



**PENGAMAN PARKIR MOBIL MENGGUNAKAN
SENSOR JARAK BERBASIS MIKROKONTROLER**

TUGAS AKHIR

Untuk Memperoleh Gelar Ahli Mada pada
Program Diploma III Teknik Elektro
Jurusan Teknik Elektro – Fakultas Teknik
Universitas Negeri Semarang

PERPUSTAKAAN
UNNES

Oleh
Heru Rianto
5350306042

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
2010**

PENGESAHAN

Tugas Akhir ini telah dipertahankan dihadapan Panitia Ujian Tugas Akhir Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang pada tanggal Februari 2010

Panitia :

Ketua

Sekretaris

Drs.Djoko Adi Widodo, M.T

Drs. Agus Murnomo, M.T

NIP 195909271986011001

NIP 195506061986031002

Penguji I/Pembimbing

Penguji II

Drs. Agus Murnomo, M.T

Drs.Suryono, M.T

NIP 195506061986031002

NIP 195503161985031001

Mengetahui

Dekan Fakultas Teknik

Drs. Abdurrahman, M. Pd

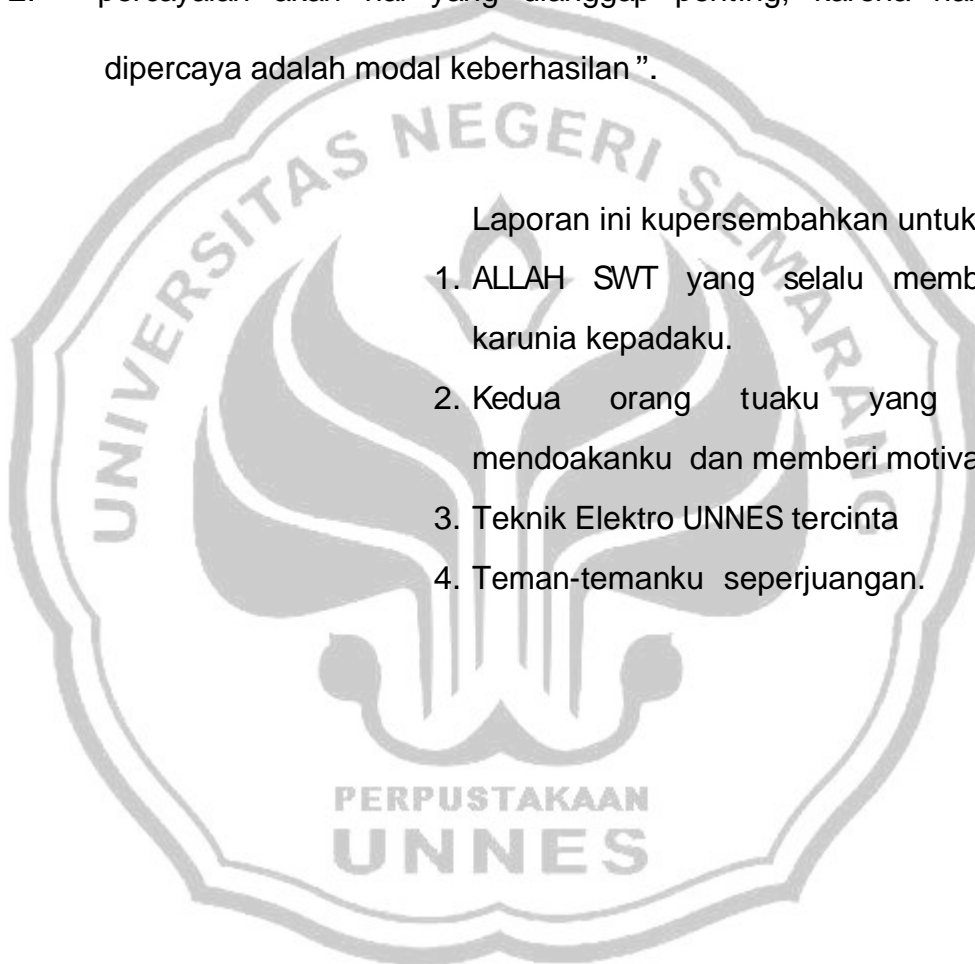
NIP 196009031985031002

MOTTO DAN PERSEMBAHAN

1. “ memberi semua yang kita punya bukanlah hal yang terbaik, tapi memberi yang dibutuhkan adalah yang terbaik”.
2. “ percayalah akan hal yang dianggap penting, karena hal yang dipercaya adalah modal keberhasilan”.

Laporan ini kupersembahkan untuk :

1. ALLAH SWT yang selalu memberikan karunia kepadaku.
2. Kedua orang tuaku yang selalu mendoakanku dan memberi motivasi.
3. Teknik Elektro UNNES tercinta
4. Teman-temanku seperjuangan.



KATA PENGANTAR

Puji syukur kupanjatkan kepada ALLAH SWT yang telah melimpahkan rahmat dan hidayahnya, sehingga penulis dapat menyelesaikan laporan tugasakhir ini dengan judul *“Pengaman parkir mobil menggunakan sensor jarak berbasis mikrokontroler”*, sebagai syarat menempuh jenjang Diploma III Teknik Elektro Universitas Negeri Semarang.

Penulisan laporan tugas akhir ini tidak lepas dari pemikiran dan bantuan dari berbagai pihak. Oleh karena itu penulis menyampaikan ucapan terima kasih kepada Yth :

1. Drs. Abdurrahman, M.Pd. selaku Dekan Fakultas Teknik Universitas Negeri Semarang.
2. Drs. Joko Adi Widodo, M.T selaku Ketua Jurusan Teknik Elektro Universitas Negeri Semarang.
3. Drs. Agus Murnomo, M.T selaku Ketua Program Studi DIII Teknik Elektro Universitas Negeri Semarang dan selaku dosen pembimbing yang telah membantu memberikan bimbingan terbaik.
4. Kedua orang tuaku yang selalu mendukungku dan mendoakanku
5. Semua pihak yang telah membantu menyelesaikan Tugas Akhir yang tidak dapat penulis sebutkan satu-persatu.

Semarang, 2010

Penulis

ABSTRAK

Rianto, Heru, 2009. "*Pengaman parkir mobil menggunakan sensor jarak berbasis mikrokontroler*". Tugas Akhir, Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang. Drs Agus Murnomo, MT

Perkembangan ilmu pengetahuan dan teknologi telah maju pesat terutama dalam bidang elektronika digital. Sistem digital berkembang dengan adanya sistem mikrokontroler. *DT-SENSE* merupakan suatu sensor ultrasonic yang bekerja mendeteksi jarak. *DT-SENSE* banyak digunakan dalam dunia otomatisasi dan robotika, misal sebagai pendeteksi lingkungan sekitar robot, sebagai pengukur jarak, dan juga sebagai pengukur tinggi suatu benda. Mobil merupakan sarana transportasi yang memudahkan manusia dalam melakukan aktifitas perjalanan. Perkembangan teknologi dan ilmu pengetahuan yang semakin canggih telah banyak menciptakan mobil-mobil mewah yang dilengkapi dengan fasilitas elektronika seperti Televisi dan mini notebook serta peralatan elektronik lainnya. Tetapi dengan fasilitas yang demikian, mobil-mobil tersebut belum dilengkapi suatu alat pengaman pada saat melakukan parkir. Sehingga dapat dimungkinkan terjadi benturan pada saat melakukan parkir, misal membentur pembatas parkir atau menyentuh mobil yang ada di sampingnya. *DT-SENSE* juga dapat digunakan sebagai pengaman jarak parkir mobil. Berangkat dari pemikiran di atas, penulis berinisiatif untuk membuat pengaman parkir mobil menggunakan sensor jarak berbasis mikrokontroler.

Penyusunan laporan tugas akhir ini berdasarkan pada metode observasi, metode interview, metode dokumentasi dan metode studi pustaka.

Hasil yang diperoleh adalah pada saat sensor mendeteksi jarak yang telah ditentukan maka akan terdengar bunyi buzzer yang menandakan jarak aman parkir. Jarak aman yang diharapkan adalah kurang dari 30cm untuk bagian depan dan 40cm untuk bagian belakang terhadap batas parkir, akan tetapi responsifitas jarak aman yang dihasilkan adalah 30,74 – 30,80cm untuk bagian depan dan 41,20 – 41,22cm untuk bagian belakang terhadap pembatas parkir.

