digital** ”. Shalawat dan salam selalu terhaturkan kepada junjungan kita Nabi Muhammad SAW yang menjadi *uswatun hasanah* bagi kita semua hingga akhir zaman.

Penyelesaian Tugas akhir ini merupakan salah satu syarat untuk memperoleh gelar sarjana Teknik Elektro pada Fakultas Teknologi Industri Universitas Islam Indonesia. Dengan selesainya Tugas akhir ini, maka seluruh aktivitas studi pada institusi tercinta ini berakhir dan membuka pintu untuk masuk ke tantangan berikutnya. Proses penyelesaian tugas akhir ini merupakan serangkaian integrasi antara teori serta praktek yang telah didapatkan semasa duduk dibangku kuliah. Ilmu dan amal menjadi poin penting akhir dari penyelesaian tugas akhir ini.

Tugas akhir ini merupakan implementasi dari proses perancangan serta pengujian sistem. Aplikasi utama dari sistem ini banyak dimanfaatkan dalam dunia industri, meskipun skala pengerjaan tugas akhir ini hanya sebatas simulasi.

Namun, begitu pentingnya fungsi dari perangkat ini, menjadikan penulis mencoba menganalisis sekaligus memodelkan sebuah perangkat yang dapat dimanfaatkan sebagai referensi dalam ilmu pengetahuan dan teknologi.

Penulis menyadari sepenuhnya bahwa isi dari tugas akhir ini masih jauh dari sempurna, namun dengan niat yang tulus dan ikhlas, penulis menerima segala macam kritik dan saran yang membangun demi kelangsungan dan kemajuan ilmu pengetahuan dan teknologi.

Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang tak terhingga kepada :

1. Bapak Tito Yuwono, S.T., M.Sc selaku Ketua Jurusan Teknik Elektro sekaligus selaku dosen pembimbing yang telah membantu dan membagi begitu banyak ilmunya kepada penulis.
2. Bapak Wahyudi Budi Pramono, ST.M.Eng. dan bapak Medila Kusriyanto, ST.M.Eng. selaku dosen pembimbing yang sangat memberikan banyak inspirasi serta motivasi dan ilmu.
3. Seluruh Dosen, mbak Umi dan karyawan FTI-UII yang bersedia dengan sabar membantu dan membagi ilmunya.
4. Bapak dan Ibu tercinta serta kakak – kakakku semua, terima kasih atas apa yang telah kalian beri selama ini.
5. Untuk kedua jagoan kecil ku Rayda Asfa Syafiq dan Nauval Kamil Al-Hakim yang selalu jadi penghibur.
6. Untuk para sahabat – sahabat yan, aliph, amir, irwan, abil, o’o, johan, bowo, hasan, bersama kalian hidup terasa lebih menyenangkan.

7. Teman-teman kost “ *Bukit Suling* “ erry, ari, sigit, tunggul, wisnu, panji, dhani, tangin, sani, gun, ino, wiwid, arif, kang endar, ardian, Semoga kita selalu bisa tertawa dan berbagi bersama selalu.
8. Seluruh pasukan elektro 04 yang menjadikanku seorang yang bermanfaat buat kalian semua. Terima kasih atas dukungan serta apa yang telah kita ciptakan bersama.
9. Seluruh fasilitas yang membantu nemenin bikin skripsi. Mas Irawan, pelayan Audio, pelayan toko Lima Satu, pelayan Sagan Elektronik, dan penjaga warnet-warnet yang setia membantu.

Akhir kata penulis sampaikan pula harapan semoga Tugas akhir ini dapat memberi manfaat yang cukup berarti khususnya bagi penulis dan bagi pembaca pada umumnya. Semoga Allah SWT senantiasa selalu memberikan rahmat dan hidayah-Nya kepada kita semua. Amiin.

*wassalamu'alaikum warahmatullahi wabarakatuh*

Yogyakarta, Juli 2010

Penulis



## Abstrak

Tujuan dilakukannya penelitian ini adalah membuat sebuah alat ukur digital untuk menggantikan peran alat ukur manual. Sistem alat ukur digital di buat untuk membantu manusia mempermudah pekerjaannya yang berkaitan dengan pengukuran jarak, alat ini dapat digunakan untuk mengukur obyek yang tidak mampu dijangkau oleh alat ukur manual seperti penggaris atau meteran. Metode pengukurannya adalah dengan mendekatkan dan menjauhkan obyek yang diukur, dengan melakukan pengukuran berulang dalam waktu yang berdekatan dan tempat yang sama. Obyek pengukuran sebaiknya mempunyai permukaan yang rata dan padat, karena sensor ultrasonik tidak mampu bekerja pada obyek yang mempunyai permukaan menyerap suara dan bergerigi tajam. Penekanan pada alat ini adalah mampu mengukur jarak suatu benda seakurat mungkin dengan error sekecil-kecilnya dengan menggunakan modul ultrasonik PING. Pada penelitian ini di gunakan ATmega8535 sebagai kontrol kerja dari sistem. Cara kerjanya yaitu sensor PING mendeteksi obyek dengan cara mengirimkan suara ultrasonik dan kemudian "mendengarkan" pantulan suara tersebut. Pada saat menunggu datangnya sinyal pantulan dari obyek, sensor akan menghasilkan sebuah pulsa. Pulsa tersebut dapat mempresentasikan jarak antara sensor dengan obyek, kemudian merupakan tugas mikrokontroler untuk melakukan perhitungan jarak yang hasilnya ditampilkan pada layar LCD. Melalui proses pengkalibrasian sensor dan pengujian alat maka didapat nilai rata-rata error yang terkecil yaitu mencapai 0.577 cm

Kata kunci : sensor ultrasonik PING, ATmega8535, LCD 16\*2, jarak.

## DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN DOSEN PEMBIMBING	ii
LEMBAR PENGESAHAN PENGUJI	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN MOTTO	v
KATA PENGANTAR	vi
ABSTRAK	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penulisan	2
1.5 Metode Penelitian	3
1.6 Sistematika Penulisan	3
<b>BAB II STUDI PUSTAKA</b>	
2.1 Studi Pustaka	5
2.2 Gelombang Ultrasonik	6

2.2.1 Karakteristik Fisik Gelombang Ultrasonik	8
2.3 Sensor Ultrasonik	10
2.3.1 Prinsip Kerja Sensor Ultrasonik	10
2.3.2 Karakteristik Sensor Ultrasonik PING	12
2.4 Mikrokontroler ATmega8535	14
2.5 LCD M1632	17
2.5.1 Tampilan M1632	18
2.6 Keypad 4x4	19
2.7 Rencana Penelitian	20
<b>BAB III PERANCANGAN SISTEM</b>	
3.1 Perancangan Alat	21
3.2 Perancangan <i>Hardware</i>	22
3.2.1 Rangkaian Keypad 4x4	23
3.2.2 Rangkaian penampil LCD	24
3.2.3 Sensor PING	25
3.2.4 Sistem Minimum Mikrokontroler ATmega8535	26
3.2.5 Rangkaian Catu Daya	27
3.3 Perancangan <i>Software</i>	28
3.4 Proses Pemrograman	30
3.4.1 Inisialisasi Program	30
3.4.2 Pengkalibrasian Sensor	31
3.4.3 Tampilan Pada LCD	32
3.4.4 Penyimpanan Data	33

<b>BAB IV HASIL PENGAMATAN DAN ANALISIS</b>	
4.1 Pengujian Alat	34
4.2 Analisa Data	42
<b>BAB V PENUTUP</b>	
5.1 Kesimpulan	44
5.2 Saran	44
<b>DAFTAR PUSTAKA</b>	45
<b>LAMPIRAN</b>	



## DAFTAR GAMBAR

Gambar 2.1	Proses Pantulan Gelombang Ultrasonik	7
Gambar 2.2	Pemantulan Sinyal Ultrasonik	11
Gambar 2.3	Modul Sensor Ultrasonik PING	12
Gambar 2.4	Proses Kerja Sensor Ping	13
Gambar 2.5	Pulsa PING	13
Gambar 2.6	Pin ATmega8535	15
Gambar 2.7	Konfigurasi Kaki M1632 ( <i>standart</i> )	18
Gambar 2.8	Keypad 4x4	19
Gambar 3.1	Skema Model Perancangan Alat	21
Gambar 3.2	Rangkaian Keypad	23
Gambar 3.3	Penampil Rangkaian LCD	24
Gambar 3.4	Cara Kerja Sensor PING	25
Gambar 3.5	Sistem Minimum Mikrokontroler ATmega8535	26
Gambar 3.6	Skema Rangkaian Catu Daya DC 5Volt	27
Gambar 3.7	<i>Flowcart</i> Program	28
Gambar 3.8	Tampilan AVR Studio	29
Gambar 3.9	Tampilan Ponyprog	30
Gambar 3.10	Tampilan Pada Layar LCD	32

Gambar 4.1	Grafik Perbandingan jarak sebelum kalibrasi	36
Gambar 4.2	Grafik Perbandingan jarak setelah proses kalibrasi	38
Gambar 4.3	Sinyal Keluaran Sensor Pada jarak 10 cm	41
Gambar 4.4	Sinyal Keluaran Sensor Pada jarak 1,5 meter	41
Gambar 4.5	Sinyal Keluaran Sensor Pada jarak 3 meter	42



## DAFTAR TABEL

Tabel 2.1	Kecepatan Bunyi Dalam Medium / Jaringan	9
Tabel 2.2	Konfigurasi Pin ATmega8535	16
Tabel 4.1	Data Pengukuran	34
Tabel 4.2	Data pengukuran interval 10 cm	37
Tabel 4.3	Data pengukuran interval 5 cm	39



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan zaman dan teknologi yang begitu pesat menuntut masyarakat untuk menciptakan alat atau sesuatu yang serba otomatis sehingga dapat memudahkan pekerjaan manusia sesuai dengan kebutuhan dan kesibukan yang semakin meningkat.

Untuk lebih memudahkan manusia dalam melakukan pekerjaan, maka banyak diciptakan alat-alat elektronik yang dapat bekerja secara otomatis. Hal ini bertujuan agar dapat lebih efisien dalam melakukan pekerjaan. Alat ukur yang sering digunakan masih berupa alat ukur manual seperti penggaris/mistar ataupun meteran, sehingga untuk mengukur benda yang posisinya sulit dijangkau oleh alat ukur manual maka di buat alat ukur digital yang mampu mengukur benda dengan jarak jauh.

Untuk membuat alat tersebut, gelombang ultrasonik di pilih sebagai media untuk mengukur jarak. Reflektivitas dari gelombang ultrasonik ini di permukaan cairan hampir sama dengan permukaan padat, tapi pada tekstil dan busa, maka jenis gelombang ini akan diserap. Gelombang ultrasonik adalah gelombang dengan besar frekuensi diatas frekuensi gelombang suara yaitu lebih dari 20 KHz. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari *transmitter* ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh *receiver* ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke



rangkaian mikrokontroller untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul).

### 1.2 Rumusan Masalah

Rumusan masalah yang akan menjadi pokok bahasan adalah bagaimana membuat suatu sistem alat ukur jarak digital yang akurat dengan sensor ultrasonik berbasis ATmega8535.

### 1.3 Batasan Masalah

Agar penulisan Tugas Akhir ini lebih terarah, maka perlu adanya batasan masalah yang dibahas. Diantaranya sebagai berikut :

1. Jarak yang diukur antara 3 cm – 300 cm
2. Penggunaan sensor ultrasonik jenis PING
3. Penggunaan ATmega8535 sebagai kendali utama
4. Penggunaan *keypad* sebagai input untuk mengaktifkan penyimpanan dan mengeluarkan data pengukuran
5. LCD sebagai penampil hasil akhir pengukuran

### 1.4 Tujuan Penulisan

Tujuan yang akan di capai dalam penulisan Tugas Akhir ini adalah membangun sebuah sistem alat ukur jarak digital dengan menggunakan sensor ultrasonik yang berbasis ATmega8535.

## 1.5 Metodologi Penelitian

Penyusunan Tugas Akhir ini mempunyai metodologi penelitian sebagai berikut:

1. Mendesain sistem yang akan dibuat.
2. Membangun *hardware* dari sistem tersebut
3. Pengujian dan analisa kerja alat dengan berdasarkan pada data yang telah didapat dari hasil pengujian.

## 1.6 Sistematika Penulisan

Dari hasil penelitian yang telah dilaksanakan, sistematika penulisan laporannya adalah sebagai berikut:

### BAB I PENDAHULUAN

Bab pendahuluan berisi tentang Latar Belakang pembuatan alat ukur jarak digital dengan menggunakan sensor ultrasonik berbasis mikrokontroller ATmega8535, Maksud dan Tujuan, Batasan Masalah, Perumusan Masalah, Metode Penelitian, serta Sistematika Penulisan pada Laporan Tugas Akhir.

### BAB II STUDI PUSTAKA

Bab studi pustaka membahas tentang penelitian yang telah dilakukan sebelumnya dan penjelasan secara terperinci mengenai teori-teori tentang gelombang ultrasonik, sensor ultrasonik, mikrokontroller ATmega8535, dan LCD yang digunakan sebagai landasan untuk pemecahan masalah.

### BAB III PERANCANGAN SISTEM

Berisi pembahasan tentang perancangan perangkat keras yang meliputi sensor sebagai media pembaca, penulisan program untuk mikrokontroller, mikrokontroler sebagai kendali utamanya dan LCD sebagai penampil hasil akhir.

### BAB IV HASIL PENGAMATAN DAN ANALISIS

Bagian ini berisi data hasil pengamatan dan analisis data serta hambatan yang dialami selama melakukan pengambilan data.

