

**SINGLE AXIS LINEAR SLIDE ACTUATOR (SLIDER) ROBOT
MENGUNAKAN STEPPER MOTOR DALAM APLIKASI KONTES
ROBOT ABU INDONESIA**

SKRIPSI

Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik



DISUSUN OLEH :

**MUHAMMAD JAKA WIMBANG WICAKSONO
NIM. 115060300111047-63**

**KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS BRAWIJAYA
MALANG
2015**

LEMBAR PERSETUJUAN

***SINGLE AXIS LINEAR SLIDE ACTUATOR (SLIDER) ROBOT
MENGUNAKAN STEPPER MOTOR DALAM APLIKASI KONTES
ROBOT ABU INDONESIA***

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

MUHAMMAD JAKA WIMBANG WICAKSONO

NIM. 115060300111047-63

Telah diperiksa dan disetujui oleh :

Pembimbing I

Pembimbing II

Ir. Dipl. -Ing Mochammad Rusli
NIP. 19630104 198701 1 001

Akhmad Zainuri, ST, MT.
NIK. 19840120 201212 1 003

LEMBAR PENGESAHAN

SINGLE AXIS LINEAR SLIDE ACTUATOR (SLIDER) ROBOT
MENGGUNAKAN *STEPPER MOTOR* DALAM APLIKASI KONTES
ROBOT ABU INDONESIA

Disusun oleh:
MUHAMMAD JAKA WIMBANG WICAKSONO
NIM. 115060300111047-63

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 20 October 2015

DOSEN PENGUJI

Dr. Eng. Panca Mudjiraharjo, ST., MT
NIP. 19700329 200012 1 001

Moch. Rif'an, ST., MT.
NIP. 19710301 200012 1 001

Ir. Nurussa'adah, MT.
NIP. 19680706 199203 2 001

Mengetahui,

Ketua Jurusan Teknik Elektro

M. Aziz Muslim, ST., MT., Ph.D.
NIP. 19741203 200012 1 001

PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT, karena atas segala petunjuk dan nikmat-Nya lah skripsi ini dapat diselesaikan. Skripsi berjudul “*Single Axis Linear Slide Actuator (Slider) Robot Menggunakan Stepper Motor Dalam Aplikasi Kontes Robot Abu Indonesia* ” ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa dalam penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

- Allah SWT atas rahmat dan hidayah yang telah diberikan,
- Rosulullah Muhammad SAW, semoga shalawat serta salam selalu tercurah kepada beliau,
- Papa dan Mama atas segala nasehat, kasih sayang, perhatian dan kesabarannya didalam membesarkan dan mendidik penulis, serta telah banyak mendoakan kelancaran penulis hingga terselesaikannya skripsi ini,
- Bapak M. Aziz Muslim, ST., MT., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Hadi Suyono, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Ali Mustofa, ST., MT. selaku Ketua Prodi Strata Satu Jurusan Teknik Elektro Universitas,
- Bapak Ir. Dipl.-Ing. Mochammad Rusli selaku pembimbing I, terima kasih atas segala bimbingan, pengarahan, ide, saran, motivasi, dan masukan yang diberikan,
- Bapak Akhmad Zainuri, ST., MT. selaku pembimbing II, terima kasih atas segala bimbingan, pengarahan, ide, saran, motivasi, dan masukan yang diberikan,
- Ibu Dr-Ing. Onny Setyawati, ST., MT., M.SC. selaku dosen Penasehat Akademik,

- Ibu Ir. Nurussa'adah, MT. selaku Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya,
- Seluruh dosen pengajar Teknik Elektro Universitas Brawijaya,
- Staff Recording Jurusan Teknik Elektro,
- Teman – teman Inverter angkatan 2011,
- Teman – teman Inverternika 2011,
- Rekan seperjuangan dalam skripsi Blek, Mi-a, Darjo, Emon, Busteler, Bowo, UGB, Desta, Manda, Mbak Jah, Bebeb , RT, Ocir , Icil ,Nurdin terima kasih atas segala bantuan yang telah diberikan,
- Rekan-rekan Tim Robot Kontes Robot Abu Indonesia, Agus, Ncrol, Fai, Kunam, Hasyim, terima kasih atas segala bantuan dan semangat yang telah diberikan,
- Rekan-rekan Asisten Laboratorium Elektroika.
- Keluarga Besar Adistya Sekar Lestari yang selama ini memberikan semangat untuk mengerjakan skripsi ini,
- Seluruh Keluarga Besar Tim Robot UB Jurusan Teknik Elektro atas segala bantuan alat, bahan, dan masukan – masukan yang telah diberikan,
- Sekret KRAI, sekret KRPAI, sekret KRSI, dan Laboratorium Elektronika yang selama ini telah menyediakan tempat bagi penulis dalam mengerjakan skripsi ini,
- Seluruh teman-teman serta semua pihak yang tidak mungkin bagi penulis untuk mencantumkan satu-persatu, terimakasih banyak atas bantuan dan dukungannya.

Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi serta bagi masyarakat.

Malang, Oktober 2015

Penulis

DAFTAR ISI

PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR TABEL.....	xii
RINGKASAN	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan	3
1.5 Sistematika Penulisan.....	3
BAB II DASAR TEORI.....	5
2.1 <i>Membrane Keypad 4x3</i>	5
2.2 <i>Stepper Motor</i>	6
2.2.1 Tipe-Tipe Dari <i>Stepper Motor</i>	6
2.2.2 Rangkaian Dasar Pengontrolan.....	11
2.3 <i>Driver L297</i>	13
2.3.1 Urutan Fasa Motor <i>Driver</i>	15
2.3.1.1 Mode <i>Full Step (One Phase On Mode)</i>	15
2.3.1.2 Mode <i>Full Step (Two phases On)</i>	16
2.3.1.3 Mode <i>Half Step</i>	17
2.4 Sensor <i>Rotary Encoder</i> Autonics E40H8-500P/R-6-L-5.....	18
2.5 Mikrokontroler Atmega 16	20
2.6 <i>LCD (Liquid Crystal Display)</i>	21
2.7 <i>Bipolar Junction Transistor</i>	22
2.7.1 Konstruksi dari Transistor.....	22

2.7.2 Operasi dari transistor	23
BAB III METODOLOGI PENELITIAN	25
3.1 Penentuan Spesifikasi <i>Slider</i> Robot	25
3.2 Perancangan Sistem	25
BAB IV PERANCANGAN DAN PEMBUATAN ALAT	29
4.1 Perancangan Sistem Robot.....	29
4.2 Perancangan Mekanik <i>Slider</i> Robot.....	29
4.3 Perancangan Perangkat Keras	30
4.3.1 Perancangan Rangkaian <i>Stepper Motor Driver</i>	30
4.3.2 Perancangan <i>System Minimum Atmega16</i>	33
4.3.4 Perancangan Perangkat Lunak	36
BAB V PENGUJIAN DAN ANALISIS.....	37
5. 1 Pengujian Catu Rangkaian.....	37
5.2 Pengujian Rangkaian <i>Stepper Motor Driver</i>	38
5.2.1 Pengujian Arus Rangkaian <i>Stepper Motor Driver</i>	38
5.2.2 Pengujian Mode Rangkaian <i>Stepper Motor Driver</i>	39
5.2.2.1 Pengujian Mode <i>Full Step</i>	39
5.2.2.2 Pengujian Mode <i>Half Step</i>	41
5.2.3 Pengujian Sudut Motor	43
5.2.3.1 Pengujian Sudut Motor Dengan Mode <i>Full Step</i>	43
5.2.3.1.1 Pengujian Resolusi Sudut Motor Dengan Mode <i>Full Step</i>	44
5.2.3.2 Pengujian Sudut Motor Dengan Mode <i>Half Step</i>	45
5.2.3.2.1 Pengujian Resolusi Sudut Motor Dengan Mode <i>Half Step</i>	47
5.2.4 Pengujian Jarak Pergeseran <i>Slider</i> Saat Motor Bergerak.....	48
5.2.4.1 Pengujian Jarak Pergeseran <i>Slider</i> Saat Motor Bergerak Dengan Mode <i>Full Step</i>	48

5.2.4.1.1 Pengujian Resolusi Jarak Pergeseran Slider Saat Motor Bergerak Dengan Mode Full Step	50
5.2.4.2 Pengujian Jarak Pergeseran <i>Slider</i> Saat Motor Bergerak Dengan Mode <i>Half Step</i>	51
5.2.4.2.1 Pengujian Resolusi Jarak Pergeseran <i>Slider</i> Saat Motor Bergerak Dengan Mode <i>Half Step</i>	53
5.3 Pengujian Waktu Pergerakan <i>Slider</i>	53
5.4 Pengujian Rangkaian <i>Rotary Encoder</i>	54
5.4.1 Pengujian Sudut <i>Rotary Encoder</i>	54
5.4.1.1 Pengujian Resolusi Sudut <i>Rotary Encoder</i>	56
5.4.2 Pengujian Jarak <i>Slider</i> dengan <i>Rotary Encoder</i>	57
5.4.2.1 Pengujian Resolusi Jarak <i>Slider</i> Dengan <i>Rotary Encoder</i>	59
5.5 Pengujian Keseluruhan.....	60
BAB VI KESIMPULAN DAN SARAN.....	63
6.1 Kesimpulan	63
6.2 Saran.....	63
DAFTAR PUSTAKA	64
LAMPIRAN I	65
LAMPIRAN II.....	68
LAMPIRAN III	79

DAFTAR GAMBAR

Gambar 1.1 Lapangan Abu Robocon 2013..... 2

Gambar 2.1 Susunan Skematik *Membrane Keypad* 4x3..... 5

Gambar 2.2 *Membrane Keypad* 4x3. 5

Gambar 2.3 *Stepper Motor Variable Reluctance*..... 7

Gambar 2.4 *Stepper Motor Unipolar*..... 8

Gambar 2.5 *Stepper Motor bipolar*..... 9

Gambar 2.6 *Stepper Motor Bifilar*. 10

Gambar 2.7 *Stepper Motor Hybrid*. 10

Gambar 2.8 Rangkaian Kontrol Motor *Variable Reluctance*..... 11

Gambar 2.9 Rangkaian Kontrol Motor Unipolar. 12

Gambar 2.10 Rangkaian Kontrol Motor Bipolar. 12

Gambar 2.11 *Pin* Konfigurasi dari L297..... 13

Gambar 2.12 Pergerakan *Stepper Motor* Unipolar dengan Mode *One Phase On*. 15

Gambar 2.13 Gelombang *Stepper Motor* Unipolar dengan Mode *One Phase On*.15

Gambar 2.14 Pergerakan *Stepper Motor Unipolar* dengan Mode *Two Phases On*
..... 16

Gambar 2.15 Gelombang *Stepper Motor Unipolar* dengan Mode *Two Phases On*
..... 16

Gambar 2.16 Pergerakan *Stepper Motor Unipolar* dengan Mode *Half step*..... 17

Gambar 2.17 Gelombang *Stepper Motor Unipolar* dengan Mode *Half Step* 18

Gambar 2.18 *Rotary Encoder Autonics*. 19

Gambar 2.19 *Line Driver Output Sensor Rotary Encoder*..... 19

Gambar 2.20 Konfigurasi *Pin* Atmega 16..... 20

Gambar 2.21 LCD (*Liquid Crystal Display*) LM162A..... 22

Gambar 2.22 Transistor (a) PNP (b) NPN 23

Gambar 2.23 *Forward Bias* pada Salah Satu *Junction* Transistor dengan
Pembawa Mayoritas 23

Gambar 2.24 *Forward Bias* pada Salah Satu *Junction* Transistor dengan
Pembawa Minoritas..... 23

Gambar 2.25 Aliran Pembawa Mayoritas dan Minoritas Pada Transistor PNP. ... 24

Gambar 4.1 Diagram Blok Perancangan Sistem Secara Keseluruhan..... 29



Gambar 4.2 <i>Slider</i>	30
Gambar 4.3 Diagram Blok IC L297.....	30
Gambar 4. 4 Sebagian Rangkaian <i>Stepper Motor Driver</i>	33
Gambar 4.5 Rangkaian Keseluruhan <i>Driver Motor</i>	33
Gambar 4.6 <i>System Minimum</i> Mikrokontroler Atmega16.	35
Gambar 4.7 Skematik Perancangan LCD.	35
Gambar 4.8 <i>Flowchart</i> Program.	36
Gambar 5. 1 Diagram Blok Pengujian Catu Rangkaian.	37
Gambar 5. 2 Sinyal <i>Output</i> Pengujian Catu Rangkaian.....	37
Gambar 5. 3 Skematik Rangkaian Catu.	37
Gambar 5. 4 Skematik Pengujian Arus <i>Driver</i>	38
Gambar 5. 5 Blok Pengujian <i>Stepper Motor Driver</i>	39
Gambar 5. 6 Sinyal <i>Input CLK</i> dan A Pada Mode <i>Full Step</i>	39
Gambar 5. 7 Sinyal <i>Output Fasa A</i> dan D Pada Mode <i>Full Step</i>	39
Gambar 5. 8 Sinyal <i>Output fasa D</i> dan B Pada Mode <i>Full Step</i>	40
Gambar 5. 9 Sinyal <i>output fasa B</i> dan C Mode <i>Full Step</i>	40
Gambar 5. 10 Sinyal <i>Full Step</i> Berdasarkan <i>Datasheet</i>	40
Gambar 5. 11 Sinyal <i>Input CLK</i> dan A Pada mode <i>Half Step</i>	41
Gambar 5. 12 Sinyal <i>Output Fasa A</i> dan D Pada Mode <i>Half Step</i>	41
Gambar 5. 13 Sinyal <i>Output Fasa D</i> dan B Mode <i>Half Step</i>	41
Gambar 5. 14 Sinyal <i>Output Fasa B</i> dan C Mode <i>Half Step</i>	42
Gambar 5. 15 Sinyal <i>Half Step</i> Berdasarkan <i>Datasheet</i>	42
Gambar 5. 16 Blok Pengujian Sudut Motor	43
Gambar 5. 17 Grafik Pengujian Sudut <i>Stepper Motor</i> Pada Mode <i>Full Step</i>	43
Gambar 5. 18 Pengujian Sudut <i>Stepper Motor</i>	47
Gambar 5. 19 Grafik Pengujian Sudut <i>Stepper Motor</i> Pada Mode <i>Half Step</i>	47
Gambar 5. 20 Blok Pengujian Jarak Pergeseran <i>Slider</i> Saat Motor Bergerak.	48
Gambar 5. 21 Grafik Pengujian Jarak Pergeseran <i>Slider</i> Dengan Mode <i>Full Step</i>	49
Gambar 5. 22 Pengujian Jarak Pergeseran <i>Slider</i> Dengan Mode <i>Half Step</i>	51
Gambar 5. 23 Blok Pengujian Sudut <i>Rotary Encoder</i>	54
Gambar 5. 24 Grafik Pengujian Sudut <i>Rotary Encoder</i>	56



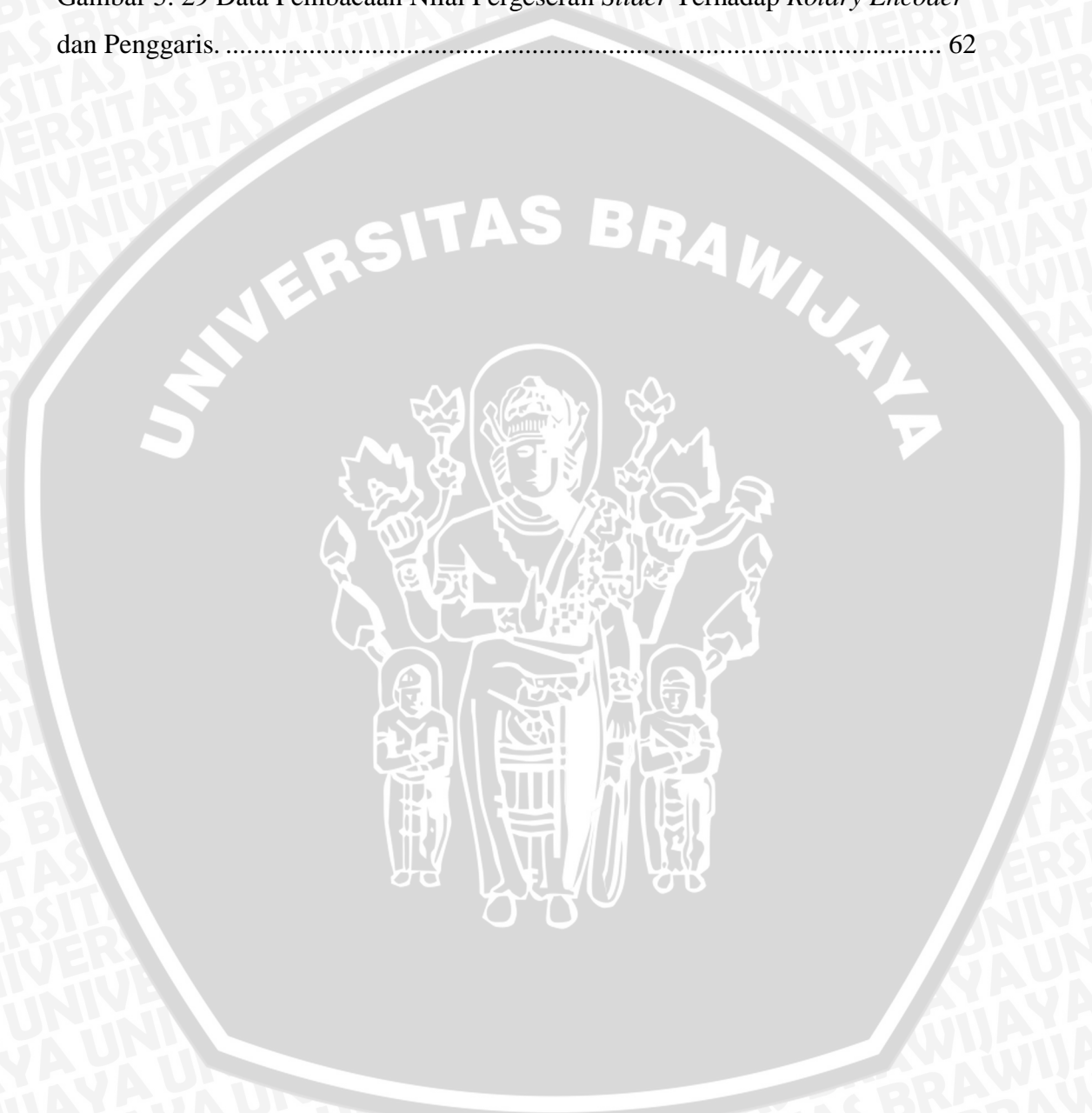
Gambar 5. 25 Pengujian Sudut *Rotary Encoder*. 56