Kesimpulan perencanaan dan pembuatan pesawat simulasi ini adalah bahwa Mikrokontroler *ATMega8535* bisa digunakan sebagai pengaman jarak parkir dengan menggunakan *DT-SENSE* sebagai sensor jarak, Sehingga dapat mengetahui jarak aman untuk melakukan parkir dengan menggunakan buzzer sebagai tanda peringatan. Keuntungan dari penggunaan mikrokontroler sebagai pengendalinya adalah mudah dalam pemrograman, mengubah dan koreksi kesalahan programnya. Mudah dalam pemeliharaan dan perbaikan.

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	ii
MOTTO DAN PERSEMBAHAN.....	iii
KATA PENGANTAR.....	iv
ABSTRAK	v
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	ix
BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Permasalahan.....	2
C. Batasan Masalah.....	2
D. Tujuan.....	2
E. Manfaat.....	2
BAB II PEMBUATAN DAN PEMBAHASAN	
A. Landasan Teori	3
1. Mikrokontroler	3
a. Pengertian Mikrokontroler.....	3
b. Mikrokontroler Atmega8535.....	5
2. Sensor Ultrasonic	10
a. Pengertian	10
b. Prinsip Kerja Sensor Ultrasonic.....	10

B. Metode Pembuatan Alat	12
1. Perencanaan Pembuatan Perangkat Lunak (<i>software</i>).....	12
2. Perencanaan Pembuatan Perangkat Keras (<i>Hardware</i>).....	13
a. Perancangan Pembuatan Perangkat Mikrokontroler.....	13
b. Perancangan Pembuatan <i>Downloader</i>	16
c. Prosedur Pembuatan.....	18
C. Pengujian	19
D. Pembahasan	20
BAB III PENUTUP	
A. Kesimpulan.....	21
B. Saran.....	21
DAFTAR PUSTAKA	22
LAMPIRAN	

DAFTAR GAMBAR

Gambar 01. Konfigurasi PIN ATmega8535	6
Gambar 02. Arsitektur ATmega8535	9
Gambar 03. Prinsip Kerja Sensor Ultrasonic	11
Gambar 04. <i>Flow chart</i> Alat Penagaman Parkir	12
Gambar 05. Skema Rangkaian Sistem Minimum Mikroontroler.....	13
Gambar 06. a. <i>Layout</i> PCB Sistem Minimum.....	15
b. Tata letak komponen.....	15
Gambar 07. Skema rangkaian Downloader.....	16
Gambar 08. a. <i>Layout</i> PCB Downloader.....	17
b. Tata letak komponen.....	17



DAFTAR TABEL

Tabel 1. Deskripsi PIN ATmega8535.....	8
Tabel 2. Hasil Pengujian Kerja Alat.....	19



BAB I

PENDAHULUAN

A. Latar Belakang

Mobil merupakan sarana transportasi yang memudahkan manusia dalam melakukan aktivitas perjalanan. Mobil juga dapat mengakibatkan kerugian bagi pengendaranya karena kelalaian pengendara itu sendiri. Sebagai contoh pada saat melakukan parkir, apabila tidak hati-hati dalam mengemudikan mobil seringkali mobil menabrak pembatas parkir. Hal ini mungkin disebabkan karena kurang jelasnya jarak pandang pengemudi dalam melihat jarak parkir.

Perkembangan teknologi dan ilmupengetahuan yang semakin canggih ini telah banyak menciptakan mobil-mobil mewah yang dilengkapi dengan fasilitas elektronika seperti Televisi dan mini notebook serta peralatan elektronik lainnya. Akan tetapi dengan fasilitas yang demikian, mobil-mobil tersebut belum dilengkapi suatu alat pengaman pada saat melakukan parkir. Sehingga dapat dimungkinkan terjadi kecelakaan atau kerugian pada saat melakukan parkir, misal menabrak pembatas parkir atau menyentuh mobil yang ada di sampingnya.

Sensor tipe “DT-SENSE” merupakan suatu sensor yang dapat mendeteksi jarak. Prinsip kerja sensor ultrasonik yaitu Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari pemancar (*transmitter*) ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh penerima (*receiver*) ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler

Berangkat dari masalah tersebut diatas maka penulis berinisiatif untuk membuat suatu alat pengaman parkir menggunakan sensor tipe “DT-SENSE” sensor sebagai pendeteksi jarak aman saat melakukan parkir.

B. Permasalahan

Berdasarkan pada latar belakang diatas yang mungkin akan menjadi masalah adalah Apakah alat pengaman parkir responsif sesuai jarak yang telah ditentukan dan mampu memberikan keamanan bagi pengemudi ?

C. Batasan Masalah

1. Sistem program menggunakan mikrokontroller ATmega8535 untuk mengoperasikan dan membaca data sensor ultrasonic
2. Sensor yang digunakan adalah sensor ultrasonic tipe DT-SENSE
3. Alat pengaman parkir bekerja dengan membunyikan alarm apabila mendeteksi jarak yang telah ditentukan
4. Alat ini hanya digunakan pada mobil roda empat

D. Tujuan

Tujuan yang ingin dicapai adalah merancang dan membuat alat pengaman parkir yang responsive sesuai dengan jarak yang telah ditentukan, sehingga dapat memberikan keamanan bagi pengemudi.

E. Manfaat

Adapun manfaat dari pembuatan alat pengaman parkir mobil menggunakan sensor jarak berbasis mikrocontroler adalah:

1. Sebagai informasi pengembangan ilmu pengetahuan dan teknologi di dunia pendidikan khususnya dan masyarakat pada umumnya
2. Digunakan sebagai sarana pembelajaran otomatisasi didalam bidang instrumentasi kendali.
3. Sebagai sarana pengaman parkir mobil roda empat

BAB II PEMBUATAN DAN PEMBAHASAN

A. Landasan Teori

1. Mikrokontroler

a. Pengertian Mikrokontroler

Mikrokontroler adalah salah satu bagian dasar suatu sistem komputer. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan komputer *mainframe*, mikrokontroler dibangun dari elemen-elemen dasar yang sama. Secara sederhana, komputer akan menghasilkan *output* spesifik berdasarkan *input* yang diterima dan program yang dikerjakan. (<http://mikrokontroler.tripod.com/6805/bab1.htm>)

Seperti umumnya komputer, mikrokontroler sebagai alat yang mengerjakan perintah-perintah yang diberikan oleh manusia. Artinya, bagian terpenting dan utama dari suatu sistem komputerisasi adalah program itu sendiri yang dibuat oleh seorang *programmer*. Program ini memerintahkan komputer untuk melakukan jalinan yang panjang dari aksi-aksi sederhana untuk melakukan tugas yang lebih kompleks yang diinginkan oleh *programmer*.

Sistem dengan mikrokontroler umumnya menggunakan piranti *input* yang jauh lebih kecil seperti saklar atau *keypad* kecil.

Hampir semua *input* mikrokontroler hanya dapat memproses sinyal *input digital* dengan tegangan yang sama dengan tegangan logika dari sumber. Tegangan positif sumber umumnya adalah 5 volt. Padahal dalam dunia nyata terdapat banyak sinyal *analog* atau sinyal dengan tegangan level yang bervariasi. Karena itu ada piranti *input* yang mengkonversikan sinyal *analog* menjadi sinyal *digital* sehingga komputer bisa mengerti dan menggunakannya. Ada beberapa

mikrokontroler yang dilengkapi dengan piranti konversi ini, yang disebut dengan ADC, dalam satu rangkaian terpadu.

