

**RANCANG BANGUN ROBOT BERODA PEMINDAH
KERANJANG MENGGUNAKAN KOMPAS DIGITAL DAN
INCREMENTAL ROTARY ENCODER BERBASIS ATMEGA16**

SKRIPSI



**Disusun Oleh :
HUDDIN
08.12.217**

**PROGRAM STUDI TEKNIK ELKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

2013

REPUBLIC OF INDONESIA
MINISTRY OF EDUCATION AND CULTURE
GENERAL DIRECTORATE OF HIGHER EDUCATION
DIRECTORATE OF QUALITY ASSURANCE

OFFICE

SECRETARY

PROF. DR. H. ...

2013

REPUBLIC OF INDONESIA
MINISTRY OF EDUCATION AND CULTURE
GENERAL DIRECTORATE OF HIGHER EDUCATION
DIRECTORATE OF QUALITY ASSURANCE

LEMBAR PERSETUJUAN

**RANCANG BANGUN ROBOT BERODA PEMINDAH
KERANJANG MENGGUNAKAN KOMPAS DIGITAL DAN
INCREMENTAL ROTARY ENCODER BERBASIS ATMEGA16**

SKRIPSI

*Disusun dan Diajukan Untuk melengkapi dan Memenuhi Persyaratan Guna Mencapai
Gelara Sarjana Teknik Strata Satu (S-1)*

Disusun oleh :

HUDDIN

08.12.217

Diperiksa dan Disetujui

Dosen Pembimbing I



M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358

Dosen Pembimbing II



Sotyohadi, ST
NIP.Y. 1039700309

Mengetahui,

Ketua Jurusan Teknik Elektro S-1



M. Ibrahim Ashari, ST, MT
NIP.P. 1030100358

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : **HUDDIN**
Nim : **08.12.217**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Elektronika**
Masa Bimbingan : **Semester Ganjil 2012-2013**
Judul : **RANCANG BANGUN ROBOT BERODA PEMINDAH
KERANJANG MENGGUNAKAN KOMPAS DIGITAL
DAN INCREMENTAL ROTARY ENCODER BERBASIS
ATMEGA16**

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Program Strata Satu (S-1)

Pada Hari : Selasa
Tanggal : 19 February 2013
Dengan Nilai : 80,4 (A) *✓*

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Sekretaris Majelis Penguji

Dr. Eng. Aryuanto Soetedjo, ST, MT
NIP.Y.1030800417

ANGGOTA PENGUJI

Dosen Penguji I

Irmalia Suryani F. ST, MT
NIP.P.1030000365

Dosen Penguji II

Ir. Eko Nurcahyo
NIP.P.1031200456

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : HUDDIN
NIM : 08 12 217
Program Studi : TEKNIK ELEKTRO S-1
Konsentrasi : TEKNIK ELEKTRONIKA S-1

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, 28-Maret-2013
Yang membuat Pernyataan,

HUDDIN
08 12 217

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan laporan Skripsi ini dengan baik dan lancar.

Laporan Skripsi ini merupakan salah satu persyaratan akademik dalam menyelesaikan program Strata 1 Jurusan Teknik Elektro, Konsentrasi Elektronika, Institut Teknologi Nasional Malang. Adapun judul laporan Skripsi ini adalah:

RANCANG BANGUN ROBOT BERODA PEMINDAH KERANJANG MENGUNAKAN KOMPAS DIGITAL DAN INCREMENTAL ROTARY ENCODER BERBASIS ATMEGA16

Selanjutnya pada kesempatan ini penulis juga menyampaikan rasa terimakasih yang sebesar-besarnya kepada pihak-pihak yang telah banyak membantu penulis selama penyusunan tugas akhir, diantaranya :

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang
2. Bapak Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
3. Bapak M. Ibrahim Ashari, ST, MT selaku Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang
4. Bapak Dr. Eng. Aryuanto Soetedjo, ST, MT selaku Sekertaris Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang
5. Bapak M. Ibrahim Ashari, ST, MT selaku Dosen Pembimbing I
6. Bapak Sotyohadi, ST selaku Dosen Pembimbing II
7. Bapak, Ibu dan saudara saya yang telah memberikan dukungan baik materi, moral maupun spritual dan untuk selalu berdoa dan berusaha beserta nasehat yang telah diberikan sampai saat ini
8. Seluruh dosen dan pegawai Institut Teknologi Nasional Malang
9. Semua teman-teman Komunitas Robotika ITN Malang, dan teman-teman seperjuangan elektro '08 semuanya
10. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu

Penulis berharap agar laporan laporan Skripsi ini dapat memberikan banyak manfaat bagi semua pihak yang membutuhkan, khususnya bagi rekan-rekan mahasiswa. Penulis menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan, oleh karena itu mohon maaf apabila dalam laporan ini terdapat hal-hal yang kurang berkenan dihati para pembaca.

Penulis juga mengharap koreksi, kritik serta saran-saran yang bermanfaat demi kesempurnaan laporan Laporan Skripsi ini.

Malang, 17 Februari 2013

Penulis

DAFTAR ISI

COVER	i
Lembar Persetujuan	ii
Abstrak	iii
Kata Pengantar	iv
Daftar Isi	vi
Daftar Gambar	viii
Daftar Tabel	x
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	2
1.4. Batasan Masalah	2
1.5. Sistematika Penulisan	2
BAB II TINJAUAN PUSTAKA	
2.1. Mikrokontroler	4
2.2. Konfigurasi Pinout ATmega16	5
2.3. Arsitektur Mikrokontroler ATmega16	10
2.4. Kompas Digital CMPS03	11
2.5. Incremental Rotary Encoder	14
2.6. Ultrasonik Ping	16
2.7. Saklar Tekan/Push Botton	18
2.8. Alphanumeric LCD	19
2.9. Penguat Motor	20
2.10. Motor DC	20
BAB III PERANCANGAN DAN ANALISA SISTEM	
3.1. Gambar Umum Desain Alat	22
3.2. Fungsi Masing-masing Rancangan Alat	23
3.3. Prinsip Kerja Alat	23
3.4. Perancangan Perangkat Keras	24
3.5. Flowchart	31
3.6. Software CodeVision AVR	33

BAB IV PENGUJIAN ALAT DAN PEMBAHASAN HASIL

4.1. Pengujian Alphanumeric LCD	36
4.2. Pengujian Kompas Digital CMPS03	37
4.3. Pengujian Incremental Rotary Encoder	40
4.4. Pengujian Ultrasonik Ping	43
4.5. Pengujian Penguat Motor	45
4.6. Pengujian Sistem	47

BAB V PENUTUP

5.1. Kesimpulan	53
5.2. Saran	53

DAFTAR PUSTAKA	54
-----------------------------	-----------

LAMPIRAN	55
-----------------------	-----------

DAFTAR GAMBAR

Gambar 2.1. ATmega 16	4
Gambar 2.2. Konfigurasi Pinout ATmega16	5
Gambar 2.3. Diagram Blok Arsitektur Mikrokontroler ATmega16	10
Gambar 2.4. Kompas Digital CMPS03	11
Gambar 2.5. I2C Communication Protocol	12
Gambar 2.6. Incremental Rotary Encoder	15
Gambar 2.7. Gelombang Keluaran Incremental Rotary Encoder	15
Gambar 2.8. Ultrasonic ping	16
Gambar 2.9. Prinsip Kerja Sensor Ultrasonik Ping	17
Gambar 2.10. Prinsip Kerja Pemantulan Sensor Ultrasonik Ping	18
Gambar 2.11. Simbol Saklar Tekan	18
Gambar 2.12. Modul LCD Dimensi 2x16	19
Gambar 2.13. Pin IC L298	20
Gambar 2.14. Motor DC	21
Gambar 3.1. Blok Diagram Sistem Kendali Robot	22
Gambar 3.2. Tempat Modul Kompas Digital CMPS03	24
Gambar 3.3. Skema Rangkaian Incremental Rotary Encoder	25
Gambar 3.4. Modul Ultrasonik Ping	26
Gambar 3.5. Skema Rangkaian Saklar Tekan/Push Botton	27
Gambar 3.6. Skema Rangkaian LCD Modul	28
Gambar 3.7. Skema Rangkaian Penguat Motor DC	29
Gambar 3.8. Rangkaian Reset	29
Gambar 3.9. Rangkaian Clock	30
Gambar 3.10. Flowchart Saat Pengambilan Keranjang	31
Gambar 3.11. flowchart saat meletakkan keranjang	32
Gambar 3.12. Tampilan Awal CodeVision AVR	33
Gambar 3.13. Membuat Project Baru	33
Gambar 3.14. Pertanyaan Membuat Project Baru	34
Gambar 3.15. Pengaturan Konfigurasi Chip	34
Gambar 3.16. Hasil konfigurasi dengan CodeWizard CodeVision AVR	35
Gambar 4.1. Hasil pengujian Modul LCD	36

Gambar 4.2. Hasil Pengujian Kompas Digital CMPS03	38
Gambar 4.3. Hasil Perbandingan Kompas Digital Dengan Kompas Analog	39
Gambar 4.4. Proses Pengujian	41
Gambar 4.5. Hasil Pengujian <i>Incremental Rotary Encoder</i>	41
Gambar 4.6. Hasil pengujian sensor ultrasonik ping	44
Gambar 4.7. Tegangan masuk pada penguat motor	46
Gambar 4.8. Hasil tegangan penguat motor tanpa beban	46
Gambar 4.9. Hasil arus penguat motor tanpa beban	46
Gambar 4.10. Hasil tegangan penguat motor dengan beban	47
Gambar 4.11. Hasil arus penguat motor tanpa beban	47
Gambar 4.12. Arena pengujian robot	48
Gambar 4.12. Proses pengujian sistem.....	49
Gambar 4.13. Proses pengujian sistem dengan kecepatan 255 pulsa	50
Gambar 4.14. Proses pengujian sistem dengan kecepatan 200 pulsa.....	50
Gambar 4.15. Proses pengujian sistem dengan kecepatan 175 pulsa.....	50
Gambar 4.16. Proses pengujian sistem dengan kecepatan 150 pulsa.....	51
Gambar 4.17. Proses pengujian sistem dengan kecepatan 100 pulsa.....	51

DAFTAR TABEL

Table 2.1. Konfigurasi Pin Port Mikrokontroler ATmega16	7
Table 2.2. Fungsi Alternatif Konfigurasi Pin <i>PortA</i>	7
Tabel 2.3. Fungsi Alternatif Konfigurasi Pin <i>PortB</i>	8
Table 2.4. Fungsi Alternatif Konfigurasi Pin <i>PortC</i>	8
Table 2.5. Fungsi Alternatif Konfigurasi Pin <i>PortD</i>	9
Table 2.6. Kompas <i>Array 16byte Register</i>	13
Table 2.7. <i>Incremental Rotary Encoder</i> Searah Jarum Jam	16
Table 2.8. <i>Incremental Rotary Encoder</i> Berlawanan Arah Jarum Jam	16
Table 3.1. Fungsi Masing-masing Sinyal LCD	27
Tabel 4.1. Hasil pengujian kompas digital/CMPS03	37
Tabel 4.2. Perbandingan hasil kompas digital dan kompas analog	39
Tabel 4.3. Presentase nilai kesalahan (% error) dari kompas digital terhadap kompas analog	40
Tabel 4.4. Hasil pengujian <i>incremental rotary encoder</i>	41
Tabel 4.5. Perbandingan Hasil pengujian dan perhitungan Rotary Encoder	42
Tabel 4.6. Presentase nilai kesalahan (% error) dari hasil pengujian terhadap perhitungan	43
Tabel 4.7. Perbandingan hasil pengujian dan pengukuran	44
Tabel 4.8. Presentase nilai kesalahan (% error) dari hasil pengujian terhadap pengukuran	45
Tabel 4.9. Hasil pengujian penguat motor	45
Tabel 4.10. Hasil pengujian penguat motor tanpa beban dan motor dengan beban	46
Tabel 4.11. Hasil pengujian sistem meletakkan barang dengan sudut 120^0	49
Tabel 4.12. Presentase nilai kesalahan (% error) dari hasil pengujian sistem	52

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sejauh ini robot yang bergerak otonom, banyak tantangan yang dihadapi. Diantaranya adalah sejauh mana robot bisa mengatasi gangguan, dan dituntut mencapai sebuah target posisi tertentu. robot harus mampu mengetahui besarnya gangguan sehingga mampu bergerak kembali ke arah yang aman dan tetap berusaha mendekati target. kebanyakan robot tidak mengetahui dimana posisi robot sekarang, bagaimana navigasi robot untuk mencapai tujuan dengan adanya gangguan yang datang dari luar dan robot dapat sampai pada tujuan. Oleh karena itu pada tugas akhir ini dibuat robot yang dapat bernavigasi untuk mencapai tempat tujuan, dan mampu mengatasi gangguan dari luar yang menyebabkan robot berubah arah.

Mobile robot, adalah istilah yang sering digunakan untuk menyebut sebuah robot yang memiliki kemampuan menjelajahi. Tidak peduli apakah robot tersebut bergerak menggunakan roda, kaki, maupun kipas, maka robot semacam ini masuk ke dalam kategori mobile robot. Agar tidak bergerak salah arah, robot harus dilengkapi dengan sistem navigasi yang dapat memberikan informasi arah dengan baik. Sehingga robot dapat memutuskan dengan benar ke arah mana seharusnya bergerak untuk mencapai target yang diinginkan.

Sistem navigasi yang di pasang pada robot dapat berupa berbagai sensor, di antaranya sensor pembaca garis, sensor ultrasonik, sensor kompas, sensor rotary encoder dan sensor camera. Penggunaan sensor sebagai sistem navigasi yang umum adalah sensor garis. Sensor ini memiliki kekurangan dalam hal karakteristik medan yang dapat di jelajahi karena hanya dapat membaca permukaan garis yang memiliki dua kontras warna yaitu antara gelap dan terang. Sebagai koreksi kekurangan dari pada sensor garis, dalam penelitian ini penulis menggunakan sensor kompas digital dan *incremental rotary encoder*. Kedua sensor ini di gunakan sebagai sistem navigasi sesuai dengan prinsip vector, dimana sensor kompas digunakan untuk menunjukkan arah gerak sementara sensor *incremental rotary encoder* digunakan untuk menentukan besaran langkah gerak robot.

1.2. Rumusan Masalah

Bagaimana menggunakan sensor kompas digital dan *incremental rotary encoder* sebagai sistem navigasi robot beroda?

1.3. Tujuan

Merancang/membangun robot pemindah keranjang yang dapat menentukan arah secara akurat berdasarkan navigasi kompas digital dan *incremental rotary encoder*.

1.4. Batasan Masalah

Permasalahan yang akan di bahas dalam penulisan ini sebagai berikut:

- a. Menggunakan minimum sistem AVR ATMEGA16 kristal 12 MHz.
- b. Menggunakan modul Sensor kompas digital CMPS03.
- c. Menggunakan modul sensor *incremental rotary encoder* autonic resolusi 1/360.
- d. Aktuator yang di gunakan adalah motor DC
- e. Robot yang digunakan berupa robot beroda, dimensi 35x45cm.
- f. Tidak membahas mekanik pada pembuatan robot.
- g. Tidak membahas perangkat keras secara detail
- h. Pelatakan keranjang pada saat robot mengambil dan meletakan di bantu oleh manusia
- i. Robot bergerak mengikuti peraturan arena yang di buat sebagai berikut:
 - Arena 3x4m
 - Posisi robot di letakan dalam *loading* arena
 - Posisi pengambilan barang
 - Posisi meletakan barang

1.5. Sistematika Penulisan

Sistematika dalam pembahasan ini dimaksudkan untuk memberikan gambaran secara garis besar tentang apa yang dikemukakan dalam pokok bahasan. Adapun susunan sistematika laporan ini sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisikan antara lain tentang latar belakang, rumusan masalah, tujuan, batasan masalah dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisikan tentang landasan teori diantaranya: mikrokontroler, kompas digital, *incremental rotary encoder*, ultrasonic PING, saklar tekan dan driver motor.

BAB III PERANCANGAN dan ANALISA SISTEM

Bab ini berisikan mengenai analisa sensor kompas digital, *incremental rotary encoder*, ultrasonik PING, saklar tekan dan driver motor dalam menyelesaikan tugas akhir.

BAB IV PENGUJIAN ALAT dan PEMBAHASAN HASIL

Bab ini berisikan implementasi dari perancangan alat yang telah di buat serta pengujian terhadap alat tersebut.

BAB V PENUTUP

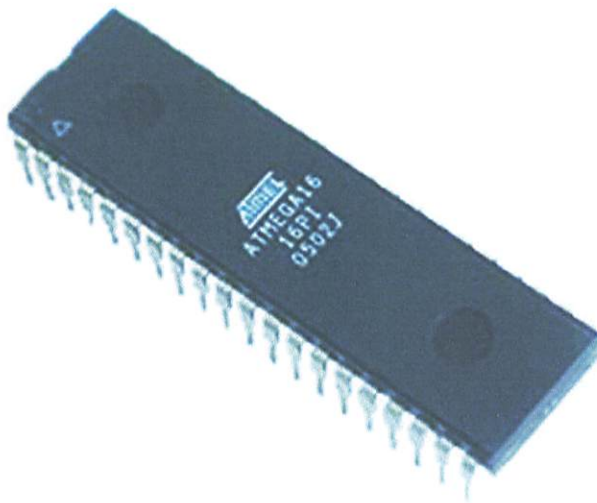
Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1. Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer lengkap dalam satu serpih (*chip*). Mikrokontroler lebih dari sekedar sebuah mikroprosesor karena sudah terdapat atau berisikan ROM (*Read-Only Memory*), RAM (*Read-Write Memory*), beberapa bandar masukan maupun keluaran, dan beberapa *peripheral* seperti pencacah/pewaktu, ADC (*Analog to Digital converter*), DAC (*Digital to Analog converter*) dan serial komunikasi.



Gambar 2.1. ATMega 16

Salah satu mikrokontroler yang banyak digunakan saat ini yaitu mikrokontroler AVR. AVR adalah mikrokontroler RISC (*Reduce Instruction Set Compute*) 8 bit berdasarkan arsitektur Harvard. Secara umum mikrokontroler AVR dapat dikelompokkan menjadi 3 kelompok, yaitu keluarga AT90Sxx, ATMega dan ATtiny. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral* dan fiturnya.^[2]

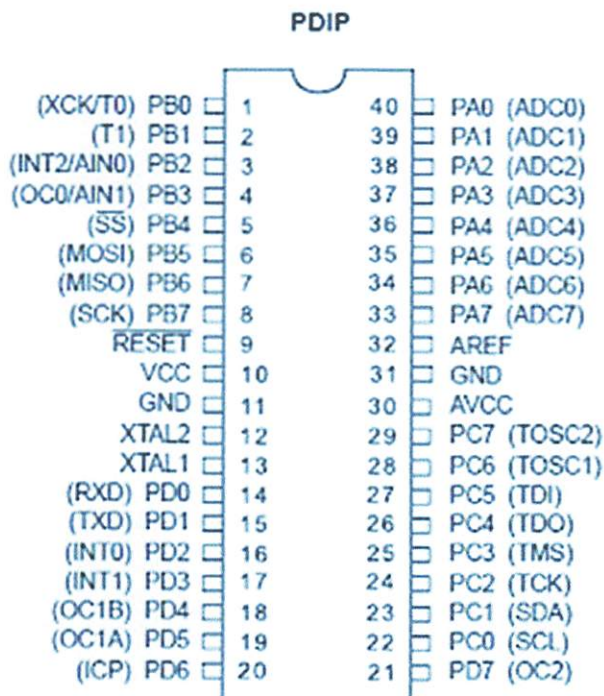
Mikrokontroler ini menggunakan arsitektur Harvard yang memisahkan memori program dari memori data, baik bus alamat maupun bus data, sehingga pengaksesan program dan data dapat dilakukan secara bersamaan (*concurrent*).^[6]

Secara garis besar mikrokontroler ATMega16 terdiri dari :

- a. Arsitektur RISC dengan throughput mencapai 16 MIPS pada frekuensi 16Mhz.