Gambar 5. 26 Pengujian Jarak *Slider* Dengan *Rotary Encoder*. 57

Gambar 5. 27 Grafik Pengujian Jarak *Slider* Dengan *Rotary Encoder*. 59

Gambar 5. 28 Pengujian Jarak *Slider* Dengan *Rotary Encoder*. 59

Gambar 5. 29 Data Pembacaan Nilai Pergeseran *Slider* Terhadap *Rotary Encoder* dan Penggaris. 62



DAFTAR TABEL

Tabel 2.1 <i>Absolute Maximum Rating</i> dari L297	13
Tabel 2.2 Penjelasan Konfigurasi <i>Pin IC L297</i>	14
Tabel 2.3 Logika dari Mode <i>One Phase On</i>	16
Tabel 2.4 Logika dari Mode <i>Two Phases On</i>	17
Tabel 2. 5 Logika dari Mode <i>Half Step</i>	18
Tabel 2. 6 Konfigurasi <i>Pin Rotary Encoder</i>	19
Tabel 2. 7 Fungsi Khusus <i>Port B</i>	20
Tabel 5. 1 Hasil Pengujian Arus <i>Driver</i>	38
Tabel 5. 2 Logika Fasa Mode <i>Full Step</i>	40
Tabel 5. 3 Logika Fasa Mode <i>Half Step</i>	42
Tabel 5. 4 Pengujian Sudut Dengan Mode <i>Full Step</i>	44
Tabel 5. 5 Pengujian Resolusi Sudut Motor Dengan Mode <i>Full Step</i>	45
Tabel 5. 6 Pengujian Sudut Dengan Mode <i>Half Step</i>	46
Tabel 5. 7 Pengujian Resolusi Sudut Motor Dengan Mode <i>Half Step</i>	48
Tabel 5. 8 Pengujian Jarak Pergeseran <i>Slider</i> Dengan Mode <i>Full Step</i>	49
Tabel 5. 9 Pengujian Resolusi Pergeseran <i>Slider</i> Saat Motor Bergerak Dengan Mode <i>Full Step</i>	50
Tabel 5. 10 Pengujian Jarak Pergeseran <i>Slider</i> Dengan Mode <i>Half Step</i>	52
Tabel 5. 11 Pengujian Resolusi Pergeseran <i>Slider</i> Saat Motor Bergerak Dengan Mode <i>Half Step</i>	53
Tabel 5. 12 Pengujian Waktu Pergerakan <i>Slider</i>	54
Tabel 5. 13 Pengujian Sudut <i>Rotary Encoder</i>	55
Tabel 5. 14 Pengujian Resolusi Sudut <i>Rotary Encoder</i>	57
Tabel 5. 15 Pengujian Jarak <i>Slider</i> Dengan <i>Rotary Encoder</i>	58
Tabel 5. 16 Pengujian Resolusi Jarak <i>Slider</i> Dengan <i>Rotary Encoder</i>	60
Tabel 5. 17 Nilai Pergeseran <i>Slider</i> Terhadap <i>Rotary Encoder</i> dan Penggaris	61



RINGKASAN

Muhammad Jaka Wimbang Wicaksono, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya, Juli 2015, *Single Axis Linear Slide Actuator (Slider) Robot Menggunakan Stepper Motor Dalam Aplikasi Kontes Robot Abu Indonesia*, Dosen Pembimbing : Ir.Dipl.-Ing Mochammad Rusli dan Akhmad Zainuri, ST., MT..

Kontes Robot Abu (KRAI) adalah perlombaan resmi yang diselenggarakan tiap tahunnya oleh DIKTI. Perlombaan ini di ikuti oleh seluruh Universitas di Indonesia. *Rule* dari perlombaan KRAI adalah bersumber dari perlombaan se-Asia-Pasific yaitu ABU Robocon (*Asia-Pasific Broadcasting Union Robot Contest*). Peraturan ABU Robocon berbeda tiap tahunnya sesuai negara mana yang menjadi tuan rumah. *Single axis linear slide actuator (slider)* sering digunakan dalam perlombaan KRAI apabila robot membutuhkan perpanjangan lengan dalam melaksanakan tugasnya. Efisiensi gerakan sangat dibutuhkan pada pergerakan *slider*. *Slider robot* menggunakan aktuator yaitu *stepper motor*. *Rotary encoder* sebagai pembaca posisi *slider* dan *keypad* sebagai *setpoint*. *Stepper motor* yang digunakan memiliki arus perfasa sebesar 1,5 A/phase dan *step angle* sebesar 1,8^o. Dalam pengujian di dapatkan jarak per pulsa sebesar 0,44mm/pulsa untuk mode *full step* dan 0,21875 mm/pulsa untuk mode *half step* dan nilai pembacaan jarak *rotary encoder* adalah 3,3 % dan kesalahan pembacaan penggaris sebesar 1,67 %.

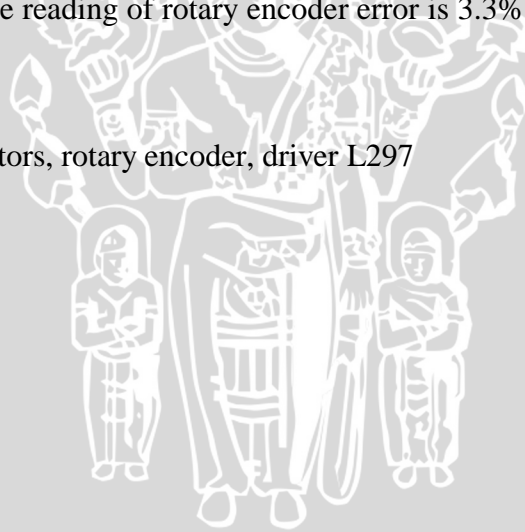
Kata kunci : *stepper motor* , *rotary encoder* , *driver L297*

SUMMARY

Muhammad Jaka Wimbang Wicaksono, Department of Electrical Engineering, Faculty of Engineering, Universitas Brawijaya, in September 2015, Single Axis Linear Slide Actuator (Slider) Robot Using Stepper Motor Applications In Abu Indonesian Robot Contest, Supervisor: Ir.Dipl.-Ing Mochammad Rusli and Akhmad Zainuri, ST., MT ..

Abu Robot Contest (KRAI) is an official competition organized annually by the Directorate of Higher Education. The event participated by all universities in Indonesia. Rule of race KRAI is sourced from the event in Asia-Pacific, namely ABU Robocon (Asia-Pacific Broadcasting Union Robot Contest). ABU Robocon have different rule each year according which Country hosted. Single axis linear slide actuator slide (slider) is often used in the competition KRAI when robot arm requires an extension in their duties. Efficiency of movement is needed on the movement of the slider. Slider robot using stepper motor as actuator. Rotary encoder as the position censor of the slider and keypad as setpoint. Stepper motor which used have current per phase 1.5 A / phase and step angle of 1.8° / phase. In this research distance per pulse of a full step mode is 0,44mm / pulse and for half step mode is 0.21875 mm / pulse and distance sense rotary value per pulse is 1.7543 mm / pulse. The reading of rotary encoder error is 3.3% and the reading of ruler error is 1.67%.

Keywords: stepper motors, rotary encoder, driver L297



BAB I PENDAHULUAN

1.1 Latar Belakang

Teknologi robotika sangat berperan dalam peningkatan efisiensi kehidupan manusia. Peran serta teknologi robotika bisa dilihat dalam kehidupan sehari-hari. Selama ini teknologi robotika semakin berkembang, mulai *mobile* robot sampai *humanoid* robot. Perkembangan teknologi ini tidak luput dari peran institusi pendidikan ataupun para pengembang. Kontes robot internasional Asia Pacific Broadcasting Union Robot Contest (ABU Robocon) diadakan setiap tahun dan pertama diadakan pada tahun 2002. Setiap tahun aturan lomba selalu berganti dan menuntut solusi pemecahan permasalahan yang paling baik. Salah satu permasalahan tersebut adalah pergerakan robot dengan efektifitas yang tinggi. Peraturan lomba ABU Robocon 2013 misalnya, robot harus meletakkan token ke dalam wadah berbentuk lingkaran dengan jarak yang berbeda-beda. Peraturan lomba ABU Robocon tahun 2014 robot harus melewati rintangan berupa tangga dimana robot dituntut untuk bisa memanjat dan melakukan perpanjangan agar bisa naik ke atas tangga yang tentunya sangat sulit. (Dikti) Berdasarkan permasalahan tersebut kita dituntut untuk membuat teknologi pergerakan *slider* robot yang lebih efektif, salah satu diantaranya adalah dengan *stepper motor*. Lapangan ABU Robocon 2013 ditunjukkan pada Gambar 1.1.

Stepper motor adalah sebuah perangkat elektromekanis yang mengubah pulsa elektrik menjadi pergerakan mekanik yang diskrit. *shaft* atau *spindle* pada *stepper motor* berputar secara *step* diskrit bertahap ketika *pulse* diberikan dengan urutan yang tepat. Rotasi motor memiliki beberapa hubungan langsung terhadap *pulse* masukan. Urutan *pulse* yang diberikan berkaitan langsung dengan rotasi *shaft* motor. Kecepatan rotasi *shaft* secara langsung berkaitan dengan frekuensi dari *pulse* masukan dan lama rotasi secara langsung berkaitan dengan jumlah pulsa yang diberikan. (Solarbotics, 2003).

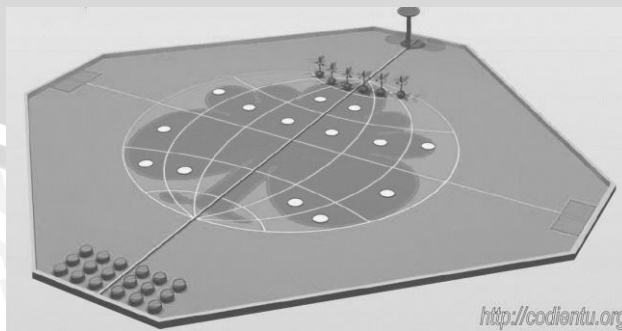
Salah satu keuntungan yang paling signifikan pada *stepper motor* adalah kemampuannya untuk di kontrol secara akurat dengan sistem *open loop*. Sistem kontrol *open loop* maksudnya tidak membutuhkan *feedback* posisi. Tipe dari



sistem kontrol ini menghilangkan kebutuhan dari penggunaan alat sensor dan *feedback* yang mahal seperti *optical encoder*. Posisi dari *stepper motor* dapat diketahui dengan cara memantau pulsa masukan. (Solarbotics, 2003) Torsi *stepper motor* tidak sebesar motor DC, namun *stepper motor* mempunyai tingkat presisi yang sangat tinggi dalam pergerakannya. Kecepatan gerak *stepper motor* dinyatakan dalam *step per second* atau jumlah *step* gerakan dalam setiap detiknya. (Kustija, 2012) Kontrol pergerakan dalam bidang industri telah menggunakan motor DC secara luas karena mudah untuk mencapai kinerja tinggi dengan kontrol yang tepat. Motor DC memiliki rasio torsi dan rasio inersia yang tinggi. Kekurangan dari motor DC adalah biaya *maintenance* yang tinggi, sikat pada motor yang harus sering di bersihkan. (Bodson Mark, 1993) Sehingga penggunaan *stepper motor* adalah pilihan alternatif dalam dunia *motion control*.

Dalam perancangan digunakan sensor *rotary encoder* yang digunakan hanya sebagai pembaca jarak dari pergeseran *slider* tanpa memberikan umpan balik pada *stepper motor*. *Rotary encoder* adalah divais elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan sebagai gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder*. (Lukman, 2013)

Skripsi ini dirancang menggunakan *keypad* sebagai *input setpoint* berupa jarak perpindahan *slider* dan diproses menggunakan mikrokontroler menggunakan bahasa C. *Driver* yang dirancang menggunakan divais L297 yang ditambahkan transistor. *Driver* ini mempunyai 4 *pulse output* yang berupa sinyal kotak dengan fasa yang berbeda antara satu dan yang lain.



Gambar 1.1 Lapangan Abu Robocon 2013.

Sumber: Robocon (2013:5).

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, dapat disusun rumusan masalah sebagai berikut :

- 1) Bagaimana merancang, mengimplementasikan, dan menganalisa pergerakan *slider* menggunakan *stepper motor*.
- 2) Bagaimana merancang, mengimplementasikan, dan menganalisa rangkaian *driver stepper motor*.
- 3) Bagaimana mengimplementasikan penggunaan *rotary encoder*.

1.3 Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan penelitian akan diberikan batasan sebagai berikut :

- 1) Karakteristik *stepper motor* yang digunakan adalah $1,8^\circ/\text{pulsa}$.
- 2) Kecepatan motor konstan.
- 3) Perpanjangan *slider* maksimal 40 cm dengan *range* pergerakan 1 cm.
- 4) Sistem pengendalian menggunakan *open loop control*.
- 5) *Rotary encoder* bukan sebagai *feedback* sistem.
- 6) Pengujian *slider* dilakukan dengan beban 0, 1, 2, 3, 4 gram.

1.4 Tujuan

Tujuan skripsi ini adalah merancang dan membangun sistem pergerakan *slider* robot yang bergerak dengan keakuratan yang tinggi dengan menggunakan sistem *open loop* dan menggunakan mikrokontroler sebagai pemroses utamanya.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini sebagai berikut :

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, ruang lingkup, tujuan, dan sistematika pembahasan.

BAB II Dasar Teori

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan *slider* robot.

BAB III Metodologi

Berisi tentang tata cara yang dipakai dalam melakukan perancangan, pengujian, dan analisis data.

BAB IV Perancangan

Perancangan dan perealisasiian *slider* robot yang meliputi spesifikasi, perencanaan diagram blok, prinsip kerja dan realisasi *slider* robot.

BAB V Pengujian dan Analisis

Memuat aspek pengujian meliputi penjelasan tentang cara pengujian dan hasil pengujian. Aspek analisis meliputi penilaian atau komentar terhadap hasil-hasil pengujian. Pengujian dan analisis terhadap *slider* robot yang telah direalisasikan berdasarkan masing-masing blok dan sistem secara keseluruhan.

BAB VI Kesimpulan dan Saran

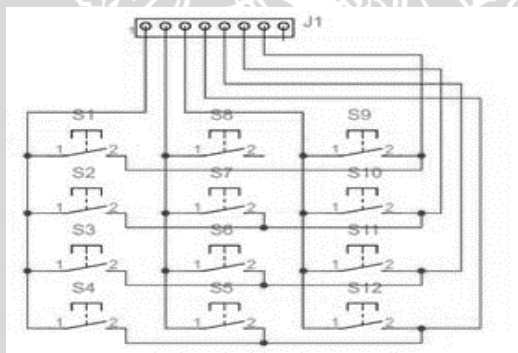
Memuat intisari hasil pengujian dan menjawab rumusan masalah serta memberikan rekomendasi untuk perbaikan kualitas penelitian dimasa yang akan datang.