Mikrokontroler ATmega8535 merupakan salah satu mikrokontroler buatan AVR yang memiliki fasilitas-fasilitas yang cukup lengkap, diantaranya :

- 1) *Flash* adalah suatu jenis *Read Only Memory* yang biasanya diisi dengan program hasil buatan manusia yang harus dijalankan oleh mikrokontroler.
- 2) RAM (*Random Acces Memory*) merupakan memori yang membantu CPU untuk penyimpanan data sementara dan pengolahan data ketika program sedang *running*.
- 3) EEPROM (*Electrically Erasable Programmable Read Only Memory*) adalah memori untuk penyimpanan data secara permanen oleh program yang sedang *running*.
- 4) Port I/O adalah kaki untuk jalur keluar atau masuk sinyal sebagai hasil keluaran ataupun masukan bagi program.
- 5) *Timer* adalah modul dalam *hardware* yang bekerja untuk menghitung waktu / pulsa.
- 6) UART (*Universl Asynchronous Receive Transnit*) adalah jalur komunikasi data khusus secara *serial asynchronous*.
- 7) PWM (*Pulse Width Modulation*) adalah fasilitas untuk membuat modulasi pulsa.
- 8) ADC (*Analog to Digital Converter*) adalah fasilitas untuk dapat menerima sinyal *analog* dalam *range* tertentu untuk kemudian dikonversi menjadi suatu nilai *digital* dalam *range* tertentu.
- 9) SPI (*Serial Peripheral Interface*) adalah jalur komunikasi data khusus secara *serial synchronous*.
- 10) ISP (*In System Programming*) adalah kemampuan khusus mikrokontroler untuk dapat diprogram langsung dalam sistem rangkaiannya dengan membutuhkan jumlah pin yang minimal.

b. Mikrokontroler ATmega8535

Seperti yang telah dijelaskan sebelumnya, ATmega8535 memiliki fasilitas yang cukup lengkap. Dari kapasitas memori program dan memori data yang cukup besar, *interupsi*, *timer / counter*, PWM, USART, TWI, *analog comparator*, EEPROM *internal* dan juga ADC *internal* semuanya ada dalam ATmega8535.

Selain itu kemampuan kecepatan eksekusi yang lebih tinggi menjadi alasan bagi banyak orang untuk lebih memilih menggunakan mikrokontroler jenis AVR daripada mikrokontroler pendahulunya yaitu keluarga MCS-51. Dengan demikian sangatlah tepat untuk mempelajari mikrokontroler keluarga AVR dengan ATmega8535.

(Bejo 2008: 10)

Mikrokontroler ATmega8535 merupakan salah satu mikrokontroler buatan AVR yang memiliki bagian-bagian, diantaranya :

1) Fitur ATmega8535

Berikut ini adalah fitur-fitur yang dimiliki oleh ATmega8535 :

- a) 130 macam intruksi, yang hampir semuanya dieksekusi dalam satu siklus *clock*.
- b) 32 x 8 bit *register* serba guna.
- c) Kecepatan mencapai 16 MIPS dengan *clock* 16 MHz.
- d) 8 Kbyte *Flash* memori, yang memiliki fasilitas *In-System Programming*.
- e) 512 Byte *internal* EEPROM.
- f) 512 Byte SRAM.
- g) *Programming Lock*, fasilitas untuk mengamankan kode program.
- h) 2 buah *timer / counter* 8 bit dan 1 buah *timer / counter* 16 bit.
- i) 4 *channel output* PWM.
- j) 8 *channel* ADC 10-bit.
- k) *Serial* USART.

Tabel 1. Susunan kaki ATmega8535

PIN	KETERANGAN
1-8	<p>Port B, merupakan Port I/O 8 bit dua arah (<i>bit-directional</i>) dengan resistor <i>pull-up internal</i>. Selain sebagai Port I/O 8 bit Port B juga dapat difungsikan secara individu sebagai berikut :</p> <ul style="list-style-type: none"> • PB7 : SCK (<i>SPI Bus Serial Clock</i>) • PB6 : MISO (<i>SPI Bus Master Input / Slave Output</i>) • PB5 : MOSI (<i>SPI Bus Master Output / Slave Input</i>) • PB4 : SS (<i>SPI Slave Select Input</i>) • PB3 : AIN1 (<i>Analog Comparator Negatif Input</i>) OC0 (<i>Output Compare Timer / Counter 0</i>) • PB2 : AIN0 (<i>Analog Comparator Positif Input</i>) INT2 (<i>External Interrupt 2 Input</i>) • PB1 : T1 (<i>Timer / counter 1 External Counter Input</i>) • PB0 : T0 (<i>USART External Clock Input / Output</i>)
9	RESET, merupakan pin reset yang akan bekerja bila diberi pulsa rendah (<i>aktif low</i>) selama minimal 1.5 μ s
10	VCC, catu daya <i>digital</i>
11	GND, Ground untuk catu daya <i>digital</i>
12	XTAL2, merupakan <i>output</i> dari penguat oscillator pembalik
13	XTAL1, merupakan <i>input</i> ke penguat oscillator pembalik dan <i>input</i> ke <i>internal clock</i> .
14-21	<p>Port D, merupakan Port I/O 8 bit dua arah (<i>bi-directional</i>) dengan resistor <i>pull-up internal</i>. Selain sebagai Port I/O 8 bit Port D juga dapat difungsikan secara individu sebagai berikut :</p> <ul style="list-style-type: none"> • PD7 : OC2 (<i>Output Compare Timer / Counter 2</i>) • PD6 : ICP1 (<i>Timer / Counter 1 input Capture</i>) • PD5 : OC1A (<i>Output Compare A Timer / Counter 1</i>)

	<ul style="list-style-type: none"> • PD4 : OC1B (<i>Output Compare B Timer / Counter 1</i>) • PD3 : INT1 (<i>External Interrupt 1 Input</i>) • PD2 : INT0 (<i>External Interrupt 0 Input</i>) • PD1 : TXD (<i>USART transmit</i>) • PD0 : RXD (<i>USART receive</i>)
22-29	<p>Port C, merupakan Port I/O 8 bit dua arah (<i>bi-directional</i>) dengan resistor <i>pull-up internal</i>. Selain sebagai Port I/O 8 bit 4bit Port C juga dapat difungsikan secara individu sebagai berikut :</p> <ul style="list-style-type: none"> • PC7 : TOC2 (<i>Timer Ocillator 2</i>) • PC6 : TOC1 (<i>Timer Ocillator 1</i>) • PC1 : SDA (<i>Serial Data Input / Output, I²C</i>) • PC0 :SCL (<i>Serial Clock, I²C</i>)
30	AVCC, merupakan catu daya yang digunakan untuk masukan <i>analog</i> ADC yang terhubung ke Port A
31	GND, Ground untuk catu daya <i>analog</i>
32	AREF, merupakan tegangan referensi <i>analog</i> untuk ADC
33-40	Port A, merupakan Port I/O 8 bit dua arah (<i>bi-directional</i>) dengan resistor <i>pull-up internal</i> . Selain sebagai Port I/O 8 bit Port A juga dapat berfungsi sebagai masukan 8 <i>channel</i> ADC

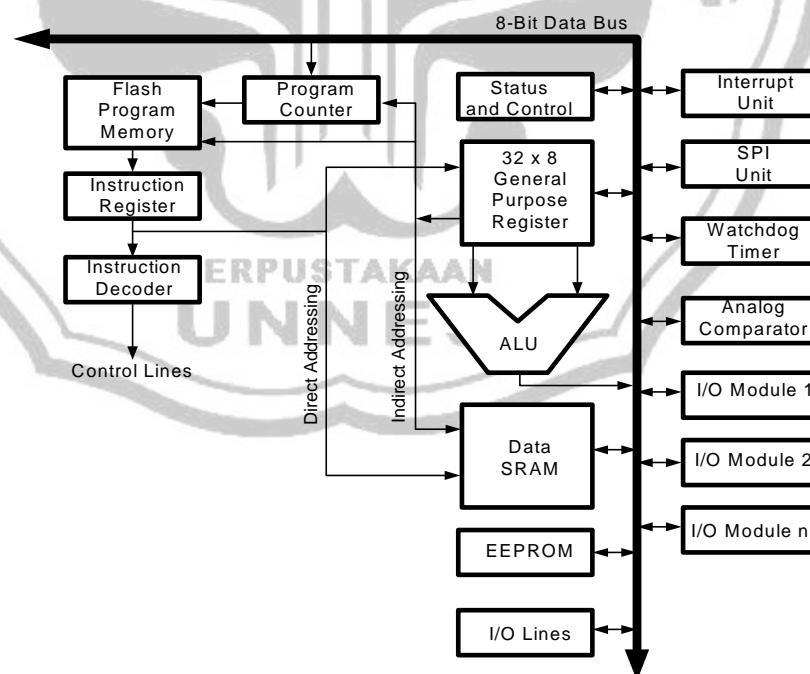
3) Arsitektur ATMega8535

Mikrokontroler ATMega8535 memiliki arsitektur Harvard, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan unjuk kerja dan paralelisme. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil (*pre – fetched*) dari memori program.