- b. Memiliki kapasitas Flash memori 16Kbyte, EEPROM 512 Byte, dan SRAM 1Kbyte
- c. Saluran I/O 32 buah, yaitu Bandar A, Bandar B, Bandar C, dan Bandar D.
- d. CPU yang terdiri dari 32 buah register.
- e. User interupsi internal dan eksternal
- f. Bandar antarmuka SPI dan Bandar USART sebagai komunikasi serial
- g. Fitur Peripheral
 - Dua buah 8-bit *timer/counter* dengan prescaler terpisah dan mode *compare*
 - Satu buah 16-bit timer/counter dengan prescaler terpisah, mode *compare*, dan mode *capture*
 - *Real time counter* dengan osilator tersendiri
 - Empat kanal PWM dan Antarmuka komparator analog
 - 8 kanal, 10 bit ADC
 - *Byte-oriented Two-wire Serial Interface*
 - *Watchdog timer* dengan osilator internal

2.2. Konfigurasi Pinout ATmega16



Gambar 2.2. Konfigurasi Pinout ATmega16

Pin-pin pada ATmega16 dengan kemasan 40-pin DIP (*dual inline package*) ditunjukkan oleh gambar 1. Guna memaksimalkan performa, AVR menggunakan *arsitektur Harvard* (dengan memori dan bus terpisah untuk program dan data). *Port* sebagai *input/output* digital.

ATmega16 mempunyai empat buah port yang bernama *PortA*, *PortB*, *PortC*, dan *PortD*. Keempat *port* tersebut merupakan jalur *bidirectional* dengan pilihan *internal pull-up*. Tiap *port* mempunyai tiga buah register bit, yaitu *DDxn*, *PORTxn*, dan *PINxn*. Huruf 'x' mewakili nama huruf dari port sedangkan huruf 'n' mewakili nomor bit. Bit *DDxn* terdapat pada *I/O address DDRx*, bit *PORTxn* terdapat pada Bit *DDxn* dalam register *DDRx* (*Data Direction Register*) menentukan arah pin. Bila *DDxn* diset 1 maka Px berfungsi sebagai pin output. Bila *DDxn* diset 0 maka Px berfungsi sebagai pin *input*. Bila *PORTxn* diset 1 pada saat pin terkonfigurasi sebagai pin *input*, maka resistor *pull-up* akan diaktifkan. Untuk mematikan resistor *pull-up*, *PORTxn* harus diset 0 atau pin dikonfigurasi sebagai pin *output*. *Pin port* adalah *tri-state* setelah kondisi *reset*. Bila *PORTxn* diset 1 pada saat pin terkonfigurasi sebagai pin *output* maka pin *port* akan berlogika 1. Dan bila *PORTxn* diset 0 pada saat pin terkonfigurasi sebagai pin output maka pin *port* akan berlogika 0. Saat mengubah kondisi port dari kondisi *tri-state* (*DDxn=0, PORTxn=0*) ke kondisi *output high* (*DDxn=1, PORTxn=1*) maka harus ada kondisi peralihan apakah itu kondisi *pull-up enabled* (*DDxn=0, PORTxn=1*) atau kondisi *output low* (*DDxn=1, PORTxn=0*).

Biasanya, kondisi *pull-up enabled* dapat diterima sepenuhnya, selama lingkungan impedansi tinggi tidak memperhatikan perbedaan antara sebuah *strong high driver* dengan sebuah *pull-up*. Jika ini bukan suatu masalah, maka bit *PUD* pada register *SFIOR* dapat diset 1 untuk mematikan *semua pull-up* dalam semua *port*. Peralihan dari kondisi *input* dengan *pull-up* ke kondisi *output low* juga menimbulkan masalah yang sama. Kita harus menggunakan kondisi *tri-state* (*DDxn=0, PORTxn=0*) atau kondisi *output high* (*DDxn=1, PORTxn=0*) sebagai kondisi transisi.^[6]

Table 2.1. Konfigurasi Pin Port Mikrokontroler ATmega16

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	In	No	Tri-state (Hi-Z)
0	1	0	In	Yes	Pxn will source current if ext.pulled low
0	1	1	In	No	Tri-state (Hi-Z)
1	0	X	Out	No	Output Low (Sink)
1	1	X	Out	No	Output High (Source)

Berikut penjelasan dari fungsi pada masing – masing konfigurasi *Port* mikrokontroler ATmega16 :

- *PortA* :menyediakan input analog untuk fitur ADC, selain itu *portA* juga bisa digunakan sebagai jalur *bi-directionalInput/Output* jika fitur ADC tidak digunakan.

Table 2.2. Fungsi Alternatif Konfigurasi Pin *PortA*

Port Pin	Alternate Function
PA7	ADC7 (ADC7 input channel 7)
PA6	ADC6 (ADC7 input channel 6)
PA5	ADC5 (ADC7 input channel 5)
PA4	ADC4 (ADC7 input channel 4)
PA3	ADC3 (ADC7 input channel 3)
PA2	ADC2 (ADC7 input channel 2)
PA1	ADC1 (ADC7 input channel 1)
PA0	ADC0 (ADC7 input channel 0)

Dari data pada tabel 2.2 dapat diketahui bahwa *PortA* dapat berfungsi sebagai jalur input AD Converter 8 channel. Bila tidak digunakan dalam mode pemrograman alternatif tersebut maka *PortA* dapat digunakan sebagai *biderctionalInput/Output*.

- *PortB* :menyediakan fitur *bi-directionalInput/Output* dan beberapa fungsi pin dengan fitur khusus, antara lain *timer/counter*, komparator analog, serta fitur SPI.

Tabel 2.3. Fungsi Alternatif Konfigurasi Pin *PortB*

Port Pin	Alternate Function
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Master Input/Slave Output)
PB5	MOSI (SPI Master Output/Slave Input)
PB4	SS (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB0	T0 (Timer/Counter0 External Counter Input) XCK (USART External Clock Input/Output)

Dari data tabel 2.3 *PortB* 4-7 merupakan konfigurasi jalur data SPI. Sedangkan *PortB* 0-3 merupakan konfigurasi timer/counter. Bila tidak digunakan pada mode pemrograman tersebut maka *PortB* dapat digunakan menjadi *bidirectional Input/Output*.

- *PortC* : menyediakan fitur *bi-directional Input/Output* dan beberapa fungsi pin dengan fitur khusus, antara lain TWI, komparator analog, serta *Timer Oscillator*.

Table 2.4. Fungsi Alternatif Konfigurasi Pin *PortC*

Port Pin	Alternate Function
PC7	TOSC2 (Timer Oscillator Pin2)
PC6	TOSC1 (Timer Oscillator Pin1)
PC1	SDA (Two-wire Serial Bus Data Input/Output Line)
PC0	SCL (Two-wire Serial Bus Clock Line)

Dari data tabel 2.4 *PortC* 0-1 merupakan konfigurasi jalur data TWI (*Two Wire Interface*). Sedangkan *PortC* 6-7 merupakan konfigurasi timer/counter. Bila tidak

digunakan pada mode pemrograman tersebut maka *PortC* dapat digunakan menjadi *bidirectional Input/Output*.

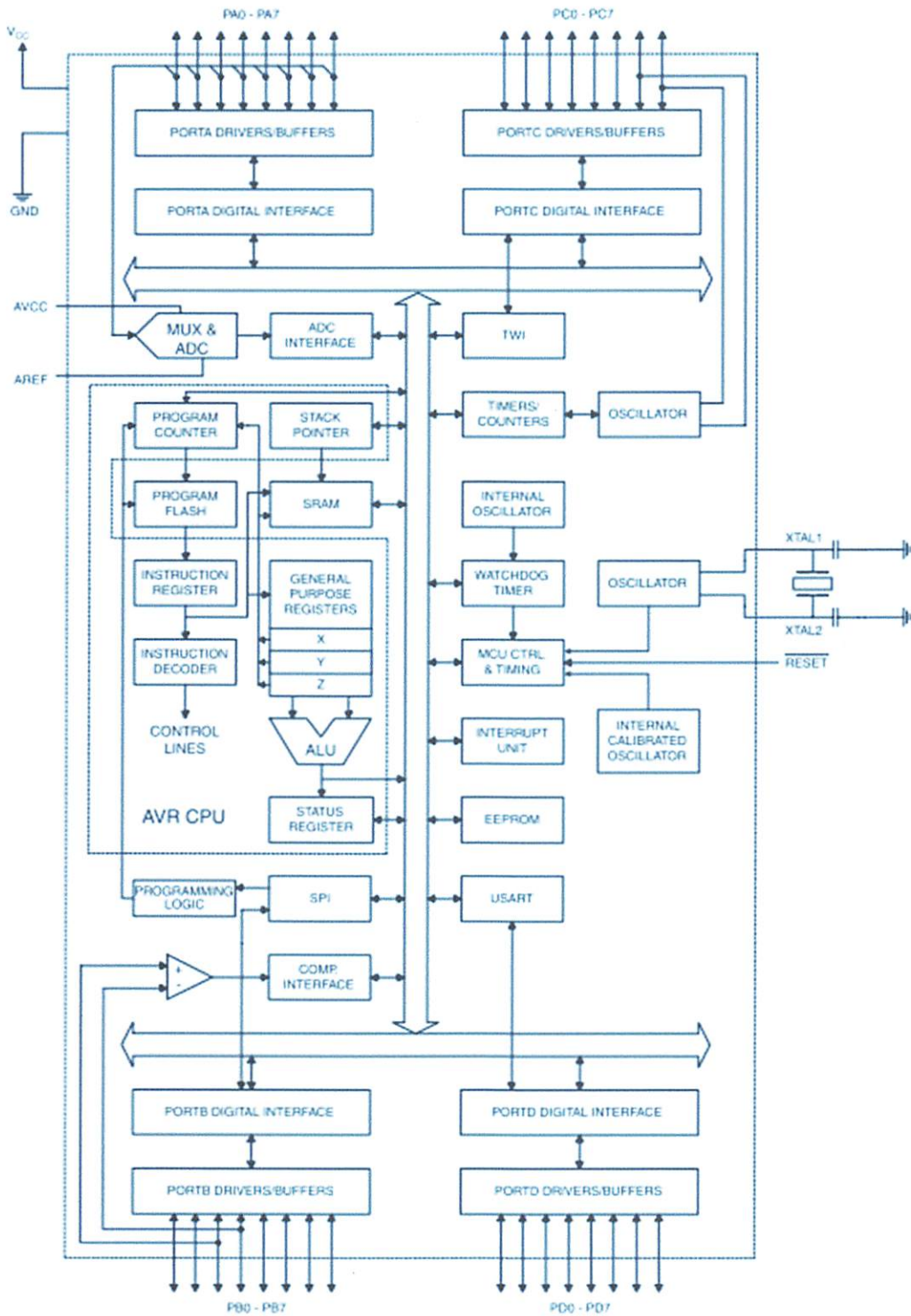
- *PortD* : menyediakan fitur *bi-directional Input/Output* dan beberapa fungsi pin dengan fitur khusus, antara lain komparator analog, interupsi eksternal, serta komunikasi serial.

Table 2.5. Fungsi Alternatif Konfigurasi Pin *PortD*

Port Pin	Alternate Function
PD7	OC (Timer/Counter2 Output Compare Match Output)
PD6	ICP1 (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt1 Input)
PD2	INT0 (External Interrupt0 Input)
PD1	TxD (USART Output Pin)
PD0	RxD (USART Input Pin)

Dari data tabel 2.5 *PortD* 4-7 merupakan konfigurasi *timer*. Sedangkan *PortD* 2-3 merupakan konfigurasi fungsi interupsi. Sementara *PortD* 0-1 merupakan konfigurasi fungsi komunikasi serial. Bila tidak digunakan pada mode pemrograman tersebut maka *PortD* dapat digunakan menjadi *bidirectional Input/Output*.

2.3. Arsitektur Mikrokontroler ATmega16



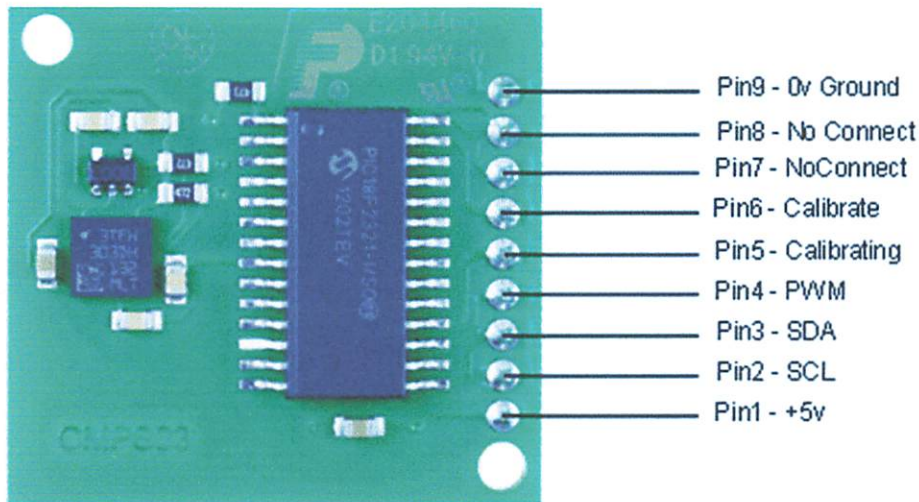
Gambar 2.3. Diagram Blok Arsitektur Mikrokontroler ATmega16

Dari gambar tersebut, dapat dilihat bahwa mikrokontroler ATmega16 merupakan sistem mikroprosesor 8 bit berbasis RISC dengan kecepatan sampai 16MHz memiliki bagian – bagian sebagai berikut; saluran *Input/Output* sebanyak 32

buah (*PortA*, *PortB*, *PortC*, dan *PortD*), ADC (*Analog to Digital Converter*) 8 saluran dengan resolusi 10 bit, tiga buah *Timer/Counter* dengan kemampuan pembandingan (*Analog Comparator*), CPU yang terdiri dari 32 Register dan *Watchdog timer* dengan *Internal Oscillator*, 4 channel PWM (Atmel, 2010). Selain itu mikrokontroler ini juga memiliki SRAM internal sebesar 512Bytes, *flash memory* sebesar 16KiloBytes dengan kemampuan *Read-While-Write*, unit interupsi internal dan eksternal, *Port* antarmuka SPI, EEPROM sebesar 1KiloBytes yang dapat diprogram saat operasi, dan *Port* USART untuk komunikasi serial.^[6]

2.4. Kompas Digital CMPS03

Kompas digital ini hanya memerlukan suplai tegangan sebesar 5 Vdc dengan arus minimal 15mA. Pada CMPS03, arah mata angin dibagi dalam bentuk derajat yaitu : Utara (0°), Timur (90°), Selatan (180°) dan Barat (270°). Ada dua cara untuk mendapatkan informasi arah dari kompas digital ini yaitu dengan membaca sinyal PWM (*Pulse Width Modulation*) pada pin 4 atau dengan membaca data interface I2C pada pin 2 dan 3.^[8]

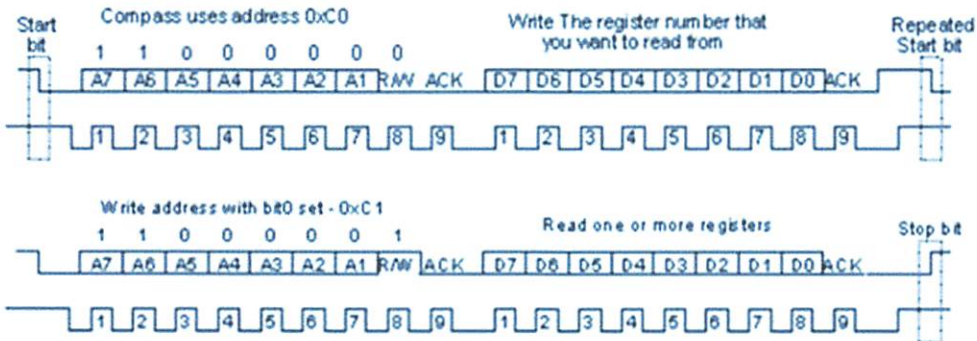


Gambar 2.4. Kompas Didital CMPS03

Sinyal PWM adalah sinyal yang lebar dari pulsanya termodulasi dengan lebar positif dari pulsanya yang mewakili sudut. Lebar pulsa bervariasi dari 1ms (0°) untuk 36.99ms ($359,9^{\circ}$) – dalam kata lain $100\mu\text{s} / ^{\circ}$ dengan 1 mS offset. Sinyal menjadi rendah untuk 65ms antara pulsa, sehingga waktu siklus adalah $65\text{ms} + \text{lebar pulsa}$ – yaitu. 66ms-102ms. Osilator dihasilkan oleh timer 16 bit di prosesor memberikan resolusi 1US, namun tidak direkomendasikan pengukuran ini untuk sesuatu yang lebih baik dari $0,1^{\circ}$ ($10\mu\text{s}$). Pastikan Anda menghubungkan pin I2C, SCL dan SDA, untuk

masukan +5v jika Anda menggunakan PWM, karena tidak ada pull-up resistor pada pin ini.

Pin 2,3 adalah untuk antarmuka I2C dan dapat digunakan untuk mendapatkan pembacaan langsung dari modul. Jika antarmuka I2C tidak digunakan maka pin ini harus di pull-up (untuk +5 v) melalui beberapa resistor 1K ohm.



Gambar 2.5. *I2C Communication Protocol*

I2C protokol komunikasi dengan modul kompas adalah sama dengan eeprom populer seperti 24C04. Pertama mengirim start bit, modul Alamat (0xC0) dengan read/write bit low, sesuai nomor register yang ingin Anda baca. Ini diikuti dengan awal yang berulang dan modul alamat lagi dengan read/write bit high (0xC1). Anda sekarang membaca satu atau dua byte untuk 8bit atau 16bit register masing-masing. Register 16bit yang membaca *byte* tinggi terlebih dahulu. Kompas memiliki *array 16byte* dari register, beberapa di antaranya ganda sebagai register 16 bit sebagai berikut:

Table 2.6. Kompas Array 16byte Register

Register	Function
0	Software Revision Number
2.3	Compass Bearing as byte,i.e.0-255 for a full circle
4.5	Internal Test-sensor1 difference signal – 16 bit signed word
6.7	Internal Test-sensor2 difference signal – 16 bit signed word
8.9	Internal Test-calibration value 1 – 16 bit signed word
10.11	Internal Test-calibration value 2 – 16 bit signed word
12	Unused – Read as Zero
13	Unused – Read as Zero
14	Unused – Read as Undefined
15	Calibrate command – write 255 to perform calibration step. See text

Antarmuka I2C tidak mempunyai pull-up resistor di modulnya, ini harus disediakan tempat lain, kemungkinan besar dengan bus master. Pullup diwajibkan pada kedua SCL dan jalur SDA, tapi hanya sekali untuk seluruh bus, bukan pada setiap modul. Disarankan nilai 1k8 jika Anda akan bekerja sampai 400KHz dan 1k2 atau bahkan 1k jika Anda pergi ke 1MHz. Kompas ini dirancang untuk bekerja sampai *clock speed* standar (SCL) dari 100KHz, namun kecepatan *clock* dapat ditingkatkan sampai 1MHz menyediakan tindakan pencegahan berikut ini diambil; Pada kecepatan di atas sekitar 160KHz CPU tidak dapat merespon cukup cepat untuk membaca I2C data. Oleh karena itu delay kecil 50uS harus dimasukkan kedua sisi penulisan register *address*. Tidak ada delay yang diperlukan di tempat lain dalam urutan. Dengan melakukan ini, sudah diuji modul kompas hingga 1.3MHz *clock speed* SCL.^[8]

Kalibrasi dibutuhkan setidaknya satu kali. Penting untuk dijaga bahwa keadaan sensor selalu datar terhadap permukaan bumi (flat, tidak miring) dan jauhkan dari benda bermagnet.