BAB II DASAR TEORI

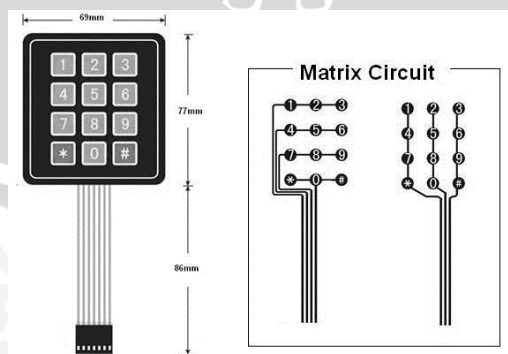
2.1 Membrane Keypad 4x3

Membrane keypad 4x3 adalah susunan dari beberapa buah saklar tekan (*push button*) yang disusun secara matrix. *Keypad 12-tombol* ini berguna bagi proyek-proyek mikrokontroler dengan fitur yang ringan dan *interface* yang mudah ke berbagai mikrokontroler. *Keypad* menggunakan kombinasi dari 4 baris dan 3 kolom. Satu tombol menghubungkan satu kolom dengan satu baris. Cara untuk menentukan tombol yang di tekan adalah dengan mengatur 4 kolom sebagai keluaran berlogika *high* atau *low* dan atur 3 baris sebagai masukan. Pada saat tombol di tekan terjadi perubahan logika. Perubahan ini direspon oleh mikrokontroler sebagai posisi tombol apa yang ditekan. (Parallax) Susunan skematik *membrane keypad 4x3* ditunjukkan pada Gambar 2.1 dan Gambar *membrane keypad 4x3* ditunjukkan pada Gambar 2.2.



Gambar 2.1 Susunan Skematik Membrane Keypad 4x3.

Sumber: Deo (2011).



Gambar 2.2 Membrane Keypad 4x3.

Sumber: Deo (2011).

2.2 Stepper Motor

Stepper motor mengisi tempat yang unik dalam dunia pengontrolan motor. Motor ini biasanya digunakan dalam pengukuran dan aplikasi kontrol. Contoh aplikasi termasuk *printer ink jet*, mesin CNC dan pompa volumetrik. Beberapa fitur umum untuk semua jenis *stepper motor* membuat penggunaan *stepper motor* menjadi ideal untuk berbagai macam aplikasi. Douglas (2004:1) Fitur ini sebagai berikut :

1. *Brushless- Stepper motor* adalah motor *brushless*. Komutator dan sikat dari motor-motor konvensional adalah beberapa dari komponen yang rawan kegagalan, dimana tercipta busur (*arc*) listrik yang tidak di inginkan atau berbahaya dalam beberapa lingkungan.
2. *Load independent- Stepper motor* akan berputar pada sebuah kecepatan yang diberikan selama beban tidak melebihi kapasitas dari *stepper motor*.
3. *Open loop positioning – stepper motor* dapat bergerak secara bertahap atau per *step* . Selama motor bergerak didalam torsi yang sesuai spesifikasi. Posisi dari *shaft* (poros) dapat diketahui setiap saat tanpa menggunakan mekanisme umpan balik.
4. *Holding torque- stepper motor* dapat menahan poros secara seimbang.
5. *Excellent response – respon* yang sangat baik terhadap mulai , berhenti dan berbalik arah.

2.2.1 Tipe-Tipe Dari Stepper Motor

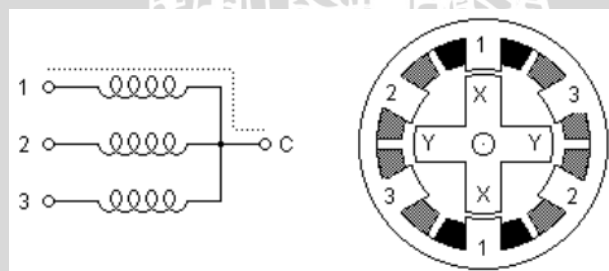
Ada 3 jenis dari *stepper motor* : magnet permanen, *variable reluctance* dan *hybrid*. Motor magnet permanen mempunyai rotor magnet sedangkan motor *variable reluctance* mempunyai rotor ber-gigi halus. Stator atau bagian yang tidak dapat bergerak berisi beberapa lilitan. Susunan dari lilitan adalah faktor utama yang membedakan berbagai jenis dari *stepper motor* dari sudut pandang elektrik. Dari perspektif elektrik dan kontrol, motor *variable reluctance* berbeda jauh dengan tipe yang lain. Kedua jenis motor magnet permanen dan motor *hybrid* menggunakan lilitan *unipolar*, lilitan *bipolar* dan lilitan *bifiliar*. Douglas (2004:1)

2.2.1.1 Motor Variable Reluctance

Motor *variable reluctance* (motor *variable switched reluctance*) mempunyai 3 sampai 5 lilitan yang terhubung dengan terminal *common*. Gambar 2.3

menunjukkan penampang dari 3 lilitan, 30 derajat per *step* motor *variable reluctance*. Rotor di dalam motor ini mempunyai 4 gigi dan 6 kutub dengan lilitan pada kutub yang berlawanan. Gigi rotor yang bertanda X tertarik oleh lilitan ketika di catu. Penyebabnya adalah fluks magnet yang dihasilkan di sekitar kumparan dan rotor. Rotor mengalami torsi dan menggerakkan motor segaris dengan kumparan yang tercatu, mengurangi jalur fluks. Motor bergerak searah jarum jam ketika lilitan 1 dimatikan dan lilitan 2 di catu. Gigi rotor di tandai dengan Y yang tertarik oleh lilitan 2. Hal ini menghasilkan pergerakan 30 derajat searah jarum jam dengan Y segaris terhadap lilitan 2. Gerakan searah jarum jam yang terus menerus di dapatkan dari mematikan dan mengaktifkan pencatuan. Urutan pengontrolan berikut akan memutarakan motor di Gambarkan pada Gambar 2.3 searah jarum jam dengan 12 *step* atau satu putaran.

Gambar 2.3 menggambarkan yang paling dasar dari *stepper motor variable reluctance*. Dalam prakteknya, motor ini biasanya mempunyai kutub kumparan dan gigi yang lebih banyak untuk sudut *step* yang lebih kecil. Jumlah kutub bisa dibuat lebih besar dengan menambahkan lilitan, sebagai contoh, menggerakkan 4 atau 5 lilitan tetapi untuk sudut *step* yang kecil. Solusi yang biasa dilakukan adalah dengan menggunakan kutub bergigi yang bekerja melawan gigi rotor. Motor *variable reluctance* menggunakan pendekatan ini tersedia dengan sudut *step* mendekati 1 derajat. Douglas (2004:2)



Gambar 2.3 Stepper Motor Variable Reluctance.

Sumber : Douglas (2004:2).

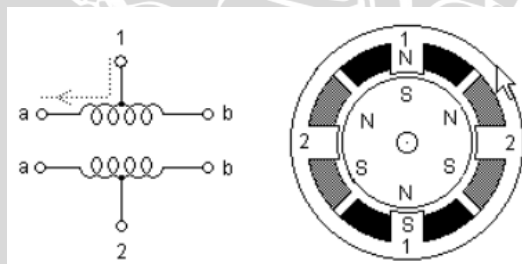
2.2.1.2 Motor Unipolar

Stepper motor unipolar terdiri dari dua lilitan, masing-masing dengan *center tap*. *Center tap* memiliki 2 kabel yang terpisah (ditunjukkan Gambar 2.4) atau terhubung satu sama lain secara internal ataupun terhubung dengan satu kabel yang menjulur keluar. Sebagai hasilnya, motor *unipolar* mempunyai 5 atau 6

kabel. Tanpa memperhatikan jumlah kabel, motor *unipolar* di picu dengan cara yang sama. Bagian kabel *center tap* dihubungkan ke sumber *power* dan ujung dari kumparan dihubungkan ke *ground* secara bergantian.

Stepper motor unipolar, sama seperti motor magnet permanen dan *hybrid*, memiliki prinsip kerja yang berbeda dari motor *variable reluctance*. Daripada beroperasi dengan meminimalkan panjang jalur fluks antara kutub stator dan gigi rotor, dimana arah aliran arus melalui lilitan stator tidak relevan, motor tersebut beroperasi dengan menarik kutub utara dan selatan dari rotor magnet permanen ke kutub stator. Arah arus yang melewati lilitan stator menunjukkan kutub rotor mana yang tertarik oleh kutub stator. Arah arus pada motor *unipolar* tergantung pada bagian lilitan mana yang dipicu. Secara fisik, sebagian dari lilitan adalah *wound* yang paralel satu sama lain. Oleh karena itu, satu lilitan bertindak sebagai kutub utara atau selatan tergantung pada bagian yang mana yang di picu.

Gambar 2.4 menunjukkan penampang dari motor *unipolar* 30 derajat per *step*. Lilitan motor bernomer 1 di distribusikan antara kutub stator atas dan bawah, saat lilitan bernomer 2 di distribusikan antara kutub motor kiri dan kanan. Rotor adalah magnet permanen dengan 6 kutub, 3 kutub utara dan 3 kutub selatan. Douglas (2004:2)



Gambar 2.4 *Stepper Motor Unipolar*.

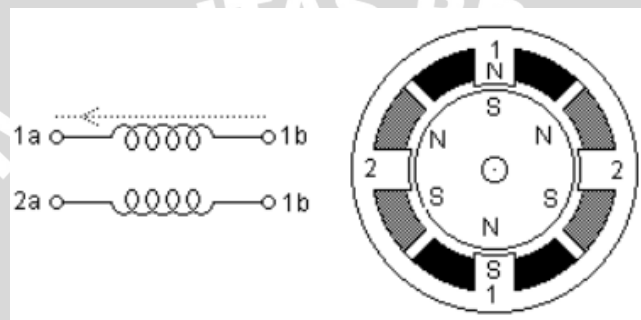
Sumber : Douglas (2004:2).

2.2.1.3 Motor *Bipolar*

Stepper motor bipolar terdiri dari 2 lilitan dan 4 kabel. Tidak seperti motor unipolar, motor bipolar tidak mempunyai *center tap*. Keuntungan tidak mempunyai *center tap* adalah arus mengalir melalui seluruh lilitan pada satu waktu bukan hanya sebagian. Sebagai hasilnya, motor *bipolar* menghasilkan lebih banyak torsi daripada motor *unipolar* pada ukuran motor yang sama. *Bipolar*

motor di bandingkan dengan *unipolar* motor , sirkuit kontrol yang lebih rumit diperlukan oleh motor *bipolar*.

Arus mengalir pada motor *bipolar* secara *bidirectional*. Hal ini membutuhkan perubahan polaritas pada ujung kumparan. Ditunjukkan pada Gambar 2.5, arus akan mengalir dari kiri ke kanan lilitan 1 ketika 1a positif dan 1b negatif. Arus mengalir dengan arah yang berbeda ketika kutub pada ujung berganti polaritas. Sebuah sirkuit kontrol, dikenal sebagai *H-Bridge*, digunakan untuk menukar polaritas pada ujung kumparan. Setiap motor *bipolar* mempunyai 2 lilitan, oleh karena itu , 2 sirkuit kontrol *H-Bridge* dibutuhkan pada setiap motor.



Gambar 2.5 *Stepper Motor bipolar*.

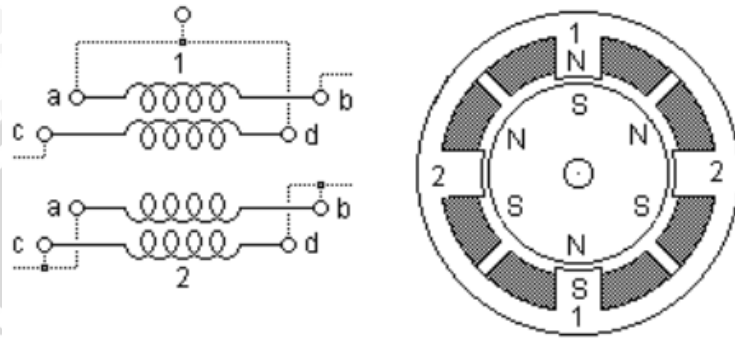
Sumber : Douglas (2004:3).

Gambar 2.5 menggambarkan motor bipolar 30 derajat per *step*. Lilitan motor 1 di distribusikan antara kutub stator atas dan bawah, ketika lilitan 2 di distribusikan antara kutub stator kanan dan kiri. Rotor menggunakan magnet permanen dengan 6 kutub, 3 selatan dan 3 utara diaturnya. Sama seperti motor *unipolar*, motor *bipolar* bisa berputar secara step dengan 2 urutan kontrol yang berbeda. Menggunakan + dan - untuk mengindikasikan kutub dari *power* yang diberikan pada terminal motor dan 0 menunjukkan tidak ada sumber tegangan yang diberikan. Douglas (2004:3)

2.2.1.4 Motor *Bifilar*

Istilah *bifilar* berarti 2 ulir. Motor dengan lilitan bifilar adalah sama pada rotor dan stator terhadap motor bipolar dengan satu pengecualian setiap lilitan terbuat dari 2 lilitan kabel yang paralel satu sama lain. Sebagai hasilnya, motor *bifilar* memiliki 8 kabel bukan 4 kabel dibandingkan dengan motor *bipolar*. Motor *bifilar* di picu bisa sebagai *bipolar* atau *unipolar*. Untuk menggunakan motor *bifilar* sebagai motor *unipolar*, 2 kabel dari masing-masing lilitan yang

berhubungan seri dan titik hubung digunakan sebagai *center tap*. Lilitan 1 yang ditunjukkan Gambar 2.6 menunjukkan susunan kabel lilitan *unipolar*. Untuk menggunakan *bifilar* sebagai motor *bipolar*, 2 kabel dari masing-masing lilitan disambungkan secara paralel atau seri. Lilitan 2 pada Gambar 2.6 menunjukkan hubungan paralel, hubungan paralel memungkinkan beroperasi dengan arus tinggi, dan hubungan seri memungkinkan beroperasi dengan tegangan tinggi.

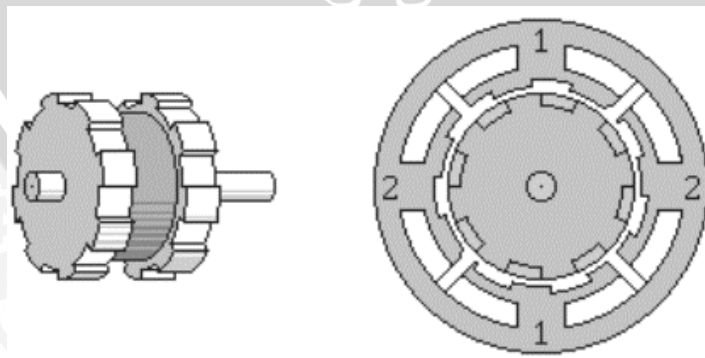


Gambar 2.6 Stepper Motor Bifilar.

Sumber : Douglas (2004:4).

2.2.1.5 Motor Hybrid

Motor *hybrid* memiliki prinsip kerja seperti kedua *stepper motor* magnet permanen dan *variable reluctance*. Rotor untuk *stepper motor hybrid* memiliki banyak gigi. Sama seperti motor *variable reluctance*, dan mengandung magnet konsentris aksial disekitar poros. Gigi dari rotor menyediakan jalur yang membantu memandu fluks magnet untuk ke lokasi yang dipilih pada celah udara. Magnet konsentris meningkatkan karakteristik torsi dinamis dari motor ketika di bandingkan dengan kedua jenis *variable reluctance* dan magnet permanen. *Stepper motor hybrid* dipicu seperti motor *unipolar* dan *bipolar*.



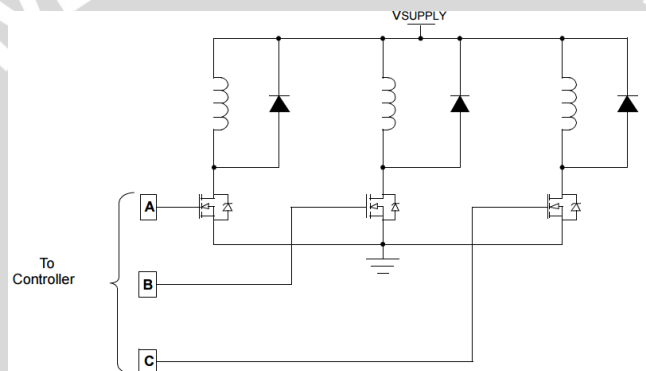
Gambar 2.7 Stepper Motor Hybrid.

Sumber : Douglas (2004:4).

2.2.2 Rangkaian Dasar Pengontrolan

2.2.2.1 Variable Reluctance

Motor *variable reluctance* mempunyai lilitan yang banyak, biasanya 3 sampai 5, yang semuanya terikat bersama-sama pada satu ujung. Lilitan di aktifkan satu per satu dalam urutan tertentu untuk memutar motor. Gambar 2.8 menunjukkan rangkaian dasar untuk memicu motor *variable reluctance*. Dengan catatan dioda dipasang diseluruh lilitan. Seperti beban induktif yan lain, karena tegangan diaktifkan melintasi lilitan, arus dalam lilitan mengalir. Ketika MOSFET di non aktifkan akan terjadi lonjakan tegangan yang dapat mematikan transistor. Dioda melindungi MOSFET dari lonjakan tegangan dengan asumsi dioda berukuran mencukupi kriteria.