Konsep inilah yang memungkinkan instruksi–instruksi dapat dieksekusi dalam setiap satu siklus *clock*.

32 x 8 bit *register* serba guna digunakan untuk mendukung operasi pada *Aritmetic Logic Unit* (ALU) yang dapat dilakukan dalam satu siklus. Enam dari *register* serba guna dapat digunakan sebagai tiga buah *register pointer* 16 bit pada mode pengalamatan tak langsung untuk mengambil data pada ruang memori data.

Hampir semua perintah AVR memiliki format 16 bit (*word*). Setiap alamat memori program terdiri dari instruksi 16 bit atau 32 bit. Selain *register* serba guna, terdapat *register* lain yang terpetakan dengan teknik *memory mapped I/O* selebar 64 Byte. Beberapa *register* ini digunakan untuk fungsi khusus antara lain sebagai *register* kontrol *Timer / Counter*, *Interupsi*, ADC, USART, SPI, EEPROM dan fungsi I/O lainnya. *Register – register* ini menempati memori pada alamat 0x20h – 0x5Fh. Gambar arsitektur ATmega8535 terlihat pada Gambar 2.



Gambar 2. Arsitektur ATmega8535

(Sumber: datasheet ATmega8535)

2. Sensor Ultrasonik

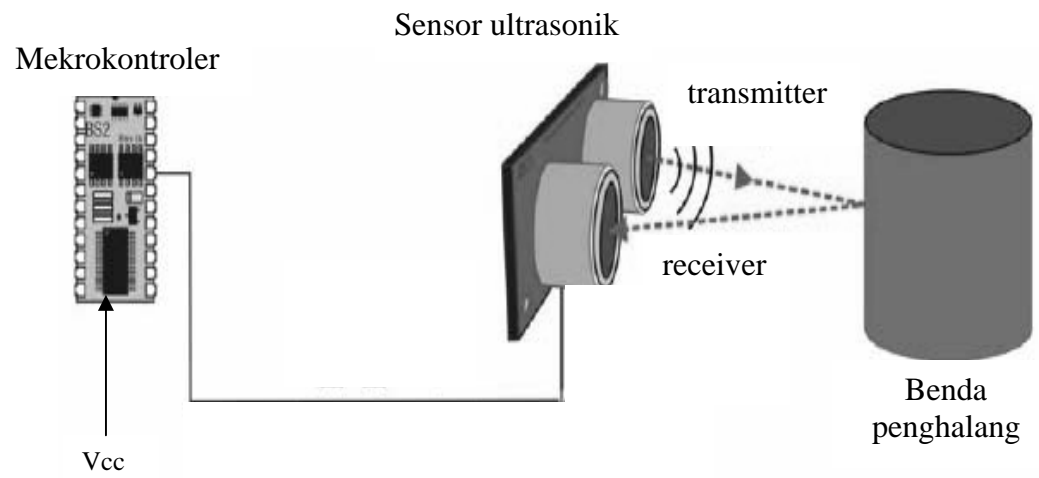
a. Pengertian

DT-SENSE ULTRASONIC merupakan modul pengukur jarak non-kontak yang sangat mudah dihubungkan dengan berbagai sistem berbasis mikrokontroler. Untuk memicu dan membaca data pengukuran dengan *DT-SENSE ULTRASONIC* hanya memerlukan 1 buah pin mikrokontroler.

Sebuah modul *DT-SENSE ULTRASONIC* terdiri dari sebuah Ultrasonik Ranger dan dapat dihubungkan dengan 2 buah sensor Infrared Ranger . Ultrasonik Ranger menghasilkan pulsa atau data keluaran yang menyatakan jarak yang ditempuh oleh sinyal tersebut sebelum menyentuh sebuah obyek dan memantul kembali. Keluaran analog dari sensor Infrared Ranger diubah oleh modul *DT-SENSE ULTRASONIC AND INFRARED RANGER* menjadi berbentuk pulsa (Modul Sensor Ultrasonik).

b. Prinsip Kerja Sensor Ultrasonik

Prinsip kerja sensor ultrasonik yaitu Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari pemancar (*transmitter*) ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh penerima (*receiver*) ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler, seperti pada Gambar 3.



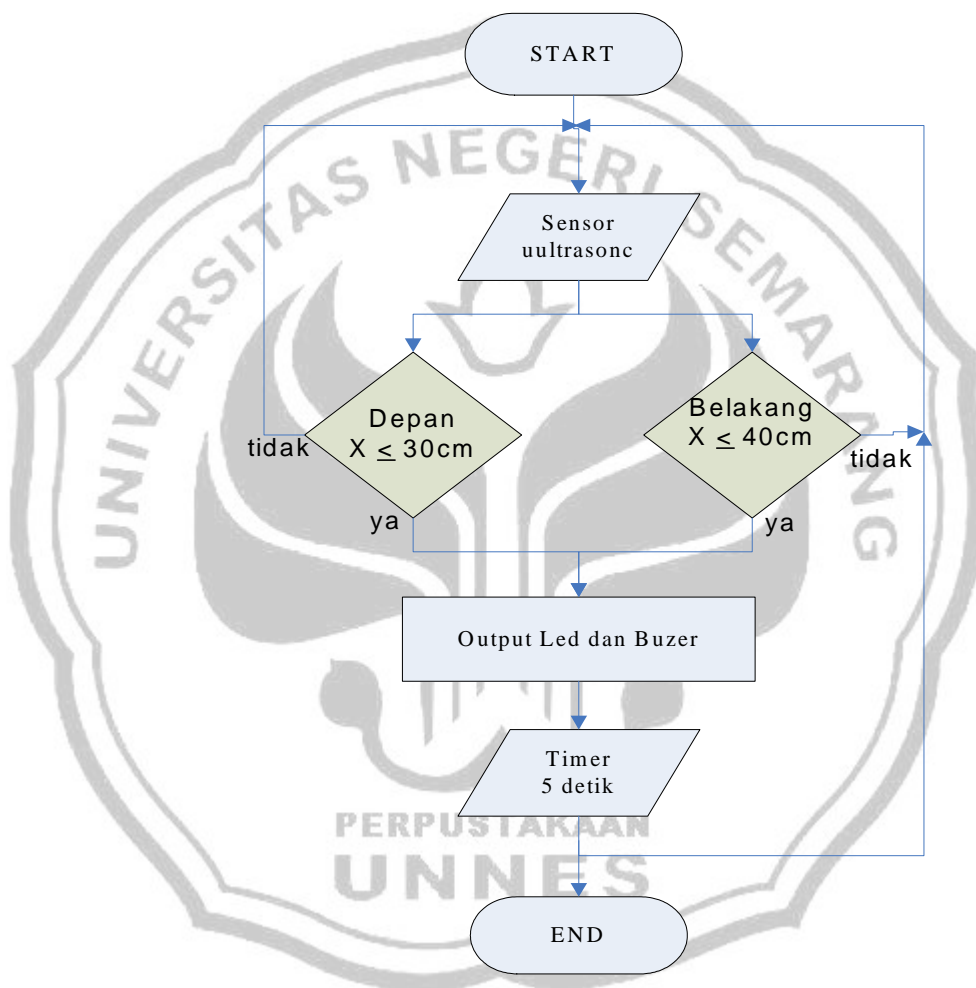
Gambar 3. Prinsip kerja sensor ultrasonik
(Sumber: Ping Ultrasonic Sensor Overview)



B. Metode Pembuatan Alat

1. Perencanaan pembuatan perangkat lunak (*software*)

Pembuatan *flow chart* dimaksudkan untuk memudahkan pembuatan program, karena dengan *flow chart* bisa diketahui alur kerja alat. Gambar 4. menunjukkan gambar diagram dasar kerja alat