I2C Method >Untuk mengkalibrasi menggunakan I2C bus, Anda hanya perlu untuk menuliskan 255 (0xff) pada register 15 untuk empat kutub utama Utara, Timur, Selatan dan Barat. Nilai 255 terhapus secara internal dengan otomatis setelah setiap poin

terkalibrasi. Poin kompas dapat diatur dalam urutan apapun, tapi semua empat poin harus dikalibrasi. Misalnya :

1. Pastikan kompas CMPS03 dalam keadaan datar, menunjuk ke Utara. Write 255 pada register 15
2. Pastikan kompas CMPS03 dalam keadaan datar, menunjuk ke Timur. Write 255 pada register 15
3. Pastikan kompas CMPS03 dalam keadaan datar, menunjuk ke Selatan. Write 255 pada register 15
4. Pastikan kompas CMPS03 dalam keadaan datar, menunjuk ke Barat. Write 255 pada register 15

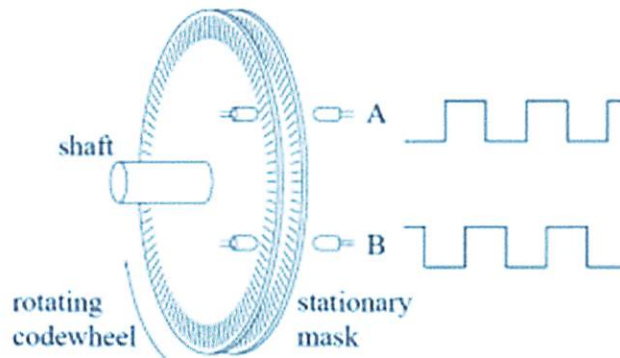
Pin Method > Pin 6 digunakan untuk mengkalibrasi kompas. Pin input kalibrasi (pin 6) sudah mempunyai pullup pada modul dan dapat dibiarkan tidak tersambung setelah kalibrasi. Untuk mengkalibrasi kompas Anda hanya perlu mengatur pin kalibrasi dengan logic low and lalu high kembali untuk setiap poin utama kompas Utara, Timur, Selatan dan Barat. Sederhananya tekan switch yang sudah terhubung dari pin6 ke 0V (Ground). Poin kompas dapat diatur dalam urutan apapun, tapi semua empat poin harus dikalibrasi. Misalnya :

1. Pastikan kompas CMPS03 dalam keadaan datar, menunjuk ke Utara. Tekan dan lepas switch
2. Pastikan kompas CMPS03 dalam keadaan datar, menunjuk ke Timur. Tekan dan lepas switch
3. Pastikan kompas CMPS03 dalam keadaan datar, menunjuk ke Selatan. Tekan dan lepas switch
4. Pastikan kompas CMPS03 dalam keadaan datar, menunjuk ke Barat. Tekan dan lepas switch

2.5. *Incremental Rotary Encoder*

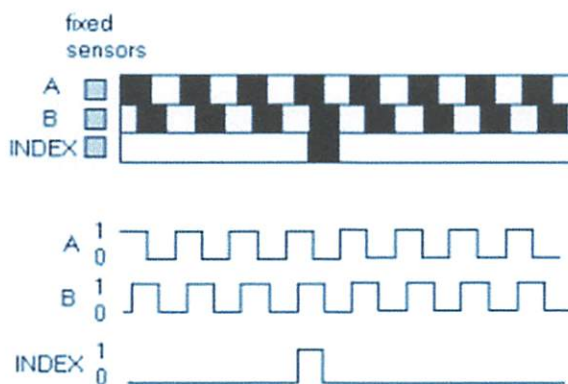
Incremental Rotary encoder terdiri dari dua track atau single track dan dua sensor yang disebut channel A dan B (Gambar 2.5). Ketika poros berputar, deretan pulsa akan muncul di masing-masing channel pada frekuensi yang proporsional dengan kecepatan putar sedangkan hubungan fasa antara *channel A* dan B menghasilkan arah putaran. Dengan menghitung jumlah pulsa yang terjadi terhadap resolusi piringan maka

putaran dapat diukur. Untuk mengetahui arah putaran, dengan mengetahui *channel* mana yang *leading* terhadap *channel* satunya dapat kita tentukan arah putaran yang terjadi karena kedua *channel* tersebut akan selalu berbeda fasa seperempat putaran (*quadrature signal*). Seringkali terdapat *output channel* ketiga, disebut INDEX, yang menghasilkan satu pulsa per putaran berguna untuk menghitung jumlah putaran yang terjadi.



Gambar 2.6. *Incremental Rotary Encoder*

Pola diagram keluaran dari suatu *incremental rotary encoder* ditunjukkan pada Gambar 2.6. Resolusi keluaran dari sinyal *quadrature* A dan B dapat dibuat beberapa macam, yaitu 1x, 2x dan 4x. Resolusi 1x hanya memberikan pulsa tunggal untuk setiap siklus salah satu sinyal A atau B, sedangkan resolusi 4x memberikan pulsa setiap transisi pada kedua sinyal A dan B menjadi empat kali resolusi 1x. Arah putaran dapat ditentukan melalui level salah satu sinyal selama transisi terhadap sinyal yang kedua. Pada contoh resolusi 1x, A = arah bawah dengan B = 1 menunjukkan arah putaran searah jarum jam, sebaliknya B = arah bawah dengan A = 1 menunjukkan arah berlawanan jarum jam.^[10]



Gambar 2.7. Gelombang Keluaran *Incremental Rotary Encoder*

Table 2.7. *Incremental Rotary Encoder* Searah Jarum Jam

Phase	A	B
1	1	0
2	1	1
3	0	1
4	0	0

Table 2.8. *Incremental Rotary Encoder* Berlawanan Arah Jarum Jam

Phase	A	B
1	1	0
2	0	0
3	0	1
4	1	1

2.6. Ultrasonik Ping (parallax)

Sensor jarak ultrasonik ping adalah sensor 40 khz produksi parallax yang banyak digunakan untuk aplikasi atau kontes robot cerdas. Kelebihan sensor ini adalah hanya membutuhkan 1 sinyal (SIG) selain jalur 5 v dan ground.^[3] Perhatikan gambar 2.7 dibawah ini :



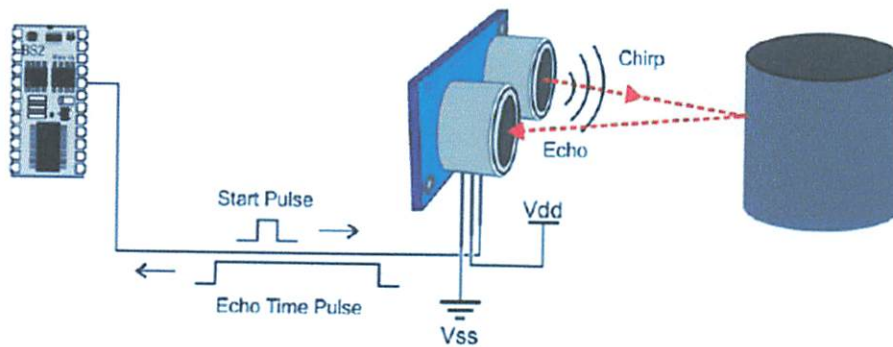
Gambar 2.8. Ultrasonic ping

Sensor PING mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 KHz) selama $t = 200 \text{ us}$ kemudian mendeteksi pantulannya. Sensor PING memancarkan gelombang ultrasonik sesuai dengan kontrol dari mikrokontroller pengendali (pulsa trigger dengan tout min 2 us)

Spesifikasi sensor ultrasonik PING :

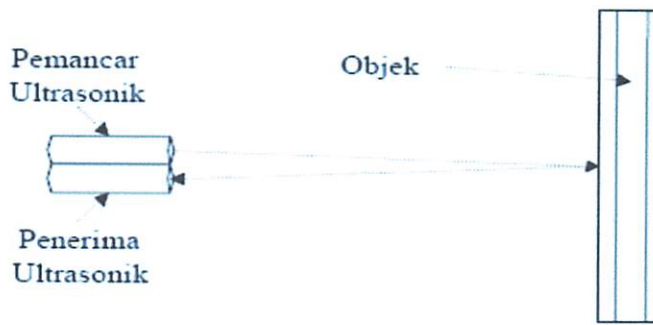
- Kisaran pengukuran 3 cm – 3 m
- Input trigger – positive TTL pulse, 2 us min, 5 us tipikal
- Echo hold off 750 us dari of trigger pulse
- Delay before next measurement 200 us
- Brust indikator LED menampilkan aktivitas sensor

Sistem mikrokontroler ATmega16 dan *software basic stamp Editor* diperlukan untuk memprogram mikrokontroler dan mencoba sensor ini. Keluaran dari pin SIG ini yang dihubungkan ke salah satu pin mikrokontroler. Berikut contoh aplikasi sensor PING pada mikrokontroler, dimana pin SIG terhubung ke PORTD, dan memberikan tegangan 5VDC dan ground. Fungsi **SIGOUT** untuk mentrigger ping. Perhatikan gambar 2.9 di bawah ini.^[3]



Gambar 2.9. Prinsip Kerja Sensor Ultrasonik Ping

Besar amplitudo sinyal elektrik yang dihasilkan unit sensor penerima tergantung dari jauh dekatnya objek yang dideteksi serta kualitas dari sensor pemancar dan sensor penerima. Proses sensing yang dilakukan pada sensor ini menggunakan metode pantulan untuk menghitung jarak antara sensor dengan obyek sasaran. Jarak antara sensor tersebut dihitung dengan cara mengalikan setengah waktu yang digunakan oleh sinyal ultrasonik dalam perjalanannya dari rangkaian pengirim sampai diterima oleh rangkaian penerima, dengan kecepatan rambat dari sinyal ultrasonik tersebut pada media rambat yang digunakannya, yaitu udara.^[3] Prinsip pantulan dari sensor ultrasonik ini dapat dilihat pada gambar 2.10 sebagai berikut :



Gambar 2.10. Prinsip Kerja Pemantulan Sensor Ultrasonik Ping

2.7. Saklar Tekan/*Push Botton*

Saklar tekan merupakan jenis saklar yang dioperasikan dengan cara menekan sebuah tombol. Terdapat dua jenis saklar semacam ini, yaitu jenis *push to make* (PTM) dan *push to break* (PTB).



Gambar 2.11. Simbol Saklar Tekan

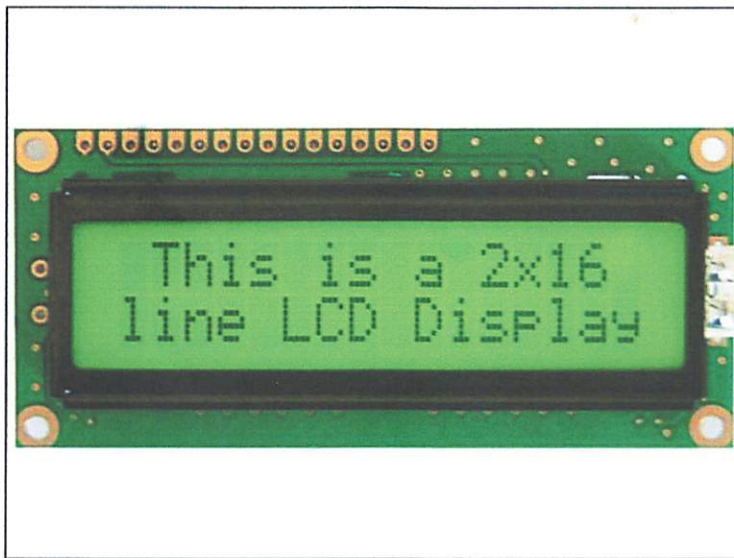
Kebanyakan diantara saklar tekan merupakan jenis PTM. Dengan menekan tombol saklar, kontak-kontak akan tertekan hingga saling bersentuhan dan saklar menutup. Sedangkan saklar tekan jenis PTB, kontak-kontaknya adalah kontak normal tertutup, namun akan dipaksa membuka ketika tombol ditekan.

Masing-masing saklar tekan jenis PTM dan PTB, dapat bekerja untuk membentuk atau memutuskan sambungan selama sekejap atau menguncinya (*latching*). Sebuah saklar yang membentuk atau memutuskan sambungan selama sekejap hanya akan menutup atau membuka selama tombol masih ditekan. Ketika tombol dilepaskan saklar akan kembali ke posisi semula.

Pada saklar sambungan yang mengunci, tombol akan tetap berada pada posisi tertekan setelah pertama kali ditekan. Kontak-kontak saklar akan tetap menutup atau membuka, bergantung pada jenis saklar yang bersangkutan. Anda harus menekan tombol itu sekali lagi untuk membuka kunci dan mengembalikan tombol ke posisi normalnya.^[11]

2.8. *Alphanumeric LCD*

LCD *display module* terdiri dari dua bagian, yang pertama yaitu panel LCD sebagai media penampil informasi dalam bentuk huruf/angka yang terdiri dari baris dan kolom tergantung spesifikasi dimensi yang diberikan. Hal ini mempengaruhi jumlah karakter yang dapat ditampilkan LCD. Bagian kedua merupakan sebuah sistem elektronik yang dibentuk dengan mikrokontroler yang ada dibalik panel LCD, berfungsi mengatur tampilan informasi serta berfungsi mengatur komunikasi dengan port mikrokontroler (master) yang memakai LCD itu(Surya, 20011:14). Dengan demikian pemakain LCD cukup dengan mengirimkan sinyal kode – kode ASCII dari informasi yang ditampilkan seperti layaknya memakai sebuah printer.



Gambar 2.12. Modul LCD Dimensi 2x16

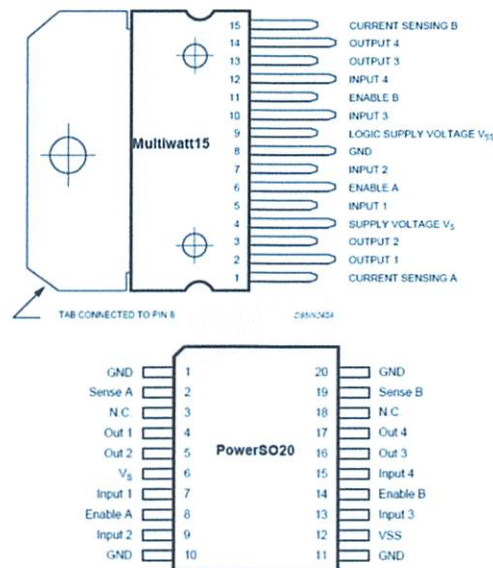
Untuk LCD dengan dimensi 2x16 memiliki 8 jalur(DB0...DB7) data sebagai jalur komunikasi dengan mikrokontroler. Semua jalur ini nantinya dipakai sebagai media penulisan dan pembacaan untuk kode ASCII yang dikirimkan mikrokontroler serta penerimaan kode – kode itu pada LCD untuk diolah menjadi tampilan pada *display* LCD maupun sebagai pengaturan kerja dari LCD tersebut. Selain itu dilengkapi pula dengan E, R/W, dan RS seperti layaknya komponen yang kompatibel dengan mikroprosesor. Kombinasi ini sesuai dengan arsitektur yang diciptakan pabrikan semisal motorola dan intel. Selain itu kode – kode tersebut juga merepresentasikan sinyal sebagai pengatur proses kerja LCD.^[5]

2.9. Penguat Motor (IC L298)

Pada dasarnya gerakan motor dikontrol oleh mikrokontroler namun karena arus yang keluar dari mikrokontroler tidak dapat menggerakkan motor maka diperlukan rangkaian luar yang mampu menggerakkan motor. Oleh karena itu IC L298 digunakan sebagai rangkaian *driver* untuk menjalankan motor berdasarkan perintah dari mikrokontroler.

IC L298 merupakan sebuah *driver* untuk motor dc maupun *motor stepper* dengan konfigurasi seperti gambar 2.12. Satu buah IC L298 bisa digunakan untuk mengontrol dua buah motor dc. L298 mampu beroperasi pada tegangan 2,5V sampai 50V. Besar arus output yang dihasilkan sampai 4A dan mempunyai proteksi terhadap temperatur yang berlebihan.

Pin *Enable A* dan *B* untuk mengendalikan jalan atau kecepatan motor, pin Input 1 sampai 4 untuk mengendalikan arah putaran. Pin *Enable* diberi VCC 5V untuk kecepatan penuh.^[9]

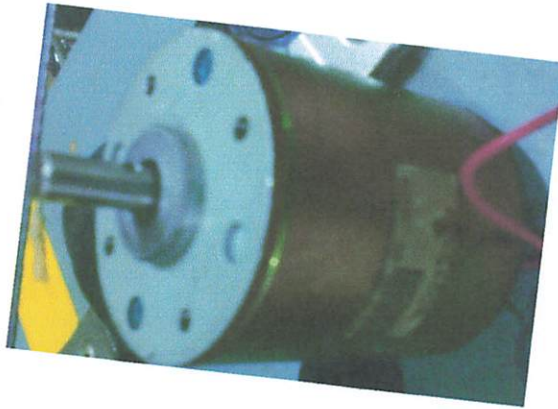


Gambar 2.13. Pin IC L298

2.10. Motor DC

Motor DC adalah piranti elektronik yang mengubah energi listrik menjadi energi mekanik berupa gerak rotasi. Pada motor DC terdapat jangkar dengan satu atau lebih kumparan terpisah. Tiap kumparan berujung pada cincin belah (komutator). Dengan adanya insulator antara komutator, cincin belah dapat berperan sebagai saklar kutub ganda (*double pole, double throw switch*). Motor DC bekerja berdasarkan prinsip gaya Lorentz, yang menyatakan ketika sebuah konduktor beraliran arus diletakkan dalam

medan magnet, maka sebuah gaya (yang dikenal dengan gaya Lorentz) akan tercipta secara ortogonal diantara arah medan magnet dan arah aliran arus.



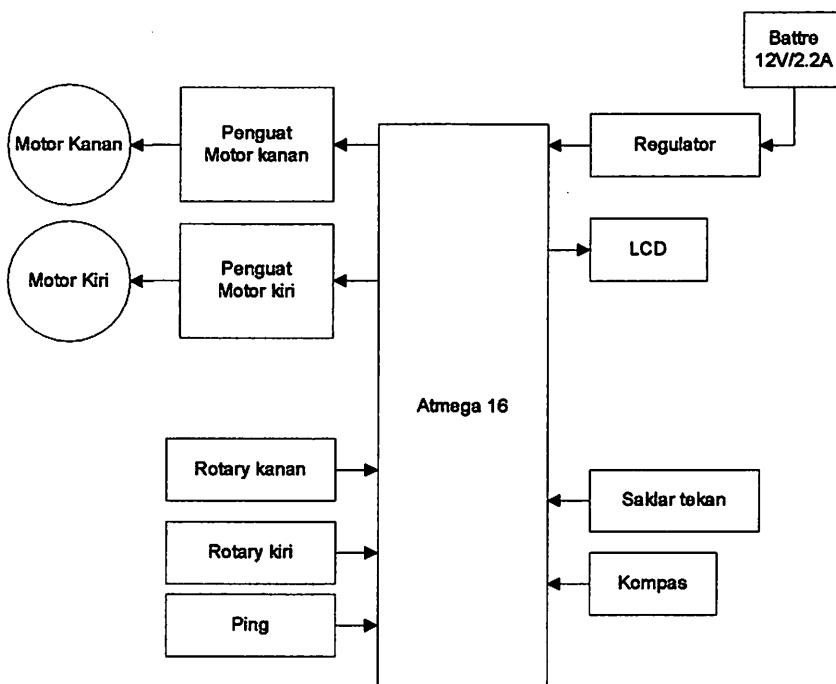
Gambar 2.14. Motor DC

BAB III PERANCANGAN DAN ANALISA SISTEM

Bab ini menjelaskan mengenai analisa sistem dan perancangan alat. Analisa ditunjukkan untuk memberikan analisa secara umum tentang perancangan alat dan memberikan solusi terhadap permasalahan yang dihadapi. Dari analisa ini diharapkan dapat menghasilkan suatu ide perancangan yang sesuai sebagai solusi dari analisa masalah yang dihadapi.