### BAB V PENUTUP

Bab penutup membahas tentang kesimpulan dan saran-saran yang menunjang agar alat tersebut dapat berkerja lebih optimal, sehingga tugas akhir ini dapat dikembangkan lebih lanjut, dengan harapan dapat digunakan untuk mendukung penerapannya dalam kehidupan masyarakat.

## BAB II

### STUDI PUSTAKA

#### 2.1 Studi Pustaka

Penelitian tentang pembuatan alat ukur digital pernah dilakukan *Untung Firdaus* dengan nomor mahasiswa 00524044 dengan judul tugas akhir “*Pembuatan Alat Ukur Jarak Digital Menggunakan Gelombang Ultrasonik*”. Pada dasarnya prinsip dasarnya prinsip kerja dari alat ini adalah dengan menghitung perbedaan waktu saat gelombang ultrasonik dikirim oleh sensor dan gelombang ultrasonik yang dipantulkan oleh obyek diterima sensor, sehingga didapat perbandingan lurus dengan jarak antara sensor dengan obyek yang akan diukur. Namun alat ini dirasa masih kurang akurat dalam membaca jarak yang diukur, artinya antara jarak yang sebenarnya dengan jarak yang terbaca pada alat ukur masih ada selisih nilai, *error* yang didapat oleh alat tersebut antara 4% - 10%. Selain sebagai alat ukur jarak, sensor ultrasonik juga dapat digunakan sebagai pengendali kecepatan kendaraan, sebagai navigasi robot mobil dimana sensor ultrasonik memberikan masukan kepada mikrokontroler agar robot tidak menabrak dinding atau penghalang dan menentukan arah gerak robot, dan dapat juga sebagai pengukur level air pada simulasi pendeteksi dini banjir.

Dengan melihat nilai *error* yang masih cukup besar maka dilakukan penelitian lanjutan untuk memperbaiki keakuratan alat. Langkah yang dilakukan dalam memperbaiki alat yaitu dengan mengganti mikrokontroler, sensor

ultrasonik, layar LCD, dan tentu saja dengan beberapa penambahan fitur-fitur yang belum ada.

Untuk mikrokontroler menggunakan jenis AVR, hal ini dikarenakan AVR memiliki arsitektur RISC 8 bit, di mana semua instruksi di kemas dalam kode 16-bit (16-bit word) dan sebagian instruksi dieksekusi dalam 1 (satu) siklus clock, berbeda dengan instruksi dengan MCS51 yang membutuhkan 12 siklus clock. Dengan menggunakan ATmega8535 diharapkan alat akan bekerja lebih maksimal dan mendapatkan hasil yang sesuai dengan jarak yang sebenarnya. Layar LCD 16\*2 dipilih sebagai penampil hasil akhir pengukuran, dan sensor Ping dipilih sebagai media pembaca atau pengukur jarak yang diinginkan. penelitian ini di buat dengan judul “ *Peningkatan Akurasi Pada Alat Ukur Jarak Digital* “. Sehingga dengan adanya alat ini pendeteksian jarak akan lebih akurat.

## 2.2 Gelombang Ultrasonik

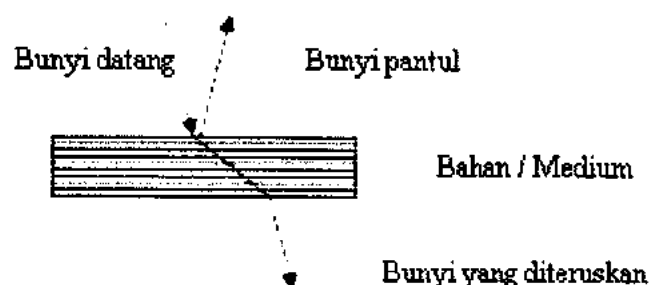
Ultrasonik adalah suara atau getaran dengan frekuensi yang terlalu tinggi untuk bisa didengar oleh telinga manusia, yaitu kira-kira di atas 20 KHz. Hanya beberapa hewan, seperti lumba-lumba menggunakannya untuk komunikasi, sedangkan kelelawar menggunakan gelombang ultrasonik untuk navigasi. Dalam hal ini, gelombang ultrasonik merupakan gelombang ultra (di atas) frekuensi gelombang suara (sonik).

Reflektivitas dari gelombang ultrasonik ini di permukaan cairan hampir sama dengan permukaan padat, tapi pada tekstil dan busa, maka jenis gelombang ini akan diserap.

Frekuensi yang diasosiasikan dengan gelombang ultrasonik pada aplikasi elektronik dihasilkan oleh getaran elastis dari sebuah kristal kuarsa yang diinduksikan oleh resonans dengan suatu medan listrik bolak-balik yang dipakaikan (efek *piezoelektrik*). Kadang gelombang ultrasonik menjadi tidak periodik yang disebut derau (*noise*), dimana dapat dinyatakan sebagai superposisi gelombang-gelombang periodik, tetapi banyaknya komponen adalah sangat besar. Kelebihan gelombang ultrasonik yang tidak dapat didengar, bersifat langsung dan mudah difokuskan. Jarak suatu benda yang memanfaatkan tunda (*delay*) gelombang pantul dan gelombang datang seperti pada sistem radar dan deteksi gerakan oleh sensor pada robot atau hewan. Secara matematis besarnya jarak dapat dihitung sebagai berikut :

$$s = \frac{v \times t}{2} \quad (2.1)$$

Dimana  $s$  adalah jarak dalam satuan meter,  $v$  adalah kecepatan rambat bunyi dengan medium udara yaitu 344 m/detik dan  $t$  adalah waktu tempuh dalam satuan detik. Ketika gelombang ultrasonik menumbuk suatu penghalang maka sebagian gelombang tersebut akan dipantulkan sebagian diserap dan sebagian yang lain akan diteruskan. Proses ini ditunjukkan pada gambar berikut :



**Gambar 2.1** Proses Pantulan Gelombang Ultrasonik

### 2.2.1 Karakteristik Fisik Gelombang Ultrasonik.

Jika gelombang bolak-balik terjadi terus menerus secara periodik maka akan menghasilkan deretan gelombang periodik dimana pada setiap gerak periodik, partikel-partikel yang berada pada titik-titik yang sama pada gelombang tersebut akan berada dalam fase yang sama.

Jarak antara dua nilai puncak gelombang yang berurutan (gelombang transversal) atau jarak dari dua bagian pemampatan gelombang yang berurutan (gelombang longitudinal) disebut panjang gelombang ( $\lambda$ ). Waktu yang dibutuhkan untuk menempuh satu gelombang penuh atau waktu yang ditempuh sepanjang gelombang tersebut disebut periode (T). hubungan antara panjang gelombang dengan periode ini adalah :

$$\lambda = c \cdot T \quad (2.2)$$

Frekuensi gelombang (f) adalah banyaknya getaran yang terjadi per satuan waktu periode sehingga persamaan 2.2 dapat juga ditulis dalam bentuk frekuensi terhadap laju gelombang bunyi dalam medium (c) yakni

$$\lambda = c / f \quad (2.3)$$

Besarnya panjang gelombang ini sangat penting untuk menentukan batas resolusi pencitraan sistem. Dua bentuk struktur yang berdekatan panjang gelombangnya tidak bisa diidentifikasi secara terpisah pada pencitraan ultrasonik.

Kecepatan bunyi dalam medium bergantung pada kerapatan ( $\rho$ ) dan kompresibilitas medium (B). Material dengan molekul berat cenderung bergerak lebih lambat dibandingkan molekul ringan saat terjadinya perubahan tekanan

dalam medium. Material yang sangat kompresibel seperti gas akan meneruskan gelombang bunyi lebih lambat sehingga penambahan intensitas atau kompresibilitas cenderung akan menurunkan kecepatan bunyi. Jadi besarnya nilai kecepatan bunyi adalah :

$$c = f \times \lambda \quad (2.4)$$

**Tabel 2.1** Kecepatan Bunyi dalam Medium / Jaringan

Medium / Jaringan	Kecepatan Bunyi (m/s)
Udara	344
Lemak	1.450
Air	1.495
Jaringan lunak ( rata-rata )	1.540
Ginjal	1.561
Otot	1.585
Tulang	4.080

Pada tabel memperlihatkan kecepatan bunyi melalui beberapa medium, dimana kecepatan bunyi bergantung kepada kerapatan dan kompresibilitas medium. Material dengan molekul berat, kecepatan bunyinya cenderung lebih lambat dibandingkan molekul ringan. Dan material yang sangat kompresibel seperti gas dengan jarak simpangan molekul yang panjang akan merambatkan gelombang menjadi lebih lambat.



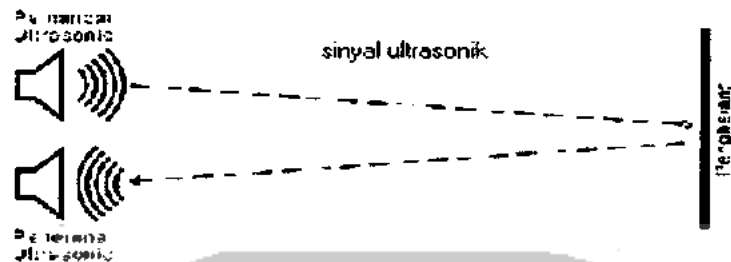
## 2.3 Sensor Ultrasonik

Sensor ultrasonik adalah sebuah sensor yang mengubah besaran fisis (bunyi) menjadi besaran listrik. Pada sensor ini gelombang ultrasonik dibangkitkan melalui sebuah benda yang disebut piezoelektrik. Piezoelektrik ini akan menghasilkan gelombang ultrasonik dengan frekuensi 40 kHz ketika sebuah osilator diterapkan pada benda tersebut. Sensor ultrasonik secara umum digunakan untuk suatu pengungkapan tak sentuh yang beragam seperti aplikasi pengukuran jarak. Alat ini secara umum memancarkan gelombang suara ultrasonik menuju suatu target yang memantulkan balik gelombang ke arah sensor. Kemudian sistem mengukur waktu yang diperlukan untuk pemancaran gelombang sampai kembali ke sensor dan menghitung jarak target dengan menggunakan kecepatan suara dalam medium.

### 2.3.1 Prinsip Kerja Sensor Ultrasonik

Seperti telah disebutkan bahwa sensor ultrasonik terdiri dari rangkaian pemancar ultrasonik yang disebut *transmitter* dan rangkaian penerima ultrasonik yang disebut *receiver*. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari *transmitter* ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh *receiver* ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul).

Prinsip kerja dari sensor ultrasonik dapat ditunjukkan dalam gambar dibawah ini :



Gambar 2.2 Pemantulan Sinyal Ultrasonik.

Prinsip kerja dari sensor ultrasonik adalah sebagai berikut :

1. Sinyal dipancarkan oleh pemancar ultrasonik. Sinyal tersebut berfrekuensi diatas 20kHz, biasanya yang digunakan untuk mengukur jarak benda adalah 40kHz. Sinyal tersebut di bangkitkan oleh rangkaian pemancar ultrasonik.
2. Sinyal yang dipancarkan tersebut kemudian akan merambat sebagai sinyal / gelombang bunyi dengan kecepatan bunyi yang berkisar 344 m/s. Sinyal tersebut kemudian akan dipantulkan dan akan diterima kembali oleh bagian penerima ultrasonik.
3. Setelah sinyal tersebut sampai di penerima ultrasonik, kemudian sinyal tersebut akan diproses untuk menghitung jaraknya. Jarak dihitung berdasarkan rumus :

$$s = 344.t/2 \quad (2.5)$$

Dimana s adalah jarak antara sensor ultrasonik dengan bidang pantul, dan t adalah selisih waktu antara pemancaran gelombang ultrasonik sampai diterima kembali oleh bagian penerima ultrasonik.

### 2.3.2 Karakteristik Sensor Ultrasonik PING

*PING Ultrasonic Range Finder*, adalah modul pengukur jarak dengan ultrasonik buatan *paralax Inc.* yang didesain khusus untuk teknologi robotika. Dengan ukuran yang cukup kecil (2,1 cm x 4,5 cm), sensor ini dapat mengukur jarak antara 3 cm sampai 300cm.



Gambar 2.3 Modul Sensor Ultrasonik PING

Pada dasarnya, sensor ini terdiri dari sebuah *chip* pembangkit sinyal 40KHz, sebuah *speaker* ultrasonik dan sebuah mikropon ultrasonik. *Speaker* ini mengubah sinyal 40 KHz menjadi suara sementara mikropon ultrasonik berfungsi untuk mendeteksi pantulan suaranya. Sensor PING mendeteksi obyek dengan cara mengirimkan suara ultrasonik dan kemudian “mendengarkan” pantulan suara tersebut. Pada saat menunggu datangnya sinyal pantulan dari obyek, sensor akan menghasilkan sebuah pulsa. Pulsa tersebut dapat merepresentasikan jarak antara sensor dengan obyek. Selanjutnya mikrokontroller cukup mengukur lebar pulsa tersebut dan mengkonversinya dalam bentuk jarak dengan perhitungan sebagai berikut :

Namun harus di ingat bahwa sensor ini tidak dapat mengukur obyek yang permukaannya dapat menyerap suara atau *sound demper* lainnya. Pengukuran jarak juga akan kacau jika permukaan obyek tidak rata atau bergerigi dengan sudut tajam.

#### 2.4 Mikrokontroller ATmega 8535

Mikrokontroller AVR memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (*16-bits word*) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus *clock*. Mikrokontroller tipe ini memiliki fitur tambahan yang tidak dimiliki oleh AT89S51/52 yaitu adanya fitur seperti ADC 10 bit, PWM, kemampuan timer dan kemampuan lain yang menjadikan mikrokontroller ini lebih canggih.