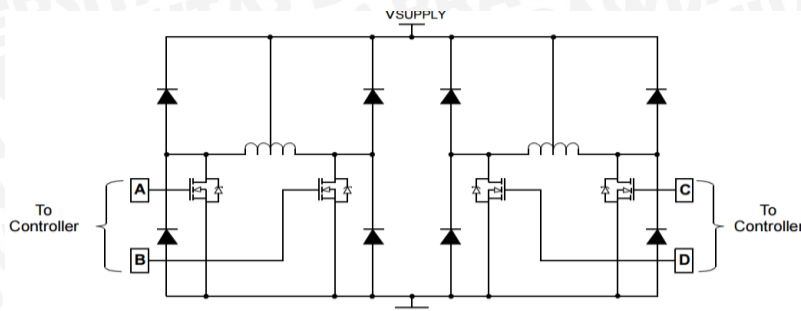


Gambar 2.8 Rangkaian Kontrol Motor *Variable Reluctance*.

Sumber : Douglas (2004:8).

2.2.2.2 Unipolar

Rangkaian kontrol dasar untuk motor unipolar, di tunjukkan pada Gambar 2.9, mirip dengan motor *variable reluctance*. Dengan catatan disetiap MOSFET di tambahkan dioda ekstra. Diperlukan karena induktor adalah *center tap* pada motor unipolar. Ketika satu dari lilitan di *pull down*, lilitan yang lain akan naik dan sebaliknya. Dioda ini mencegah tegangan jatuh dari seluruh MOSFET. Beberapa MOSFET mempunyai dioda integral yang memungkinkan arus balik mengalir tanpa hambatan, terlepas dari tegangan gerbang. Dioda harus mempunyai kapasitas arus yang mencukupi untuk menampung seluruh arus motor, dioda yang bawah pada Gambar 2.9 bisa di hilangkan. Semua dioda harus memiliki kecepatan *switch* yang sebanding dengan kecepatan transistor.



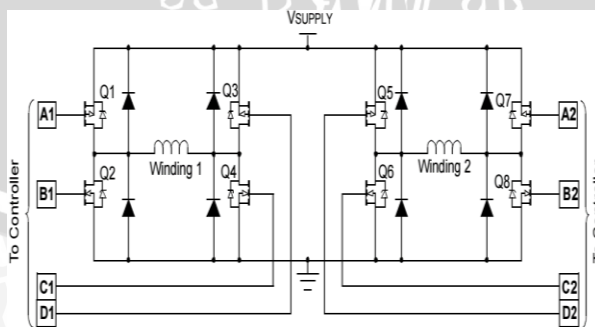
Gambar 2.9 Rangkaian Kontrol Motor Unipolar.

Sumber : Douglas (2004:8).

2.2.2.3 Bipolar

Rangkaian dasar untuk memicu lilitan dari motor bipolar adalah *H-bridge*, ditunjukkan pada Gambar 2.10. Sebuah *H-Bridge* bisa di atur untuk memungkinkan arus mengalir keseluruhan lilitan. Berdasarkan pada Gambar 2.9 arus akan mengalir dari kiri ke kanan lilitan 1, ketika MOSFET Q1 dan Q4 di aktifkan saat Q2 dan Q3 tidak aktif. Arus akan mengalir dari kanan ke kiri ketika Q2 dan Q3 aktif saat Q1 dan Q4 tidak aktif.

H-Bridge memiliki satu bahaya yang melekat yang harus disebutkan. Dalam situasi transistor pada sisi yang sama dari jembatan diaktifkan pada waktu yang sama. Hal ini akan menyebabkan *short circuit* yang merusak rangkaian. Perhatian yang khusus harus di buat untuk membuat pergantian MOSFET dari keadaan ON ke OFF secara bergantian. Dioda pada posisi paralel dimana melindungi setiap MOSFET dari lonjakan tegangan yang di sebabkan oleh *switching* dari induktor. Dioda – dioda ini harus memiliki kapasitas yang mencukupi agar tidak merusak dioda itu sendiri.



Gambar 2.10 Rangkaian Kontrol Motor Bipolar.

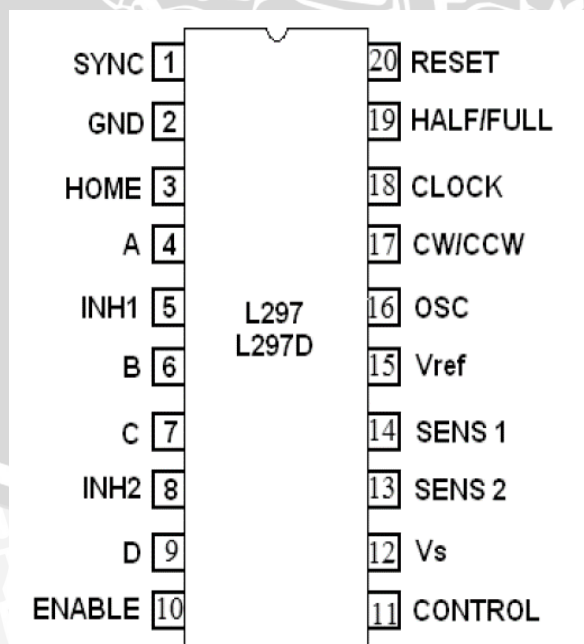
Sumber : Douglas (2004:9).

2.3 Driver L297

L297 adalah IC pengendali *stepper motor* yang menghasilkan 4 sinyal *drive* untuk *stepper motor* bipolar 2 *phase* dan *stepper motor* unipolar 4 *phase* yang dikendalikan oleh mikrokontroler. Motor bisa di *drive* dalam mode *half step*, normal, dan *wave drive*. Rangkaian PWM *chopper* yang terpasang di dalam *chip* memungkinkan mode *switch* pengontrol arus pada lilitan. Fitur dari perangkat ini adalah hanya membutuhkan sinyal *clock*, *direction*, dan masukan mode, karena fasanya telah dihasilkan secara internal sehingga beban mikroprosesor berkurang drastis. Tabel 2.1 menunjukkan *absolute maximum rating* dari L297. Gambar 2.11 menunjukkan *pin* konfigurasi dari L297. Penjelasan konfigurasi pin IC L297 ditunjukkan dalam Tabel 2.1.

Tabel 2.1 *Absolute Maximum Rating* dari L297

Symbol	Parameter	Value	Unit
V_s	Supply Voltage	10	V
V_i	Input signals	7	V
P_{tot}	Total power dissipation ($T_{amb}=70^{\circ}C$)	1	W
T_{stg}, T_j	Storage and junction temperature	-40 to + 150	$^{\circ}C$



Gambar 2.11 *Pin* Konfigurasi dari L297.

Sumber : ST (1996:1).

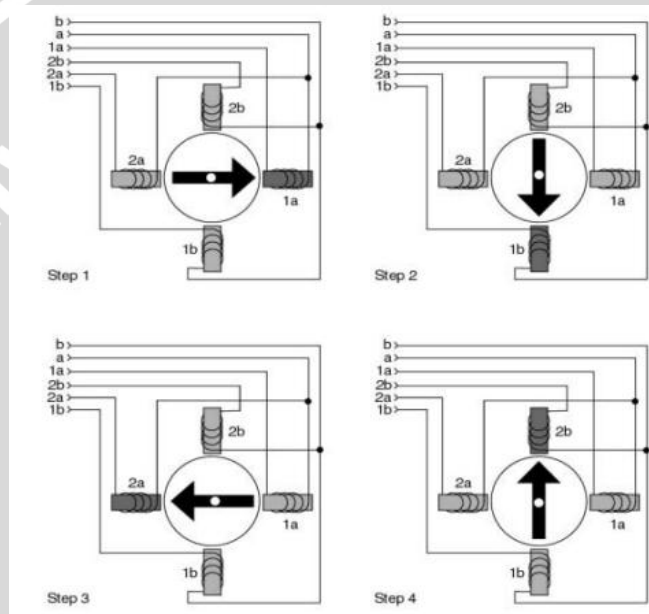
Tabel 2.2 Penjelasan Konfigurasi *Pin* IC L297

Nama Pin	Fungsi
SYNC	<i>Output</i> dari oscillator internal di dalam L297. Jika digunakan lebih dari satu L297 maka pin SYNC akan dihubungkan dengan pin SYNC pada IC L297 yang lain
GND	<i>Ground</i>
HOME	Pin ini akan mengeluarkan logika <i>high</i> jika motor <i>stepper</i> sudah kembali ke posisi semula
A	Sinyal <i>drive</i> fasa A
INH1	Kontrol <i>inhibit</i> bersifat aktif <i>low</i> untuk fasa motor A dan B
B	Sinyal <i>drive</i> fasa B
C	Sinyal <i>drive</i> fasa C
INH2	Kontrol <i>inhibit</i> bersifat aktif <i>low</i> untuk fasa motor C dan D
D	Sinyal <i>drive</i> fasa D
ENABLE	Jika pin ini mendapatkan logika <i>low</i> maka INH1, INH2, A, B, C, D juga akan <i>low</i>
CONTROL	<i>Input</i> kontrol yang menentukan aksi pemotong
Vs	Tegangan <i>Supply</i> sebesar 5 V
SENS2	<i>Input</i> untuk <i>me-load</i> tegangan saat ini dari fasa motor C dan D
SENS1	<i>Input</i> untuk <i>me-load</i> tegangan saat ini dari fasa motor A dan B
Vref	Tegangan referensi untuk rangkaian pemotong
OSC	Rangkaian RC dengan menghubungkan hambatan ke Vcc dan kapasitor ke <i>ground</i> terhubung pada pin ini untuk menentukan nilai dari pemotong
CW/CCW	<i>Input</i> kontrol arah pergerakan <i>stepper motor</i> . Jika <i>high</i> maka searah jarum jam dan jika <i>low</i> akan berlawanan jarum jam
CLOCK	<i>Clock</i> untuk setiap pergerakan <i>stepper motor</i> . Step terjadi ketika pulsa berubah dari <i>low</i> menjadi <i>high</i>
HALF/FULL	Jika logika <i>high</i> maka akan dipilih mode <i>half step</i> , dan bila logika <i>low</i> akan dipilih mode <i>full step</i>

2.3.1 Urutan Fasa Motor Driver

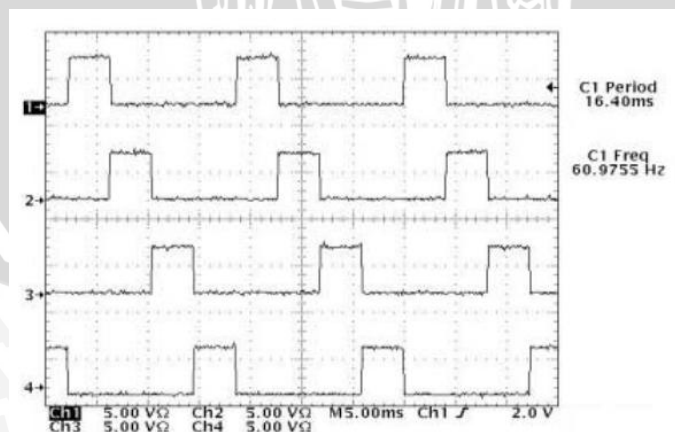
2.3.1.1 Mode Full Step (One Phase On Mode)

Dalam mode *one phase on*, kumparan yang berurutan di catu agar berputar. Mode *one phase* menghasilkan perputaran yang halus dan konsumsi daya yang rendah di bandingkan mode yang lain. Gambar 2.12 dan 2.13 menunjukkan Gambar pergerakan dan gelombang *stepper motor* unipolar dengan mode *one phase on* secara berurutan. Berdasarkan Gambar 2.12 dan 2.13 didapatkan Tabel logika dengan mode *one phase on* yang ditunjukkan pada Tabel 2.3.



Gambar 2.12 Pergerakan *Stepper Motor* Unipolar dengan Mode *One Phase On*.

Sumber : Atmel (2002:4).



Gambar 2.13 Gelombang *Stepper Motor* Unipolar dengan Mode *One Phase On*.

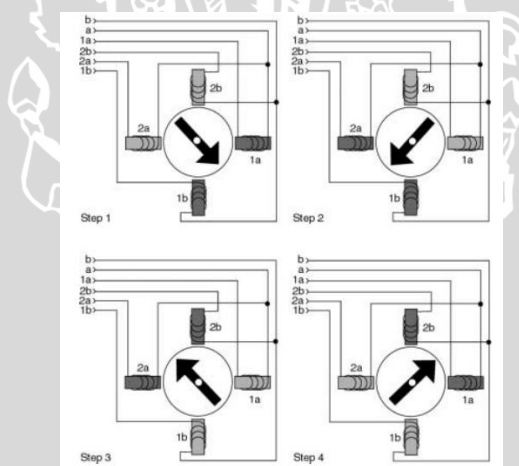
Sumber : Atmel (2002:4).

Tabel 2.3 Logika dari Mode *One Phase On*

Step	1a	1b	2a	2b
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

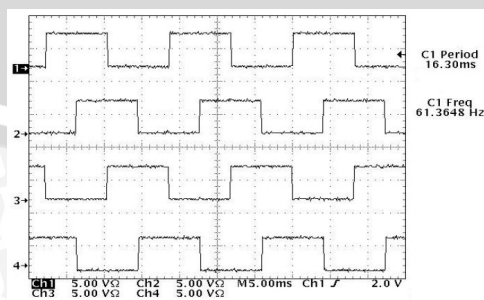
2.3.1.2 Mode *Full Step (Two phases On)*

Pada mode *two phases* sepasang kumparan berurutan di catu secara bergantian agar berputar. Pergerakannya tidak selembut mode *one phase on*, konsumsi daya lebih besar tetapi torsi lebih besar. Gambar 2.14 dan 2.15 menunjukkan gambar pergerakan dan gelombang *stepper motor* unipolar dengan mode *two phases on* secara berurutan. Berdasarkan Gambar 2.14 dan 2.15 didapatkan Tabel logika dengan mode *Two phases on* yang ditunjukkan pada Tabel 2.4.



Gambar 2.14 Pergerakan *Stepper Motor Unipolar* dengan Mode *Two Phases On*

Sumber : Atmel (2002:5).



Gambar 2.15 Gelombang *Stepper Motor Unipolar* dengan Mode *Two Phases On*

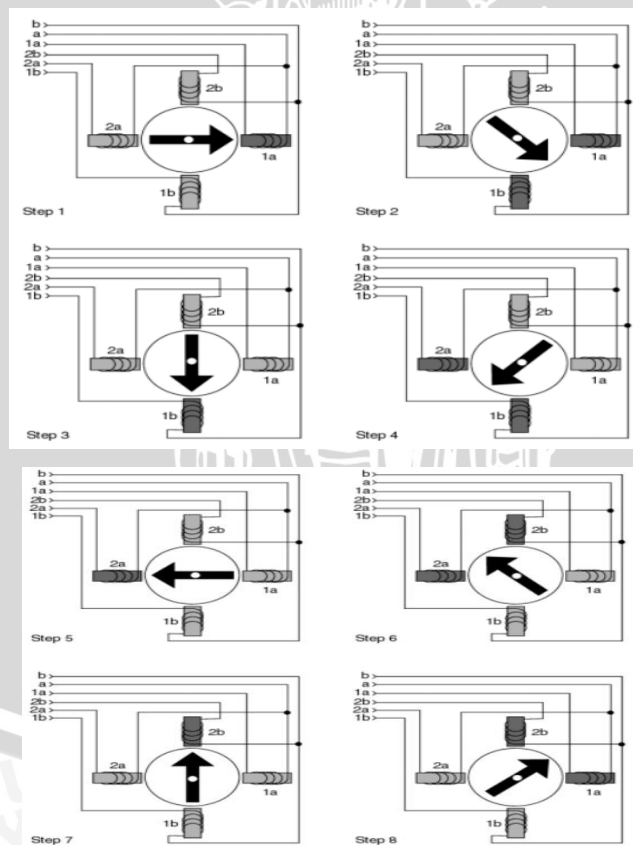
Sumber : Atmel (2002:5).

Tabel 2.4 Logika dari Mode *Two Phases On*.