3.1. Gambaran Umum Desain Alat

Sistem yang akan di buat dalam perancangan robot ini adalah mikrokontroler berbasis sistem minimum ATmega AVR yang berupa minimum sistem dengan spesifikasi inti mikrokontroler dari keluaran Atmel Mega AVR dan peripheral I/O sebagai indikator simulasi praktis terhadap fitur – fitur yang ada pada mikrokontroler tersebut. Nantinya sejumlah program akan ditanamkan pada chip inti mikrokontroller yang kemudian akan mengolah data sesuai dengan hubungan antarmuka antara masing – masing peripheral I/O dengan jalur I/O mikrokontroler dan sensor sebagai input untuk navigasi robot.



Gambar 3.1. Blok Diagram Sistem Kendali Robot

3.2. Fungsi Masing-masing Rancangan Alat

Dari diagram tersebut diatas dapat diterangkan bahwa desain modul praktikum mikrokontroler yang akan dibuat memiliki bagian – bagian yang berfungsi sebagai berikut :

1. Minimum sistem yang berbasis ATMEGA AVR, dimana jenis ATMEGA AVR yang digunakan adalah jenis ATMEGA16 dengan jenis paket DIP(*Dual In-line Package*) 40 pin. Minimum sistem ini berfungsi sebagai suatu sistem inti otak dari segala proses pengolahan data dan juga penyedia fungsi pin – pin I/O.
2. *peripheral I/O* sebagai pemeberi kontrol input maupun indikator dari proses *output* pengolahan data yang dilakukan mikrokontroler melalui koneksi antarmuka dengan pin – pinnya. Adapun bagian – bagian dari *peripheral I/O* tersebut memiliki masing – masing fungsi :
 - 2.1 Kompas digital/CMPS03 : indikasi input sebagai pendeteksi arah
 - 2.2 *Incremental Rotary encoder* : indikasi input pendeteksi jarak tempuh robot
 - 2.3 Ultrasonik Ping : indikasi input pendeteksi jarak
 - 2.4 Saklar tekan : indikasi input sebagai kontrol pergerakan robot
 - 2.5 LCD Display 16 x 2: indikasi output yang lebih kompleks dengan membentuk kombinasi karakter dalam dimensi 2 baris dan 16 kolom
 - 2.6 Penguat Motor: indikasi output sebagai kontrol pergerakan motor DC

3.3. Prinsip Kerja Alat

Prinsip kerja alat mobile robot beroda ini secara umum adalah memberikan masing-masing kode program pada sistem kontrol minimum sistem AVR sesuai dengan antarmuka *peripheral I/O board* yang terpasang melalui koneksi antara pin mikrokontroler dengan *peripheral I/O board* tersebut. Dalam hal ini *peripheral I/O board* ada yang berfungsi sebagai *input* dan ada pula yang berfungsi sebagai *output*.

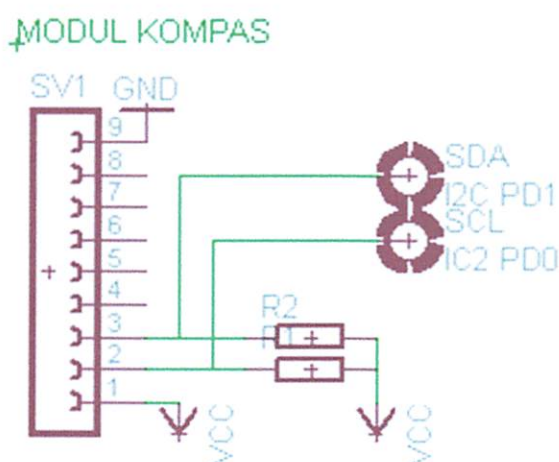
Dalam perancangan skripsi ini penulis membuat pengujian yang terdiri dari simulasi masing–masing bagian *peripheral I/O board*. Adapun beberapa simulasi tersebut adalah :

1. Pengujian sensor kompas digital
2. Pengujian sensor *incremental Rotary encoder*
3. Pengujian sensor ultrasonik ping
4. Pengujian penguat motor
5. Pengujian sistem

3.4. Perancangan Perangkat Keras

3.4.1. Perancangan Rangkaian Kompas Digital

Disini saya menggunakan PCB lubang untuk membuat dudukan modul kompas digital, sekaligus saya beri pin header untuk supply dan pin I2C untuk memudahkan koneksi dengan mikrokontroler. Gambar rangkaiannya seperti gambar 3.2 di bawah ini.



Gambar 3.2. Tempat Modul Kompas Digital CMPS03

Kompas digital ini hanya memerlukan *supply* tegangan sebesar 5VDC dengan arus minimal 15mA. Pada kompas digital, arah mata angin di bagi dalam bentuk derajat: Utara (0^0), Timur (90^0), Selatan (180^0) dan Barat (270^0). Untuk mendapatkan informasi dari kompas digital ini dengan membaca data interface I2C, SCL dan SDA di koneksikan pada *peripheral* I/O dengan kabel konektor ke pin mikrokontroler (PORTD).

Menentukan arah sudut mata angin, maka menggunakan persamaan:

$$kompas = (data * 1.40625) + 1.40625 \dots \dots \dots (3.4-1)$$

$$kompas = (255 * 1.40625) + 1.40625 = 360^0$$

data merupakan perubahan nilai yang di hasilkan sinyal pwm melalui modul sensor kompas. Perhitungan rumus di atas menghasilkan besaran nilai dalam satuan derajat($^{\circ}$).

3.4.2. Perancangan *incremental rotari encoder*

Untuk menggerakkan robot baik maju atau mundur dengan jarak tertentu biasanya dengan menghitung waktu. Dengan kecepatan dan durasi tertentu maka akan di dapat ditentukan bahwa robot telah bergerak dengan jarak tertentu melalui persamaan : $S = v * t$. Namun kecepatan motor pada robot tidaklah selalu stabil seiring dengan perubahan kapasitas *battery* sebagai sumber daya. Pada saat kapasitas *battery* menurun maka kecepatan putar motor akan berkurang sehingga perhitungan nilai t (durasi) tidaklah akurat, untuk perhitungan jarak yang lebih akurat dapat dilakukan dengan menghitung gerakan roda seberapa jauh roda berputar. Proses modul *incremental rotary encoder* ini piringan rotary memiliki resolusi 360 pulsa dan diameter piringan 4cm dengan lubang piringan 266, maka dengan menggunakan persamaan :

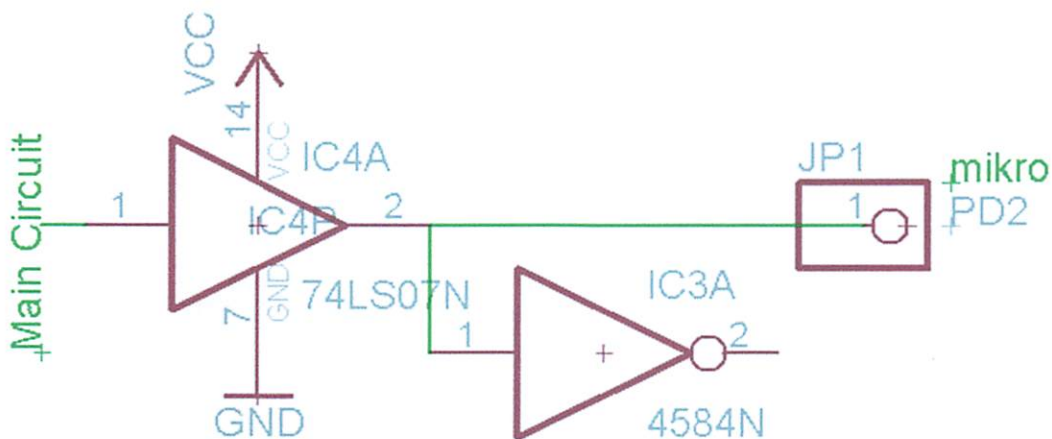
$$S = \pi * d \dots\dots\dots(3.4-2)$$

$$S = 3,14 * 4cm = 12,56cm$$

Jumlah pulsa tiap jarak tempuh menggunakan persamaan:

$$\text{jumlah pulsa} = \frac{\text{jarak tempuh}}{\text{keliling pringan rotary}} * \text{jumlah lubang} \dots\dots\dots(3.4-3)$$

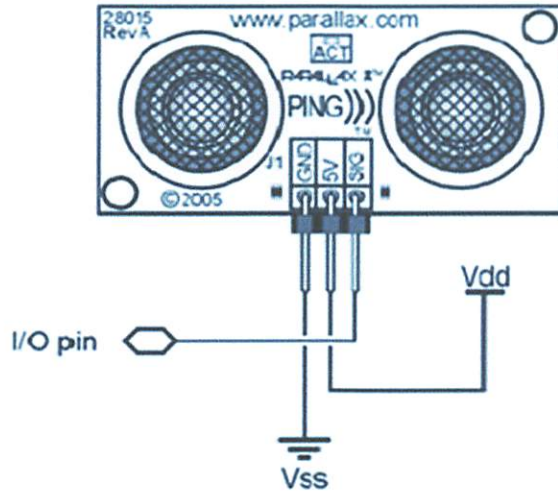
$$\text{jumlah pulsa} = \frac{17}{12,56} * 266 = 360$$



Gambar 3.3. Skema Rangkaian *Incremental Rotary Encoder*

3.4.3. Perancangan ultrasonik ping

Sensor yang digunakan berupa modul ultrasonik ping, sensor ini merupakan sensor yang menggunakan prinsip *time of flight* (waktu suara untuk pergi dan pulang) untuk mengukur jarak.



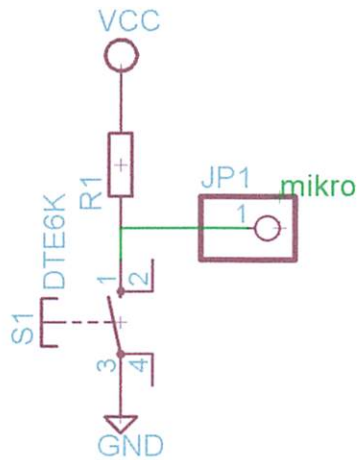
Gambar 3.4. Modul Ultrasonik Ping

Rangkaian yang terlihat pada gambar 3.4, Ultrasonik parallax mempunyai 3 kaki *interface*, *GND*, *Vcc* dan *SIG*. Namun hanya kaki *SIG* yang terhubung pada *peripheral I/O* dengan kabel konektor ke pin mikrokontroler (PORTD.6), mikrokontroler diset sebagai *output* dan memberikan data ke ultrasonik ping. Kaki *Vcc* pada ultrasonik ping ini di beri *supply voltage* sebesar 5 volt, sedangkan *GND* terhubung dengan *ground*.

3.4.4. Perancangan saklar tekan

Untuk memberi masukan kedalam pin mikrokontroler dengan cara termudah adalah dengan menggunakan saklar tekan. Prinsip kerja yang sederhana untuk membaca tombol ini nantinya dapat dikembangkan untuk membaca pemrograman.

tombol merupakan piranti masukan mekanis yang berfungsi untuk memberikan inputan data ke mikrokontroler.



Gambar 3.5. Skema Rangkaian Saklar Tekan/*Push Botton*

Metode perancangan dengan skema gambar 3.5, memungkinkan kita untuk memberi inputan pada pin mikrokontroler dengan pull-up eksternal. Sehingga apabila kita menghubungkan pada portC, pembacaan logika dengan skema rangkaian ini yaitu apabila saklar terbuka akan berlogika 1 dan saat tertutup akan berlogika 0.

3.4.5. Perancangan LCD 2x16

Dalam perancangan ini menggunakan sebuah layar LCD dengan dimensi 16x2. Dimana hal ini berarti LCD terdiri dari tampilan 2 baris yang masing – masing barisnya berisikan 16 karakter. Berikut fungsi masing – masing pin sinyal LCD.^[6]

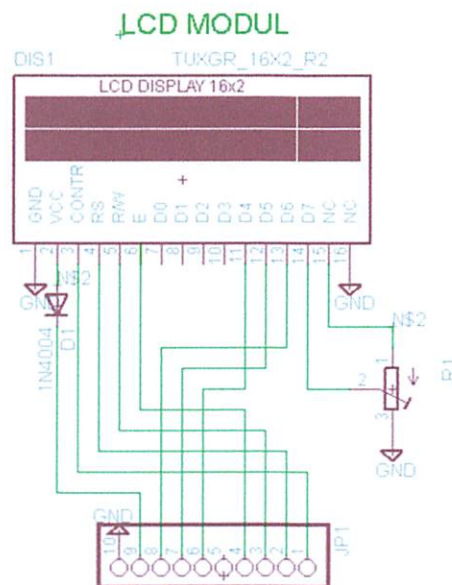
Table 3.1. Fungsi Masing-masing Sinyal LCD

Nama sinyal	Fungsi
DB0-DB7	Merupakan saluran data, berisi perintah dan data yang akan di tampilkan di LCD
Enable	Sinyal operasi awal, sinyal ini mengaktifkan data tulis atau baca
R/W	Sinyal seleksi tulis atau baca 0 : tulis 1 : baca
RS	Sinyal pemilih register 0 : instruksi register (tulis) 1 : data register (baca dan tulis)

Secara garis besar pemrograman LCD diatur oleh 3 sinyal yaitu RS, R/W, Enable dan 8 buah saluran data. Karena pada tugas akhir ini LCD difungsikan sebagai komunikasi 4 bit, jadi saluran data yang dipakai hanya D4 s/d D7. VR pada pin 14 digunakan untuk mengatur kontras LCD, sedangkan pin 2 dihubungkan dengan diode agar tegangan yang masuk sesuai dengan datasheet yaitu 4,3 V.

$$\begin{aligned} \text{Tegangan jatuh diode} &= 0,7\text{V} \\ V_{cc} &= 5\text{V} \\ \text{Maka tegangan input} &= V_{cc} - \text{tegangan jatuh diode} \\ &= (5 - 0,7) \text{ V} = 4,3 \text{ V} \end{aligned}$$

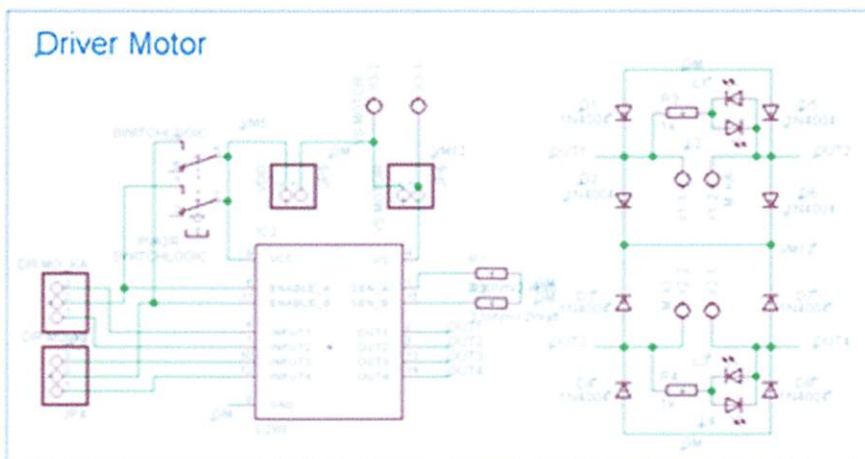
Berikut skema rangkaian antarmuka LCD



Gambar 3.6. Skema Rangkaian LCD Modul

3.4.6. Perancangan Penguat Motor DC

Rangkaian penguat motor DC ini di gunakan untuk mengatur arah putaran motor dan PWM untuk mengatur kecepatan pada motor, PWM sangat tepat dan mudah dibandingkan dengan menaik atau menurunkan tegangan pada motor DC. dengan hanya mengganti lebar pulsa saja kita dapat mengatur kecepatan motor DC dengan kecepatan yang sangat variatif dan lebih detail serta halus.

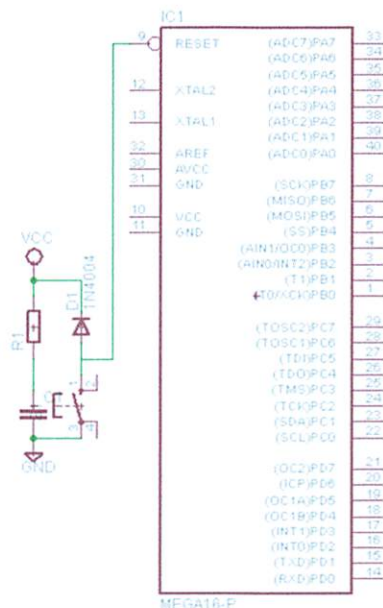


Gambar 3.7. Skema Rangkaian Penguat Motor DC

3.4.5. Perancangan Minimum Sistem ATmega16

3.4.5.1. Perancangan Rangkaian Reset dan Clock

Perancangan rangkaian reset pada mikrokontroler ATmega16 memberikan logika low pada pin reset mikrokontroler ATmega16. Rangkaian reset ini diperoleh dari *application note AVR Design Consideration* dari ATMEL. Berikut ialah gambar rancangan rangkaian reset pada ATmega16:



Gambar 3.8. Rangkaian Reset

Osilator pada rangkaian minimum sistem ATmega16 menggunakan kristal 12MHz dan kapasitor 22 pF. Nilai kapasitor ini diperoleh dari tabel datasheet tentang penggunaan kapasitor untuk rangkaian osilator / sistem clock pada ATmega16. Sesuai

Databook ATmega16 dengan penggunaan kristal 12MHz dan UBRR 17 maka didapat perhitungan sebagai berikut :

$$UBRR = \frac{f_{osc}}{16 \times Baud} - 1 \dots\dots\dots(3.4.5-1)$$

$$Baud = \frac{12000000}{16 \times (17 + 1)} - 1$$

$$Baud = \frac{12000000}{288} - 1$$

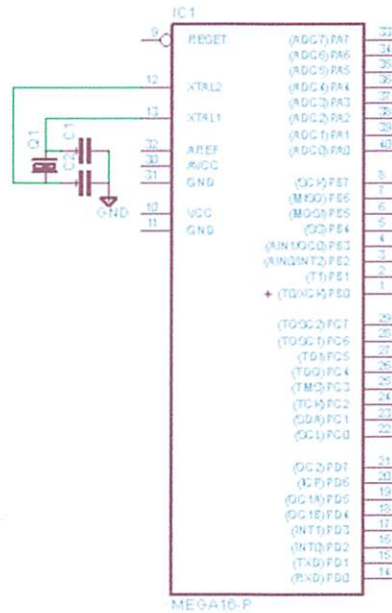
$$Baud = 41665.67$$

Keterangan :

UBRR = nilai register *UBRRH* dan *UBRRL*(0 – 4095 desimal)

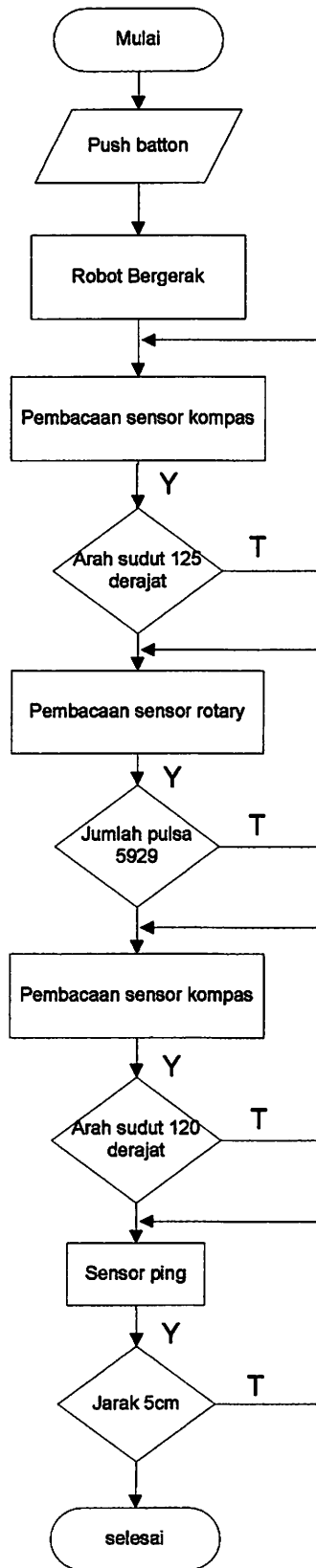
fosc = frekuensi osilator

Baud = Baud rate(bit per detik)