Fitur dasar yang dimiliki Mikrokontroller AVR ATmega8535 adalah sebagai berikut :

1. Port I/O 32 jalur (Port A, Port B, Port C, Port D masing-masing 8 bit).
2. ADC 10 bit 8 channel.
3. 3 buah timer/counter.
4. 32 register dalam CPU
5. Watchdog Timer dengan osilator internal.
6. Flash PEROM 8 kb.
7. EEPROM 512 bytes.

8. SRAM 512 bytes.
9. Interupsi Eksternal dan Internal.
10. Interfacing dengan komparator analog.
11. Port USART untuk komunikasi serial.

Mikrokontroler AVR ATmega8535 terdiri dari 40 Pin, yang konfigurasiya sebagai berikut :

Pin	Function	Pin	Function
1	(XCK/T0) PB0	40	PA0 (ADC0)
2	(T1) PB1	39	PA1 (ADC1)
3	(INT2/AIN0) PB2	38	PA2 (ADC2)
4	(OC0/AIN1) PB3	37	PA3 (ADC3)
5	(SS) PB4	36	PA4 (ADC4)
6	(MOSI) PB5	35	PA5 (ADC5)
7	(MISO) PB6	34	PA6 (ADC6)
8	(SCK) PB7	33	PA7 (ADC7)
9	RESET	32	AREF
10	VCC	31	GND
11	GND	30	AVCC
12	XTAL2	29	PC7 (TOSC2)
13	XTAL1	28	PC6 (TOSC1)
14	(RXD) PD0	27	PC5
15	(TXD) PD1	26	PC4
16	(INT0) PD2	25	PC3
17	(INT1) PD3	24	PC2
18	(OC1B) PD4	23	PC1 (SDA)
19	(OC1A) PD5	22	PC0 (SCL)
20	(ICP1) PD6	21	PD7 (OC2)

**Gambar 2.6** Pin ATmega8535

Fungsi dari masing-masing pin Mikrokontroler AVR ATmega8535 adalah sebagai berikut :

**Tabel 2.2.** Konfigurasi Pin ATmega8535

No Pin	Nama	Fungsi
1	PB0 (XCK/TO)	Port B.0 / Timer-Counter 0 dan clock eksternal untuk USART (XCK)
2	PB1 (T1)	Port B.1 / Timer-Counter 1
3	PB2 (INT2/AIN0)	Port B.2 / Input (+) Analog komparator (AIN0) dan interupsi eksternal 2 (INT2)
4	PB3 (OC0/AIN1)	Port B.3 / Input (-) Analog komparator (AIN1) dan output pembanding timer/counter (OC0)
5	PB4 (SS)	Port B.4 / SPI Slave Select Input (SS)
6	PB5 (MOSI)	Port B.5 / SPI Bus Master Out Slave In
7	PB6 (MISO)	Port B.6 / SPI Bus Master In Slave Out
8	PB7 (SCK)	Port B.7 / sinyal clock serial SPI
9	RESET	Me-reset Mikrokontroler
10	VCC	Catu daya (+)
11	GND	Sinyal ground terhadap catu daya
12 - 13	XTAL 2 - XTAL 1	Sinyal input clock eksternal (kristal)
14	PD0 (RXD)	Port D.0 / penerima data serial
15	PD1 (TXD)	Port D.1 / pengirim data serial
16	PD2 (INT0)	Port D.2 / Interupsi eksternal 0

17	PD3 (INT1)	Port D.3 / Interupsi eksternal 1
18	PD4 (OC1B)	Port D.4 / Pembanding Timer-Counter 1
19	PD5 (OC1A)	Port D.5 / Pembanding Timer-Counter 1
20	PD6 (ICP1)	Port D.6 / Timer-Counter 1 Input
21	PD7 (OC2)	Port D.7 / Pembanding Timer-Counter 2
22	PC0 (SCL)	Port C.0 / Serial bus clock line
23	PC1 (SDA)	Port C.0 / Serial bus data input-output
24 - 27	PC2 – PC5	Port C.0
28	PC6 (TOSC1)	Port C.0 / Timer osilator 1
29	PC7 (TOSC2)	Port C.0 / Timer osilator 2
30	AVCC	Tegangan ADC
31	GND	Sinyal ground ADC
32	AREFF	Tegangan referensi ADC
33 - 40	PA0 (ADC0) – PA7 (ADC7)	Port A.0 – Port A.7 dan input untuk ADC (8 channel : ADC0 – ADC7)

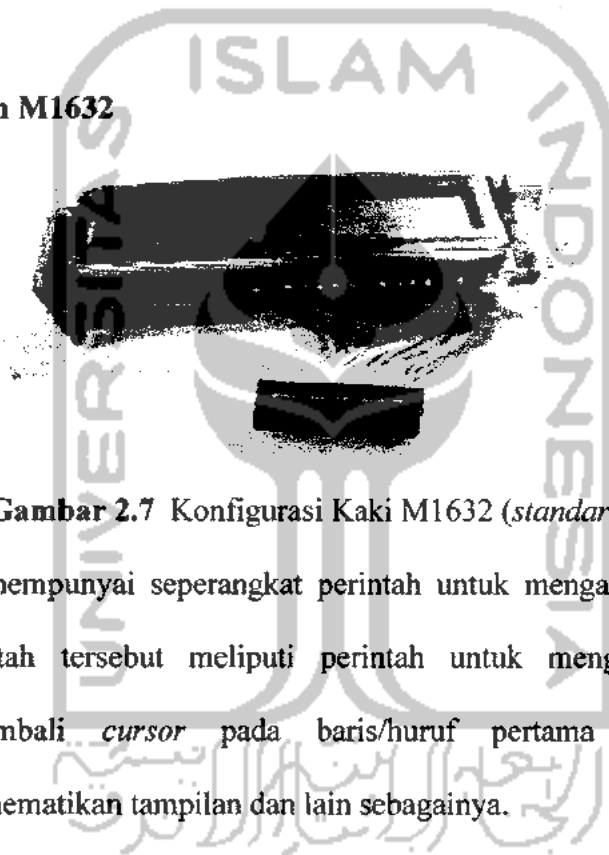
## 2.5 LCD M1632

LCD *Display Module M1632* buatan Seiko Instrument Inc. Terdiri dari dua bagian, yang pertama merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf/angka dua baris, masing–masing baris bisa menampung 16 huruf/angka.

Bagian kedua merupakan sebuah sistem yang dibentuk dengan mikrokontroller yang ditempelkan dibalik panel LCD, berfungsi untuk mengatur

tampilan informasi serta berfungsi mengatur komunikasi M1632 dengan mikrokontroler yang memakai tampilan LCD tersebut. Dengan demikian pemakaian M1632 menjadi sederhana, sistem lain yang memakai M1632 cukup mengirimkan kode-kode ASCII dari informasi yang ditampilkan seperti layaknya.

### 2.5.1 Tampilan M1632



**Gambar 2.7** Konfigurasi Kaki M1632 (*standart*)

M1632 mempunyai seperangkat perintah untuk mengatur tata kerjanya, perangkat perintah tersebut meliputi perintah untuk menghapus tampilan, meletakkan kembali *cursor* pada baris/huruf pertama baris pertama, menghidupkan/mematikan tampilan dan lain sebagainya.

Setelah diberi catu daya, ada beberapa langkah persiapan yang harus dikerjakan dulu agar M1632 bisa dipakai, langkah- langkah tersebut antara lain adalah :

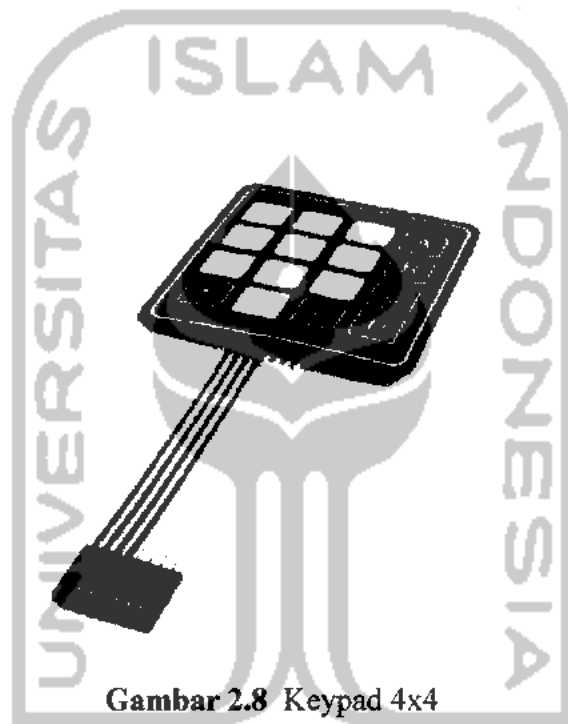
1. Menunggu dulu selama 15 mili-detik atau lebih.
2. Mengirimkan perintah 30h, artinya transfer data antar M1632 dan mikrokontroler dilakukan dengan mode 8-bit.
3. Menunggu selama 4.1 mili-detik atau lebih.
4. Mengirimkan sekali lagi perintah 30h.



5. Menunggu lagi selama 100 mikro-detik atau lebih.

Setelah langkah-langkah tersebut M1632 barulah bisa menerima data dan menampilkannya dengan baik. Pada awalnya tampilan akan nampak kacau, dengan demikian perlu dikirim perintah menghapus tampilan dan sebagainya.

## 2.6 Keypad 4x4



Gambar 2.8 Keypad 4x4

*Keypad 4x4* adalah *keypad* yang mempunyai 4 baris tombol mendatar dan 4 baris tombol menurun, pada prinsipnya sistem kerja *keypad* seperti tombol *pushbutton* yang mana bila tombol ditekan akan menghubungkan rangkaian dengan sumber tegangan, jika tombol dilepas maka sambungan akan terputus.

*Keypad 4x4* digunakan sebagai tombol input berupa huruf dan angka yang dirubah menjadi data digital, digunakan untuk memasukan data mikrokontroller ATmega8535, papan *keypad* terdiri dari 16 saklar *pushbutton* yang dirangkai

menjadi satu, digunakan secara bergantian sehingga menghasilkan *output* data digital yang diinginkan.

## 2.7 Rencana Penelitian

Pada rencana penelitian ini dimulai dengan membuat rancangan *hardware* terlebih dahulu, kemudian memilih komponen-komponen yang akan digunakan dalam penelitian ini. Selanjutnya mengumpulkan data yang akan diolah pada sistem ini kemudian membuat program pada sistem. Langkah selanjutnya membandingkan antara hasil pengukuran digital dengan pengukuran analog.

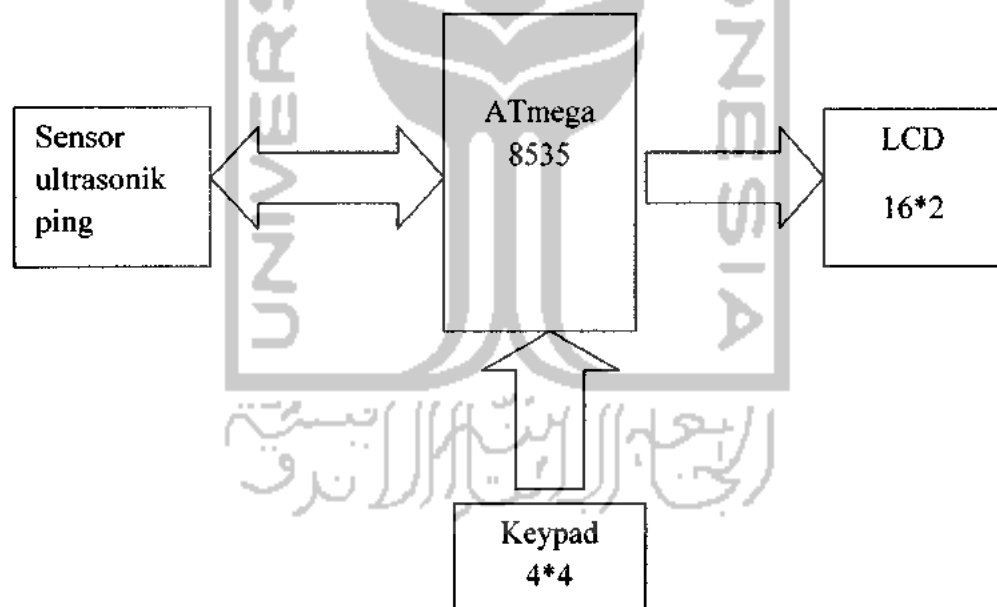
Pada penelitian ini yang akan diukur adalah keluaran dari sensor ultrasonik yang berupa jarak pengukuran sebuah benda. Analisa yang dilakukan dari sistem berupa membandingkan data hasil keluaran digital dengan hasil keluaran analog serta *software* yang akan digunakan.