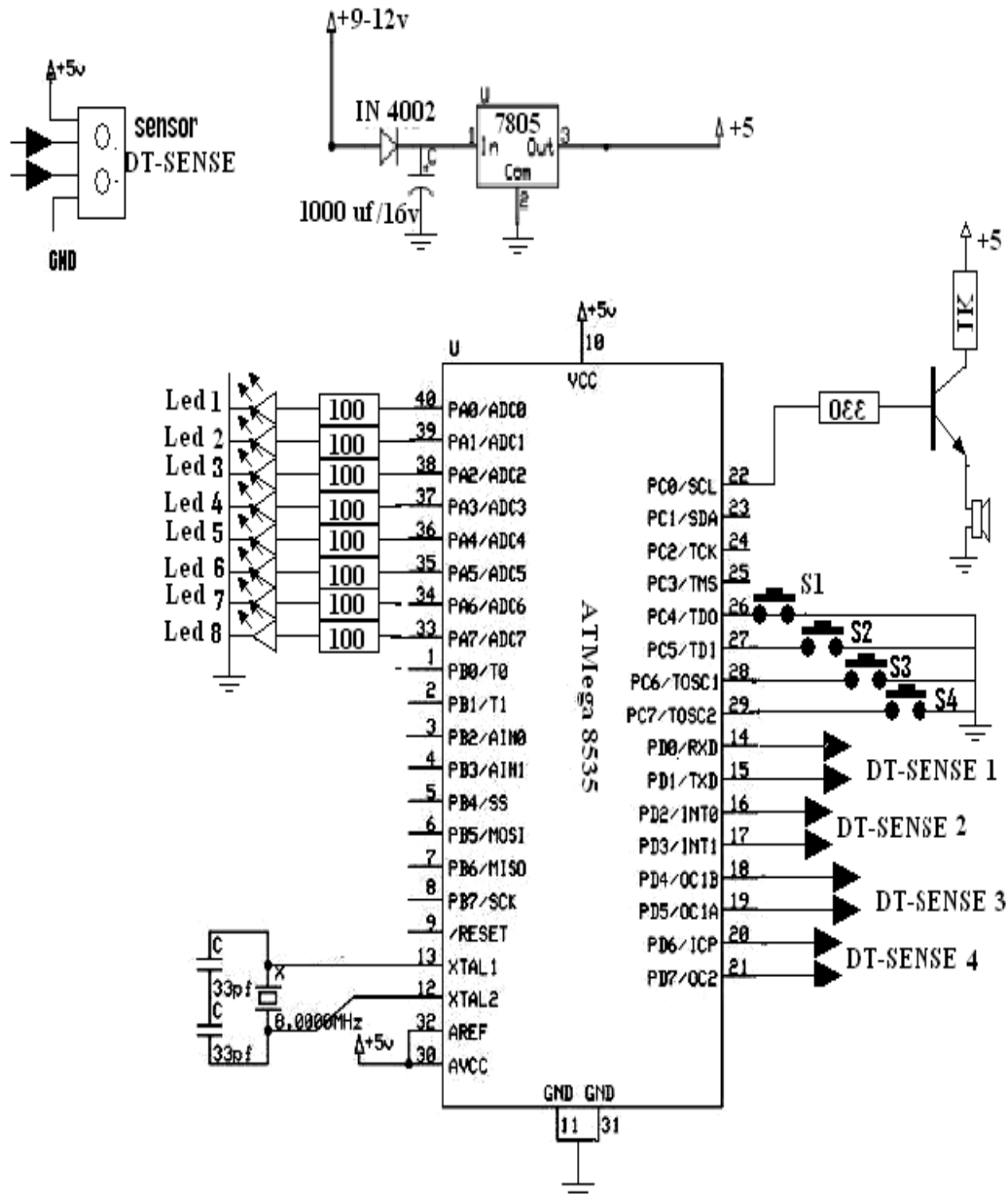


Gambar 4. *Flow chart* dasar kerja alat

2. Perancangan Pembuatan perangkat keras (*hardware*)

Langkah pertama dalam perancangan *hardware* adalah membuat desain skema rangkaian serta menentukan alat dan bahan yang akan digunakan

a. Perancangan Pembuatan Perangkat Mikrokontroler



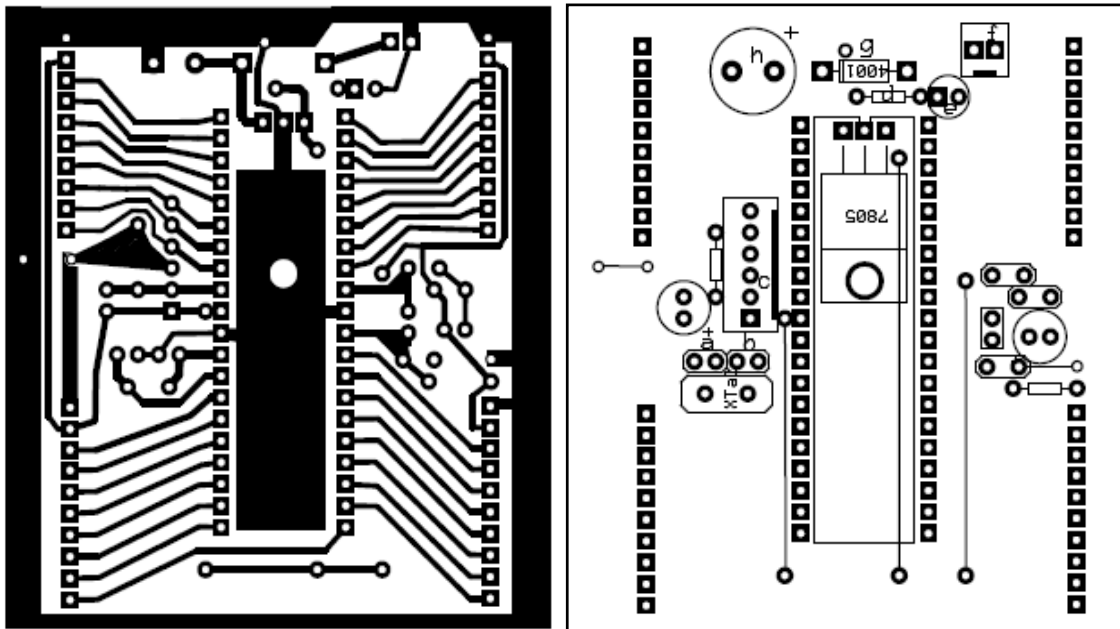
Gambar 5. Skema Rangkaian Sistem Minimum Mikrokontroler

1) Alat yang digunakan

- a) Spidol Permanent
- b) Bor dan mata bor
- c) Palu
- d) Cutter
- e) Tang potong
- f) Penggaris
- g) Solder
- h) Toolset lengkap

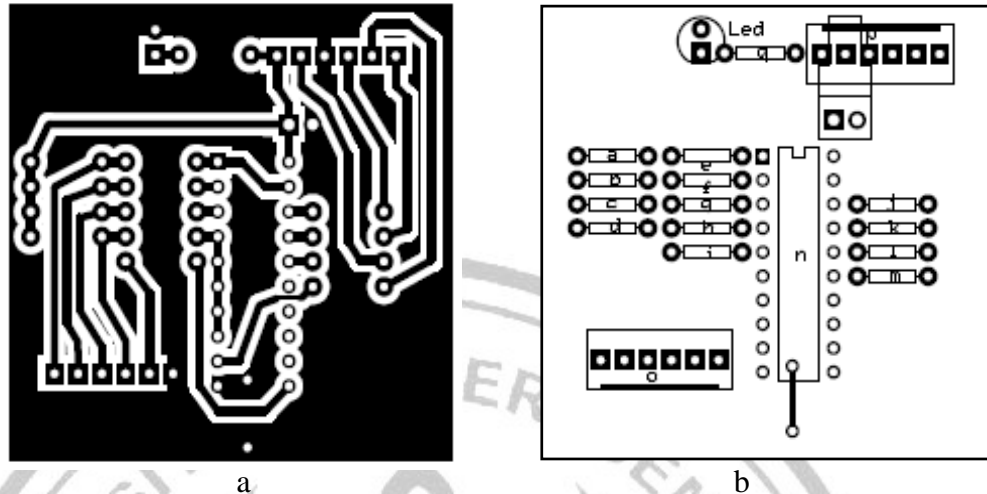
2) Komponen yang diperlukan:

- a) Resistor 100 Ω ¼ W.....8 buah
- b) Resistor 330 Ω ¼ W.....1 buah
- c) Led ½ W.....9 buah
- d) IC ATmega 8535.....1 buah
- e) IC LM 7805.....1 buah
- f) Kristal 11.059200Mhz.....1 buah
- g) Capacitor 1000uf/16v..... 1 buah
- h) Capacitor 30 pf.....2 buah
- i) Conektor 2pin.....1 buah
- j) Conektor 6pin.....1 buah
- k) Sakelar Push On.....4 buah
- l) Dioda IN 4002.....1 buah
- m) Transistor BD139.....1 buah
- n) Buzzer 5 Volt.....1 buah



Keterangan Gambar 6.b:

- a,b : Kapasitor 30 j
- c : Konektor 6 pin
- d : Resistor 330 Ω $\frac{1}{4}$ W
- e : Led
- f : Konektor 2 pin
- g : Dioda 1 A
- h : Kapasitor 100 μ F/16V



a
 b
 Gambar 8 . a. *Layout PCB downloader*
 b. Tata letak komponen

Keterangan Gambar 8.b:

a,b,c,d : Resistor 4K7? ¼ W

e,f,g,h,q : Resistor 1K? ¼ W

i,j,k,l,m : Resistor 100? ¼W

q : Resistor 330 ? ¼ W

n : IC 74LS541 + soket

o,p : Conektor 6 pin

c.

d. Prosedur Pembuatan

- 1) Menyiapkan alat-alat yang akan digunakan
- 2) Menggambar jalur

Teknik yang digunakan adalah teknik sablon setrika. Langkah pertama adalah menggambar lay out PCB seperti pada gambar 5a dan 7a. Gambar tersebut di foto kopi di kertas transparan (mika) dan selanjutnya gambar disetrika di atas papan PCB sampai gambar tersebut menempel di papan PCB. Pada saat akan melakukan setrika kita harus membersihkan papan PCB terlebih dahulu agar gambar dapat menempel dengan sempurna

- 3) Pelarutan Papan PCB

Gambar di papan PCB yang telah selesai dibuat kemudian dilarutkan dengan cairan *ferridclorida* (FhCl) untuk menghilangkan lapisan tembaga yang tidak tertutup gambar. Pada saat melakukan pelarutan wadah yang digunakan harus selalu di goyang-goyang.

- 4) Pelubangan Papan PCB

Proses pelubangan adalah untuk membuat tempat memasang komponen yang akan digunakan.

- 5) Memasang Komponen ke Papan PCB

Pemasangan komponen harus sesuai dengan tata letak yang telah ditentukan sebelumnya. Untuk lebih jelasnya dapat melihat gambar 5b dan gambar 7b.

- 6) Menyolder

Untuk menghasilkan hasil solderan yang baik sebaiknya menggunakan mata solder yang bersih dan lancip agar mempermudah dalam penyolderan. Hasil solderan yang baik adalah lancip dan mengkilap.

- 7) Pemotongan Kaki Komponen

Pemotongan kaki komponen dilakukan untuk meratakan kaki komponen agar menjadi rapi

C. Pengujian

Berdasarkan sumber Ary Darmadji, majalah AutoBild tanggal 2-5 Agustus 2003 edisi 7-suplemen (<http://library.monx007>) jarak aman parkir adalah 1 meter terhadap pembatas parkir. Namun kali ini kita menentukan sendiri jarak yang akan dibuat simulasi pengaman parkir sesuai keinginan. misalnya dengan membuat jarak aman 40cm untuk bagian belakang dan 30cm untuk bagian depan. Tabel 2. Adalah hasil pengujian alat pengaman parkir

Tabel 2. hasil pengujian alat pengaman parkir

Pengujian Ke.	Depan (diharapkan 30cm)		Belakang(diharapkan 40cm)	
	Kanan (cm)	Kiri (cm)	Kanan (cm)	Kiri (cm)
1	31,0	30,6	41,2	41,2
2	30,8	30,8	41,2	41,3
3	30,8	30,7	41,3	41,3
4	30,8	30,8	41,2	41,2
5	30,8	30,6	41,1	41,2
6	30,6	30,8	41,2	41,2
7	30,6	30,7	41,0	41,2
8	30,8	30,8	41,2	41,2
9	31,0	30,8	41,3	41,2
10	30,8	30,8	41,3	41,2
jumlah	308	307,4	412	412,2

D. Pembahasan

Berdasarkan hasil pengujian alat terjadi ketidak cocokan antara hasil percobaan dengan hasil yang di inginkan, sehingga terjadi selisih antara harga sebenarnya dengan harga yang ditunjukkan. Setelah melakukan pengukuran sebanyak 10 kali maka akan di cari nilai rata-rata dengan menggunakan persamaan sebagai berikut:

$$\bar{X} = \frac{X_1 + X_2 + X_3 + X_4 + \dots + X_n}{n} = \frac{\sum X}{n}$$

Dimana:

\bar{X} = nilai rata-rata hasil pengukuran

X_1, X_2, X_n = hasil pengukuran yang dilakukan

n = banyaknya pengukuran

Dengan menggunakan persamaan diatas dan berdasarkan data pada table 2. Maka akan di dapatkan nilai rata-rata untuk tiap sensornya.

1. Depan kanan

$$\bar{X} = \frac{31 + 30,8 + 30,8 + 30,8 + 30,8 + 30,6 + 30,6 + 30,8 + 31 + 30,8}{10} = \frac{308}{10}$$

$$\bar{X} = 30,8 \text{ cm}$$

2. Depan kiri

$$\bar{X} = \frac{30,6 + 30,8 + 30,7 + 30,8 + 30,6 + 30,8 + 30,7 + 30,8 + 30,8 + 30,8}{10} = \frac{307,4}{10}$$

$$\bar{X} = 30,74 \text{ cm}$$

3. Belakang kanan

$$\bar{X} = \frac{41,2 + 41,2 + 41,3 + 41,2 + 41,1 + 41,2 + 41,0 + 41,2 + 41,3 + 41,3}{10} = \frac{412}{10}$$

$$\bar{X} = 41,2 \text{ cm}$$

4. Belakang kiri:

$$\bar{X} = \frac{41,2 + 41,2 + 41,3 + 41,3 + 41,2 + 41,2 + 41,2 + 41,2 + 41,2 + 41,2}{10} = \frac{412,2}{10}$$

$$\bar{X} = 41,22 \text{ cm}$$

BAB III

PENUTUP

A. Kesimpulan

Berdasarkan percobaan yang telah dilakukan dan pembahasan maka dapat bahwa mikrokontroler ATmega8535 dapat digunakan untuk pengendali sensor ultrasonik tipe DT-SENSE yang bermanfaat untuk mengamankan mobil pada saat melakukan parkir.

Alat pengaman pengaman parkir ini akan memberikan kode berupa bunyi buzzer yang menandakan jarak tidak aman dengan tingkat responsifitas:

Depan kanan	: responsif pada jarak kurang dari 30,80 cm
Depan kiri	: responsif pada jarak kurang dari 30,74 cm
Belakang kanan	: responsif pada jarak kurang dari 41,20 cm
Belakang kiri	: responsif pada jarak kurang dari 41,22 cm

B. Saran

Untuk mendapatkan hasil deteksi jarak yang lebih reponsif maka diperlukan sensor pendeteksi jarak dengan tingkat kepekaan yang lebih tinggi

DAFTAR PUSTAKA

- Arikunto, Suharsimi. 2002. *Dasar – dasar evaluasi pendidikan*. Jakarta : PT. RinekaCipto.
- Bejo, Agus. 2008. *C&AVR Rahasia Kemudahan Bahasa C dalam Mikokontroler ATmega8535*. Yogyakarta : GRAHA ILMU.
- Budiharto, Widodo. 2007. *Panduan Praktikum Mikrokontroler AVR ATmega16*. Jakarta : PT. Elex Media Komputindo.
- Sugiharto, Agus. 2002. *Penerapan Dasar Tranduser Dan Sensor*. Yogyakarta : Kanisius.