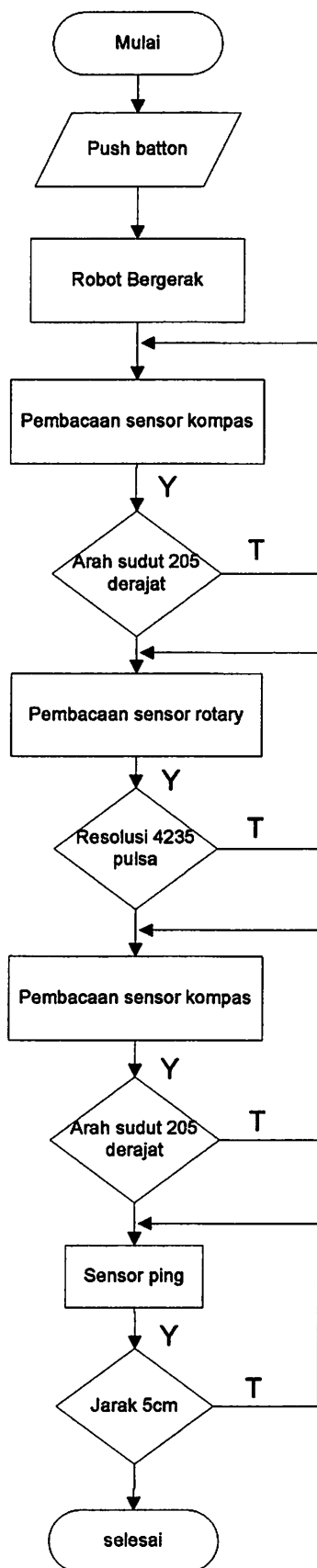


Gambar 3.9. Rangkaian Clock

3.5. Flowchart



Gambar 3.10. Flowchart Saat Pengambilan Keranjang

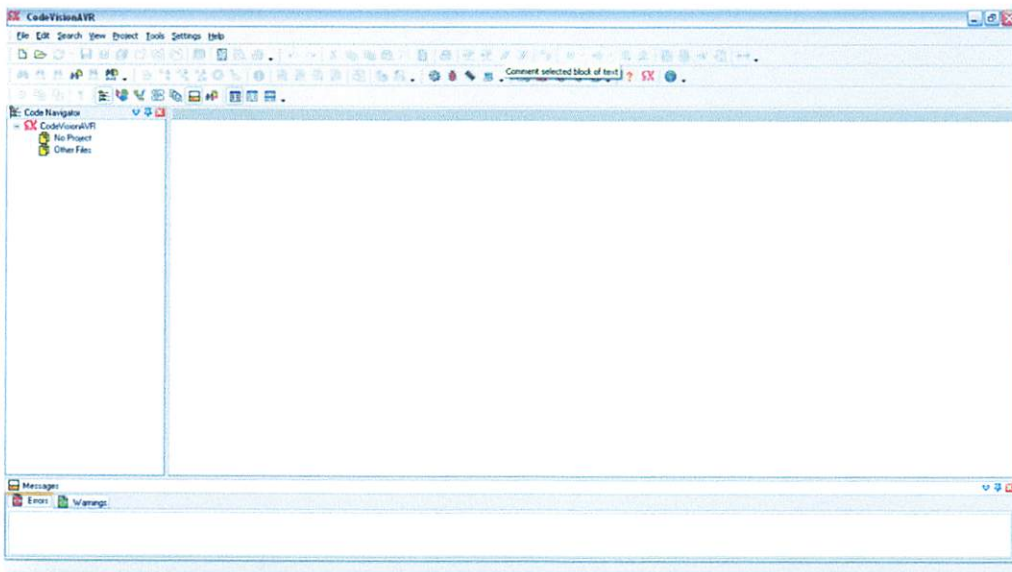


Gambar 3.11. flowchart saat meletakkan keranjang

3.6. *Software CodeVision AVR*

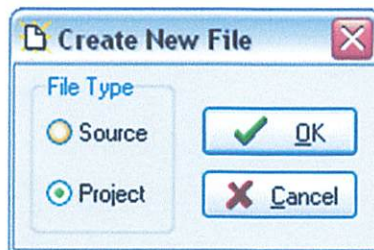
Software dalam penyusunan tugas akhir ini menggunakan *CodeVision AVR*, dimana *software* ini hampir sama dengan IDE (*IntergratedDevelopment Environment*) lainnya. *CodeVision AVR* dilengkapi dengan *source code editor*, *compiler*, *linker*, dan dapat memanggil Atmel AVR Studio untuk *debugernya*. *CodeVision AVR* versi evaluasi dapat di-download pada www.hpinfotech.ro. *CodeVision AVR* versi evaluasi dapat kita pakai secara gratis.

Untuk memulai menjalankan program setelah melakukan proses instalasi buka *CodeVision AVR* melalui **Start |All Program || CodeVision|| CodeVision AVR C compiler.**



Gambar 3.12. Tampilan Awal *CodeVision AVR*

Kemudian untuk memulai new project pilih file| new| pilih file type-> project.

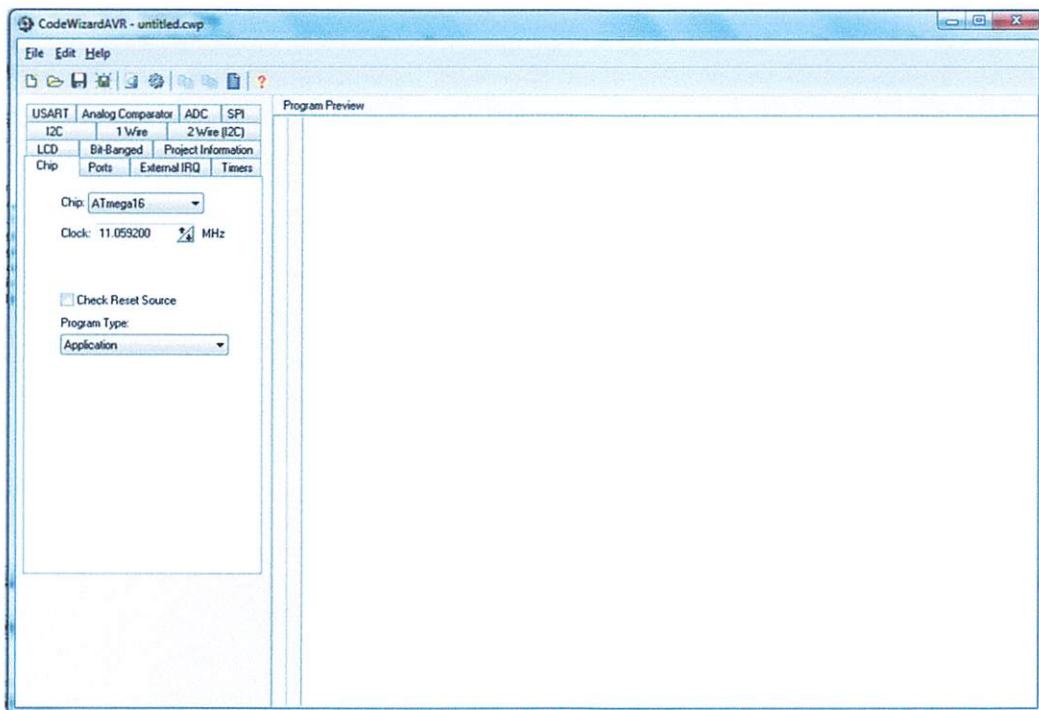


Gambar 3.13. Membuat Project Baru



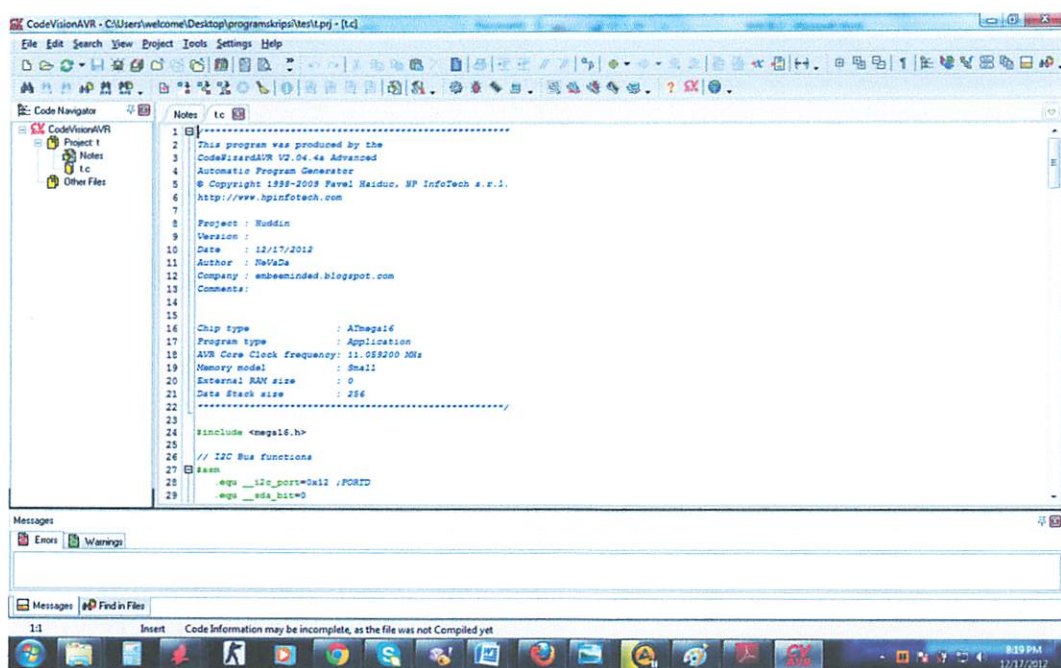
Gambar 3.14. Pertanyaan Membuat Project Baru

Akan muncul tampilan seperti diatas dan akan menanyakan apakah akan menggunakan codevision avr wizard untuk membuat project baru, selanjutnya pilih saja yes. Kemudian akan tampil konfigurasi USART, Analog Komparator, ADC, I2C, LCD, Bit-banged, project informatison, chip, port, external RQ dan Timer. Kita tinggal mengatur program yang akan kita buat melalui codevision avr wizard ini. Misalnya untuk konfiurasi chip: ATmega32, clock: 11.059200 Mhz. Untuk pengaturan port sebagai input/output pilih port, dan seterusnya.



Gambar 3.15. Pengaturan Konfigurasi Chip

Jika kita sudah menkonfigurasi project pilih file| generate. Kemudian beri nama file source (*.c), file project (*.Prj) dan file project codewizard(*.cwp). Setelah diberi nama tekan save pada masing – masing pilihan jendela *windows*. Misalkan saya mencoba dengan memberi nama coba pada masing – masing pilihan, maka tampilannya sebagai berikut :



Gambar 3.16. Hasil konfigurasi dengan *CodeWizard CodeVision AVR*

BAB IV

PENGUJIAN ALAT DAN PEMBAHASAN HASIL

Dalam bab ini membahas tentang pengujian alat dan pembahasan hasil yang telah di rancang, dimana meliputi perangkat keras dan perangkat lunak untuk mengetahui sistem yang akan di rancang sesuai terhadap fungsi yang di harapkan , di lakukan terhadap sistem tersebut. Berikut di jelaskan mengenai prosedur dan hasil pengujian.

4.1. Pengujian LCD

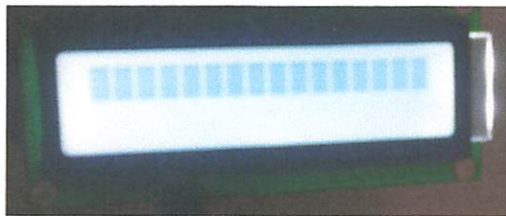
4.1.1. Peralatan yang digunakan

1. Modul LCD
2. Kabel konektor
3. DC power supply 5VDC

4.1.2. Langkah-langkah pengujian

Koneksikan pin modul LCD pada power supply yang mempunya nilai 5VDC dan ground

4.1.3. Hasil pengujian



Gambar 4.1. Hasil pengujian Modul LCD

4.1.4. Analisa pengujian

Sesuai dengan hasil pengujian dapat ditarik sebuah analisa bahwa LCD tiap kolom dan baris bekerja dengan baik. Nantinya pada librari *codevision* AVR hanya untuk menampilkan karakter LCD pada baris dan kolom digunakan dalam perintah “`lcd gotoxy`” sedangkan untuk menampilkan tulisan digunakan perintah “`lcd pustf`”.

4.2. Pengujian Kompas Digital CMPS03

4.2.1. Peralatan yang digunakan

1. Minimum sistem AVR
2. Modul CMPS03
3. Kompas Analog
4. Modul LCD
5. Kabel konektor
6. Downloader
7. DC power supply
8. PC/Laptop sebagai media pemrograman

4.2.2. Langkah-langkah pengujian

1. Hidupkan minimum sistem *board* dengan sumber tegangan 5VDC
2. Koneksikan pin CMPS03 pada *peripheral I/O* dengan kabel konektor ke pin mikrokontroler (PORTD)
3. Koneksikan pin LCD pada *peripheral I/O* dengan kabel konektor ke pin mikrokontroler (PORTA)
4. Membuat program CMPS03 melalui *CodeVision AVR* dan mendownload program tersebut ke mikrokontroler

4.2.3. Hasil pengujian

Tabel 4.1. Hasil pengujian kompas digital/CMPS03

No	Arah mata angin utara	Data	Kompas
		Lebar pulsa	Digital
1	Timur Laut	31	45
2	Timur	63	90
3	Tenggara	95	135
4	Selatan	127	180
5	Barat Daya	159	225
6	Barat	191	270
7	Barat Laut	223	315



Gambar 4.2. Hasil Pengujian Kompas Digital CMPS03

4.2.4. Analisa pengujian

Dari hasil pengujian di gunakan persamaan berikut untuk menentukan perubahan sudut arah mata angin.^[8]

$$\text{Kompas} = (\text{data} * 1.40625) + 1.40625 \dots\dots\dots(4.2.4-1)$$

Dari rumus di atas, data merupakan perubahan nilai yang di hasilkan sinyal pwm melalui modul sensor kompas. Perhitungan rumus di atas menghasilkan besaran nilai dalam satuan derajat(⁰)

Contoh :

1. Pada saat data yang di hasilkan bernilai 31, maka nilai kompas =
 $(31 * 1.40625) + 1.40625 = 45(^0)$
2. Pada saat data yang di hasilkan bernilai 63, maka nilai kompas =
 $(63 * 1.40625) + 1.40625 = 90(^0)$
3. Pada saat data yang di hasilkan bernilai 95, maka nilai kompas =
 $(95 * 1.40625) + 1.40625 = 135(^0)$
4. Pada saat data yang di hasilkan bernilai 127, maka nilai kompas =
 $(127 * 1.40625) + 1.40625 = 180(^0)$
5. Pada saat data yang di hasilkan bernilai 159, maka nilai kompas =
 $(159 * 1.40625) + 1.40625 = 225(^0)$
6. Pada saat data yang di hasilkan bernilai 191, maka nilai kompas =
 $(191 * 1.40625) + 1.40625 = 270(^0)$
7. Pada saat data yang di hasilkan bernilai 223, maka nilai kompas =
 $(223 * 1.40625) + 1.40625 = 315(^0)$

Tabel 4.2. Perbandingan hasil kompas digital dan kompas analog

No	Arah mata angin	Data	Kompas	
		Lebar pulsa	Digital (°)	Analog (°)
1	Timur Laut	31	45	45
2	Timur	63	90	85
3	Tenggara	95	135	125
4	Selatan	127	180	180
5	Barat Daya	159	225	235
6	Barat	191	270	275
7	Barat Laut	223	315	310



Gambar 4.3. Hasil Perbandingan Kompas Digital Dengan Kompas Analog

Mencari presentase nilai kesalahan (% error) dari hasil pengujian terhadap perhitungan yang akan di rumuskan sebagai berikut:

$$error = \left| \frac{digital-analog}{analog} \right| * 100\% \dots\dots\dots(4.2.4-2)$$

1. = $\left| \frac{45-45}{45} \right| * 100\% = 0\%$
2. = $\left| \frac{90-85}{95} \right| * 100\% = 5,26\%$
3. = $\left| \frac{135-125}{125} \right| * 100\% = 8\%$
4. = $\left| \frac{180-180}{180} \right| * 100\% = 0\%$
5. = $\left| \frac{225-235}{235} \right| * 100\% = 4,25\%$
6. = $\left| \frac{270-275}{275} \right| * 100\% = 1,81\%$
7. = $\left| \frac{315-310}{310} \right| * 100\% = 1,61\%$

Tabel 4.3. Presentase nilai kesalahan (% error) dari kompas digital terhadap kompas analog

No	Arah mata angin utara	Data	Kompas		% error
		Lebar pulsa	Digital (°)	Analog (°)	
1	Timur Laut	31	45	45	0
2	Timur	63	90	85	5,25
3	Tenggara	95	135	125	8
4	Selatan	127	180	180	0
5	Barat Daya	159	225	235	4,25
6	Barat	191	270	275	1,81
7	Barat Laut	223	315	310	1,61
Rata-rata					2,98

Berdasarkan tabel 4.3. Hasil pengujian modul kompas CMPS03 memiliki faktor kesalahan rata sebearar 2,98 %.

4.3. Pengujian *Incremental Rotary Encoder*

4.3.1. Peralatan yang digunakan

1. Minimum sistem AVR
2. Modul *Incremental Rotary Encoder*
3. Modul LCD
4. Kabel konektor
5. Downloader
6. DC power supply
7. PC/Laptop sebagai media pemrograman

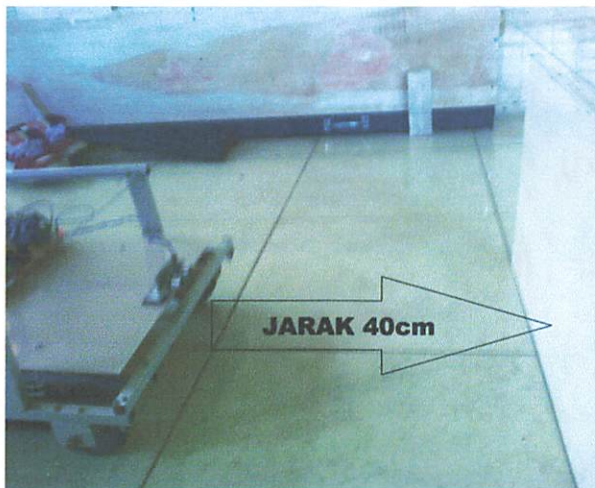
4.3.2. Langkah-langkah pengujian

1. Hidupkan minimum sistem *board* dengan sumber tegangan 5VDC
2. Koneksikan pin Rotary Encoder pada *peripheral I/O* dengan kabel konektor ke pin mikrokontroler (PORTD)
3. Koneksikan pin LCD pada *peripheral I/O* dengan kabel konektor ke pin mikrokontroler (PORTA)
4. Membuat program *Incremental Rotary Encoder* melalui *CodeVision AVR* dan mendownload program tersebut ke mikrontroler

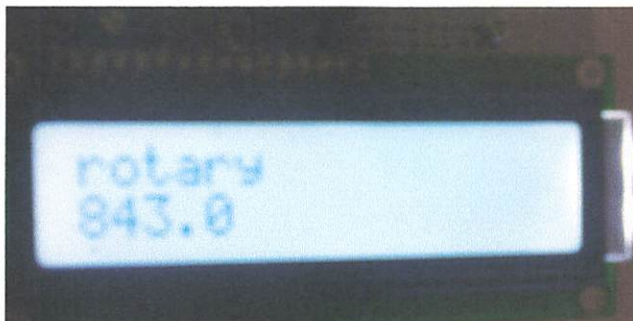
4.3.3. Hasil pengujian

Tabel 4.4. Hasil pengujian *incremental rotary encoder*

No	Jarak (cm)	Keliling (cm)	Lubang piringan rotary	Pengujian
1	40	12,56	266	843
2	80			1702
3	120			2545
4	160			3393
5	200			4242



Gambar 4.4. Proses Pengujian



Gambar 4.5. Hasil Pengujian *Incremental Rotary Encoder*

4.3.4. Analisa pengujian

Dari hasil pengujian di gunakan rumus berikut untuk menentukan jumlah pulsa yang dihasilkan modul sensor *incremental rotary encoder*. Diameter piringan rotary 4cm dan resolusi piringan 360 satu kali putaran.^[4] Persamaan dari keliling piringan rotary sebagai berikut:

$$S = \pi * d \dots\dots\dots(4.3.4-1)$$

$$S = 3,14 * 4cm = 12,56cm$$

$$jumlah\ pulsa = \frac{jarak\ tempuh}{keliling\ piringan\ rotary} * jumlah\ lubang \dots\dots\dots(4.3.4-2)$$

Dari rumus di atas, jarak merupakan perubahan nilai yang di hasilkan melalui modul sensor *incremental rotary encoder*. Perhitungan rumus di atas menghasilkan besaran nilai jumlah pulsa.