## BAB III

### PERANCANGAN SISTEM

#### 3.1 Perancangan Alat

Dalam pembuatan sistem yang lengkap, harus mengetahui terlebih dahulu gambaran umum tentang rancangan yang akan dibuat. Hal ini perlu dilakukan agar pembuatan alat lebih terkonsep. Secara garis besar model rancangan alat pengukur jarak adalah sebagai berikut :



**Gambar 3.1** Skema Model Perancangan Alat

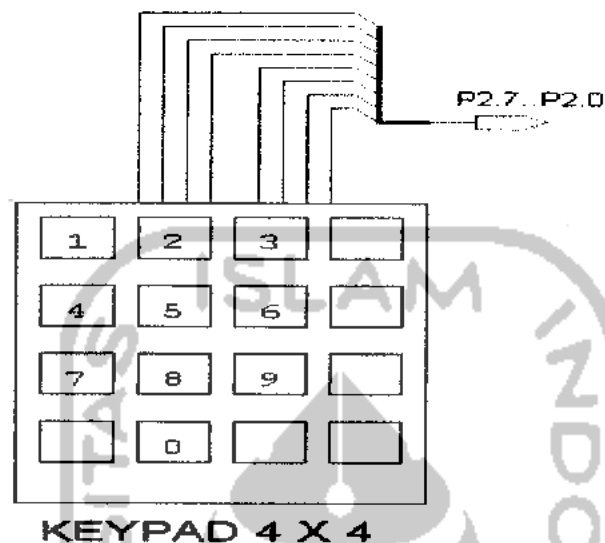
Pada dasarnya sistem ini terdiri dari 3 bagian penting, bagian yang pertama adalah sensor ultrasonik. Sensor ini bekerja dengan cara memancarkan gelombang ultrasonik ke obyek benda yang akan diukur dan kemudian gelombang yang dipantulkan oleh obyek diterima kembali oleh sensor. Dimana sensor akan

menghasilkan sebuah pulsa yang menpresentsikan jarak benda. Untuk selanjutnya mikrokontroller ATmega8535 akan memproses sinyal masukan yang di dapat dari sensor ultrasonik dengan bantuan *software* yang telah diinstruksikan untuk memabaca, memproses dan mengeluarkan hasilnya yang akan ditampilkan pada bagian terakhir yaitu pada layar LCD. Tampilan pada LCD berupa jarak obyek dengan alat dalam satuan millimeter (mm). Sedangkan untuk *keypad* merupakan fitur tambahan, yang berfungsi untuk mengaktifkan, penyimpanan data hasil pengukuran, menghapus data, dan mengeluarkan data yang telah disimpan untuk dapat ditampilkan pada layar LCD.

### 3.2 Perancangan Hardware

Pada perancangan *hardware* penelitian ini, seluruh rangkaian elektronik baik yang merupakan rangkaian kontroler maupun utilitas, semuanya terhubung dengan *wiring* secara fisik. Perancangan *hardware* ini didukung oleh rangkaian-rangkaian elektris yang membantu kerja mikrokontroller sebagai pengendali utama, seperti: sistem minimum, *power supply* serta rangkaian elektris lainnya yang menjalankan sistem secara keseluruhan. Berikut merupakan komponen-komponen yang digunakan.

### 3.2.1 Rangkaian Keypad 4x4



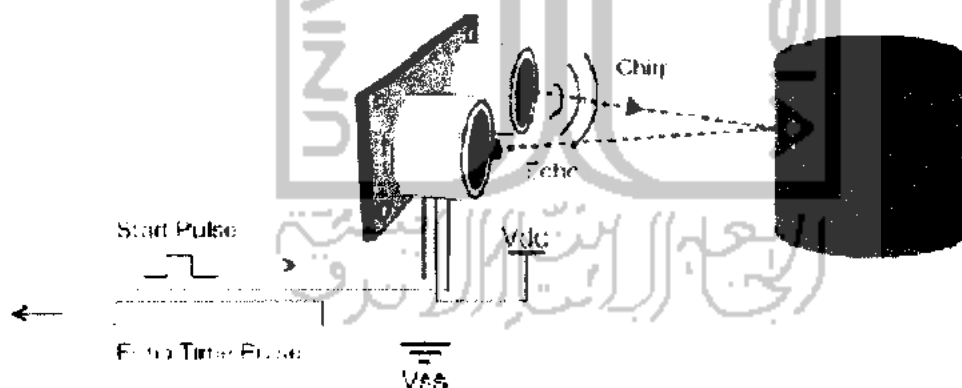
**Gambar 3.2 Rangkaian Keypad**

*Keypad* digunakan sebagai media *input*, dalam satu rangkaian *keypad* mempunyai 16 tombol *push button*, masing-masing tombol mempunyai fungsi yang berbeda, yaitu :

1. Tombol A : Untuk memulai pengukuran
2. Tombol B : Untuk mengakses slot eeprom berikutnya sebagai tempat penyimpanan data (*forward*)
3. Tombol C : Untuk membaca data yang telah disimpan (*backward*)
4. Tombol D : untuk menyimpan data
5. Tombol \* : Untuk mengakses data yang ingin kita tampilkan
6. Tombol # : Untuk menampilkan data yang dipilih
7. Tombol 0 – 9 : Sebagai media pemilih data (maksimal 2 *digit*, misal 01-99)
8. Pushbutton : sebagai penghapus data

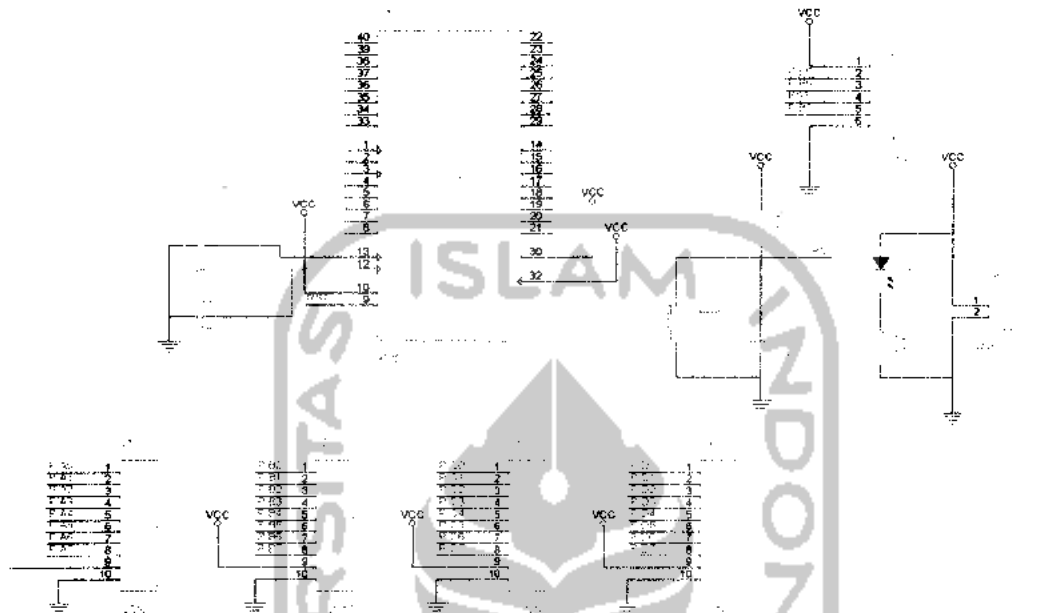
### 3.2.3 Sensor PING

Sensor PING mendeteksi jarak obyek dengan cara memancarkan gelombang ultrasonik (40 KHz) selama  $t$  (200 $\mu$ s) kemudian mendeteksi pantulannya. Sensor PING memancarkan gelombang sesuai dengan kontrol dari mikrokontroller pengendali. Gelombang ultrasonik ini melalui udara dengan kecepatan 344 meter per 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* akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk 2x jarak ukur dengan obyek.



**Gambar 3. 4** Cara Kerja Sensor PING

### 3.2.4 Sistem Minimum Mikrokontroller ATmega8535



**Gambar 3.5** Sistem Minimum Mikrokontroller ATmega 8535

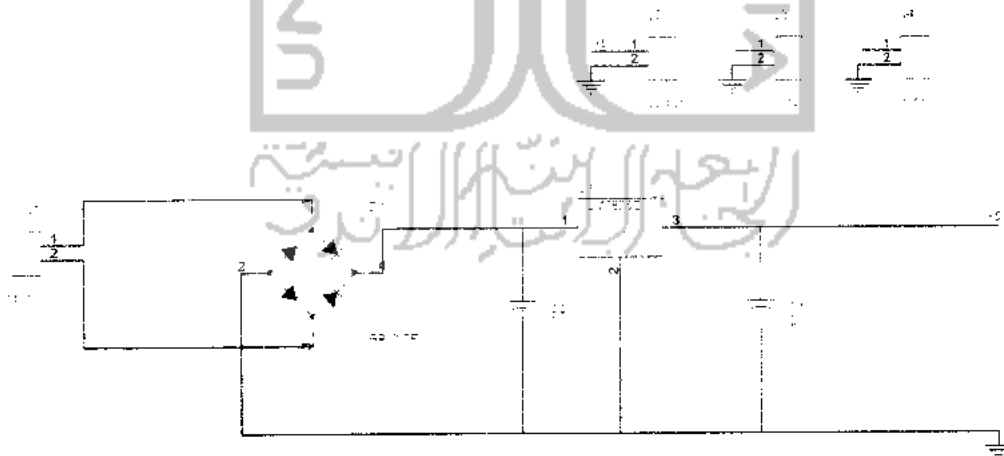
Mikrokontroler AVR memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (*16-bits word*) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus *clock*. Rangkaian osilator pada sistem ini digunakan oleh mikrokontroller sebagai sinyal denyut (*clock*). Dengan memanfaatkan kristal internal sebesar 4MHz sebagai pembangkit sinyal denyut (*clock*).

Dengan mengkoneksikan LCD M1632 dengan port A pada mikrokontroller, untuk pin yang digunakan adalah pin 2 sampai dengan pin 8 pada port A. Port D di gunakan sebagai pengkoneksi antara sensor PING dengan mikrokontroller, sensor PING hanya membutuhkan 1 pin sebagai media

koneksinya yaitu melalui pin SIG yang berfungsi sebagai input sekaligus output data, sedangkan 2 pin lainnya adalah sebagai sumber catu daya 5 volt dan *ground*.

### 3.2.5 Rangkaian Catu Daya

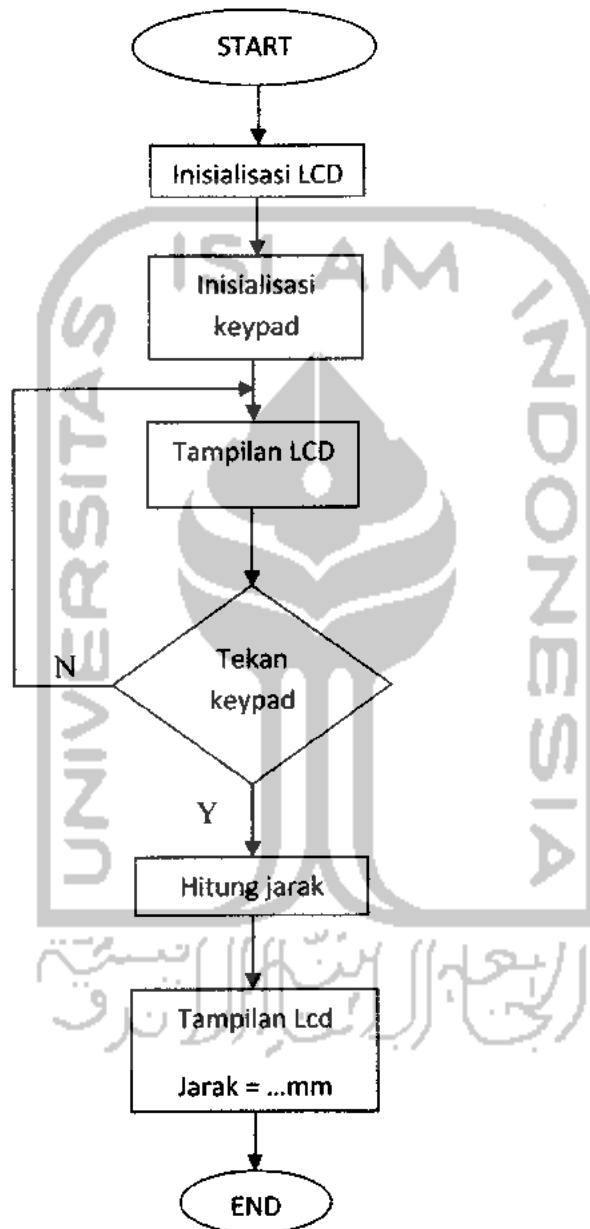
Setiap rangkaian elektronik tentunya membutuhkan catu daya, sehingga perancangan catu daya menjadi hal yang sangat penting, agar rangkaian ini dapat memberikan kebutuhan arus dan tegangan yang sesuai. Selain arus dan tegangan yang sesuai, hal lain yang perlu diperhatikan adalah kestabilan dari tegangan dan arus tersebut. Pada alat catu daya yang digunakan terdiri dari +5V, dan 220 AC. Untuk rancangan catu daya +5V berikut menggunakan IC LM7805 sebagai regulator tegangannya, sehingga tegangan outputnya lebih stabil. Selain itu dilengkapi dengan kapasitor C1 dan C2 sebagai filter tegangan.