File: 1.c, Date: 18/03/2010, Time: 7:49:02

This program was produced by the
CodeWizardAVR V1.24.0 Standard
Automatic Program Generator
© Copyright 1998-2003 HP InfoTech s.r.l.
<http://www.hpinfotech.ro>
e-mail:office@hpinfotech.ro

Chip type : ATmega8535
Program type : Application
Clock frequency : 12.000000 MHz
Memory model : Small
External SRAM size : 0
Data Stack size : 128
*****/

```
#include <mega8535.h>
#include <delay.h>
```

```
#define buzzer1 PORTC.0
#define buzzer2 PORTC.0
#define buzzer3 PORTC.0
#define buzzer4 PORTC.0
```

```
#define sw_4 PINC.4
#define sw_3 PINC.5
#define sw_2 PINC.6
#define sw_1 PINC.7
```

```
#define led_1 PORTA.1
#define led_2 PORTA.0
#define led_3 PORTA.6
#define led_4 PORTA.7
```

```
#define led_akt1 PORTA.2
#define led_akt2 PORTA.3
#define led_akt3 PORTA.5
#define led_akt4 PORTA.4
```

```
typedef unsigned int uint16_t;
unsigned int counter;
```

```
uint16_t US,tinggi;
```

```
// Alphanumeric LCD Module functions
```

```
#asm
.equ __lcd_port=0x18
#endasm
#include <lcd.h>
```

```
// Declare your global variables here
```

```
void sendData()
```

```
{
```

```
unsigned char Posisi;
```

```
tinggi=US;
```

```
Posisi=0;
lcd_gotoxy(Posisi,0);
lcd_putchar('j');
lcd_gotoxy(Posisi+1,0);
lcd_putchar('a');
lcd_gotoxy(Posisi+2,0);
lcd_putchar('r');
lcd_gotoxy(Posisi+3,0);
lcd_putchar('a');
```


File: 1.c, Date: 18/03/2010, Time: 7:49:02

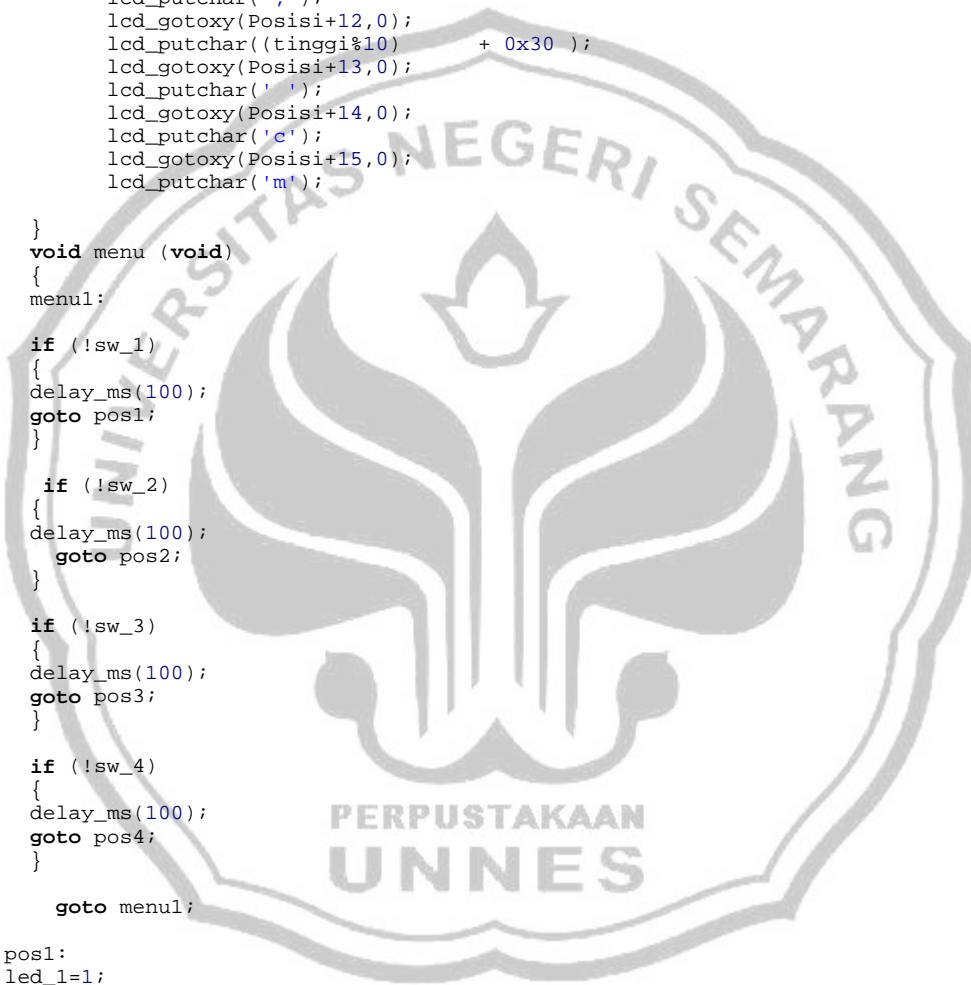
```
    lcd_gotoxy(Posisi+4,0);
    lcd_putchar('k');
    lcd_gotoxy(Posisi+5,0);
    lcd_putchar(' ');
    lcd_gotoxy(Posisi+6,0);
    lcd_putchar('=');
    lcd_gotoxy(Posisi+7,0);
    lcd_putchar(' ');
    lcd_gotoxy(Posisi+8,0);
    lcd_putchar((tinggi/1000)%10 + 0x30 );
    lcd_gotoxy(Posisi+9,0);
    lcd_putchar((tinggi/100)%10 + 0x30 );
    lcd_gotoxy(Posisi+10,0);
    lcd_putchar((tinggi%100)/10 + 0x30 );
    lcd_gotoxy(Posisi+11,0);
    lcd_putchar(',');
    lcd_gotoxy(Posisi+12,0);
    lcd_putchar((tinggi%10) + 0x30 );
    lcd_gotoxy(Posisi+13,0);
    lcd_putchar(' ');
    lcd_gotoxy(Posisi+14,0);
    lcd_putchar('c');
    lcd_gotoxy(Posisi+15,0);
    lcd_putchar('m');
}
void menu (void)
{
    menu1:
    if (!sw_1)
    {
        delay_ms(100);
        goto pos1;
    }
    if (!sw_2)
    {
        delay_ms(100);
        goto pos2;
    }
    if (!sw_3)
    {
        delay_ms(100);
        goto pos3;
    }
    if (!sw_4)
    {
        delay_ms(100);
        goto pos4;
    }
    goto menu1;

pos1:
led_1=1;
goto baca_sensor1;
goto pos1;

pos2:
led_2=1;
goto baca_sensor2;
goto pos2;

pos3:
led_3=1;
goto baca_sensor3;
goto pos3;

pos4:
led_4=1;
goto baca_sensor4;
goto pos4;
```



File: 1.c, Date: 18/03/2010, Time: 7:49:02

```

baca_sensor1:

#define SIG_out    PORTD.0
#define SIG_in     PIND.0
#define SIG_dir    DDRD.0
#define ready      PIND.1

    SIG_dir = 1;    // set SIG pin as output

    SIG_out = 0;    //
    delay_us(50);  // send start pulse
    SIG_out = 1;    //

    SIG_dir = 0;    // set SIG pin as input

    TCNT1=0;
    counter=0;
    while (SIG_in && counter<30000)    // wait for return pulse
        counter++;
    TCCR1B=0x02;    // start timer (timer period = 2uS)
    while (!(SIG_in) && !(TIFR&0x80)); //TIFRx80 :bit overflow:bit ini tidak
// selalu sama letaknya di setiap UC AVR
    TCCR1B=0x00;    // stop timer
    US = TCNT1;    // save timer value to variable US

    US = US/14;    // convert to milimeters

    //sendData();
    if (US<=300){buzzer1=1;led_akt1=1;delay_ms(5000);buzzer1=0;led_akt1=0;
delay_ms(5000);}
else {buzzer1=0;led_akt1=0;}

////////////////////////////////////

if (!sw_1)
{
    delay_ms(100);
    goto matil;
}

if (!sw_2)
{
    delay_ms(100);
    goto pos2;
}

if (!sw_3)
{
    delay_ms(100);
    goto pos3;
}

if (!sw_4)
{
    delay_ms(100);
    goto pos4;
}

////////////////////////////////////

    delay_ms(500);

    goto baca_sensor1;

    /*****/
matil:
    led_1=0;
    led_akt1=0;
    menu();

    goto matil;
/*****/