1. $= \frac{40cm}{12,56cm} * 266 = 847$
2. $= \frac{80cm}{12,56cm} * 266 = 1694$
3. $= \frac{120cm}{12,56cm} * 266 = 2541$
4. $= \frac{160cm}{12,56cm} * 266 = 3388$
5. $= \frac{200cm}{12,56cm} * 266 = 4235$

Tabel 4.5. Perbandingan Hasil pengujian dan perhitungan Rotary Encoder

No	Jarak (cm)	Roda	Lubang piringan rotary	Hasil resolusi	
				pengujian	perhitungan
1	40	12,56	266	843	847
2	80			1702	1694
3	120			2545	2541
4	160			3393	3383
5	200			4264	4235

Mencari presentase nilai kesalahan (% error) dari hasil pengujian terhadap perhitungan yang akan di rumuskan sebagai berikut:

$$error = \left| \frac{pengujian - perhitungan}{perhitungan} \right| * 100\% \dots\dots\dots(4.3.4-3)$$

1. $= \left| \frac{861-872}{872} \right| * 100\% = 1,26\%$
2. $= \left| \frac{1711-1742}{1742} \right| * 100\% = 1,78\%$
3. $= \left| \frac{2559-2617}{2617} \right| * 100\% = 2,21\%$
4. $= \left| \frac{3414-3488}{3488} \right| * 100\% = 2,12\%$

$$5. = \left| \frac{4264-4363}{4363} \right| * 100\% = 2,26\%$$

Tabel 4.6. Presentase nilai kesalahan (% error) dari hasil pengujian terhadap perhitungan

No	Jarak (cm)	Roda	Lubang piringan rotary	Hasil resolusi		% error
				pengujian	perhitungan	
1	40	12,56	266	843	847	0,47
2	80			1702	1694	0,47
3	120			2545	2541	0,15
4	160			3393	3383	0,29
5	200			4264	4235	0,68
Rata-rata						0,41

Berdasarkan tabel 4.6. Hasil pengujian modul rotary encoder memiliki faktor kesalahan rata-rata sebesar 0,41%.

4.4. Pengujian Ultrasonik Ping

4.4.1. Peralatan yang digunakan

1. Minimum sistem AVR
2. Modul Ultrasonik Ping
3. Modul LCD
4. Kabel konektor
5. Downloader
6. DC power supply
7. PC/Laptop sebagai media pemrograman

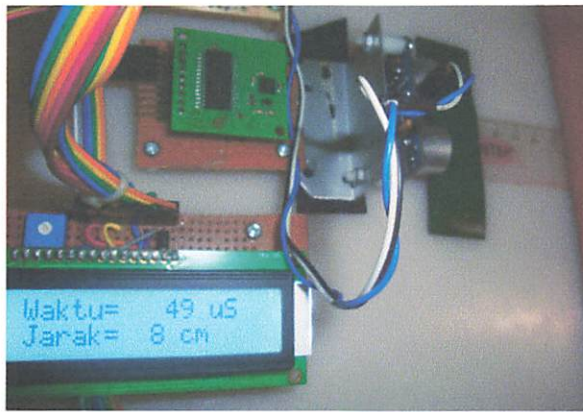
4.4.2. Langkah-langkah pengujian

1. Hidupkan minimum sistem *board* dengan sumber tegangan 5VDC
2. Koneksikan pin Ultrasonik Ping pada *peripheral I/O* dengan kabel konektor ke pin mikrokontroler (PORTD)
3. Koneksikan pin LCD pada *peripheral I/O* dengan kabel konektor ke pin mikrokontroler (PORTA)
4. Membuat program Ultrasonik Ping melalui *CodeVision AVR* dan mendownload program tersebut ke mikrontroler

4.4.3. Hasil pengujian

Tabel 4.7. Perbandingan hasil pengujian dan pengukuran

No	Jarak (cm)	
	pengujian	Alat Ukur
1	4	4
2	6	5,5
3	8	7,5
4	10	9,3
5	12	11,4



Gambar 4.6. Hasil pengujian sensor ultrasonik ping

4.4.4. Analisa pengujian

Mencari presentase nilai kesalahan (% error) dari hasil pengujian terhadap perhitungan yang akan di rumuskan sebagai berikut:

$$error = \left| \frac{pengujian - alat\ ukur}{alat\ ukur} \right| * 100\% \dots\dots\dots(4.4.4-1)$$

1. $= \left| \frac{4cm - 4cm}{4cm} \right| * 100\% = 0$
2. $= \left| \frac{6cm - 5,5cm}{5,5cm} \right| * 100\% = 8,3$
3. $= \left| \frac{8cm - 7,5cm}{7,5cm} \right| * 100\% = 6,25$
4. $= \left| \frac{10cm - 9,3cm}{9,3cm} \right| * 100\% = 7$
5. $= \left| \frac{12cm - 11,4cm}{11,4cm} \right| * 100\% = 5$

Tabel 4.8. Presentase nilai kesalahan (% error) dari hasil pengujian terhadap pengukuran

No	Jarak (cm)		% error
	pengujian	Alat Ukur	
1	4	4	0
2	6	5,5	8,3
3	8	7,5	6,25
4	10	9,3	7
5	12	11,4	5
Rata-rata			5,31

Berdasarkan tabel 4.8. hasil pengujian modul ultrasonik ping memiliki faktor kesalahan rata sebearar 5,31 %.

4.5. Pengujian Penguat Motor

4.5.1. Peralatan yang digunakan

1. AVO meter
2. Penguat motor
3. Motor DC
4. Kabel konektor
5. DC power supply 5VDC

4.5.2. Langkah-langkah pengujian

1. Koneksikan pin penguat motor pada *power supply*
2. Koneksikan motor ke port penguat motor

4.5.3. Hasil pengujian

Tabel 4.9. Hasil pengujian penguat motor

pwm	Input		Output
	P1.1	P1.2	putaran motor
0	0	0	tidak berputar
255	1	0	searah jarum jam
255	0	1	bellawanan arah jarum jam
255	1	1	tidak berputar

Tabel 4.10. Hasil pengujian penguat motor tanpa beban dan motor dengan beban

Motor	Tegangan (V)	Arus (A)
Tanpa beban	13,71	1,27
dengan beban	13,66	3,33



Gambar 4.7. Tegangan masuk pada penguat motor



Gambar 4.8. Hasil tegangan penguat motor tanpa beban



Gambar 4.9. Hasil arus penguat motor tanpa beban



Gambar 4.10. Hasil tegangan penguat motor dengan beban



Gambar 4.11. Hasil arus penguat motor tanpa beban

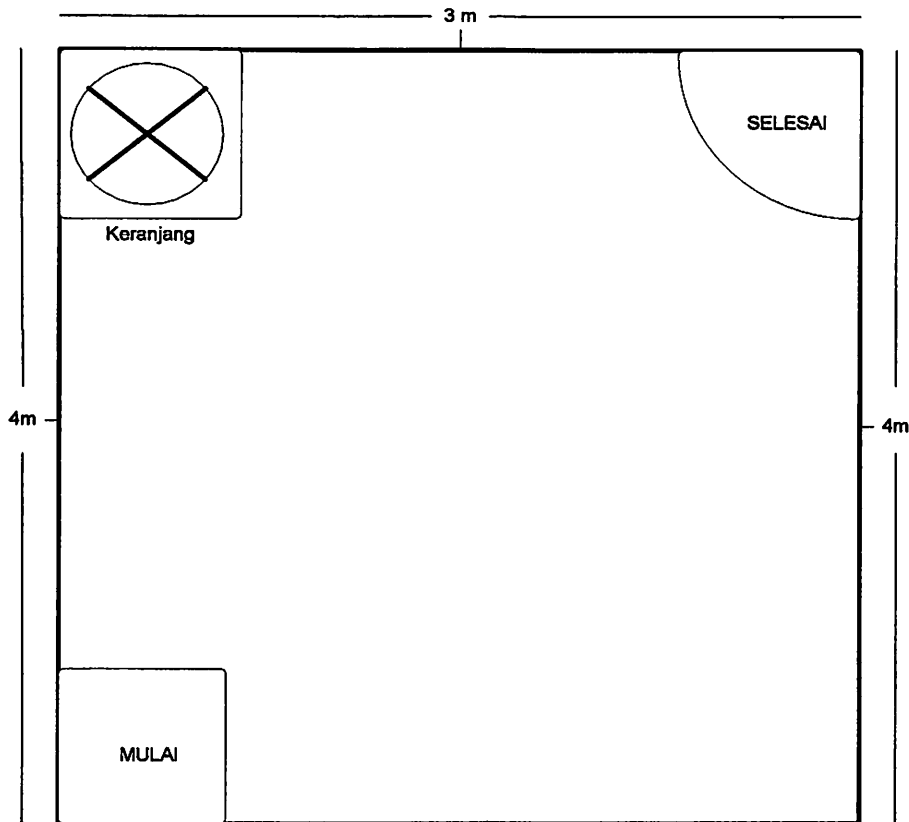
4.5.4. Analisa pengujian

Tabel 4.9 Menunjukkan bahwa penguat motor L298 memiliki kecepatan yang bisa di kurangi. Apabila kecepatan 0, P1.1 *low* dan P1.2 *low* maka motor tidak akan berputar dan apabila kecepatan 255, P1.1 *higt* dan P1.2 *low* maka motor akan bergerak searah jarum jam.

Tabel 4.10 Hasil dari pengukuran tegangan dan arus motor dc di dapat tegangan sebesar 13,71 *Volt* pada saat motor tanpa beban dengan tegangan input 13,89 *Volt*. Hal ini terjadi karena *drop* tegangan pada tahanan jangkar motor. Sedangkan pada saat motor dibebani dengan beban tertentu maka tegangan motor berkurang karena pengaruh dari tahanan jangkar dan daya perlawanan pada beban.

Dan besarnya arus yang mengalir ke motor berubah-ubah tergantung dari beban. Arus yang mengalir pada motor kondisi tanpa beban 1,27 *Ampere*, sedangkan arus yang mengalir pada motor dengan beban semakin meningkat sebesar 3,33 *Ampere*.

4.6. Pengujian sistem



Gambar 4.12. Arena pengujian robot

4.6.1. Langkah-langkah pengujian

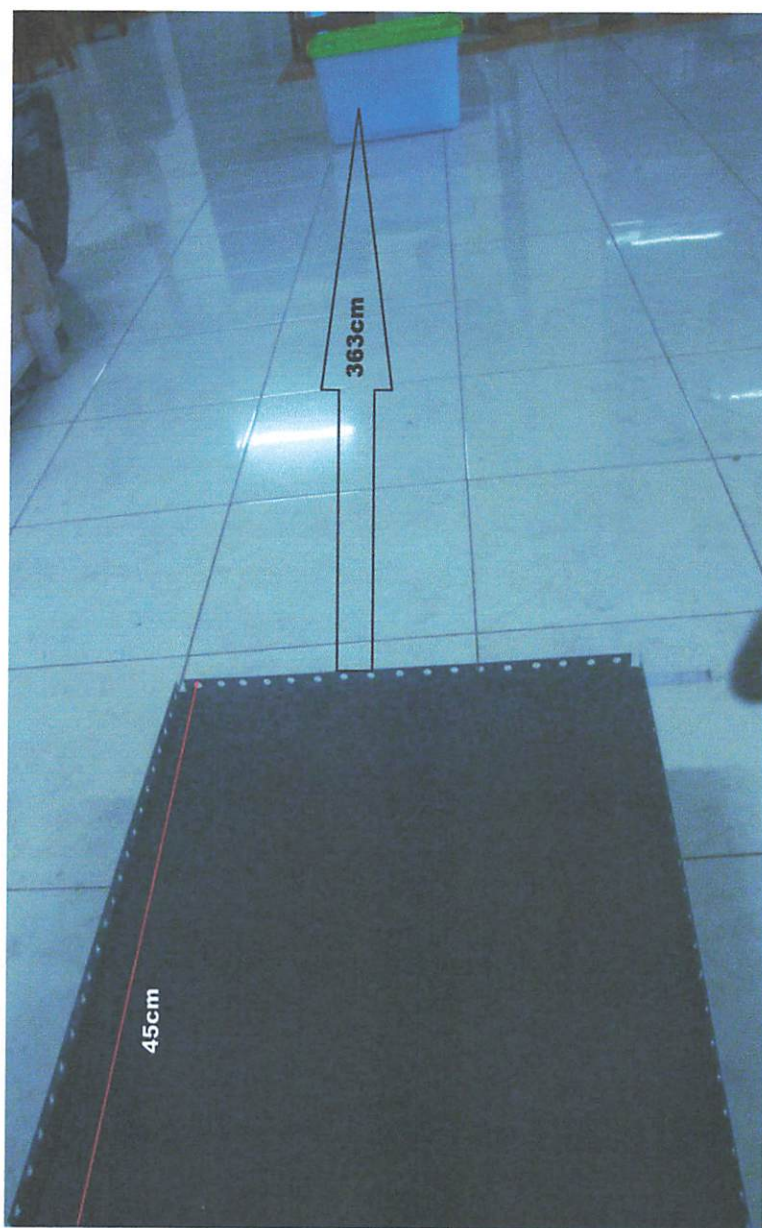
1. Robot pada posisi mulai
2. Tekan tombol atau saklar tekan nomer 1 untuk menggerakkan robot menuju posisi barang barang yang akan di ambil
3. Letakan barang pada robot lalu tekan tombol atau saklar tekan nomer 2 untuk menggerakkan robot menuju posisi meletakkan barang

4.6.2. Hasil pengujian

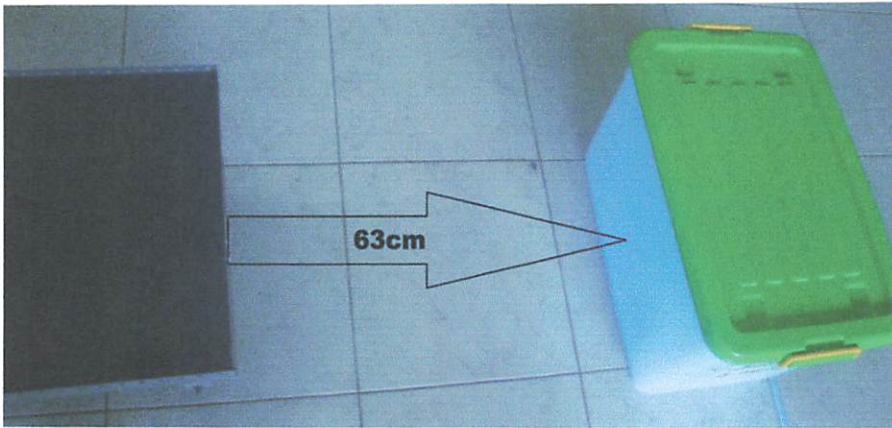
Pengujian sistem dengan jarak arena 400cm, robot 45cm dan keranjang 32 cm.

Tabel 4.11. Hasil pengujian sistem meletakkan barang dengan sudut 120°

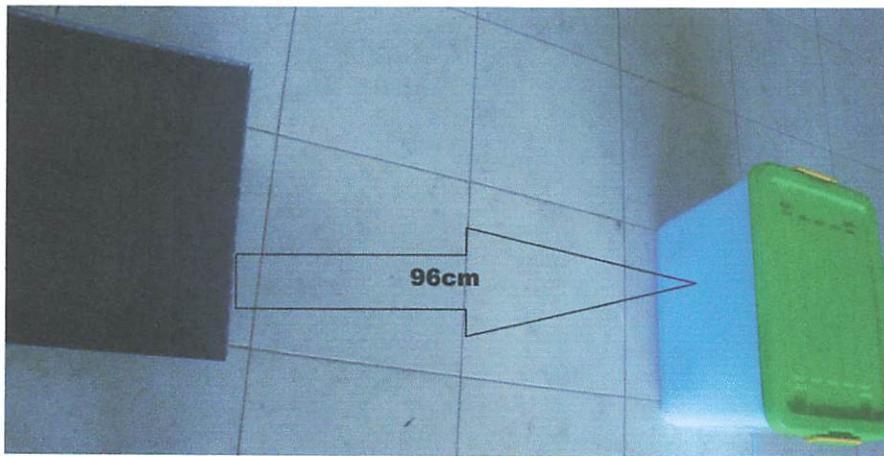
No	PWM	Sudut ($^{\circ}$)	Ping (cm)	Rotary	
				Input	Output
1	255	120	28	4235	7094
2	200	120	40	4235	6840
3	175	120	53	4235	6565
4	150	120	63	4235	6353
5	100	120	96	4235	5654



Gambar 4.12. Proses pengujian sistem



Gambar 4.16. Proses pengujian sistem dengan kecepatan 150 pulsa



Gambar 4.17. Proses pengujian sistem dengan kecepatan 100 pulsa

4.6.3. Analisa pengujian

Mencari presentase nilai kesalahan (% error) dari hasil pengujian sistem yang akan di rumuskan sebagai berikut:

$$error = \left| \frac{input-output}{input} \right| * 100\% \dots\dots\dots(4.6.3-1)$$

$$1. error = \left| \frac{4235-7094}{4235} \right| * 100\% = 67,51\%$$

$$2. error = \left| \frac{4235-6840}{4235} \right| * 100\% = 61,51\%$$

$$3. error = \left| \frac{4235-6565}{4235} \right| * 100\% = 55,01\%$$

$$4. error = \left| \frac{4235-6353}{4235} \right| * 100\% = 50,01\%$$

$$5. error = \left| \frac{4235-5654}{4235} \right| * 100\% = 33,51\%$$

Tabel 4.12. Presentase nilai kesalahan (% error) dari hasil pengujian sistem

No	PWM	Sudut (^o)	Ping (cm)	Rotary		% Error
				Input	Output	
1	255	120	28	4235	7094	67,51
2	200	120	40	4235	6840	61,51
3	175	120	53	4235	6565	55,01
4	150	120	63	4235	6353	50,01
5	100	120	96	4235	5654	33,51

Berdasarkan tabel 4.12. Hasil pengujian sistem mempunyai nilai kesalahan yang tidak sama di keranakan nilai PWM yang di input kan mulai dari 100 pulsa sampai 255 pulsa, maka nilai kesalahan yang di dapat sebesar 67,51% dengan kecepatan 255 pulsa dan 33,51% dengan kecepatan 100 pulsa. Oleh karena itu semakin kecil nilai kecepatan motor maka semakin kecil pula nilai kesahanan. Bedasarkan sistem navigasi robot menggunakan kompas digital dan *incremental rotary encoder* bekerja kurang baik karena nilai kesalahan yang di dapat sensor *inremental rotary encoder* sangat besar. Oleh kerana itu sensor ultrasonik ping sangat membantu untuk mengetahui jarak objek yang di tentukan dan agar robot bisa bernavigasi dengan baik.