**Gambar 3.6** Skema Rangkaian Catu Daya DC 5 volt



### 3.3 Perancangan Software

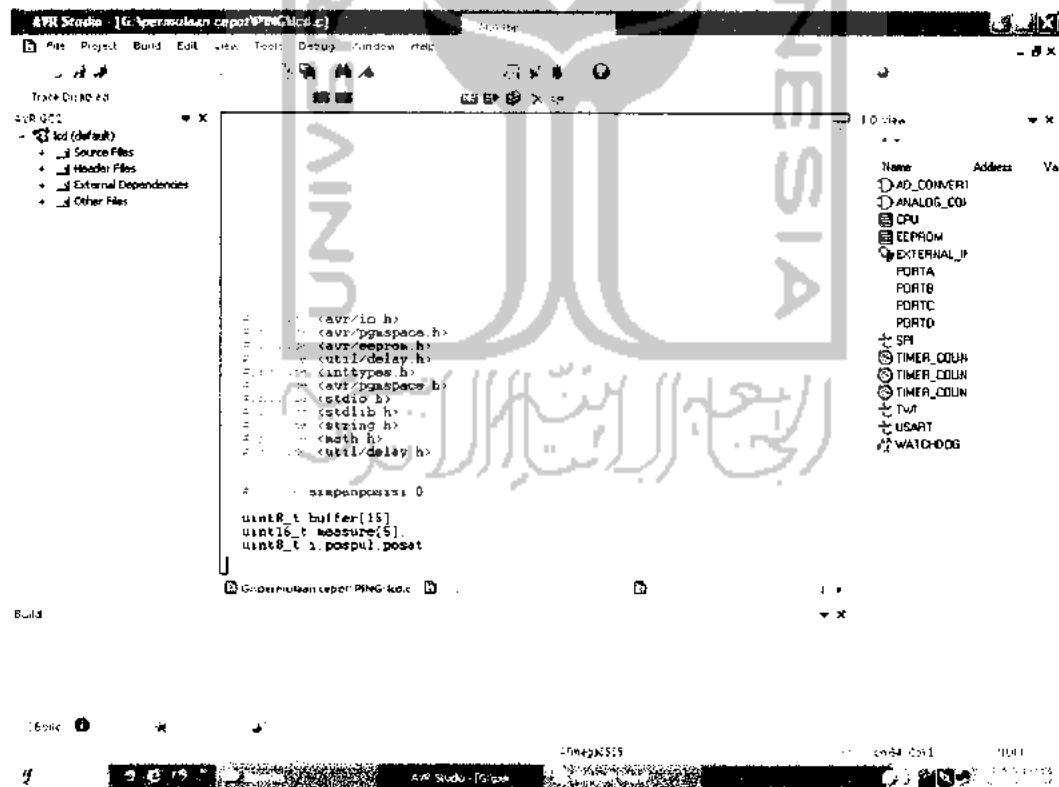


**Gambar 3.7** Flowcart Program

Untuk mempermudah dan memfokuskan pembuatan *software* maka perlu adanya suatu perencanaan awal, salah satunya dengan membuat *flowcart* program.

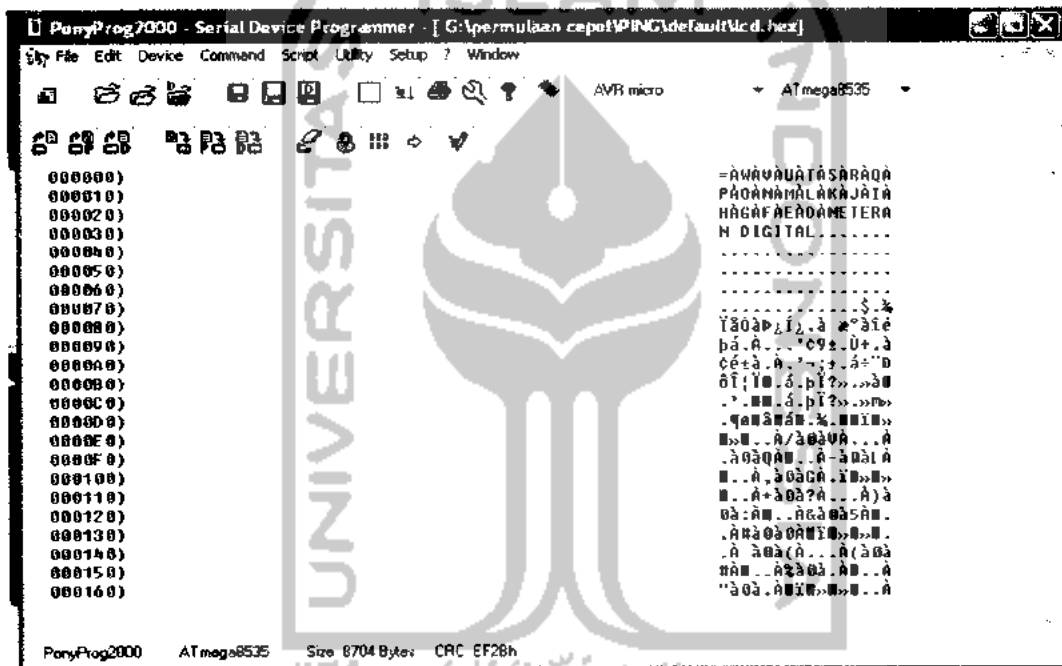
Langkah pertama adalah penginisialisasian komponen-komponen yang nantinya akan digunakan dalam *hardware*. Setelah proses inisialisasi selesai, masuk ke tahap selanjutnya yaitu proses penampilan hasil pada layar LCD berupa tulisan “METERAN DIGITAL P = 01 JRK = mm”. kemudian menuju keproses selanjutnya yaitu pada tombol *keypad* yang berfungsi sebagai media untuk mengaktifkan fungsi-fungsi dari alat, dan yang terakhir merupakan tampilan hasil jarak pengukuran alat.

Sebagai media penulisan dan *compiler software* digunakan aplikasi AVR Studio. Berikut adalah tampilan dari aplikasi AVR Studio :



Gambar 3.8 Tampilan AVR Studio

Setelah *software* selesai di *compile* dan tidak terdapat *error* maka proses selanjutnya adalah men-*download software* ke mikrokontroler melalui port serial dengan menggunakan aplikasi Ponyprog. Aplikasi ini dipilih karena penggunaannya relatif lebih mudah dan praktis. Berikut tampilan dari aplikasi Ponyprog:



Gambar 3.9 Tampilan Ponyprog

### 3.4 Proses Pemrograman

#### 3.4.1 Inisialisasi Program

```
#include <avr/io.h>
#include <avr/pgmspace.h>
#include <avr/eeprom.h>
#include <inttypes.h>
#include <avr/pgmspace.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <util/delay.h>
```

Instruksi – instruksi diatas merupakan intruksi penginisialisasian program. Diantaranya inisialisasi mikrokontroller jenis AVR, *eeprom* dari mikrokontroller, inisialisasi *delay* atau waktu tunda, dan operasi matematik untuk melakukan perhitungan yang dilakukan oleh mikrokontroler. *File include* berfungsi memberitahu *compiler* agar membaca *file* yang di *include*-kan lebih daulu agar mengenali definisi–definisi yang digunakan dalam program sehingga tidak dianggap *error*.

### 3.4.2. Pengkalibrasian Sensor

Dengan melakukan pengkalibrasian sensor yang terpusat pada mikrokontroller, agar pembacaan sensor tidak meleset terlalu jauh dari jarak sebenarnya atau agar dapat mendapatkan hasil se akurat mungkin. Berikut cara pengkalibrasian sensor ping melalui AVR studio:

```
uint16_t baca_ping(void){
    DDRD |= (1<<INP);
    PORTD |= (1<<INP);
    delay_us(7);
    PORTD &= ~(1<<INP);
    DDRD &= ~(1<<INP);
    PORTD |= (1<<INP);
    while (bit_is_clear(PIND,INP)) {};
    while (!bit_is_clear(PIND,INP))
    {
        count++;
    }
    jarak= count * 0.34442 / 2 * 0.734 – 0.7742; //rumus mencari jarak
    count=0;
    return jarak;
}
```

$DDRD \mid = (1 \ll INP)$  menunjukan input diberikan pada setiap *bit* di port D, kemudahan menggeser tiap *bit* ke kiri dan tiap pasangan bit di *or*-kan sehingga

menghasilkan 0 jika keduanya berisi 0, dan 1 untuk yang lainnya. Kemudian di tunda selama 7 mikro detik, selanjutnya bit pada port D di *and*-kan dan di balik isi tiap *bit*-nya agar bernilai sama seperti awalnya. Setelah itu ditunggu sampai pin D set, langkah berikutnya adalah melakukan perhitungan jarak yang diproses di dalam mikro.

### 3.4.3 Tampilan Pada LCD

Untuk mengetahui hasil dari pengukuran maka hasil dari pengukuran ditampilkan pada layar LCD. Berikut program untuk menampilkan hasil pengukuran pada LCD:

```
sprintf(buffer, "P=%02i;JRK:%imm \0",ukurke,average);
LCDGotoXY(0,1);
```

Perintah *sprint* berfungsi untuk menyimpan data *string* ke dalam memori SRAM. Dalam hal ini memori yang di tuju adalah memori yang ditunjukan variable *array buffer*. Sedangkan untuk perintah *LCDGoto* digunakan untuk melompat ke baris program tertentu yang telah diberi label.



**Gambar 3.10** Tampilan Pada Layar LCD

### 3.4.4 Penyimpanan Data

Untuk penyimpanan data pengukuran digunakan memori *eeprom* pada ATmega8535. Memori *eeprom* dapat digunakan untuk menyimpan data pada saat *chip running* dan tidak dapat terhapus meskipun catu daya mati. Untuk menuliskan data pada *eeprom* menggunakan perintah *eeprom\_write type data* dan untuk membaca data pada memori *eeprom* menggunakan perintah *eeprom\_read type data*.

Contoh penulisan program pada *eeprom*:

```
if (keypad()==11){
    while(keypad()==11){
        ukurke=pospul*10+posat;
        eeprom_write_byte((uint8_t*)simpanposisi,ukurke);
        pos=ukurke*2;
        average = eeprom_read_word((uint16_t*)pos);
        sprintf(buffer,"P=%02i;JRK:%imm
\0",ukurke,average);//RPM/T);
        LCDGotoXY(0,1);
        LCDstring(buffer,15);
    }
}
```

Untuk langkah pertama tentukan jenis data yang akan disimpan, kemudian membuat nama variabel penyimpanan agar mudah untuk mengingat. Dan untuk pengalamatannya sudah menjadi tugas dari *compiler*. Sedangkan untuk proses pembacaan memory ada yang harus diperhatikan yaitu tipe data variabel penampung harus sama dengan tipe data yang yang di baca. Setelah proses pembacaan dilakukan maka selanjutnya data akan ditampilkan pad layar LCD.

## BAB IV

### HASIL PENGAMATAN DAN ANALISIS

Pada proses alat ukur digital dilakukan beberapa proses pengujian, diantaranya adalah pengujian sensor ultrasonik PING dan pengujian alat.

#### 4.1 Pengujian Alat

Pengujian sensor PING dilakukan dengan membandingkan jarak hasil pengukuran sensor dengan jarak sebenarnya yang di ukur dengan menggunakan meteran. Hasil pengukuran pada tabel berikut merupakan hasil dari perhitungan rata-rata dari 5 kali pengukuran yang dilakukan oleh sensor PING, dimana proses perhitungannya telah dilakukan didalam mikrokontroler.

Sebelum melakukan proses kalibrasi, maka perlu adanya proses penentuan konstanta sebagai konstanta pengkalibrasi, berikut data sebelum sensor di kalibrasi :

$$\text{jarak} = \text{count} * 0.34442 / 2 ;$$

**Tabel 4.1** Data Pengukuran

No	Jarak sebenarnya (cm)	Jarak pada layar LCD / alat (cm)
1	10	14.9
2	20	28.2
3	30	41.2
4	40	56.8

Tabel 4.1 Tabel lanjutan data pengukuran

No	Jarak sebenarnya (cm)	Jarak pada layar LCD / alat (cm)
5	50	69.8
6	60	82.6
7	70	95.9
8	80	109.8
9	90	123.6
10	100	137.3
11	110	151.3
12	120	163.9
13	130	177.8
14	140	191.1
15	150	204.7
16	160	219.1
17	170	233.9
18	180	246.6
19	190	260.6
20	200	273.7
21	210	287.4
22	220	300.4
23	230	315.1
24	240	327.4
25	250	342.6
26	260	354.7
27	270	368.8
28	280	382.2
29	290	395.7
30	300	410.7



Dengan menggunakan pendekatan regresi linear maka dapat diperoleh nilai a dan b pada rumus diatas. Berikut perhitungannya :

$$a = \frac{[\sum XY \times N - \sum X \times \sum Y]}{[\sum x^2 \times N - (\sum X)^2]}$$

$$= 0,7340$$

Dan

$$b = \frac{[\sum x^2 \times \sum Y - \sum X \times \sum XY]}{[\sum x^2 \times N - (\sum X)^2]}$$

$$= -0,7742$$

Dimana X adalah jarak pada layar LCD dan Y adalah jarak sebenarnya. Berikut adalah grafik perbandingannya :



**Gambar 4.1** Grafik perbandingan jarak sebelum kalibrasi

Setelah mendapat nilai kalibrasi melalui rumus diatas maka, nilai tersebut dimasukan kedalam rumus perhitungan sebagai konstanta kalibrasi.

$$jarak = count * 0.34442/2 * 0.734 + (-0.7742);$$

dengan menggunakan rumus perhitungan diatas maka di dapat data pengukuran kalibrasi sebagai berikut:

**Tabel 4.2** Data Pengukuran Interval 10 cm

No	Jarak sebenarnya (cm)	Jarak pada layar LCD / alat (cm)
1	10	10,4
2	20	19,6
3	30	29,9
4	40	40,4
5	50	50,3
6	60	59,4
7	70	69
8	80	78,7
9	90	89
10	100	98,4
11	110	108
12	120	117,4
13	130	127,6
14	140	137,2
15	150	147,3
16	160	156,9
17	170	167,5
18	180	176,5
19	190	186,2

**Tabel 4.2** Tabel lanjutan Data Pengukuran interval 10 cm

No	Jarak sebenarnya (cm)	Jarak pada layar LCD / alat (cm)
20	200	196.3
21	210	206
22	220	216.5
23	230	226.6
24	240	235.9
25	250	245.6
26	260	257.7
27	270	265.8
28	280	275.4
29	290	288.2
30	300	291