Step	1a	1b	2a	2b
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

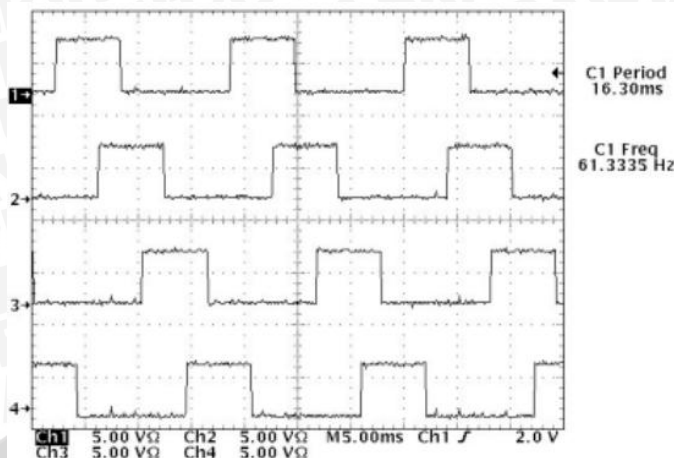
2.3.1.3 Mode *Half Step*

Logika *half step* adalah campuran dari logika *one phase on* dan *two phases on*. Keuntungan utamanya adalah mode ini meningkatkan jumlah step 2 kali lipat dari mode yang lain. Gambar 2.16 dan 2.17 menunjukkan Gambar pergerakan dan gelombang *stepper motor* unipolar dengan mode *half step* secara berurutan. Berdasarkan Gambar 2.16 dan 2.17 didapatkan Tabel logika dengan mode *half step* yang ditunjukkan pada Tabel 2.5.



Gambar 2.16 Pergerakan *Stepper Motor Unipolar* dengan Mode *Half step*

Sumber : Atmel (2002:7).



Gambar 2.17 Gelombang *Stepper Motor Unipolar* dengan Mode *Half Step*

Sumber : Atmel (2002 : 8).

Tabel 2. 5 Logika dari Mode *Half Step*

Step	1a	1b	2a	2b
1	1	0	0	1
2	1	0	0	0
3	1	1	0	0
4	0	1	0	0
5	0	1	1	0
6	0	0	1	0
7	0	0	1	1
8	0	0	0	1

2.4 Sensor *Rotary Encoder* Autonics E40H8-500P/R-6-L-5

Sensor *rotary encoder* yang digunakan pada skripsi ini merupakan produksi dari Autonics dengan tipe E40H8-500P/R-6-L-5 seperti dalam Gambar 2.18. Sensor ini memiliki lubang dengan diameter 80 mm sebagai tempat untuk mengkopel antara sensor dengan *stepper motor*. Keluaran dari sensor ini berupa pulsa dengan tegangan $5V \pm 5\%$ saat pulsa high dan 0 V saat pulsa low. Sensor ini dapat menyediakan 500 pulsa tiap 1 kali rotasi. Pulsa keluaran dari sensor ini dapat diartikan menjadi gerakan, posisi, dan arah sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder*. Sensor ini mempunyai 9 kaki atau pin. Konfigurasi dari masing-masing pin ditunjukkan dalam Tabel 2.6. Perbedaan antara masing – masing keluaran sensor *rotary encoder* dapat di lihat pada Gambar 2.19. Dalam Gambar

2.19 diketahui bahwa terdapat pergeseran fasa antara OUT A dan OUT B dengan periode pulsa yang sama, sedangkan OUT Z memiliki periode 2 kali lebih lama. Fasa yang berkebalikan juga disediakan oleh *rotary encoder* ini pada OUT A', B' dan Z'.

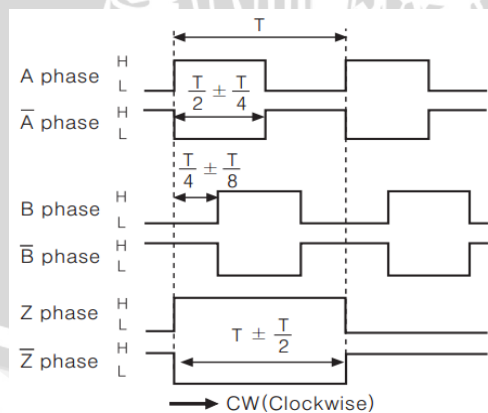


Gambar 2.18 *Rotary Encoder Autonics*.

Sumber : autonics (2013:1)

Tabel 2. 6 Konfigurasi *Pin Rotary Encoder*

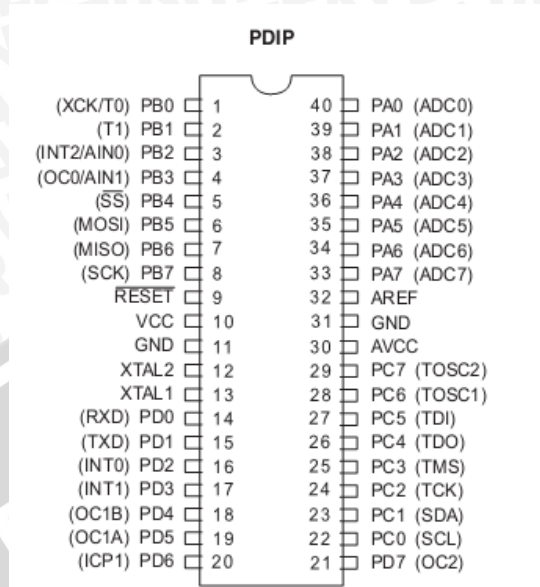
NO PIN	FUNGSI	WARNA KABEL
1	OUT A	HITAM
2	OUT A'	MERAH
3	+V	COKLAT
4	GND	BIRU
5	OUT B	PUTIH
6	OUT B'	ABU-ABU
7	OUT Z	JINGGA
8	OUT Z'	KUNING
9	F.G	PELINDUNG KABEL



Gambar 2.19 *Line Driver Output Sensor Rotary Encoder*

Sumber : autonics (2013:2).

2.5 Mikrokontroler Atmega 16



Gambar 2.20 Konfigurasi *Pin* Atmega 16.

Sumber: Atmel (2007)

Konfigurasi *pin* Atmega 16 dengan kemasan 40 *pin* DIP (*Dual Inline Package*) dapat dilihat pada Gambar 2.20. dari Gambar diatas dapat dijelaskan fungsi dari masing-masing *pin* ATMEGA 16 sebagai berikut:

1. VCC merupakan *pin* yang berfungsi sebagai masukan catu daya.
2. GND merupakan *pin* *Ground*.
3. *Port* A (PA0...PA7) merupakan *pin* *input/output* dua arah dan *pin* masukan ADC.
4. *Port* B (PB0...PB7) merupakan *pin* *input/output* dua arah dan *pin*

Tabel 2. 7 Fungsi Khusus *Port B*

<i>Pin</i>	Fungsi Khusus
PB7	SCK (<i>SPI Bus Serial Clock</i>)
PB6	MISO (<i>SPI Bus Master Input/Slave Output</i>)
PB5	MOSI (<i>SPI Bus Master Output/Slave Input</i>)
PB4	\overline{SS} (<i>SPI Slave Select Input</i>)
PB3	AIN1 (<i>Analaog Comparator Negative Input</i>)
	OC0 (<i>Timer/Counter0 Output Compare Match</i>)

Output)

PB2 AIN0 (*Analog Comparator Positive Input*)

INT2 (*External Interrupt 2 Input*)

PB1 T1 (*Timer/Counter1 External Counter Input*)

PB0 T0 T1 (*Timer/Counter0 External Counter Input*)

XCX (*USART External Clock Input/Output*)

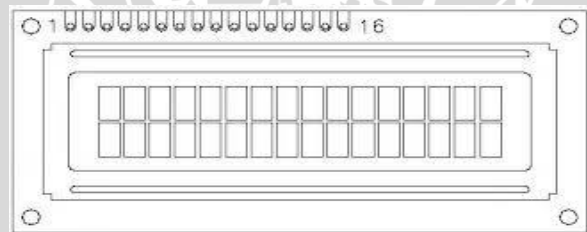
2.6 LCD (Liquid Crystal Display)

LMB162A adalah modul LCD (*Liquid Crystal Display*) *matrix* dengan konfigurasi 16 karakter dan 2 baris dengan setiap karakternya ditunjukkan dalam Gambar 2.21. *Liquid Crystal Display* atau biasa disebut LCD merupakan alat tampilan yang biasa digunakan untuk menampilkan karakter ASCII sederhana, dan Gambar-Gambar pada alat-alat digital seperti jam tangan, kalkulator dan lain-lain. Deskripsi sederhana cara kerja dari sebuah LCD *matrix* adalah sebuah *Twisted Nematic* (TN) *Liquid Crystal Display*, yang terdiri dari 2 material yang terpolarisasi, 2 buah kaca, sebuah bentuk elemen elektroda untuk menentukan pixel, dan *Integrated Circuit* (IC) untuk mengalamatkan baris dan kolom.

Perlu diketahui fungsi dari setiap kaki yang ada pada komponen tersebut.

- 1) Kaki 1 (GND) : Kaki ini berhubungan dengan tegangan 0 volt (*Ground*).
- 2) Kaki 2 (VCC) : Kaki ini berhubungan dengan tegangan +5 Volt yang merupakan tegangan untuk sumber daya.
- 3) Kaki 3 (VEE/VLCD) : Tegangan pengatur kontras LCD, kaki ini terhubung pada *output* variabel resistor. Kontras mencapai nilai maksimum pada saat kondisi kaki ini pada tegangan 0 volt.
- 4) Kaki 4 (RS) : *Register Select*, kaki pemilih register yang akan diakses. Untuk akses ke Register Data, logika dari kaki ini adalah 1 dan untuk akses ke Register Perintah, logika dari kaki ini adalah 0.
- 5) Kaki 5 (R/W) : Logika 1 pada kaki ini menunjukkan bahwa modul LCD sedang pada mode pembacaan dan

- logika 0 menunjukkan bahwa modul LCD sedang pada mode penulisan. Untuk aplikasi yang tidak memerlukan pembacaan data pada modul LCD, kaki ini dapat dihubungkan langsung ke *Ground*.
- 6) Kaki 6 (E) : *Enable Clock* LCD, kaki mengaktifkan *clock* LCD. Logika 1 pada kaki ini diberikan pada saat penulisan atau pembacaan data.
- 7) Kaki 7 – 14 (D0 – D7) : Data bus, kedelapan kaki LCD ini adalah tempat aliran data sebanyak 4 bit ataupun 8 bit mengalir saat proses penulisan maupun pembacaan data.
- 8) Kaki 15 (Anoda) : Berfungsi untuk tegangan positif dari *backlight* LCD sekitar 4,5 volt (hanya terdapat untuk LCD yang memiliki back light).
- 9) Kaki 16 (Katoda) : Tegangan negatif *back light* LCD sebesar 0 volt (hanya terdapat pada LCD yang memiliki *backlight*).



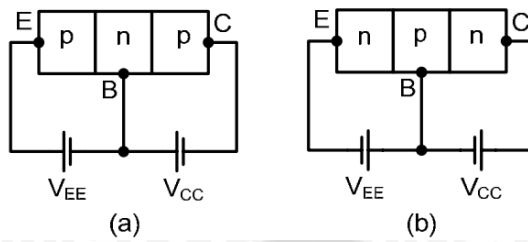
Gambar 2.21 LCD (*Liquid Crystal Display*) LM162A

Sumber: Baskara (2013:2).

2.7 Bipolar Junction Transistor

2.7.1 Konstruksi dari Transistor

Transistor merupakan piranti yang terdiri atas tiga lapisan semikonduktor, yaitu 2 buah lapisan semikonduktor tipe $-p$ dan sebuah lapisan semi konduktor tipe $-n$, atau sebaliknya. Jenis pertama dikenal sebagai transistor tipe PNP, sedang yang kedua dikenal dengan transistor NPN. Ketiga terminal yang terhubung ke semikonduktor dikenal dengan kolektor (C), basis (B) , emitter (E). seperti yang dapat dilihat seperti Gambar 2.22.

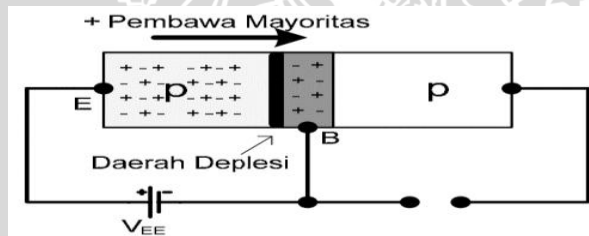


Gambar 2.22 Transistor (a) PNP (b) NPN

Sumber : Irianto (2005:1).

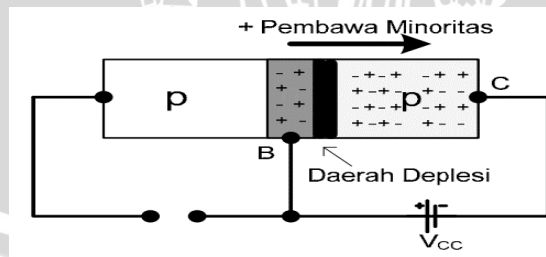
2.7.2 Operasi dari transistor

Pada Gambar 2.23 transistor PNP diGambarkan kembali tanpa tegangan bias pada basis – kolektor. Daerah deplesi mengecil karena adanya pembiasan, akibatnya terjadi aliran arus pembawa yang besar (*Majority Carrier/ Pembawa Mayoritas*) dari lapisan p ke n. Jika bias pada basis-emitter dihilangkan dan dipasang pada basis-kolektor, maka pembawa mayoritas akan hilang dan yang ada hanyalah pembawa minoritas. Seperti yang dilihat pada Gambar 2.24.



Gambar 2.23 *Forward Bias* pada Salah Satu *Junction* Transistor dengan Pembawa Mayoritas

Sumber : Irianto (2005:2).



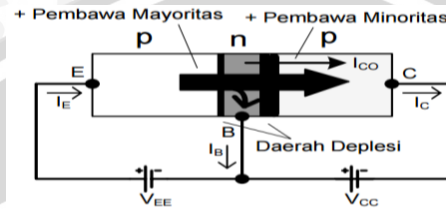
Gambar 2.24 *Forward Bias* pada Salah Satu *Junction* Transistor dengan Pembawa Minoritas

Sumber : Irianto (2005:2).

Jika kedua tegangan bias dipasang seperti Gambar 2.25 berikut, maka semua arus pembawa (pembawa mayoritas dan pembawa minoritas) akan muncul dan



melintasi daerah sambungan (*junction*) dari transistor. Lapisan n sangat tipis dan mempunyai konduktivitas rendah, maka hanya akan ada sebagian kecil dari pembawa mayoritas yang keluar melalui terminal basis (biasanya orde mikro). Sebagian besar pembawa mayoritas akan langsung terdifusi melewati *junction* yang terbias *reverse* ke dalam material tipe p yang terhubung ke terminal kolektor.



Gambar 2.25 Aliran Pembawa Mayoritas dan Minoritas Pada Transistor PNP.

Sumber : Irianto (2005:2).



BAB III

METODOLOGI PENELITIAN

3.1 Penentuan Spesifikasi *Slider Robot*

Spesifikasi *slider* robot secara umum ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi *slider* robot yang direncanakan adalah sebagai berikut:

1. Dimensi mekanik keseluruhan panjang 30 cm, lebar 35 cm, tinggi 65 cm.
2. *Slider* robot bergerak dari 1- 40 cm.
3. *Keypad* sebagai setpoint jarak pergeseran *slider*.
4. *Stepper motor* 1,8°.
5. *Driver* motor menggunakan L297.
6. Sensor *rotary encoder* dikopel terhadap *stepper motor*.

3.2 Perancangan Sistem

3.2.1 Perancangan Mekanik

Perancangan mekanik menggunakan referensi dari robot KRAI 2014. *Slider* dibuat menggunakan alumunium berukuran 1,2 x 1,2 cm. *Slider* dibuat 2 buah , kiri dan kanan. *Stepper motor* dipasang di bagian bawah *slider*, posisi *stepper motor* dibuat seimbang , berada ditengah antara *slider* kiri dan kanan. *Stepper motor* menggunakan *spur gear* dengan diameter 2,8 cm , dikopel ke *slider* menggunakan rantai sepeda dengan panjang 1 m. *rotary encoder* dikopel sesumbu terhadap *stepper motor*.

3.2.2 Perancangan dan Pembuatan Perangkat Keras (*Hardware*)

Perancangan perangkat keras pertama dilakukan dengan menyusun blok diagram perangkat keras kemudian melakukan perancangan dan analisis perhitungan elektrik rangkaian. Layout skematik rangkaian menggunakan aplikasi Eagle 5.11.0. Perancangan perangkat keras robot diantaranya:

1. *Keypad* digunakan untuk input jarak pergerakan *slider* dalam satuan cm.
2. Mikrokontroler yang digunakan adalah ATmega 16 untuk memproses pulsa ke *driver stepper motor*, dan dari *rotary encoder*.
3. LCD 2x16 yang digunakan untuk tampilan pembacaan pergeseran *slider*.

4. *Driver* motor menggunakan IC L297 sebagai kontrol logika motor dan transistor sebagai penguat arus pada *stepper motor*.
5. Motor yang digunakan adalah *stepper motor* tipe unipolar dengan sudut langkah 1,8° per langkah.