BAB V

PENUTUP

5.1. Kesimpulan

Setelah melakukan pengujian dan analisa, maka dapat disimpulkan sebagai berikut:

1. Eksekusi penggunaan sensor kompas digital dan *incremental rotary encoder* tidak bisa dilakukan bersama. Hal ini karena keduanya menggunakan perintah berbeda pada mikrokontroler.
2. Dari pengukuran dan pengujian sensor kompas digital memiliki nilai kesalahan rata-rata sebesar 2,98 %, *sensor incremental rotary encoder* memiliki nilai kesalahan rata-rata sebesar 0,41 % dan sensor ultrasonik ping memiliki nilai kesalahan rata-rata sebesar 5,31 %.
3. Dari pengujian sistem hasil yang di dapat mempunyai nilai kesalahan sebesar 67,51% dengan kecepatan 255 pulsa dan sebesar 33,51% dengan kecepatan 100 pulsa. Oleh karena itu semakin kecil nilai kecepatan motor maka semakin kecil nilai kesalahan.
4. Kompas digital dan *incremental rotary encoder* sebagai sistem navigasi robot bekerja secara akurat dengan bantuan sensor ultrasonik ping.

5.2. Saran

Adapun saran penulis untuk perkembangan penelitian selanjutnya:

1. Penggunaan kompas digital yang masih kurang maksimal dan perlu menambahkan sensor kamera atau GPS agar navigasi robot lebih akurat.
2. Robot yang sebelumnya masih menggunakan saklar tekan kurang maksimal dan perlu menambahkan keypad atau di kontrol menggunakan android untuk robot manual agar robot tidak perlu merubah program lagi.

DAFTAR PUSTAKA

- [1] Budiharto, Widodo (2010), *Robotika teori+implementasi*, Andi Yogyakarta
- [2] Hadi, Mokh sholihul (2008), *Mengenal mikrokontroler atmega16*, Malang
- [3] Husein, Setiawardhana Ali (2007), *PING Ultrasonic Sensor Overview*. Surabaya
- [4] Pranadipta, Bisma. 2011. *Perancangan dan Pembuatan Robot dengan Pengendali Motor dc Menggunakan Relative/Incremental Rotary Encoder Berbasis ATmega16*. Malang
- [5] Prayascitaram, Surya B. 2011. *Rancang Bangun Modul Pelatihan Robotika ITN Malang Berbasis Minimum Sistem ATmega16*. Malang
- [6] Putra, Agus Anggasana. 2012. *Rancang Bangun Modul Praktikum Mikrokontroler ATmega 8538/16/32 dan ATmega 128L di Laboratorium Elektronika Digital ITN Malang Berbasis Minimum Sistem AVR*. Malang
- [7] Winoto, Ardi (2010), *Mikrokontroler AVR ATmega8/32/16/8535 dan Pemrograman dengan Bahasa C pada WinAVR*, Informatika Bandung
- [8] compass CMPS03, <http://id.wikipedia.org/wiki/compassCMPS03> Tanggal akses: 7 Oktober 2012
- [9] L298, <http://www.alldatasheet.com//datasheet-pdf> Tanggal akses: 21 juli 2012
- [10] rotary encoder, <http://id.wikipedia.org/wiki/rotaryencoder> Tanggal akses: 19 Juli 2012
- [11] saklar-tekan, <http://en.wikipedia.org/saklar-tekan> Tanggal akses: 17 Januari 2013



File: tes.c, Date: 10/04/2013, Time: 1:18:33

/*

*/

This program was produced by the
CodeWizardAVR V2.04.4a Advanced
Automatic Program Generator
© Copyright 1998-2009 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfootech.com

Project : HUDDIN
Version :
Date : 21/01/2013
Author : NeVaDa
Company :
Comments:

Chip type : ATmega16
Program type : Application
AVR Core Clock frequency: 12,000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 256

*/

```
#include <mega16.h>
#include <delay.h>
#include <stdlib.h>
#include <stdio.h>

#define rotary_kanan_rev      PIND.4
#define rotary_kiri_rev      PIND.5
#define pulsa PORTD.6
#define deteksi PIND.6
#define ARAH DDRD.6
#define OUT 1
#define INP 0
#define dir_kal PORTB.0      //arah mundur
#define pwm_ka PORTB.1      //kecepatan motor kanan
#define dir_ka2 PORTB.2      //arah maju
#define dir_kil PORTB.3      //arah mundur
#define pwm_ki PORTB.4      //kecepatan motor kiri
#define dir_ki2 PORTB.5      //arah maju
#define T1 PINC.0           //tombol 1
#define T2 PINC.1           //tombol 2
#define T3 PINC.2           //tombol 3
#define T4 PINC.3           //tombol 4
#define T5 PINC.4           //tombol 5
char digit [8];

// I2C Bus functions
#asm
.equ __i2c_port=0x12 ;PORTD
.equ __sda_bit=1
.equ __scl_bit=0
#endasm
#include <i2c.h>

// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x1B ;PORTA
#endasm
#include <lcd.h>

bit putaran_kanan;
int rotary_kanan=0;
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
if (putaran_kanan--){
if (rotary_kanan_rev==0){rotary_kanan--;}
if (rotary_kanan_rev==1){rotary_kanan++;}
if (putaran_kanan==0){
if (rotary_kanan_rev==0){rotary_kanan++;}
if (rotary_kanan_rev==1){rotary_kanan--;}
}
}

bit putaran_kiri;
int rotary_kiri=0;
```

File: tes.c, Date: 10/04/2013, Time: 1:18:33

```
// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
// Place your code here
if (putaran_kiri==1){
if (rotary_kiri_rev--0) {rotary_kiri--;}
if (rotary_kiri_rev==1 ) {rotary_kiri++;}
if (putaran_kiri==0){
if (rotary_kiri_rev==0){rotary_kiri++;}
if (rotary_kiri_rev==1){rotary_kiri--;}
}

unsigned char lpwm, rpwm, x;
// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Place your code here
x++;
if(x<=lpwm)pwm_ki=1; else pwm_ki=0;
if(x<=rpwm)pwm_ka=1; else pwm_ka=0;
TCNT0=255;
}

unsigned char data;
int Kompas;
unsigned char nilai[6];
void baca_kompas()
{
i2c_start();
i2c_write(0xC0);
i2c_write(0x01);
i2c_start();
i2c_write(0x61);
data=i2c_read(0);
i2c_stop();
}

void data_kompas()
{
baca_kompas();
Kompas=((data*1.40625)+1.40625);

nilai[0]=(Kompas/100)|0x30;
nilai[1]=((Kompas%100)/10)|0x30;
nilai[2]=((Kompas%100)%10)|0x30;

nilai[3]=(data/100)|0x30;
nilai[4]=((data%100)/10)|0x30;
nilai[5]=((data%100)%10)|0x30;

lcd_gotoxy(0,0);
lcd_putsf("Data CMPS03");

lcd_gotoxy(0,1);
lcd_putsf("Kompas =");

lcd_gotoxy(12,0);lcd_putchar(nilai[3]);
lcd_gotoxy(13,0);lcd_putchar(nilai[4]);
lcd_gotoxy(14,0);lcd_putchar(nilai[5]);

lcd_gotoxy(9,1);lcd_putchar(nilai[0]);
lcd_gotoxy(10,1);lcd_putchar(nilai[1]);
lcd_gotoxy(11,1);lcd_putchar(nilai[2]);
lcd_gotoxy(12,1);lcd_putchar(0xDF);
}

void objek_pertama ()
{
data_kompas();
if (Kompas>120&&Kompas<=145) {rpwm=150;lpwm=150;dir_ka1=0;dir_ka2=1;dir_ki1=1;dir_ki2=0;}else;
if (Kompas>145&&Kompas<=170) {rpwm=175;lpwm=175;dir_ka1=0;dir_ka2=1;dir_ki1=1;dir_ki2=0;}else;
if (Kompas>170&&Kompas<=195) {rpwm=200;lpwm=200;dir_ka1=0;dir_ka2=1;dir_ki1=1;dir_ki2=0;}else;
if (Kompas>195&&Kompas<=220) {rpwm=225;lpwm=225;dir_ka1=0;dir_ka2=1;dir_ki1=1;dir_ki2=0;}else;
if (Kompas>220) {rpwm=255;lpwm=255;dir_ka1=0;dir_ka2=1;dir_ki1=1;dir_ki2=0;}else;
}
```

File: tes.c, Date: 10/04/2013, Time: 1:18:33

```
    if (kompas<120&&kompas>=95) {rpwm=150;lpwm=150;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=
1;}else;
    if (kompas<95&&kompas>=70) {rpwm=175;lpwm=175;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
    if (kompas<70&&kompas>=45) {rpwm=200;lpwm=200;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
    if (kompas<45&&kompas>=20) {rpwm=225;lpwm=225;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
    if (kompas<20) {rpwm=255;lpwm=255;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1;}else;

    if (kompas==120) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
    // if (kompas==120) {rpwm=0;lpwm=0;}
}

void ambil_keranjang ()
{
data_kompas();
    if (kompas>125&&kompas<=145) {rpwm=150;lpwm=150;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>150&&kompas<=170) {rpwm=175;lpwm=175;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>175&&kompas<=195) {rpwm=200;lpwm=200;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>200&&kompas<=220) {rpwm=225;lpwm=225;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>225) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2=0;}else;

    if (kompas<125&&kompas>=95) {rpwm=150;lpwm=150;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=
1;}else;
    if (kompas<100&&kompas>=70) {rpwm=175;lpwm=175;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=
1;}else;
    if (kompas<75&&kompas>=45) {rpwm=200;lpwm=200;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
    if (kompas<50&&kompas>=20) {rpwm=225;lpwm=225;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
    if (kompas<25) {rpwm=255;lpwm=255;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1;}else;

    //if (kompas==125) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
    if (kompas==125) {rpwm=0;lpwm=0;}
}

void belok ()
{
data_kompas();
    if (kompas>157&&kompas<=182) {rpwm=150;lpwm=150;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>182&&kompas<=207) {rpwm=175;lpwm=175;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>207&&kompas<=232) {rpwm=200;lpwm=200;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>232&&kompas<=257) {rpwm=225;lpwm=225;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>257) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2=0;}else;

    if (kompas<157&&kompas>=132) {rpwm=150;lpwm=150;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<132&&kompas>=107) {rpwm=175;lpwm=175;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<107&&kompas>=82) {rpwm=200;lpwm=200;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=
1;}else;
    if (kompas<82&&kompas>=57) {rpwm=225;lpwm=225;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
    if (kompas<57) {rpwm=255;lpwm=255;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1;}else;

    //if (kompas==157) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
    #if (kompas==157) {rpwm=0;lpwm=0;}
}

void objek_kedua ()
{
data_kompas();
    if (kompas>205&&kompas<=230) {rpwm=150;lpwm=150;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>230&&kompas<=255) {rpwm=175;lpwm=175;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
```

File: tes.c, Date: 10/04/2013, Time: 1:18:33

```
=0; }else;
    if (kompas>255&&kompas<=280) {rpwm=200;lpwm=200;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0; }else;
    if (kompas>280&&kompas<=305) {rpwm=225;lpwm=225;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0; }else;
    if (kompas>305) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2=0;}else;
    if (kompas<205&&kompas>=180) {rpwm=150;lpwm=150;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<180&&kompas>=155) {rpwm=175;lpwm=175;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<155&&kompas>=130) {rpwm=200;lpwm=200;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<130&&kompas>=105) {rpwm=225;lpwm=225;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<105) {rpwm=255;lpwm=255;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1;}else;

    //if (kompas==205) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
    if (kompas==205) {rpwm=0;lpwm=0;}
}

void taruh ()
{
data_kompas();
    if (kompas>215&&kompas<=240) {rpwm=150;lpwm=150;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>240&&kompas<=265) {rpwm=175;lpwm=175;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>265&&kompas<=290) {rpwm=200;lpwm=200;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>290&&kompas<=315) {rpwm=225;lpwm=225;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>315) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2=0;}else;

    if (kompas<215&&kompas>=190) {rpwm=150;lpwm=150;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<190&&kompas>=165) {rpwm=175;lpwm=175;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<165&&kompas>=140) {rpwm=200;lpwm=200;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<140&&kompas>=115) {rpwm=225;lpwm=225;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<115) {rpwm=255;lpwm=255;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1;}else;

    //if (kompas==215) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
    if (kompas==215) {rpwm=0;lpwm=0;}
}

void ambil ()
{
data_kompas();
    if (kompas>130&&kompas<=155) {rpwm=150;lpwm=150;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>155&&kompas<=170) {rpwm=175;lpwm=175;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>170&&kompas<=195) {rpwm=200;lpwm=200;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>195&&kompas<=220) {rpwm=225;lpwm=225;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>220) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2=0;}else;

    if (kompas<130&&kompas>=105) {rpwm=150;lpwm=150;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<105&&kompas>=80) {rpwm=175;lpwm=175;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=
1;}else;
    if (kompas<80&&kompas>=55) {rpwm=200;lpwm=200;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
    if (kompas<55&&kompas>=30) {rpwm=225;lpwm=225;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
    if (kompas<30) {rpwm=255;lpwm=255;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1;}else;

    //if (kompas==130) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
    if (kompas==130) {rpwm=0;lpwm=0;}
}
```

File: tes.c, Date: 10/04/2013, Time: 1:18:33

```
void naruh_kerangjang ()
{
    data_kompas();
    if (kompas>210&&kompas<=235) {rpwm=150;lpwm=150;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>235&&kompas<=260) {rpwm=175;lpwm=175;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>260&&kompas<=285) {rpwm=200;lpwm=200;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>285&&kompas<=310) {rpwm=225;lpwm=225;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>310) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2=0;}else;

    if (kompas<210&&kompas>=185) {rpwm=150;lpwm=150;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<185&&kompas>=160) {rpwm=175;lpwm=175;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<160&&kompas>=135) {rpwm=200;lpwm=200;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<135&&kompas>=110) {rpwm=225;lpwm=225;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<110) {rpwm=255;lpwm=255;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1;}else;

    //if (kompas==210) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
    if (kompas==210) {rpwm=0;lpwm=0;}
}

void barat ()
{
    data_kompas();
    if (kompas>270&&kompas<=290) {rpwm=125;lpwm=150;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>290&&kompas<=310) {rpwm=150;lpwm=175;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>310&&kompas<=330) {rpwm=175;lpwm=200;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>330&&kompas<=350) {rpwm=200;lpwm=225;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>350) {rpwm=225;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2=0;}else;

    if (kompas<270&&kompas>=250) {rpwm=125;lpwm=150;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<250&&kompas>=230) {rpwm=150;lpwm=175;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<230&&kompas>=210) {rpwm=175;lpwm=200;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<210&&kompas>=190) {rpwm=200;lpwm=225;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<190) {rpwm=225;lpwm=255;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1;}else;

    //if (kompas==270) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
    if (kompas==270) {rpwm=0;lpwm=0;}
}

void selatan ()
{
    data_kompas();
    if (kompas>180&&kompas<=200) {rpwm=125;lpwm=150;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>200&&kompas<=220) {rpwm=150;lpwm=175;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>220&&kompas<=240) {rpwm=175;lpwm=200;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>240&&kompas<=260) {rpwm=200;lpwm=225;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
    if (kompas>260) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2=0;}else;

    if (kompas<180&&kompas>=160) {rpwm=125;lpwm=150;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<160&&kompas>=140) {rpwm=150;lpwm=175;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<140&&kompas>=120) {rpwm=175;lpwm=200;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<120&&kompas>=100) {rpwm=200;lpwm=225;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2
=1;}else;
    if (kompas<100) {rpwm=225;lpwm=255;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1;}else;
}
```

File: tes.c, Date: 10/04/2013, Time: 1:18:33

```
//if (kompas==180) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
if (kompas==180) {rpwm=0;lpwm=0;}
}

void timur ()
{
data_kompas();
if (kompas>90&&kompas<=110) {rpwm=125;lpwm=150;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2=
0;}else;
if (kompas>110&&kompas<=130) {rpwm=150;lpwm=175;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
if (kompas>130&&kompas<=150) {rpwm=175;lpwm=200;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
if (kompas>150&&kompas<=170) {rpwm=200;lpwm=225;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2
=0;}else;
if (kompas>170) {rpwm=225;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=1;dir_ki2=0;}else;

if (kompas<90&&kompas>=70) {rpwm=125;lpwm=150;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
if (kompas<70&&kompas>=50) {rpwm=150;lpwm=175;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
if (kompas<50&&kompas>=30) {rpwm=175;lpwm=200;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
if (kompas<30&&kompas>=10) {rpwm=200;lpwm=225;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1
;}else;
if (kompas<10) {rpwm=225;lpwm=255;dir_kal=1;dir_ka2=0;dir_kil=0;dir_ki2=1;}else;

//if (kompas==90) {rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
if (kompas==90) {rpwm=0;lpwm=0;}
}

void baca_rotary ()
{
ftoa(rotary_kanan,1,digit);
lcd_gotoxy(0,0);
lcd_putsf("kanan");
lcd_gotoxy(0,1);
lcd_puts(digit);
ftoa(rotary_kiri,1,digit);
lcd_gotoxy(9,0);
lcd_putsf("kiri");
lcd_gotoxy(9,1);
lcd_puts(digit);
}

void putaran_mundur()
{
ftoa(rotary_kanan,1,digit);
lcd_gotoxy(0,0);
lcd_putsf("rotary kanan");
lcd_gotoxy(0,1);
lcd_puts(digit);
}

void putaran_maju()
{
ftoa(rotary_kiri,1,digit);
lcd_gotoxy(0,0);
lcd_putsf("rotary kiri");
lcd_gotoxy(0,1);
lcd_puts(digit);
}

void rotary()
{
baca_rotary();
if (rotary_kanan==rotary_kiri) {rpwm=100;lpwm=100;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
else;
if (rotary_kanan<rotary_kiri) {rpwm=100;lpwm=50;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
else;
if (rotary_kanan>rotary_kiri) {rpwm=50;lpwm=100;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
}

unsigned int count;
unsigned long jarak,waktu;
unsigned char baris1[16];
unsigned char baris2[16];
```