Dari data di atas maka dapat diperoleh nilai error rata-rata:

$$\begin{aligned}
 \text{Rata-rata error} &= \frac{\sqrt{\sum ei^2}}{30} \\
 &= \frac{\sqrt{299.39}}{30} \\
 &= 0.577 \text{ cm}
 \end{aligned}$$

Untuk data interval 5cm dilakukan pengukuran pada jarak 5 cm – 150 cm, berikut data hasil pengukurannya :

No	Jarak sebenarnya (cm)	Jarak pada layar LCD / alat (cm)
1	5	5.1
2	10	10.4
3	15	15.2
4	20	19.6
5	25	24.4
6	30	29.9
7	35	35
8	40	40.4
9	45	44.5
10	50	50.3
11	55	54.9
12	60	59.4
13	65	64.3
14	70	69
15	75	74.9
16	80	78.7
17	85	84.2
18	90	88.3
19	95	93.6
20	100	98.4
21	105	103.1
22	110	108
23	115	112.9
24	120	117.4
25	125	123.6
26	130	127.6

Tabel 4.3 Data Pengukuran Interval 5 cm

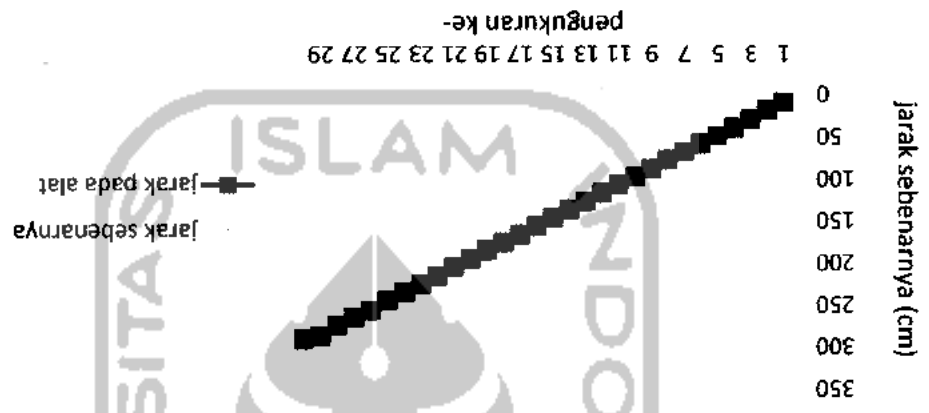
No	Jarak sebenarnya (cm)	Jarak pada layar LCD / alat (cm)
27	135	132.3
28	140	137.5
29	145	142.6
30	150	147.3

Dari data di atas maka dapat diperoleh nilai error rata-rata:

$$\text{Rata-rata error} = \frac{\sqrt{\sum e_i^2}}{30} = 0.274 \text{ cm}$$

Berikut adalah gambar grafik perbandingan jarak sebenarnya dengan jarak yang

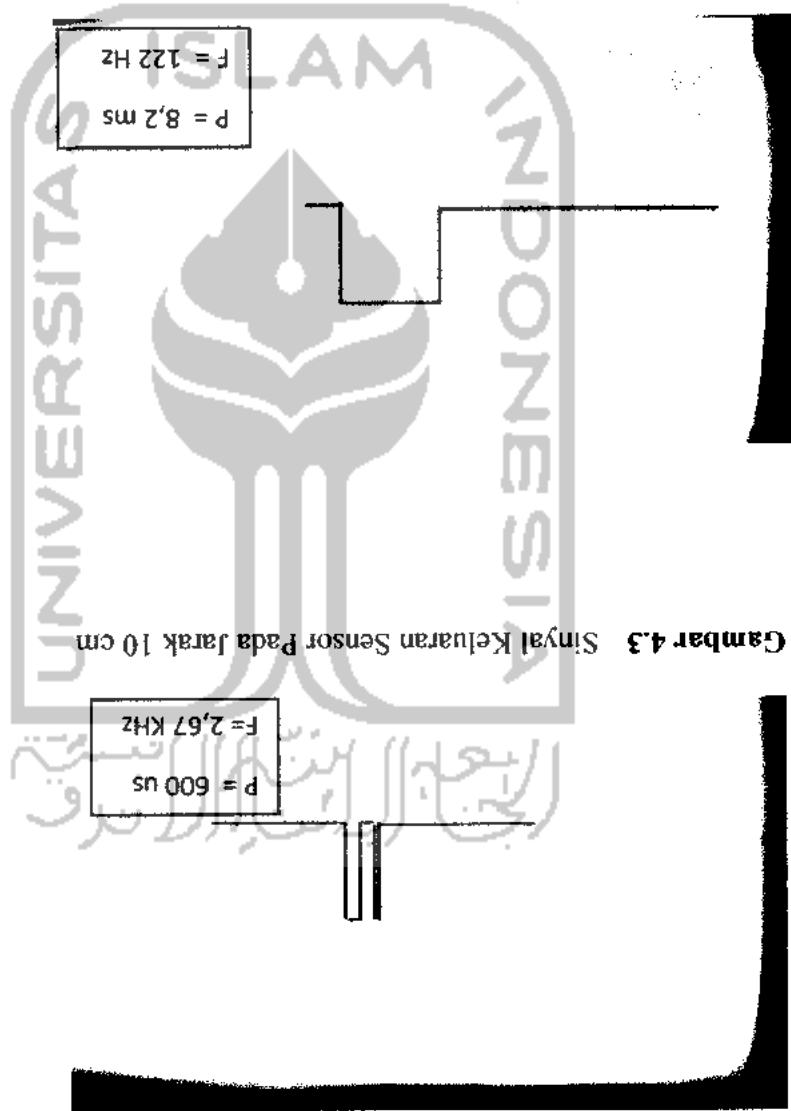
terbaca pada alat :



Gambar 4.2 grafik perbandingan alat setelah proses kalibrasi

Bentuk sinyal keluaran sensor pada saat melakukan pengukuran pada jarak

terentu adalah sebagai berikut :



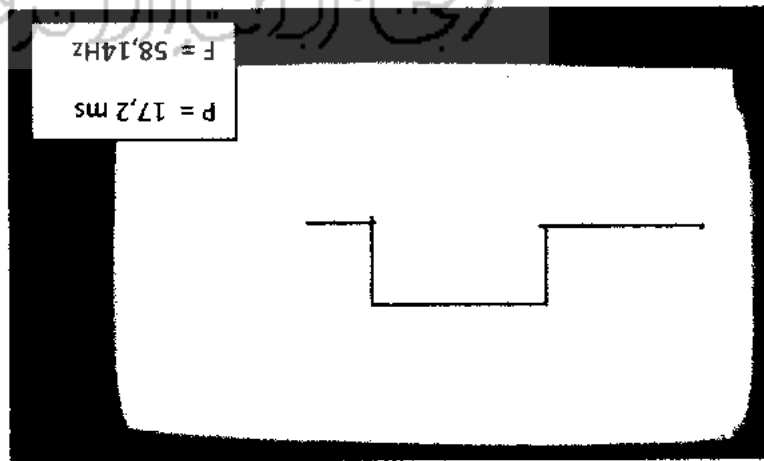
Gambar 4.4 Sinyal Keluaran Sensor pada Jarak 1,5 meter

Dari data pengukuran setelah proses kalibrasi, dapat di lihat pada jarak-jarak pendek sekitar jarak 10 cm sampai 100 cm, selisih kesalahan alat masih relatif kecil yaitu sekitar 0,4 cm sampai 1,6 cm, tetapi pada jarak menengah antara 100 cm sampai 290 cm selisih kesalahan alat berkisar antara 2 cm sampai 4,6 cm sedangkan pada jarak 300 cm selisihnya menjadi 9 cm. Hal ini dikarenakan karena beberapa faktor, yang pertama sensor PING hanya mampu membaca maksimal jarak 3 meter (300cm), yang kedua sensor akan mengalami gangguan pada saat mengukur obyek dengan sudut yang dangkal sehingga pantulannya tidak sempurna atau tidak menuju

#### 4.2 Analisa Data

Dari hasil pengujian dan pengamatan dapat diperoleh berupa sebuah sinyal pulsa kotak dengan frekuensi dan periode yang berbeda, sesuai dengan jarak sensor terhadap obyek dan tegangan puncak ke puncak adalah 5 volt. Lebar sinyal yang dihasilkan pada penerima dipengaruhi jarak sensor terhadap halangan.

Gambar 4. 5 Sinyal Keluaran Sensor Pada Jarak 3 meter



## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan perancangan sistem dan hasil analisa yang didapat maka dalam pembuatan *hardware* alat ukur jarak digital dapat disimpulkan beberapa hal, yaitu :

1. Pada pengukuran interval 10 cm nilai rata-rata *error* 0,577 cm sedangkan pada pengukuran interval 5 cm nilai rata-rata *error* 0,274 cm.
2. Alat bekerja baik pada keadaan suhu normal, atau masih kurang maksimal pada saat cuaca mendung atau tingkat kelembaban udara cukup tinggi.
3. Modul sensor PING bekerja berdasarkan prinsip pemantulan gelombang ultrasonik, terkadang gelombang ultrasonic mengalami gangguan seperti interferensi dari gelombang lain atau mendapat pantulan dari benda lain yang menyebabkan pengukuran tidak akurat.
4. Dengan adanya fitur tambahan berupa penyimpanan data hasil pengukuran, maka dapat mempermudah dalam melakukan pengamatan hasil percobaan.

#### 5.2 Saran

Untuk mengembangkan sistem dimasa yang akan datang maka dapat disarankan beberapa hal sebagai berikut :

1. Perlu adanya sumber daya yang tidak selalu terhubung dengan sumber tegangan, agar lebih praktis.



2. Sensor dapat diganti dengan sensor jenis lain yang jarak ukur atau jangkauannya lebih jauh, dan tidak terlalu terpengaruh oleh kelembaban udara.
3. Jika memungkinkan perlu adanya detector kelembaban, agar bisa mengetahui batas kelembaban udara alat mampu bekerja dengan baik.



## DAFTAR PUSTAKA

Atmel Corporation. *ATMega 8535 Datasheet*. 11 juni 2009.

[http://www.datasheetcatalog.com/datasheets\\_pdf/A/1/8/9/ATMega8535.html](http://www.datasheetcatalog.com/datasheets_pdf/A/1/8/9/ATMega8535.html)

Confile Technology. 2007. *Keypad switch 3x4 Data Sheet*.

<http://www.confiletech.com>. 12 juli 2009.

Firdaus Untung, 2006 “*Pembuatan Alat Ukur Jarak Digital Menggunakan Gelombang Ultrasonik*”, Teknik Elektro UII.

Nalwan Andi Paulus. 2004. *Panduan Praktis Penggunaan dan Antarmuka Modul LCD M1632*. Jakarta: Elek Media Komputindo.

Parallax, *Ping datasheet*, 7 september 2009

[http://www.parallax.com/Portals/0/Downloads/docs/prod/acc\\_28015-PING-v1.6.pdf](http://www.parallax.com/Portals/0/Downloads/docs/prod/acc_28015-PING-v1.6.pdf)

Winoto Ardi. 2008. *Mikrokontroler AVR ATmega8/32/16/8535 dan pemrograman dengan Bahasa C pada WinAVR*. Bandung: Informatika.

program

```
/**_S*****
```

```
*****
```

```
/
/ File Name      : 'main.c'
/ Title         : LCD demo
/ Author        : Scienceprog.com - Copyright (C) 2007
/ Created       : 2007-03-29
/ Revised      : 2007-08-28
/ Version       : 1.0
/ Target MCU    : Atmel AVR series
```