3.2.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak berupa *flowchart* (diagram alir) bahasa pemrograman yang dipakai untuk atmega16. Bahasa pemrograman yang digunakan adalah bahasa C yang nantinya dibuat dan di *compile* menggunakan *software* codevision AVR. Program dimulai dari membaca *input keypad* dengan masukan dalam satuan milimeter. Mikrokontroler memproses input jarak dan mengolah jarak menjadi pulsa untuk mengaktifkan *stepper motor*. *Rotary encoder* membaca pergerakan *slider*. pembacaan *rotary* ditampilkan oleh mikrokontroler ke LCD 2x16.

1. Pengujian *Slider* Robot

Pengujian *slider* robot dilakukan untuk memastikan agar sistem berjalan sesuai dengan perancangan. Pengujian *slider* robot meliputi pengujian perangkat keras dan perangkat lunak.

2. Pengujian Catu Rangkaian

Pengujian catu rangkaian bertujuan untuk melihat level tegangan dari rangkaian catu. Pengujian dilakukan dengan cara rangkaian catu di sambungkan ke sumber tegangan berupa baterai 12 V, setelah rangkaian catu dalam keadaan *ON* keluaran dari *regulator* 5 V di sambungkan ke *oscilloscope* untuk dilihat level tegangannya.

3. Pengujian Rangkaian *Stepper Motor Driver*

Pengujian rangkaian *stepper motor driver* di bagi menjadi 2 bagian yaitu pengujian arus dan pengujian mode *stepper motor*. Pengujian arus bertujuan untuk mengetahui apakah hasil pengujian sesuai dengan perancangan. Pengujian mode menggunakan rangkaian yang terdiri dari mikrokontroler, *stepper motor driver* dan *oscilloscope*. Pengujian ini dilakukan untuk melihat bentuk gelombang keluaran dari mikrokontroler dan dari *driver* dalam mode *drive stepper motor* yaitu *full mode* dan *half mode*. Bentuk gelombang akan di analisa dan di sesuaikan dengan yang ada di datasheet.

4. Pengujian Rangkaian *Stepper Motor*

Pengujian stepper motor dibagi menjadi dua pengujian diantaranya pengujian sudut motor dan pengujian jarak pergeseran *slider* saat motor bergerak. Pengujian sudut putaran motor dilakukan untuk mengetahui perbandingan antara jumlah pulsa dengan sudut saat motor bergerak. Pengujian sudut putaran *stepper motor* dilakukan dengan menggunakan penggaris busur untuk mengetahui sudut putaran motor. Pengambilan data sudut putaran motor dilakukan sebanyak tiga kali dalam setiap pemberian jumlah pulsa. Pengujian pergeseran *slider* saat motor bergerak dilakukan untuk mengetahui resolusi pulsa dengan jarak pergeseran *slider*. Pengujian dilakukan dengan memberikan pulsa ke *stepper motor driver* dengan *pulse generator* dari atmega16 mulai dari 10 – 200 dengan range tiap pengujian sebesar 10 pulsa kemudian sticker penggaris di pasang pada aluminium agar dapat membaca pergeseran *slider*. Pengujian dilakukan sebanyak 3 kali.

5. Pengujian Rangkaian *Rotary Encoder*

Pengujian *rotary encoder* dilakukan dua pengujian diantaranya pengujian sudut putaran *rotary encoder* dan pengujian jarak pergeseran *slider* saat *rotary encoder* bergerak. Pengujian sudut putaran *rotary encoder* dilakukan untuk mengetahui perbandingan antara jumlah pulsa dengan sudut saat *rotary encoder* bergerak. Pengujian *rotary encoder* dilakukan dengan menggunakan penggaris busur untuk mengetahui sudut putaran motor dimana *rotary encoder* diputar tiap 10 derajat sampai 360 derajat dimana pulsa keluaran *rotary encoder* dibaca menggunakan modul *serial* dan di tampilkan di PC.

Pengujian pergeseran *slider* dengan *rotary encoder* dilakukan untuk mengetahui resolusi pulsa rotary dengan jarak pergeseran *slider*. Pengujian dilakukan dengan memberikan menggeser *slider* tiap 1 cm sampai 40 cm. *Rotary encoder* akan membaca pergeseran *slider* dengan mengeluarkan pulsa yang di baca oleh modul serial dan di tampilkan ke PC.

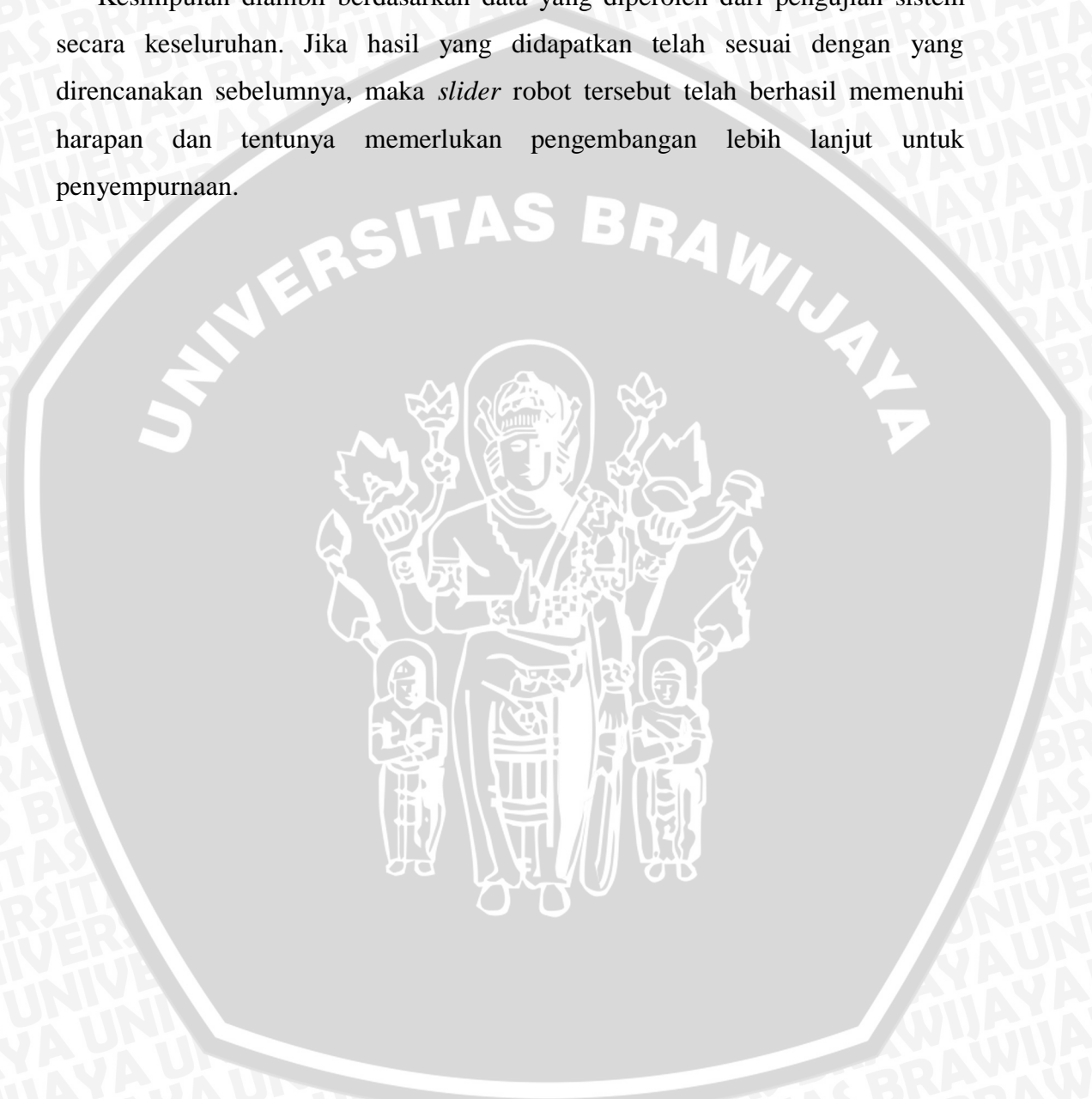
6. Pengujian Sistem Secara Keseluruhan

Pengujian ini dilakukan dengan cara menggabungkan semua bagian *slider* robot yang dibuat dan melihat kinerja *slider* robot baik perangkat keras maupun perangkat lunak. Pengujian ini bertujuan untuk mengetahui kinerja *slider* robot yang dibuat dan memberikan analisa terhadap kinerja *slider* robot. Pengujian

dilakukan mulai dari pemberian masukan dari *keypad* , *slider* akan bergerak sesuai jarak yang dimasukkan pada *keypad*. Pengambilan data dilakukan dengan menggunakan *sticker* penggaris pada aluminium dan tampilan LCD.

7. Pengambilan Kesimpulan

Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Jika hasil yang didapatkan telah sesuai dengan yang direncanakan sebelumnya, maka *slider* robot tersebut telah berhasil memenuhi harapan dan tentunya memerlukan pengembangan lebih lanjut untuk penyempurnaan.

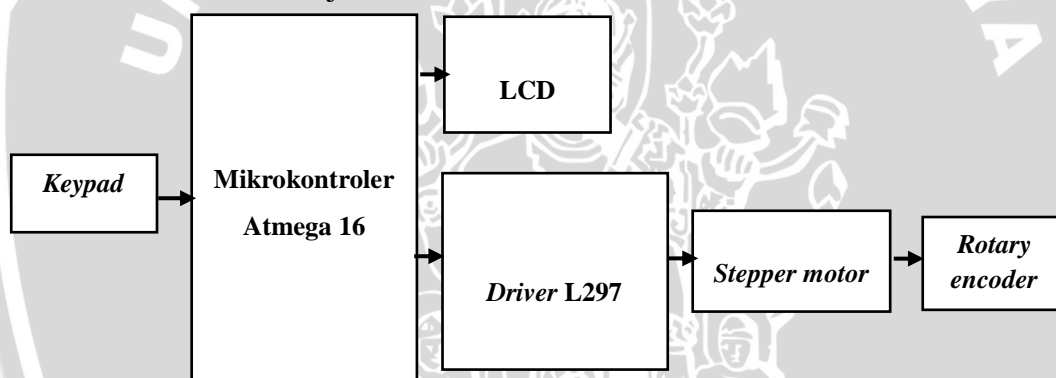


BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

4.1 Perancangan Sistem Robot

Awalnya mikrokontroler Atmega 16 akan mengolah *input* dari *keypad* yang berupa jarak pergeseran *slider* dalam satuan centimeter. Mikrokontroler Atmega 16 kemudian akan menghitung jumlah pulsa (CLK) yang akan di berikan kepada *stepper motor driver* L297 kemudian L297 akan menerjemahkan pulsa CLK menjadi 4 buah pulsa A, B, C dan D. 4 pulsa ini diteruskan ke *bipolar junction transistor* untuk dikuatkan arusnya, kemudian *stepper motor* ini akan bergerak sesuai dengan jumlah pulsa yang diberikan. Perputaran dari *stepper motor* akan di baca oleh *rotary encoder* dan di tampilkan ke LCD. Perancangan sistem keseluruhan robot ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram Blok Perancangan Sistem Secara Keseluruhan.

4.2 Perancangan Mekanik *Slider* Robot

Sistem mekanik yang baik berpengaruh besar pada pergerakan robot, oleh karena itu perancangan mekanik dalam hal ini rangka robot haruslah dibuat sepresisi mungkin. Perancangan dan pembuatan *slider* robot menggunakan *chasis* dengan bahan alumunium pada tiang utama dengan ukuran 25mm x 25mm. *Slider* (bagian yang bergerak) terbuat dari alumunium berukuran 12mm x 12 mm. *Slider* dibuat 2 buah kiri dan kanan. *Stepper motor* dipasang pada bagian bawah dimana *gear* motor tepat berada di tengah dari *slider* agar memudahkan motor menarik *slider*. Antara *stepper motor* dan *slider* di hubungkan menggunakan rantai dengan panjang 1 m. *Rotary encoder* dipasang se sumbu dengan as motor. Mekanik *slider* robot ditunjukkan dalam Gambar 4.2



(a)



(b)

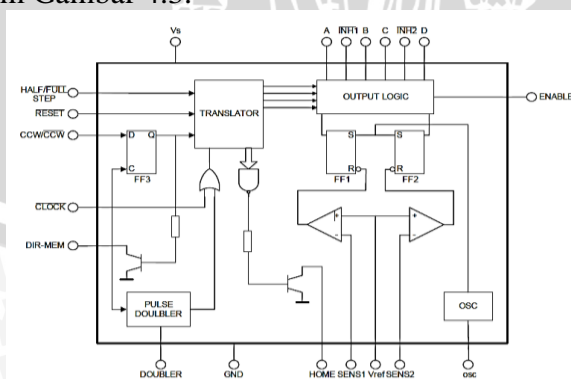
Gambar 4.2 Slider (a) Rancangan Slider Robot Tampak Depan, (b) Rancangan Slider Robot Tampak Perspektif.

4.3 Perancangan Perangkat Keras

4.3.1 Perancangan Rangkaian Stepper Motor Driver

Pada perancangan *stepper motor driver* menggunakan IC L297 sebagai kontrol logika motor dan transistor TIP41C sebagai penguat arus *driver*. IC L297 menerima *input* berupa pulsa yang diatur frekuensinya oleh mikrokontroler dan arah putaran motor yang diwakili oleh pin CW/CCW. *Output* dari L297 adalah 4 buah pin logika fasa A, B, C, D. *Output* IC L297 dihubungkan ke transistor TIP41C sebagai penguat arus dimana arus yang dibutuhkan 1.5 Ampere per fasanya.

Rangkaian IC L297 membutuhkan *input* pulsa dari mikrokontroler. Pulsa diatur frekuensinya yang berbanding lurus dengan kecepatan *stepper motor*. *Output* IC L297 berupa logika fasa A, B, C, D. Diagram blok IC L297 ditunjukkan dalam Gambar 4.3.



Gambar 4.3 Diagram Blok IC L297.

Sumber: SGS-Thomson (2001:2)

Berdasarkan Gambar 4.3 dapat dijelaskan bahwa *input* IC L297 yang digunakan dalam perancangan ini adalah pin *direction*, *clock*, dan *half/full*. *Output* yang digunakan pada perancangan adalah A, B, C, D. Pin *clock* tersambung ke *D flip flop* dan *translator* sebagai sumber *clock*. Prinsip kontrol arah putaran pada IC L297 yaitu ketika *input D flip flop* (pin *DIR*) adalah logika *high*, maka *output D flip flop* adalah 1 yang berarti motor berputar ke kanan. Apabila pin *DIR* berlogika 0 maka *output D flip flop* adalah 0 dan motor berputar ke kiri.

Pada blok *translator*, *input* logika adalah pin *half/full*, *clock*, *reset*, dan *DIR*. Dalam perancangan pin *reset* disambung ke *Vcc* karena *reset* aktif logika *low*. Tugas *translator* untuk melakukan kontrol logika pin *output* A, B, C, D dengan prinsip kerja sama dengan rangkaian *register* geser yang menggeser logika dengan kontrol arah pin *DIR* dan kontrol mode *step* pin *half/full*. *Output* dari L297 berupa empat pin logika diantaranya A, B, C, D yang aktif pada logika *high*.