File: tes.c, Date: 10/04/2013, Time: 1:18:33

```
void baca_ping ()
{
    count=0;
    ARAH=OUT;
    pulsa=1;
    delay_us(5);
    pulsa=0;
    // port as input
    ARAH=INP;
    // with pull-up
    pulsa=1;
    while (deteksi==0) {};
    while (deteksi==1)
    {
        count++;
    }
    waktu=(count*0.090422453703703703703703703704); /*waktu 1 kali eksekusi
perintah count++ = 0.090422453703703703703703703704 mikro second atau (1/frekuensi
kristal)*/
    jarak=((waktu*0.3495)/2);

    sprintf(baris1,"Waktu=%5d uS ",waktu);
    sprintf(baris2,"Jarak=%3d cm ",jarak);
    lcd_gotoxy(0,0);
    lcd_puts(baris1);
    lcd_gotoxy(0,1);
    lcd_puts(baris2);
    delay_ms(200);
    lcd_clear();
}

void tes_ping()
{
    baca_ping();
    if (jarak<=5) {rpwm=0;lpwm=0;dir_kal=0;dir_ka2=0;dir_kil=0;dir_ki2=0;}else;
    if (jarak>=5) {rpwm=75;lpwm=75;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}else;
    if (jarak>=15) {rpwm=75;lpwm=75;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}else;
    if (jarak>=25) {rpwm=100;lpwm=100;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;}
}

void reset ()
{
    rpwm=0;lpwm=0;
    rotary_kanan=0;rotary_kiri=0; lcd_clear();
}

void stop()
{
    rpwm=150;lpwm=150;dir_kal=1;dir_ka2=0;dir_kil=1;dir_ki2=0;
    delay_ms(100);
}

void maju_pertama()
{
    rpwm=255;lpwm=255;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;
}

void maju_kedua()
{
    rpwm=200;lpwm=200;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;
}

void maju_ketiga()
{
    rpwm=175;lpwm=175;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;
}

void maju_keempat()
{
    rpwm=150;lpwm=150;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;
}

void maju_kelima()
{
    rpwm=100;lpwm=100;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;
}

void maju()
{
    rpwm=100;lpwm=100;dir_kal=0;dir_ka2=1;dir_kil=0;dir_ki2=1;
```

```
}

void mundur()
{
  rpwm=100,lpwm=100;dir_kal=1;dir_ka2=0;dir_kil=1;dir_ki2=0;
}

void pertama()
{
  reset();
  for(;;){ambil_keranjang();if(kompas==125)break;}
  reset();delay_ms(10);
  for(;;){maju();putaran_maju();if(rotary_kiri>=1694)break;}
  stop();reset();delay_ms(10);
  for(;;){objek_pertama();if(kompas==120)break;}
  reset();delay_ms(10);
  for(;;){maju();putaran_maju();if(rotary_kiri>=4235)break;}
  stop();reset();delay_ms(10);
  for(;;){objek_pertama();if(kompas==120)break;}
  reset();delay_ms(10);
  for(;;){tes_ping();if(jarak<=5)break;}
  reset();delay_ms(10);
}

void kedua()
{
  reset();
  for(;;){taruh();if(kompas==215)break;}
  reset();delay_ms(10);
  for(;;){maju();putaran_maju();if(rotary_kiri>=4235)break;}
  stop();reset();delay_ms(10);
  for(;;){objek_kedua();if(kompas==205)break;}
  reset();delay_ms(10);
  for(;;){tes_ping();if(jarak<=5)break;}
  reset();delay_ms(10);
}

void ketiga()
{
  reset();
  for(;;){barat();if(kompas==270)break;}
  reset();delay_ms(100);
  for(;;){selatan();if(kompas==180)break;}
  reset();delay_ms(100);
  for(;;){timur();if(kompas==90)break;}
  reset();delay_ms(100);
  for(;;){ambil_keranjang();if(kompas==125)break;}
  reset();delay_ms(10);
  for(;;){maju();putaran_maju();if(rotary_kiri>=1694)break;}
  stop();reset();delay_ms(10);
  for(;;){objek_pertama();if(kompas==120)break;}
  reset();delay_ms(10);
  for(;;){maju();putaran_maju();if(rotary_kiri>=3383)break;}
  stop();reset();delay_ms(10);
  for(;;){objek_pertama();if(kompas==120)break;}
  reset();delay_ms(10);
  for(;;){tes_ping();if(jarak<=5)break;}
  reset();delay_ms(10);
  for(;;){mundur();putaran_mundur();if(rotary_kanan>=211)break;}
  stop();reset();delay_ms(10);
  for(;;){belok();if(kompas==157)break;}
  stop();reset();delay_ms(10);
  for(;;){maju();putaran_maju();if(rotary_kiri>=2541)break;}
  stop();reset();delay_ms(10);
  for(;;){objek_kedua();if(kompas==205)break;}
  stop();reset();delay_ms(10);
  for(;;){maju();putaran_maju();if(rotary_kiri>=1694)break;}
  stop();reset();delay_ms(10);
  for(;;){objek_kedua();if(kompas==205)break;}
  stop();reset();delay_ms(10);
  for(;;){tes_ping();if(jarak<=5)break;}
  reset();delay_ms(10);
}

void keempat()
{
  reset();
  for(;;){ambil_keranjang();if(kompas==125)break;}
  reset();delay_ms(10);
}
```

File: tes.c, Date: 10/04/2013, Time: 1:18:33

```
for(;;){maju();putaran_maju();if(rotary_kiri>=1694)break;}
stop();reset();delay_ms(10);
for(;;){ambil_keranjang();if(kompas==125)break;}
reset();delay_ms(10);
for(;;){maju();putaran_maju();if(rotary_kiri>=2541)break;}
stop();reset();delay_ms(10);
for(;;){ambil();if(kompas==130)break;}
reset();delay_ms(10);
for(;;){tes_ping();if(jarak<=5)break;}
reset();delay_ms(10);
}

void kelima()
{
reset();
for(;;){taruh();if(kompas==215)break;}
reset();delay_ms(10);
for(;;){maju();putaran_maju();if(rotary_kiri>=4235)break;}
stop();reset();delay_ms(10);
for(;;){objek_kedua();if(kompas==210)break;}
reset();delay_ms(10);
for(;;){tes_ping();if(jarak<=5)break;}
reset();delay_ms(10);
}

void pengujian_pertama()
{
reset();
for(;;){ambil_keranjang();if(kompas==125)break;}
reset();delay_ms(10);
for(;;){maju_pertama();putaran_maju();if(rotary_kiri>=4235)break;}
stop();lpwm=0;rpwm=0;
for(;;){objek_pertama();if(kompas==120)break;}
reset();delay_ms(10);
}

void pengujian_kedua()
{
reset();
for(;;){ambil_keranjang();if(kompas==125)break;}
reset();delay_ms(10);
for(;;){maju_kedua();putaran_maju();if(rotary_kiri>=4235)break;}
stop();lpwm=0;rpwm=0;
for(;;){objek_pertama();if(kompas==120)break;}
reset();delay_ms(10);
}

void pengujian_ketiga()
{
reset();
for(;;){ambil_keranjang();if(kompas==125)break;}
reset();delay_ms(10);
for(;;){maju_ketiga();putaran_maju();if(rotary_kiri>=4235)break;}
stop();lpwm=0;rpwm=0;
for(;;){objek_pertama();if(kompas==120)break;}
reset();delay_ms(10);
}

void pengujian_keempat()
{
reset();
for(;;){ambil_keranjang();if(kompas==125)break;}
reset();delay_ms(10);
for(;;){maju_keempat();putaran_maju();if(rotary_kiri>=4235)break;}
stop();lpwm=0;rpwm=0;
for(;;){objek_pertama();if(kompas==120)break;}
reset();delay_ms(10);
}

void pengujian_kelima()
{
reset();
for(;;){ambil_keranjang();if(kompas==125)break;}
reset();delay_ms(10);
for(;;){maju_kelima();putaran_maju();if(rotary_kiri>=4235)break;}
stop();lpwm=0;rpwm=0;
for(;;){objek_pertama();if(kompas==120)break;}
reset();delay_ms(10);
}
```

File: tes.c, Date: 10/04/2013, Time: 1:18:33

```
void sudut_barat()
{
  reset();
  for(;;)barat();
}

void sudut_selatan()
{
  reset();
  for(;;)selatan();
}

void sudut_timur()
{
  reset();
  for(;;)timur();
}

void pengujian_kompas()
{
  reset();
  for(;;)data_kompas();
}

void pengujian_rotary()
{
  reset();
  for(;;)putaran_maju();
}

void pengujian_ping()
{
  for(;;)baca_ping();
}

void tombol ()
{
  for ( ;; ){if (T1==0){delay_ms (20);pengujian_pertama();}else;
             if (T2==0){delay_ms (20);pengujian_kedua();}else;
             if (T3==0){delay_ms (20);pengujian_ketiga();}else;
             if (T4==0){delay_ms (20);pengujian_keempat();}else;
             if (T5--0){delay_ms (20);pengujian_kelima();}
}

// Declare your global variables here

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

  // Input/Output Ports initialization
  // Port A initialization
  // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
  // State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
  PORTA=0x00;
  DDRA=0x00;

  // Port B initialization
  // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
  // State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
  PORTB=0x00;
  DDRB=0xFF;

  // Port C initialization
  // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
  // State7=P State6=P State5=P State4=P State3=P State2=P State1=P State0=P
  PORTC=0xFF;
  DDRC=0x00;

  // Port D initialization
  // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
  // State7=P State6=P State5=P State4=P State3=P State2=P State1=P State0=P
  PORTD=0xFF;
  DDRD=0x00;

  // Timer/Counter 0 initialization
  // Clock source: System Clock
```

File: tes.c, Date: 10/04/2013, Time: 1:18:33

```
// Clock value: 11,719 kHz
// Mode: Normal top=FFh
// OCO output: Disconnected
TCCR0=0x05;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=FFFFh
// OClA output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Rising Edge
// INT1: On
// INT1 Mode: Rising Edge
// INT2: Off
GICR|=0xC0;
MCUCR=0x0F;
MCUCSR=0x00;
GIFR=0xC0;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x01;

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

// I2C Bus initialization
i2c_init();

// LCD module initialization
lcd_init(16);

// Global enable interrupts
#ifdef sei
asm("sei")
#endif

while (1)
{
    // Place your code here
    tombol();
};
}
```



PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : HUDDIN
 NIM : 0812217
 Semester : 8 (DELAPAN)
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-1
 Konsentrasi : ~~TEKNIK ELEKTRONIKA~~
~~TEKNIK ENERGI LISTRIK~~
~~TEKNIK KOMPUTER DAN INFORMATIKA~~
~~TEKNIK KOMPUTER~~
~~TEKNIK TELEKOMUNIKASI~~
 Alamat : Jl. KH. YUSUF BLOK A. / 22 PERUM PUSKOP AD

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat *SKRIPSI Tingkat Sarjana*. Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

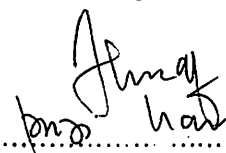
Adapun persyaratan-persyaratan pengambilan *SKRIPSI* adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.


Telah diteliti kebenaran data tersebut diatas

Recording Teknik Elektro


 (.....)

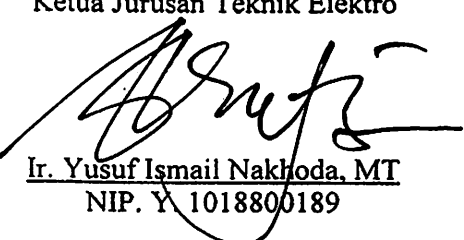
Malang, 27-MARET-2012

Pemohon


 (..... HUDDIN))

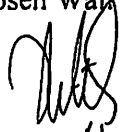
Disetujui

Ketua Jurusan Teknik Elektro


 Ir. Yusuf Ismail Nakhoda, MT
 NIP. Y 1018800189

Mengetahui

Dosen Wali


 (..... M. Ibrahim Ismail, ST MT))

Catatan :

Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

1. IR 1255 / = 3.00
2. / 138
3. praktikum? lengkap



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

INSTITUT TEKNOLOGI NASIONAL MALANG
KAMPUS I (PERSERO) MALANG
KAMPUS II (PERUSAHAAN) MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Surat : ITN-338/EL-FTI/2012
Piranti : -
Materi : BIMBINGAN SKRIPSI

Untuk : Yth. Bapak/Ibu **M. Ibrahim Ashari, ST, MT**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **HUDDIN**
Nim : **0812217**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Elektronika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Ganjil Tahun Akademik 2012 -2013 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Ketua Program Studi Teknik Elektro S-1

H. Yusuf Ismail Nakhoda, MT

NIP. Y. 1918800189



PERKUMPULAN PENGELOLA PENDIDIKAN UNIVERSITAS DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

UNIVERSITAS (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Surat : ITN-338/EL-FTI/2012
No : -
Judul : BIMBINGAN SKRIPSI

Untuk : Yth. Bapak/Ibu **Sotyohadi, ST**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **HUDDIN**
Nim : **0812217**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Elektronika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Ganjil Tahun Akademik 2012 -2013 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Ketua Program Studi Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT

NIP. Y. 1018800189



PROGRAM STUDI TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Bapak/Ibu **M. Ibrahim Ashari, ST, MT**
Dosen Teknik Elektro S-1
ITN Malang

Yang bertanda tangan dibawah

Nama : **HUDDIN**
Nim : **0812217**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Elektronika**

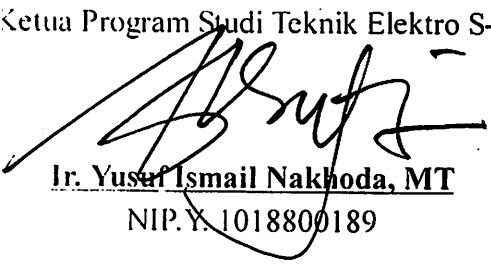
Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Pembimbing untuk penyusunan Skripsi dengan judul :

**"RANCANG BANGUN ROBOT BERODA PEMINDAH KERANJANG
MENGUNAKAN KOMPAS DIGITAL DAN INCREMENTAL ROTARY
ENCODER BERBASIS ATMEGA 16"**

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih

Mengetahui

Ketua Program Studi Teknik Elektro S-1


Ir. Yusuf Ismail Nakhoda, MT

NIP. Y. 1018800189

Hormat Kami



HUDDIN

NIM. 0812217



PROGRAM STUDI TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **HUDDIN**
Nim : **0812217**
Semester : **VIII (Delapan)**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Elektronika**

Dengan ini menyatakan bersedia/~~tidak bersedia~~*) Membimbing skripsi dari mahasiswa tersebut. dengan judul :

" RANCANG BANGUN ROBOT BERODA PEMINDAH KERANJANG MENGGUNA KOMPAS DIGITAL DAN INCREMENTAL ROTARY ENCODER BERBASIS ATMEC 16"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358

*) Coret yang tidak perlu



PROGRAM STUDI TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

Lampiran : 1 (satu) berkas
Pembimbing Skripsi

Kepada : Yth. Bapak/Ibu **Sotyohadi, ST**
Dosen Teknik Elektro S-1
ITN Malang

Yang bertanda tangan dibawah

Nama : **HUDDIN**
Nim : **0812217**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Elektronika**

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Pembimbing untuk penyusunan Skripsi dengan judul :

**"RANCANG BANGUN ROBOT BERODA PEMINDAH KERANJANG
MENGUNAKAN KOMPAS DIGITAL DAN INCREMENTAL ROTARY
ENCODER BERBASIS ATMEGA 16"**

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima ka

Mengetahui

Ketua Program Studi Teknik Elektro S-1


Ir. Yusuf Ismail Nakhoda, MT

NIR.Y. 1018800189

Hormat Kami

HUDDIN

NIM. 0812217



PROGRAM STUDI TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **HUDDIN**
Nim : **0812217**
Semester : **VIII (Delapan)**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Elektronika**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" RANCANG BANGUN ROBOT BERODA PEMINDAH KERANJANG MENGGUNA KOMPAS DIGITAL DAN INCREMENTAL ROTARY ENCODER BERBASIS ATMEC 16"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami

Sotyohadi, ST
NIP.Y. 1039700309

Catatan :

Setelah disetujui agar formulir ini Diserahkan mahasiswa/i yang bersangkutan kepada jurusan untuk diproses lebih lanjut

*) Coret yang tidak perlu



FORMULIR BIMBINGAN SKRIPSI

Nama : HUDDIN
Nim : 08.12.217
Masa Bimbingan : Semester Ganjil 2012 -2013
Judul Skripsi : Rancang Bangun Robot Beroda Pemindah Keranjang Menggunakan Kompas Digital dan Incremental Rotary Encoder Berbasis ATMEGA 16

No	Tanggal	Uraian	Paraf Pembimbing
1	25-12-2012	Konsultasi BAB I (Latar Belakang, Rumusan Masalah, Tujuan dan Batasan Masalah)	
2	05-01-2013	Konsultasi BAB II (Landasan Teori dan Makalah Seminar Proposal)	
3	09-01-2013	Konsultasi BAB III (Blok Diagram, Perancangan dan Flowchart)	
4	14-01-2013	Perbaikan BAB I, BAB II dan BAB III	
5	26-01-2013	Konsultasi BAB IV (Pengujian dan Pembahasan Hasil)	
6	01-02-2013	Perbaikan Hasil Pengujian LCD, dan Tabel Pengujian Kompas dan Rotary Encoder	
7	06-02-2013	Perbaikan pengujian secara keseluruhan, Makalah Seminar Hasil dan BAB V	
8	16-02-2013	Konsultasi Proposal Skripsi	
9			
10			

Malang,
Dosen Pembimbing I

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358



FORMULIR BIMBINGAN SKRIPSI

Nama : HUDDIN
Nim : 08.12.217
Masa Bimbingan : Semester Ganjil 2012 -2013
Judul Skripsi : Rancang Bangun Robot Beroda Pemindah Keranjang Menggunakan Kompas Digital dan Incremental Rotary Encoder Berbasis ATMEGA 16

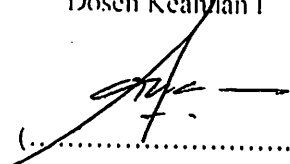
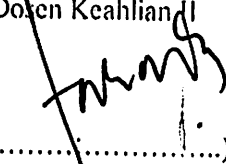
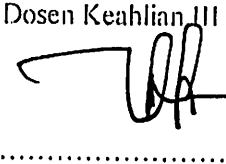
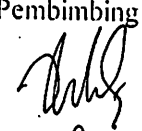
No	Tanggal	Uraian	Paraf Pembimbing
1	28-12-2012	Konsultasi BAB I (Latar Belakang, Rumusan Masalah, Tujuan dan Batasan Masalah)	
2	04-01-2013	Konsultasi BAB II (Landasan Teori dan Makalah Seminar Proposal)	
3	11-01-2013	Konsultasi BAB III (Blok Diagram, Perancangan dan Flowchart)	
4	16-01-2013	Perbaikan BAB I, BAB II dan BAB III	
5	23-01-2013	Konsultasi BAB IV (Pengujian dan Pembahasan Hasil)	
6	31-01-2013	Perbaikan Hasil Pengujian LCD, dan Tabel Pengujian Kompas dan Rotary Encoder	
7	05-02-2013	Perbaikan pengujian secara keseluruhan, Makalah Seminar Hasil dan BAB V	
8	16-02-2013	Konsultasi Proposal Skripsi	
9			
10			

Malang,
Dosen Pembimbing II

Sotvohadi, ST
NIP.Y. 1039700309



**BERITA ACARA SEMINAR PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Konsentrasi : Teknik Elektronika**

1.	Nim	: 0812217	
2.	Nama	: HUDDIN	
3.	Konsentrasi Jurusan	: Teknik Elektronika	
4.	Jadwal Pelaksanaan:	Waktu	Tempat
	16 April 2012	09.00	III.1.1
5.	Judul proposal yang diseminarkan Mahasiswa	RANCANG BANGUN ROBOT BERODA PEMINDAH KERANJANG MENGGUNAKAN KOMPAS DIGITAL, DAN INCREMENTAL ROTARY ENCODER BERBASIS-ATMEGA 16	
6.	Perubahan judul yang diusulkan oleh Kelompok Dosen Keahlian		
7.	Catatan :		
8.	Catatan :		
	Persetujuan judul Skripsi		
	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III
	 (.....)	 (.....)	 (.....)
	Mengetahui, Ketua Program Studi Teknik Elektro S-1	Disetujui, Calon Dosen Pembimbing ybs	
Ir. Yusuf Ismail Nakhoda.MT NIP. Y. 1018800189	Pembimbing I  (M. Hordani Isom) (.....)	Pembimbing II (.....)	



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Huddin
NIM :
Perbaikan melalui :

BAB 3. Perencanaan Komparasi

4. Encoder

1. Pengujian secara keseluruhan

4. Jarak

5. Kesimpulan diambil dari pengujian

penyualan PWM

Malang



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ENERGI LISTRIK

PERSETUJUAN PERBAIKAN SKRIPSI

Dari hasil ujian skripsi Jurusan Teknik Elektro jenjang strata satu (S-1) yang diselenggarakan pada :

Hari : Selasa
Tanggal : 19 Februari 2013

Telah dilakukan perbaikan skripsi oleh :

1. Nama : HUDDIN
2. NIM : 08.12.217
3. Jurusan : Teknik Elektro
4. Konsentrasi : Teknik Elektronika
5. Judul Skripsi : **RANCANG BANGUN ROBOT BERODA PEMINDAH KERANJANG MENGGUNAKAN KOMPAS DIGITAL DAN INCREMENTAL ROTARY ENCODER BERBASIS ATMEGA16**

No	Materi Perbaikan	Ket
1	Perancangan Kompas dan Rotary Encoder	
2	Pengujian Jarak dan secara keseluruhan	
3	Kesimpulan diambil di pengujian, pengujian PWM	

Dosen Penguji I

Irmalia Suryani F., ST., MT
NIP.P.1030009365

Dosen Pembimbing I

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Dosen Pembimbing II

Sotyhadi, ST
NIP.Y.1039700309