```

File: 1.c, Date: 18/03/2010, Time: 7:49:02

```
    baca_sensor2:

#define SIG_out    PORTD.2
#define SIG_in     PIND.2
#define SIG_dir    DDRD.2
#define ready     PIND.3

    SIG_dir = 1;    // set SIG pin as output

    SIG_out = 0;    //
    delay_us(50);  // send start pulse
    SIG_out = 1;    //

    SIG_dir = 0;    // set SIG pin as input

    TCNT1=0;
    counter=0;
    while (SIG_in && counter<30000)    // wait for return pulse
    counter++;
    TCCR1B=0x02;    // start timer (timer period = 2uS)
    while (!(SIG_in) && !(TIFR&0x80)); //TIFR&0x80 :bit overflow:bit ini tidak
// selalu sama letaknya di setiap UC AVR
    TCCR1B=0x00;    // stop timer
    US = TCNT1;    // save timer value to variable US

    US = US/14;    // convert to milimeters

    //sendData();
    if (US<=300){buzzer2=1;led_akt2=1;delay_ms(5000);buzzer2=0;led_akt2=0;
delay_ms(5000);}
else{buzzer2=0;led_akt2=0;}

////////////////////////////////////
if (!sw_1)
{
delay_ms(100);
goto pos1;
}

if (!sw_2)
{
delay_ms(100);
goto mati2;
}

if (!sw_3)
{
delay_ms(100);
goto pos3;
}

if (!sw_4)
{
delay_ms(100);
goto pos4;
}

////////////////////////////////////
    delay_ms(500);

    goto baca_sensor2;

    /*****/
    mati2:
    led_2=0;
    led_akt2=0;
    menu();

    goto mati2;
    /*****/

    baca_sensor3:
```

File: 1.c, Date: 18/03/2010, Time: 7:49:02

```
#define SIG_out    PORTD.4
#define SIG_in     PIND.4
#define SIG_dir    DDRD.4
#define ready      PIND.5

    SIG_dir = 1;    // set SIG pin as output

    SIG_out = 0;    //
    delay_us(50);  // send start pulsea /
    SIG_out = 1;    //

    SIG_dir = 0;    // set SIG pin as input

    TCNT1=0;
    counter=0;
    while (SIG_in && counter<30000)    // wait for return pulse
        counter++;
    TCCR1B=0x02;    // start timer (timer period = 2uS)
    while (!(SIG_in) && !(TIFR&0x80)); //TIFRx80 :bit overflow:bit ini tidak
// selalu sama letaknya di setiap UC AVR
    TCCR1B=0x00;    // stop timer
    US = TCNT1;    // save timer value to variable US

    US = US/14;    // convert to milimeters

    //sendData();
    if (US<=400){buzzer3=1;led_akt3=1;delay_ms(5000);buzzer3=0;led_akt3=0;
delay_ms(5000);}
else {buzzer3=0;led_akt3=0;}
////////////////////////////////////
if (!sw_1)
{
    delay_ms(100);
    goto pos1;
}

if (!sw_2)
{
    delay_ms(100);
    goto pos2;
}

if (!sw_3)
{
    delay_ms(100);
    goto mati3;
}

if (!sw_4)
{
    delay_ms(100);
    goto pos4;
}

////////////////////////////////////
    delay_ms(500);

    goto baca_sensor3;

    /*****/
    mati3:
    led_3=0;
    led_akt3=0;
    menu();

    goto mati3;
    /*****/

    baca_sensor4:

#define SIG_out    PORTD.6
#define SIG_in     PIND.6
```

File: 1.c, Date: 18/03/2010, Time: 7:49:02

```
#define SIG_dir    DDRD.6
#define ready     PIND.7

    SIG_dir = 1;    // set SIG pin as output

    SIG_out = 0;    //
    delay_us(50);  // send start pulse
    SIG_out = 1;    //

    SIG_dir = 0;    // set SIG pin as input

    TCNT1=0;
    counter=0;
    while (SIG_in && counter<30000)    // wait for return pulse
        counter++;
    TCCR1B=0x02;    // start timer (timer period = 2uS)
    while (!(SIG_in) && !(TIFR&0x80)); //TIFRx80 :bit overflow:bit ini tidak
// selalu sama letaknya di setiap UC AVR
    TCCR1B=0x00;    // stop timer
    US = TCNT1;    // save timer value to variable US

    US = US/14;    // convert to milimeters

    //sendData();
    if (US<=400){buzzer4=1;led_akt4=1;delay_ms(5000);buzzer4=0;led_akt4=0;
delay_ms(5000);}
else {buzzer4=0;led_akt4=0;}

////////////////////////////////////
if (!sw_1)
{
    delay_ms(100);
    goto pos1;
}

if (!sw_2)
{
    delay_ms(100);
    goto pos2;
}

if (!sw_3)
{
    delay_ms(100);
    goto pos3;
}

if (!sw_4)
{
    delay_ms(100);
    goto mati4;
}

////////////////////////////////////
    delay_ms(500);

    goto baca_sensor4;

/*****/
    mati4:
        led_4=0;
        led_akt4=0;
        menu();

    goto mati4;
/*****/

}

void main(void)
{
    // Declare your local variables here
```

File: 1.c, Date: 18/03/2010, Time: 7:49:02

```
// Input/Output Ports initialization
// Port A initialization
PORTA=0x00;
DDRA=0xFF;

// Port B initialization
PORTB=0x00;
DDRB=0x00;

// Port C initialization
PORTC=0xF0;
DDRC=0x01;

// Port D initialization
PORTD=0xBF;
DDRD=0x55;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
// Analog Comparator Output: Off
ACSR=0x80;
SFIOR=0x00;

// LCD module initialization
lcd_init(16);

led_1=0;
led_2=0;
```



File: 1.c, Date: 18/03/2010, Time: 7:49:02

```
led_3=0;
led_4=0;

menu();

while (1)
{
    // Place your code here

};
}
```



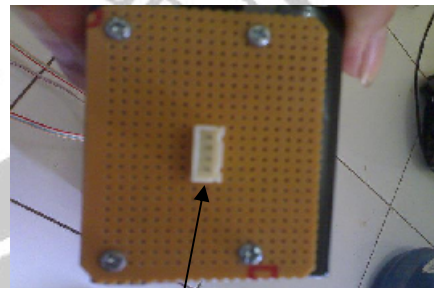


Kabel penghubung ke sensor

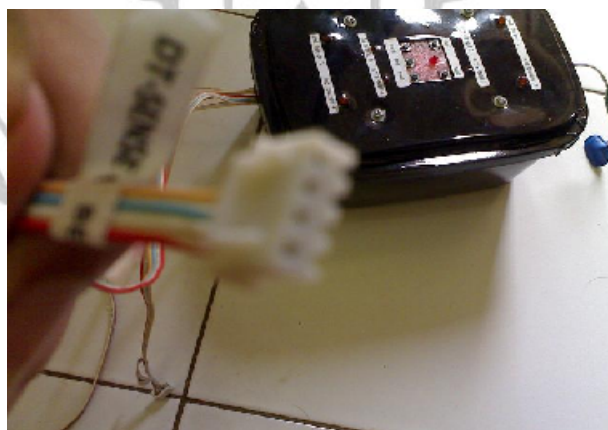
Alat Pengaman Parkir



Sensor Ultrasonic



konektor sensor Ultrasonic



Konektor Penhubung Ke Sensor Ultrasonic