IC L297 mempunyai spesifikasi *input* :

$$V_{IL} = 0,6 \text{ V}$$

$$V_{IH} = 2 \text{ V}$$

$$I_{IL} = 100 \text{ uA}$$

$$I_{IH} = 10 \text{ uA}$$

Output fasa L297 adalah *input* dari transistor TIP41C sebagai penguat arus *driver*. L297 memiliki karakteristik *output* seperti berikut :

$$V_{OL} = 0,4 \text{ V}$$

$$V_{OH} = 5 \text{ V (min. 3,9 V)}$$

$$I_{OL} = 10 \text{ mA}$$

$$I_{OH} = 5 \text{ mA}$$

Tegangan *output* dari L297 mencukupi untuk men *drive* motor , tetapi arus *output* tidak mencukupi dimana arus *output* L297 adalah 5mA-10mA. Berdasarkan *datasheet* dari *stepper motor* PH268-21B karakteristik yang dibutuhkan adalah :

$$I_C = 1,5 \text{ A}$$

$$V_{Motor} = 5,4 \text{ V}$$

Berdasarkan Grafik V_{BE} dan V_{CE} saturasi yang ditunjukkan Gambar 4.4. Pada saat $I_C = 1,5 \text{ A}$, $V_{CE} = 200 \text{ mV} = 0,2 \text{ V}$ sehingga tegangan sumber adalah sebagai berikut :

$$V_S = V_{Motor} + V_{CE} = 5,4 + 0,2 = 5,6 \text{ V}$$

Berdasarkan Gambar 4.5 di ketahui persamaan LOOP 1 adalah sebagai berikut :

$$V_S - I_C \times R_C - V_{CE} = 0$$

$$\beta = I_C / I_B$$

$$\beta = 10 \text{ (Gambar 4.4)}$$

$$I_B = I_C / \beta = I_C / 10 = 0,15 \text{ A}$$

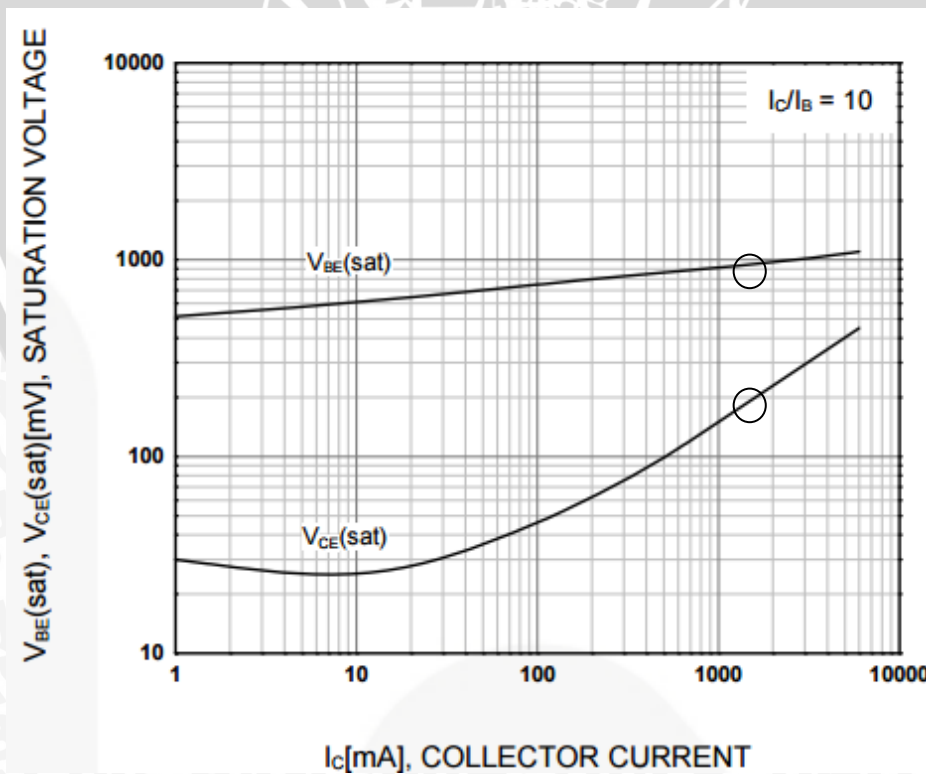
Sedangkan persamaan LOOP 2 adalah sebagai berikut :

$$V_{OH} - I_B \times R_B - V_{BE} = 0$$

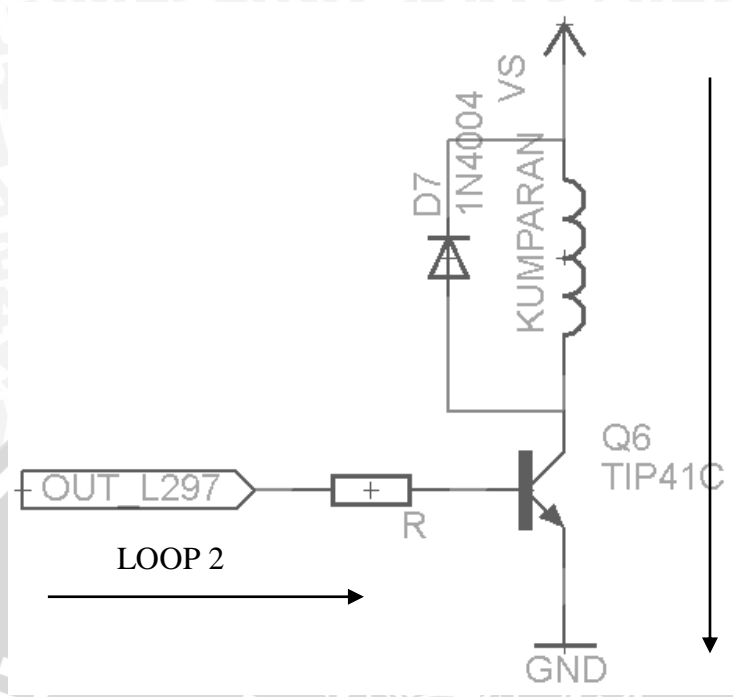
$$\text{Dimana } V_{BE} = 900 \text{ mV} = 0,9 \text{ V (Gambar 4.4)}$$

$$5 - 0,15 \times R - 0,9 = 0$$

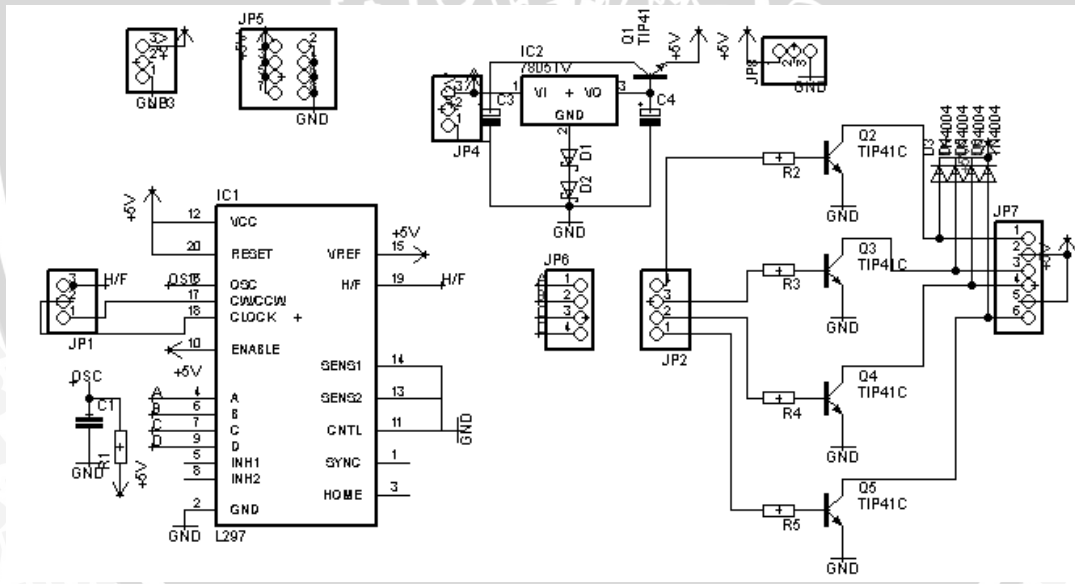
Didapatkan $R = 27,33 \Omega$, didalam perancangan yang digunakan adalah resistor berukuran 27Ω . Gambar 4.5 menunjukkan sebagian rangkaian *stepper motor driver*. Skema rangkaian *stepper motor* keseluruhan ditunjukkan dalam Gambar 4.6.



Gambar 4. 4 Grafik V_{BE} dan V_{CE} Saat Saturasi.



Gambar 4. 5 Sebagian Rangkaian *Stepper Motor Driver*.



Gambar 4. 6 Rangkaian Keseluruhan *Driver Motor*.

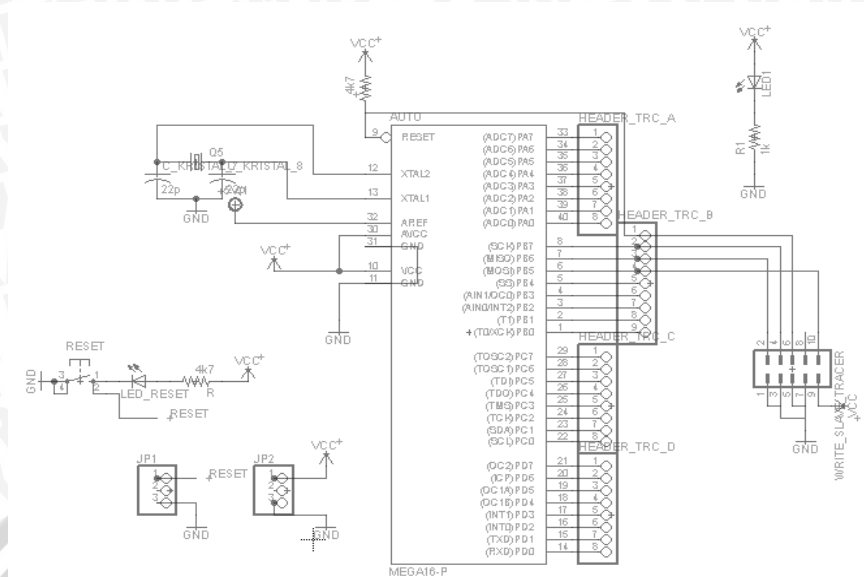
4.3.2 Perancangan *System Minimum Atmega16*

Pada Perancangan Mikrokontroler *slider robot* ini digunakan *sytem minimum atmega16*. Pada perancangan ini skema perancangan mikrokontroler atmega 16 ditunjukkan dalam Gambar 4.7. Mikrokontroler ATmega16 mempunyai 32 jalur I/O yang dapat diprogram menjadi masukan dan keluaran. Ke-32 jalur tersebut

dikelompokkan ke dalam empat *port*, yaitu *port* A, B, C, dan D. Pada perancangan ini pin yang digunakan adalah:

PIN A.0	= digunakan sebagai <i>input keypad</i>
PIN A.1	= digunakan sebagai <i>input keypad</i>
PIN A.2	= digunakan sebagai <i>input keypad</i>
PIN A.4	= digunakan sebagai <i>input keypad</i> .
PIN A.5	= digunakan sebagai <i>input keypad</i>
PIN A.6	= digunakan sebagai <i>input keypad</i>
PIN A.7	= digunakan sebagai <i>input keypad</i>
PIN B.0	= pembangkit pulsa CLK untuk <i>driver</i>
PIN B.1	= logika <i>half / full mode</i> pada <i>driver</i>
PIN B.2	= logika putaran motor <i>clockwise / counter clockwise</i>
PIN B.5	= pin MOSI (untuk komunikasi SPI dengan <i>slave</i>)
PIN B.6	= pin MISO (untuk komunikasi SPI dengan <i>slave</i>)
PIN B.7	= pin SCK (untuk komunikasi SPI dengan <i>slave</i>)
PIN C.0	= digunakan untuk RS pada LCD
PIN C.1	= digunakan untuk RD pada LCD
PIN C.2	= digunakan untuk EN pada LCD
PIN C.4	= digunakan untuk D4 pada LCD
PIN C.5	= digunakan untuk D5 pada LCD
PIN C.6	= digunakan untuk D6 pada LCD
PIN C.7	= digunakan untuk D7 pada LCD
PIN D.0	= Rx untuk transmisi UART
PIN D.1	= Tx untuk transmisi UART
PIN D.2	= <i>channel 1 rotary encoder</i>
PIN D.3	= <i>channel 2 rotary encoder</i>
XTAL1	= digunakan sebagai masukan dari rangkaian osilator kristal
XTAL2	= digunakan sebagai masukan dari rangkaian osilator kristal

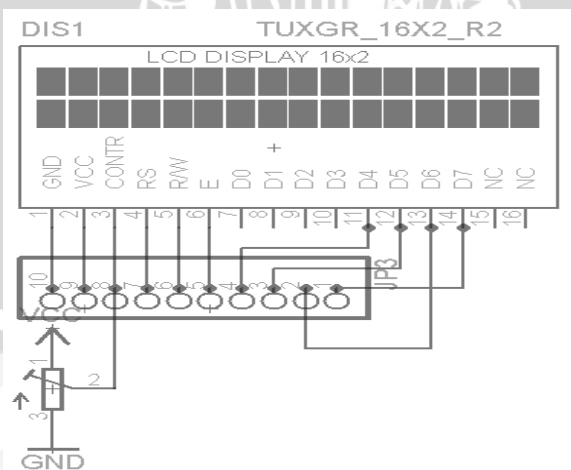




Gambar 4. 7 System Minimum Mikrokontroler Atmega16.

4.3.3 Perancangan Display LCD

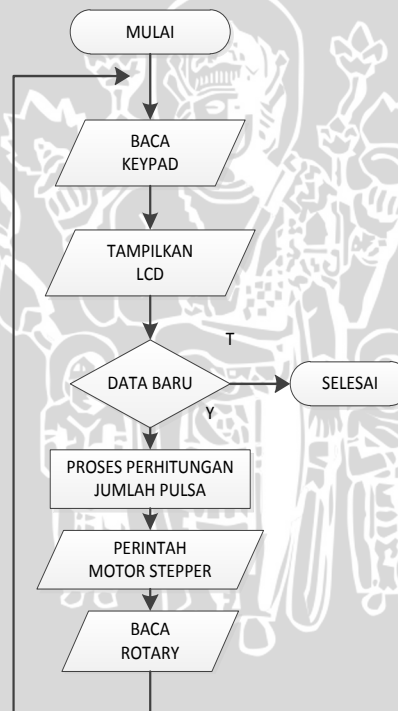
LCD digunakan untuk *display* data pembacaan pergeseran *slider* oleh *rotary* dalam satuan centimeter. Skematik perancangan LCD 2x16 ditunjukkan dalam Gambar 4.8. Berdasarkan Gambar 4.7 data LCD terhubung dengan PINC.4 sampai PINC 7. Sedangkan untuk PINC.0 sampai PINC.2 digunakan untuk RS, R/W, dan Enable. Pin Vee pada LCD dihubungkan dengan *output* variabel resistor yang bernilai 10K Ω dengan dua kaki lainnya dihubungkan dengan Vcc dan GND. Tujuan pemasangan variabel resistor 10 K Ω dilakukan untuk mengubah kontras dengan tegangan yang masuk dalam pin Vee.



Gambar 4. 8 Skematik Perancangan LCD.

4.3.4 Perancangan Perangkat Lunak

Tahapan proses yang terdapat pada sistem ini meliputi, *input keypad* pergeseran *slider* robot melalui pin I/O. Kemudian mode pergerakan diterjemahkan oleh mikrokontroler data satuan atau puluhan. Kemudian mikrokontroler memproses data tersebut menjadi pulsa yang diteruskan ke *driver*. *Driver* mengubah pulsa yang diterima menjadi pulsa kontrol *stepper motor*. *Stepper motor* menggerakkan *slider* dan pergerakan *slider* di pantau oleh *rotary encoder*. Data pembacaan *rotary encoder* di sampaikan ke mikrokontroller untuk di olah dan di tampilkan ke LCD. *Software* mikrokontroler dibuat menggunakan *compiler Code Vision AVR* buatan HP infotech. Menggunakan bahasa pemrograman yaitu bahasa C dan *downloader* menggunakan AVR USB. *Flowchart* program dapat dilihat pada Gambar 4.9.

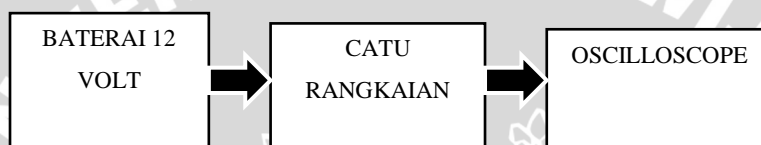


Gambar 4. 9 *Flowchart* Program.

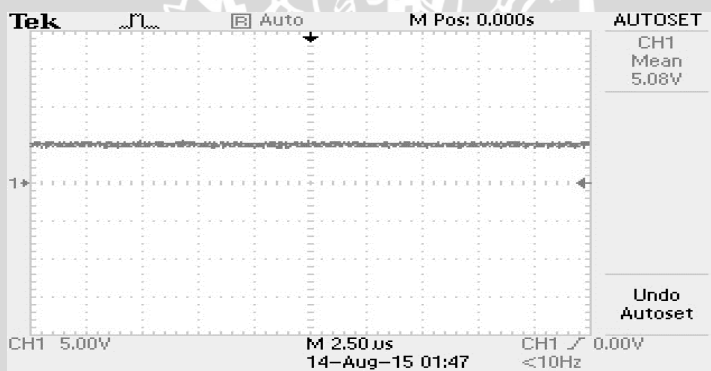
BAB V PENGUJIAN DAN ANALISIS

5.1 Pengujian Catu Rangkaian

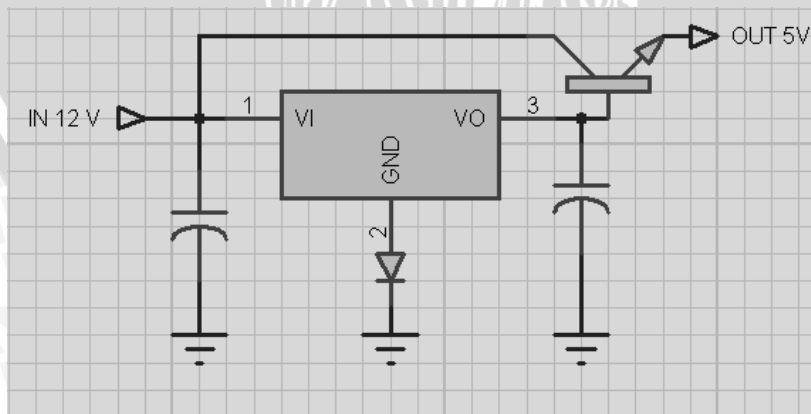
Pada pengujian catu rangkaian ini dilakukan menggunakan *oscilloscope* untuk melihat level tegangannya. Diagram blok pengujian ditunjukkan dalam Gambar 5.1. Berdasarkan Gambar 5.1 pengujian catu dilakukan dengan cara memberikan sumber tegangan berupa baterai pada catu rangkaian. Pada *pin output* dipasang *oscilloscope* dan di lihat level tegangannya. Sinyal *output* catu rangkaian ditunjukkan dalam Gambar 5.2. Skematik rangkaian ditunjukkan oleh Gambar 5.3.