This code is distributed under the GNU Public License  
which can be found at

[tp://www.gnu.org/licenses/gpl.txt](http://www.gnu.org/licenses/gpl.txt)

```
*****
```

```
*****
```

```
include <avr/io.h>
include <avr/pgmspace.h>
include <avr/eeprom.h>
include <util/delay.h>
clude <inttypes.h>
clude <avr/pgmspace.h>
clude <stdio.h>
clude <stdlib.h>
clude <string.h>
clude <math.h>
clude <util/delay.h>
```

```
fine simpanposisi 0
```

```
uint8_t buffer[15];
uint16_t measure[5];
uint8_t i, pospul, posat;
```

```
uint16_t ov_counter, pos, jarak, count, distance, average, ukurke; //define
```

```
float counter
```

```
define INP      7 //PD7
```

```
strings stored in AVR Flash memory.
uint8_t welcome[] PROGMEM="METERAN DIGITAL\0";
```

```
uint8_t keypad(void){
```

```
    //first column
```

```
    PORTB =0b11111110;
    PORTB =0b11111110;
    if (bit_is_clear(PINB, 4)) return 15;
    if (bit_is_clear(PINB, 5)) return 14;
    if (bit_is_clear(PINB, 6)) return 13;
    if (bit_is_clear(PINB, 7)) return 12;
    //third column
```

program

```
PORTB =0b11111101;
PORTB =0b11111101;
if (bit_is_clear(PINB, 4)) return 11;
if (bit_is_clear(PINB, 5)) return 9;
if (bit_is_clear(PINB, 6)) return 6;
if (bit_is_clear(PINB, 7)) return 3;
//third column

PORTB =0b11111011;
PORTB =0b11111011;
if (bit_is_clear(PINB, 4)) return 0;
if (bit_is_clear(PINB, 5)) return 8;
if (bit_is_clear(PINB, 6)) return 5;
if (bit_is_clear(PINB, 7)) return 2;

//fourth column

PORTB =0b11110111;
PORTB =0b11110111;
if (bit_is_clear(PINB, 4)) return 10;
if (bit_is_clear(PINB, 5)) return 7;
if (bit_is_clear(PINB, 6)) return 4;
if (bit_is_clear(PINB, 7)) return 1;
return 16;
}
```

```
t16_t baca_ping(void){
```

```
    DDRD |= (1<<INP);
    PORTD |= (1<<INP);
    _delay_us(7);
    PORTD &= ~(1<<INP);
    DDRD &= ~(1<<INP);
    PORTD |= (1<<INP);
    while (bit_is_clear(PIND,INP)) {};
    while (!bit_is_clear(PIND,INP))
    {
        count++;
    }
    arak= count * 0.34442/2* 0.734 - 0.7742 ; //rumus mencari
    k
    t=0;
    rn jarak;
```

```
main(void)
```

```
    LCDinit();//init LCD bit, dual line, cursor right
    LCDclr();//clears LCD
```

```
    DDRB=0x0f;
    PORTB=0xff;
    CopyStringtoLCD(welcome, 0, 0);
```

```

                                program
ukurke=eeprom_read_byte((uint8_t*)simpanposisi);
ukurke--;
eeprom_write_byte((uint8_t*)simpanposisi,ukurke);
//ukurke=ukurke*2;
pos=ukurke*2;
average = eeprom_read_word((uint16_t*)pos);
sprintf(buffer,"P=%02i;JRK:%imm

i",ukurke,average);//RPM/T);
LCDGotoXY(0,1);
LCDstring(buffer,15);
}

if(keypad()==15){
_delay_ms(100);
while(keypad()==15){}
ukurke=eeprom_read_byte((uint8_t*)simpanposisi);
ukurke++;
eeprom_write_byte((uint8_t*)simpanposisi,ukurke);
//ukurke=ukurke*2;
pos=ukurke*2;
average = eeprom_read_word((uint16_t*)pos);
sprintf(buffer,"P=%02i;JRK:%imm

,ukurke,average);//RPM/T);
LCDGotoXY(0,1);
LCDstring(buffer,15);
}

if(keypad()==10){
_delay_ms(100);
while(keypad()==10){}

LCDcursorOnBlink();
LCDGotoXY(2, 1);
while(keypad()==16){}

pospul=keypad();
LCDsendChar(keypad()+0x30);
while(keypad()<10){}

while(keypad()<10){}
while(keypad()==16){}
posat=keypad();
LCDsendChar(keypad()+0x30);
while(keypad()<10){}
while(keypad()==16){}

if (keypad()==11){
while(keypad()==11){}
ukurke=pospul*10+posat;
eeprom_write_byte((uint8_t*)simpanposisi,ukurke);
pos=ukurke*2;
average = eeprom_read_word((uint16_t*)pos);
sprintf(buffer,"P=%02i;JRK:%imm

ikurke,average);//RPM/T);
LCDGotoXY(0,1);
CDstring(buffer,15);
}

if (keypad()==10){

```

```

                                program
while(keypad()==10){
    pos=ukurke*2;
    average = eeprom_read_word((uint16_t*)pos);
    sprintf(buffer,"P=%02i;JRK:%imm
)",ukurke,average);//RPM/T);
    LCDGotoXY(0,1);
    LCDstring(buffer,15);
}

}

}

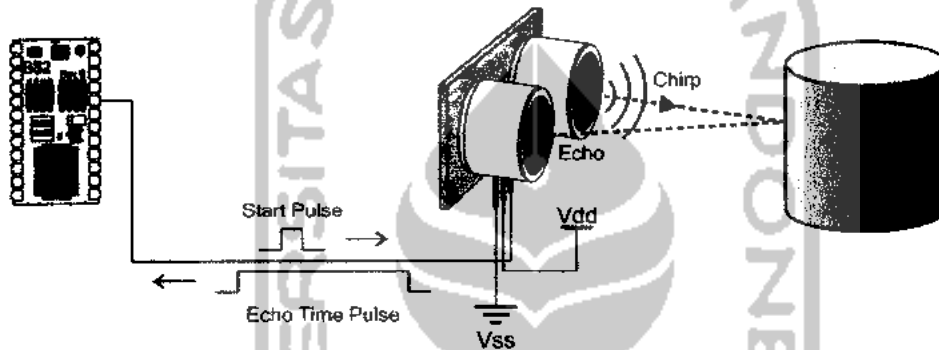
```



## PING)))™ Ultrasonic Distance Sensor (#28015)

The Parallax PING))) ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to microcontrollers such as the BASIC Stamp®, SX or Propeller chip, requiring only one I/O pin.

The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.



### Features

- Range: 2 cm to 3 m (0.8 in to 3.3 yd)
- Burst indicator LED shows sensor activity
- Bidirectional TTL pulse interface on a single I/O pin can communicate with 5 V TTL or 3.3 V CMOS microcontrollers
- Input trigger: positive TTL pulse, 2  $\mu$ s min, 5  $\mu$ s typ.
- Echo pulse: positive TTL pulse, 115  $\mu$ s minimum to 18.5 ms maximum.
- RoHS Compliant

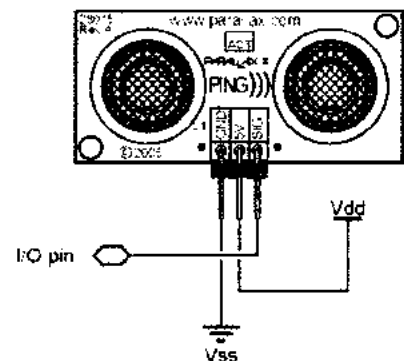
### Key Specifications

- Supply voltage: +5 VDC
- Supply current: 30 mA typ; 35 mA max
- Communication: Positive TTL pulse
- Package: 3-pin SIP, 0.1" spacing (ground, power, signal)
- Operating temperature: 0 – 70° C.
- Size: 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Weight: 9 g (0.32 oz)

### Pin Definitions

GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

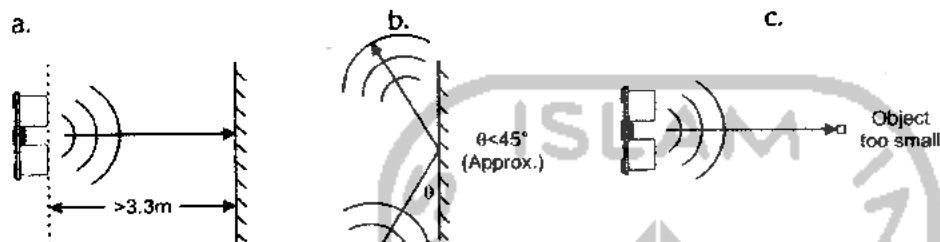
The PING))) sensor has a male 3-pin header used to supply ground, power (+5 VDC) and signal. The header may be plugged directly into solderless breadboard, or into a standard 3-wire extension cable (Parallax part #805-000012).



## Practical Considerations for Use

### Object Positioning

The PING))) sensor cannot accurately measure the distance to an object that: a) is more than 3 meters away, b) that has its reflective surface at a shallow angle so that sound will not be reflected back towards the sensor, or c) is too small to reflect enough sound back to the sensor. In addition, if your PING))) sensor is mounted low on your device, you may detect sound reflecting off of the floor.



### Target Object Material

In addition, objects that absorb sound or have a soft or irregular surface, such as a stuffed animal, may not reflect enough sound to be detected accurately. The PING))) sensor will detect the surface of water, however it is not rated for outdoor use or continual use in a wet environment. Condensation on its transducers may affect performance and lifespan of the device.

### Air Temperature

Temperature has an effect on the speed of sound in air that is measurable by the PING))) sensor. If the temperature (°C) is known, the formula is:

$$C_{air} = 331.5 + (0.6 \times T_c) \text{ m/s}$$

The percent error over the sensor's operating range of 0 to 70 ° C is significant, in the magnitude of 11 to 12 percent. The use of conversion constants to account for air temperature may be incorporated into your program (as is the case in the example BS2 program given in the Example Programs section below). Percent error and conversion constant calculations are introduced in Chapter 2 of *Smart Sensors and Applications*, a Stamps in Class text available for download from the 28029 product page at [www.parallax.com](http://www.parallax.com).

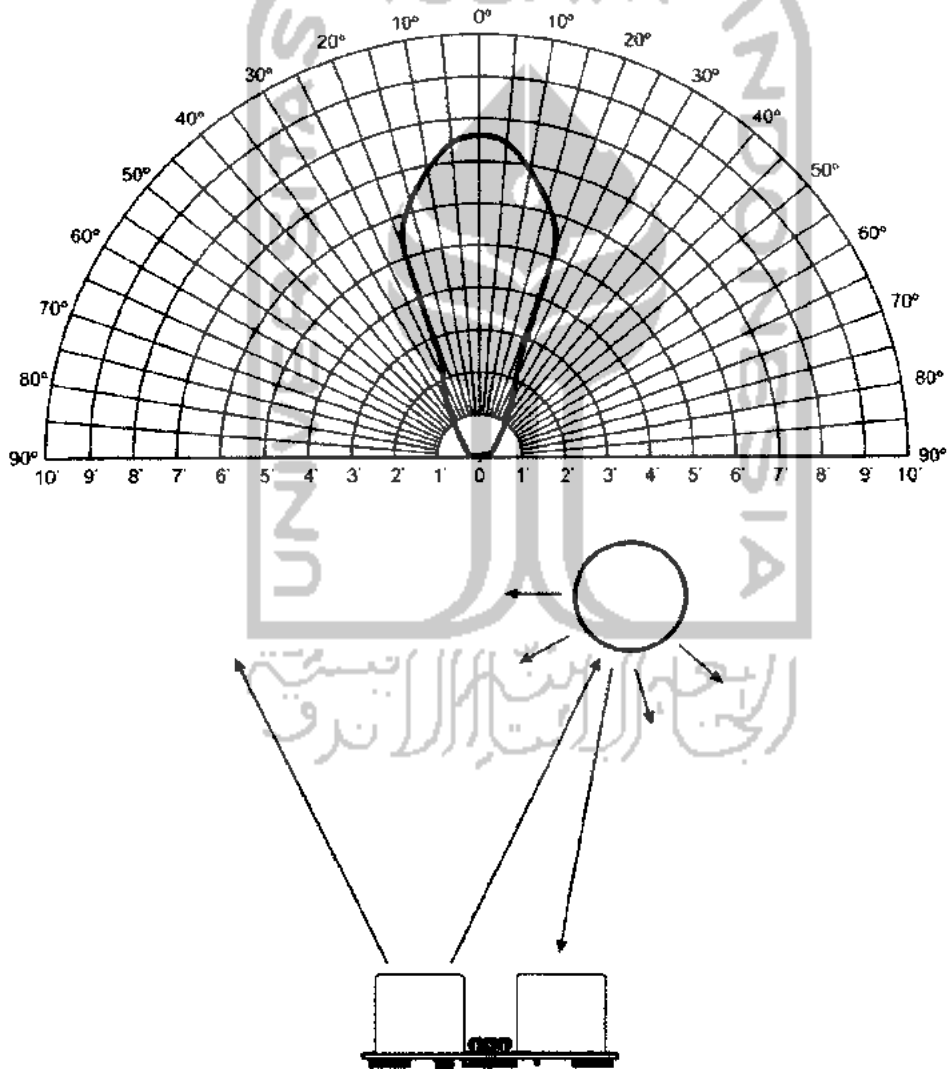


## Test Data

The test data on the following pages is based on the PING))) sensor, tested in the Parallax lab, while connected to a BASIC Stamp microcontroller module. The test surface was a linoleum floor, so the sensor was elevated to minimize floor reflections in the data. All tests were conducted at room temperature, indoors, in a protected environment. The target was always centered at the same elevation as the PING))) sensor.

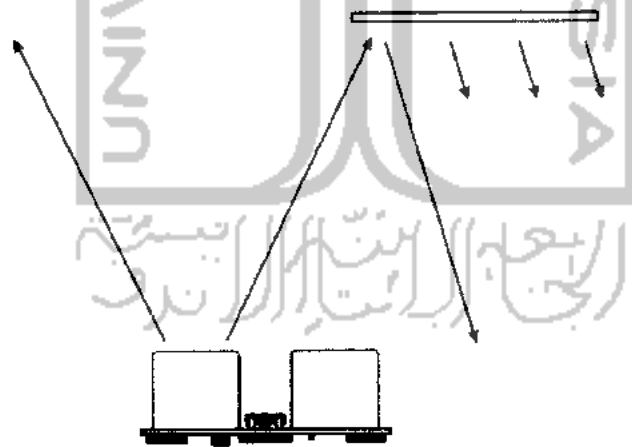
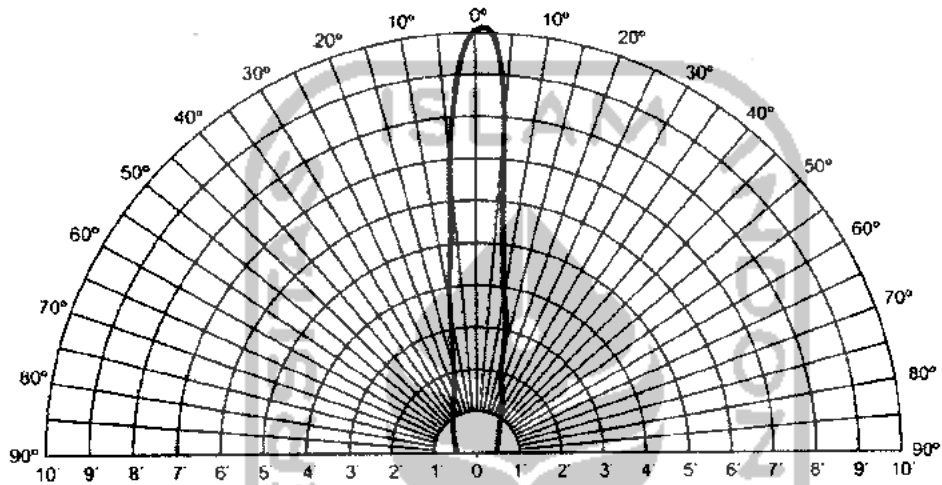
### Test 1

Sensor Elevation: 40 in. (101.6 cm)  
Target: 3.5 in. (8.9 cm) diameter cylinder, 4 ft. (121.9 cm) tall – vertical orientation



## Test 2

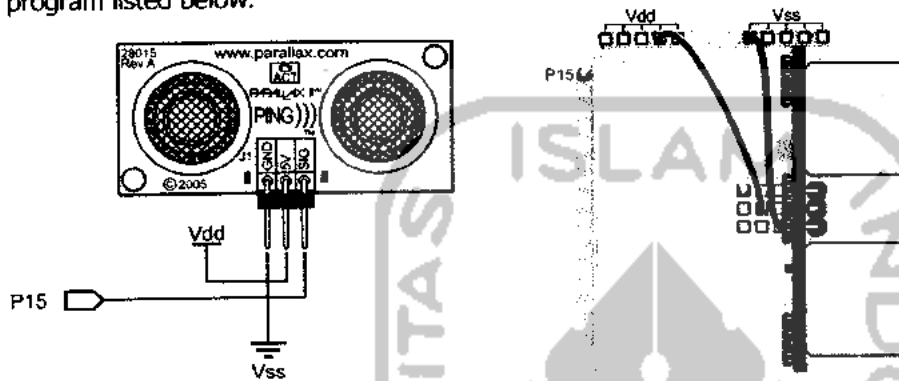
Sensor Elevation: 40 in. (101.6 cm)  
Target: 12 in. x 12 in. (30.5 cm x 30.5 cm) cardboard, mounted on 1 in. (2.5 cm) pole  
Target positioned parallel to backplane of sensor



## Example Programs and Applications

### BASIC Stamp 2

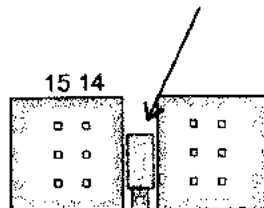
This circuit allows you to quickly connect your PING))) sensor to a BASIC Stamp<sup>®</sup> 2 via the Board of Education<sup>®</sup> breadboard area. The PING))) module's GND pin connects to Vss, the 5 V pin connects to Vdd, and the SIG pin connects to I/O pin P15. This circuit will work with the example BASIC Stamp program listed below.



### Extension Cable and Port Cautions for the Board of Education

If you are connecting your PING))) sensor to a Board of Education platform using an extension cable, follow these steps:

1. When plugging the cable onto the PING))) sensor, connect Black to GND, Red to 5 V, and White to SIG.
2. Check to see if your Board of Education servo ports have a jumper, as shown at right.
3. If your Board of Education servo ports have a jumper, set it to Vdd as shown. Then plug the cable into the port, matching the wire color to the labels next to the port.
4. If your Board of Education servo ports do not have a jumper, **do not use them with the PING))) sensor**. These ports only provide Vin, not Vdd, and this may damage your PING))) sensor. Go to the next step.
5. Connect the cable directly to the breadboard with a 3-pin header as shown above. Then, use jumper wires to connect Black to Vss, Red to Vdd, and White to I/O pin P15.



Board of Education Servo Port Jumper, Set to Vdd

## Features

High-performance, Low-power AVR<sup>®</sup> 8-bit Microcontroller  
Advanced RISC Architecture

- 130 Powerful Instructions - Most Single Clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 16 MIPS Throughput at 16 MHz
- On-chip 2-cycle Multiplier
- Non-volatile Program and Data Memories
  - 8K Bytes of In-System Self-Programmable Flash
    - Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - 512 Bytes EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 512 Bytes Internal SRAM
  - Programming Lock for Software Security**Peripheral Features**
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
    - 8-channel, 10-bit ADC
      - 8 Single-ended Channels
      - 7 Differential Channels for TQFP Package Only
      - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator**Special Microcontroller Features**
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby**and Packages**
  - 32 Programmable I/O Lines
  - 10-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad QFN/MLF**Operating Voltages**
  - 1.7 - 5.5V for ATmega8535L
  - 1.5 - 5.5V for ATmega8535**Speed Grades**
  - 8 MHz for ATmega8535L
  - 16 MHz for ATmega8535



**8-bit AVR<sup>®</sup>**  
**Microcontroller**  
**with 8K Bytes**  
**In-System**  
**Programmable**  
**Flash**

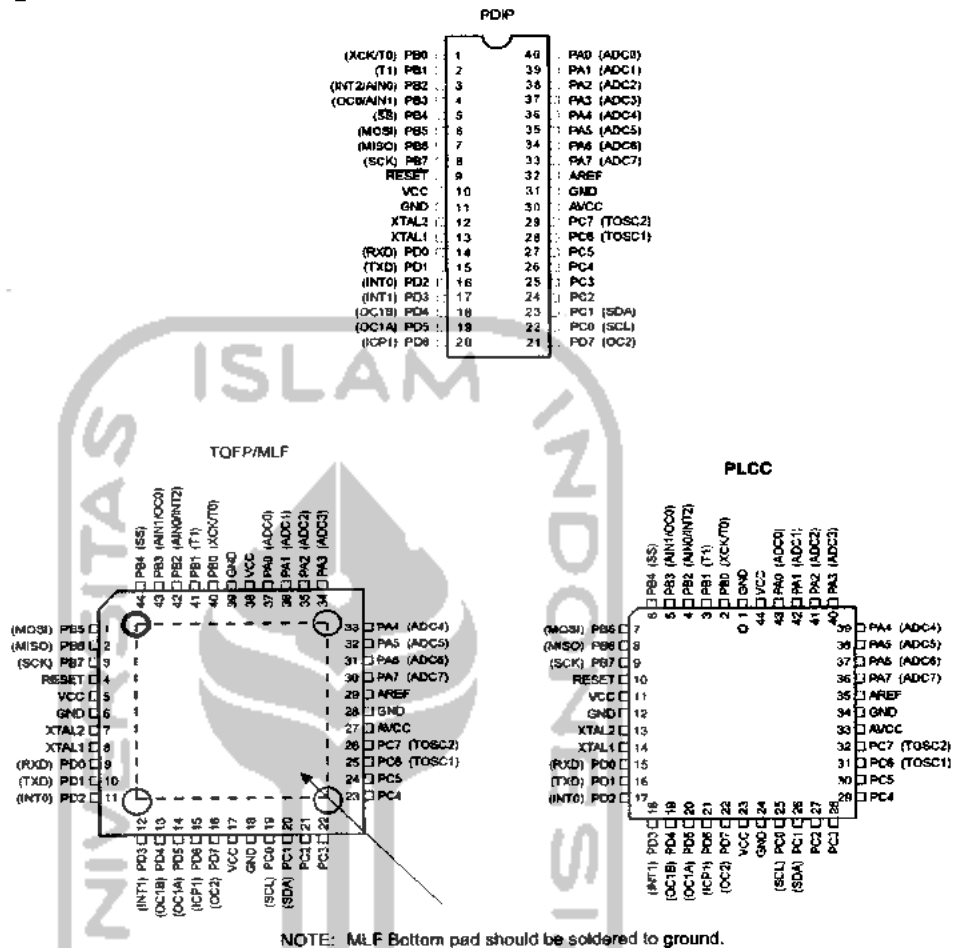
**ATmega8535**  
**ATmega8535L**





# n Configurations

Figure 1. Pinout ATmega8535



aimer

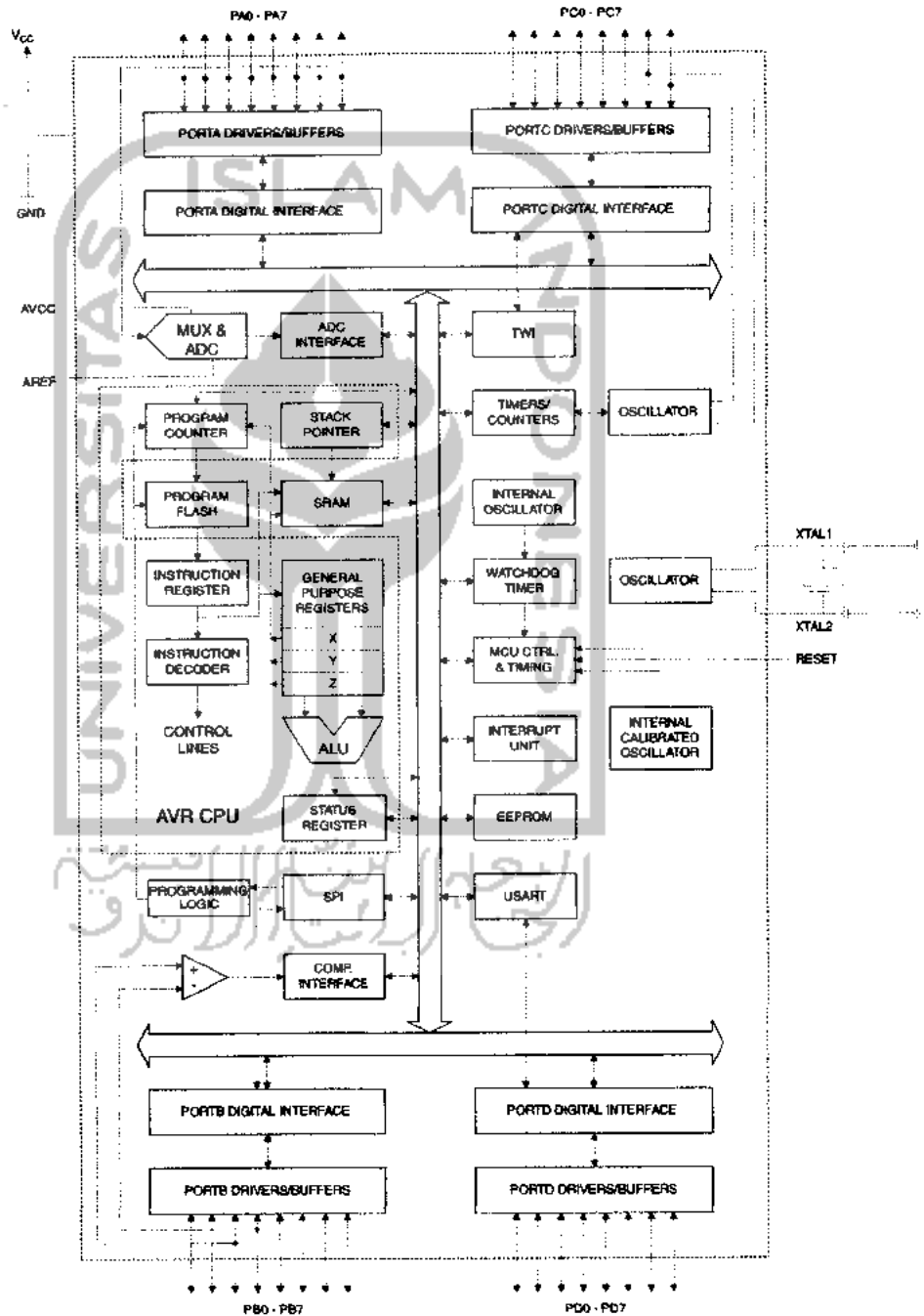
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

## Overview

The ATmega8535 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing instructions in a single clock cycle, the ATmega8535 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## Block Diagram

Figure 2. Block Diagram





The AVR core combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8535 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 512 bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain in TQFP package, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the asynchronous timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega8535 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega8535 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators, and evaluation kits.

### **S8535 Compatibility**

The ATmega8535 provides all the features of the AT90S8535. In addition, several new features are added. The ATmega8535 is backward compatible with AT90S8535 in most cases. However, some incompatibilities between the two microcontrollers exist. To solve this problem, an AT90S8535 compatibility mode can be selected by programming the S8535C fuse. ATmega8535 is pin compatible with AT90S8535, and can replace the AT90S8535 on current Printed Circuit Boards. However, the location of fuse bits and the electrical characteristics differs between the two devices.

### **S8535 Compatibility**

Programming the S8535C fuse will change the following functionality:

- The timed sequence for changing the Watchdog Time-out period is disabled. See "Timed Sequences for Changing the Configuration of the Watchdog Timer" on page 44 for details.
- The double buffering of the USART Receive Register is disabled. See "AVR USART vs. AVR UART – Compatibility" on page 144 for details.

## **ATmega8535(L)**

**Descriptions**

:	Digital supply voltage.
D	Ground.
A (PA7..PA0)	<p>Port A serves as the analog inputs to the A/D Converter.</p> <p>Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
B (PB7..PB0)	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega8535 as listed on page 59.</p>
C (PC7..PC0)	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega8535 as listed on page 63.</p>
RST	<p>Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 36. Shorter pulses are not guaranteed to generate a reset.</p> <p>Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.</p> <p>Output from the inverting Oscillator amplifier.</p> <p>AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to <math>V_{CC}</math>, even if the ADC is not used. If the ADC is used, it should be connected to <math>V_{CC}</math> through a low-pass filter.</p> <p>AREF is the analog reference pin for the A/D Converter.</p>