Gambar 5. 1 Diagram Blok Pengujian Catu Rangkaian.



Gambar 5. 2 Sinyal *Output* Pengujian Catu Rangkaian.



Gambar 5. 3 Skematik Rangkaian Catu.

Berdasarkan Gambar 5.2 di dapatkan grafik pengujian sinyal *output* catu rangkaian yang di ukur menggunakan *oscilloscope*. Grafik menunjukkan tegangan 5 V dengan Volt/div = 5 Volt/div. Tegangan 5 V di gunakan pada rangkaian mikrokontroller dan rangkaian *driver*. Berdasarkan Gambar 5.3 menunjukkan skematik rangkaian dimana *input* tegangan berupa baterai 12 Volt yang terhubung ke rangkaian *regulator* 7805 yang akan merubah tegangan masukan menjadi tegangan keluaran sebesar 5 volt.

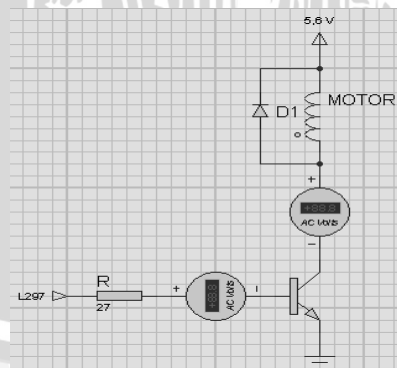
5.2 Pengujian Rangkaian Stepper Motor Driver

5.2.1 Pengujian Arus Rangkaian Stepper Motor Driver

Pengujian arus dilakukan untuk membuktikan hasil perancangan rangkaian *driver*. Pengujian dilakukan dengan mengukur arus I_C dan arus I_B saat motor di nyalakan. Pengukuran arus dilakukan menggunakan Amperemeter yang di pasang pada terminal basis dan kolektor nya. Gambar 5.4 menunjukkan skematik pengujian. Berdasarkan hasil pengujian di dapatkan Tabel 5.1.

Tabel 5. 1 Hasil Pengujian Arus *Driver*

BEBAN	ARUS (A)	
	I_B	I_C
0 gr	0.05	0.99
100 gr	0.03	0.9
200 gr	0.03	0.86
300 gr	0.03	0.9
400 gr	0.03	0.83



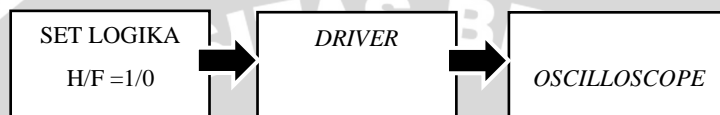
Gambar 5. 4 Skematik Pengujian Arus *Driver*.

Berdasarkan Tabel 4.1 di dapatkan hasil pengujian arus I_C dan arus I_B yang dilakukan dengan menambahkan beban pada *slider*. Arus I_B antara tidak berbeban

dengan beban relatif turun, akan tetapi arus I_B relatif konstan pada pemberian beban 100 – 400 gram. Arus I_C pada setiap pengujian relatif konstan.

5.2.2 Pengujian Mode Rangkaian Stepper Motor Driver

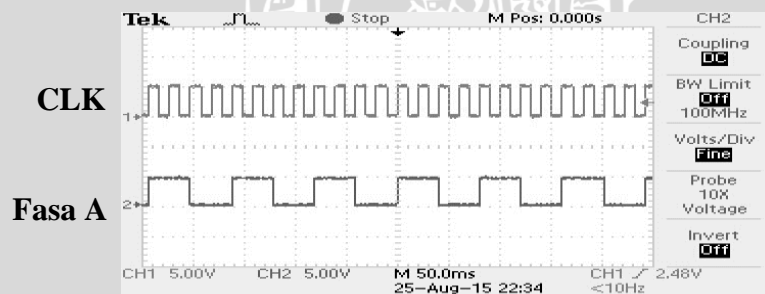
Pengujian rangkaian *stepper motor driver* dilakukan untuk membandingkan sinyal keluaran dari IC L297 dimana terdapat sinyal A , sinyal B, sinyal C dan sinyal D. Sinyal dapat dilihat menggunakan *oscilloscope*. Pengujian *driver* ini di bagi menjadi 2 yaitu pengujian mode *half step* dan *full step*. Sinyal keluaran akan di bandingkan dengan sinyal keluaran dari *datasheet*. Diagram blok pengujian ditunjukkan oleh Gambar 5.5.



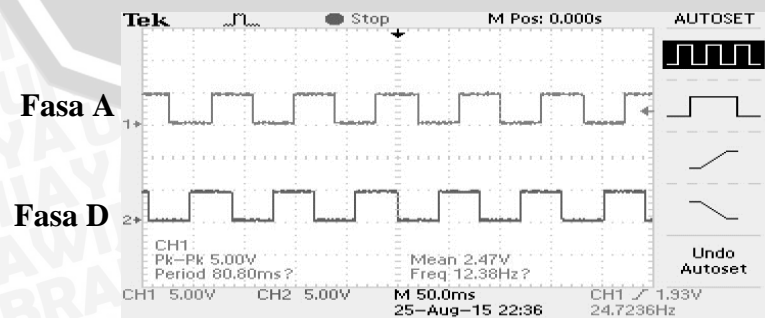
Gambar 5. 5 Blok Pengujian Stepper Motor Driver.

5.2.2.1 Pengujian Mode Full Step

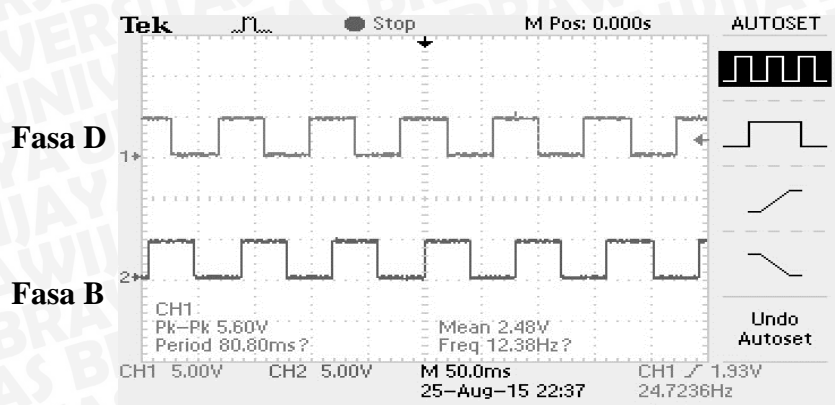
Pengujian mode *full step* dilakukan dengan cara mengatur logika H/F pada IC L297 dengan logika 0. Terdapat 1 masukan berupa sinyal CLK dan 4 keluaran yaitu sinyal A, B, C, D. Berdasarkan Gambar 5.6 , 5.7 , 5.8 dan 5.9 dapat di bandingkan terhadap *datasheet* Gambar 5.10 dimana Gambar sinyal CLK , A , B , C dan D adalah serupa. Tabel logika fasa motor *stepper* pada mode *full step* ditunjukkan dalam Tabel 5.2.



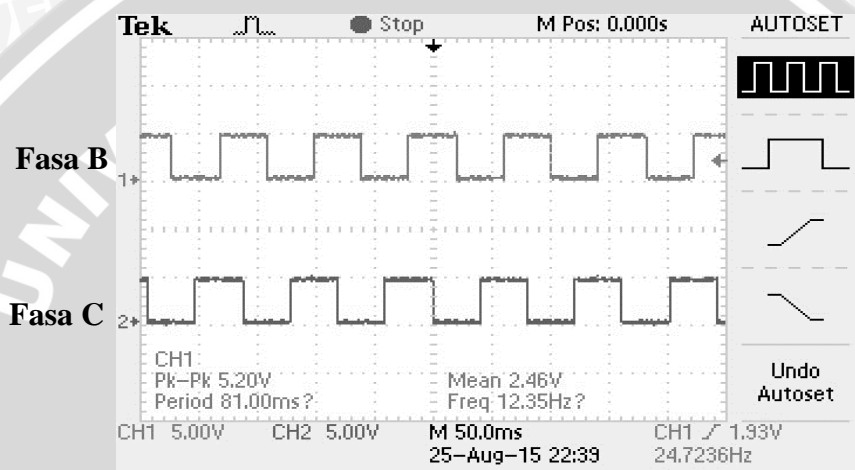
Gambar 5. 6 Sinyal Input CLK dan A Pada Mode Full Step.



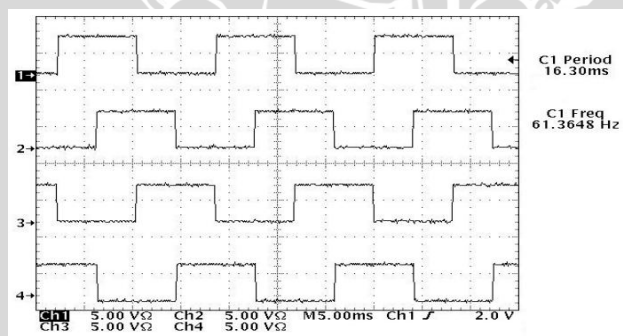
Gambar 5. 7 Sinyal Output Fasa A dan D Pada Mode Full Step.



Gambar 5. 8 Sinyal Output fasa D dan B Pada Mode Full Step.



Gambar 5. 9 Sinyal output fasa B dan C Mode Full Step.



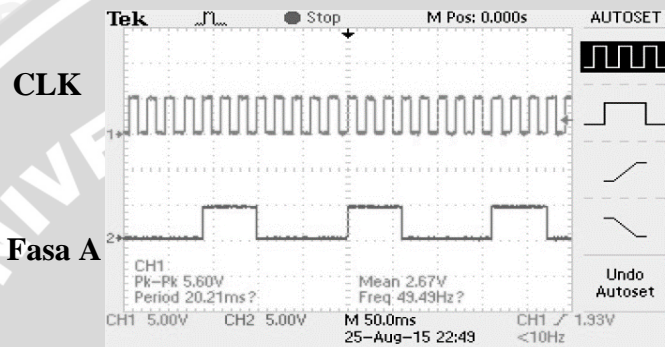
Gambar 5. 10 Sinyal Full Step Berdasarkan Datasheet .

Tabel 5. 2 Logika Fasa Mode Full Step

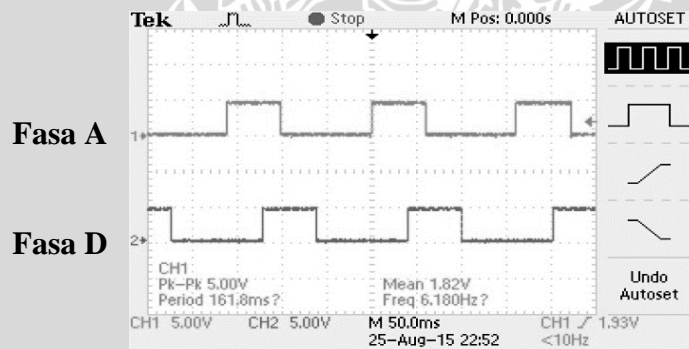
Clock	A	B	C	D
1	1	0	1	0
3	1	0	1	1
5	0	1	0	1
7	0	1	1	0

5.2.2.2 Pengujian Mode *Half Step*

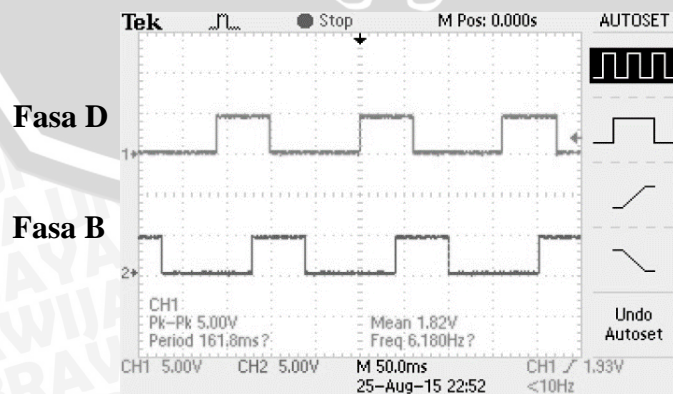
Pengujian mode *half step* dilakukan dengan cara mengatur logika H/F pada IC L297 dengan logika 1. Terdapat 1 masukan berupa sinyal CLK dan 4 keluaran yaitu sinyal A, B, C dan D. Berdasarkan Gambar 5.11 , 5.12 , 5.13 dan 5.14 dapat di bandingkan terhadap *datasheet* pada Gambar 5.15 dimana Gambar sinyal CLK , A , B , C dan D adalah serupa. Tabel logika fasa motor *stepper* pada mode *half step* ditunjukkan dalam Tabel 5.3.



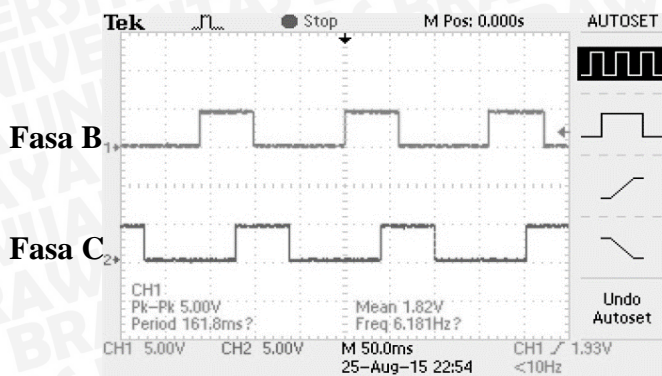
Gambar 5. 11 Sinyal *Input CLK* dan *A* Pada mode *Half Step*.



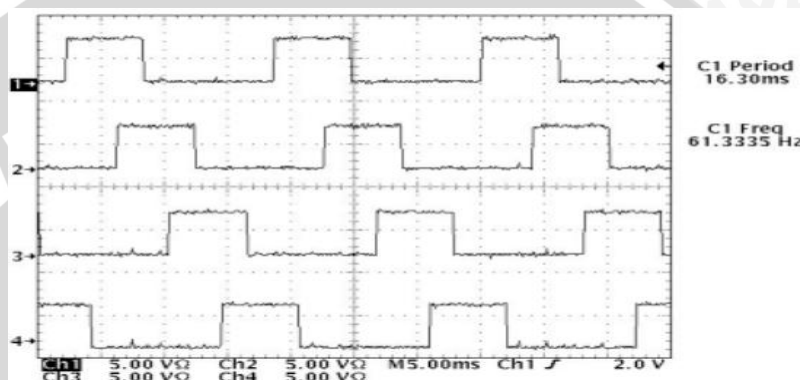
Gambar 5. 12 Sinyal *Output Fasa A* dan *D* Pada Mode *Half Step*.



Gambar 5. 13 Sinyal *Output Fasa D* dan *B* Mode *Half Step*.



Gambar 5. 14 Sinyal *Output* Fasa B dan C Mode *Half Step*.



Gambar 5. 15 Sinyal *Half Step* Berdasarkan *Datasheet* .

Sumber : ST (1996: 5)

Tabel 5. 3 Logika Fasa Mode *Half Step*

Clock	A	B	C	D
1	1	0	1	0
2	1	0	0	0
3	1	0	0	1
4	0	0	0	1
5	0	1	0	1
6	0	1	0	0
7	0	1	1	0
8	0	0	1	0

Dapat dianalisa bahwa pada mode *full step* mencapai siklus penuh dan kembali ke fasa awal lebih cepat dari pada mode *half step*. Sesuai dengan perhitungan teori untuk mode *half step* yang membutuhkan jumlah pemicuan lebih banyak daripada *full step* (dua kali *full step*).

5.2.3 Pengujian Sudut Motor

Pengujian sudut motor bertujuan untuk mengetahui resolusi sudut *stepper motor* saat menggunakan mode *full step* dan *half step*. pengujian sudut motor dilakukan dengan menggunakan penggaris busur 360° dengan ketelitian 1° .

Diagram blok pengujian ditunjukkan oleh Gambar 5.16.

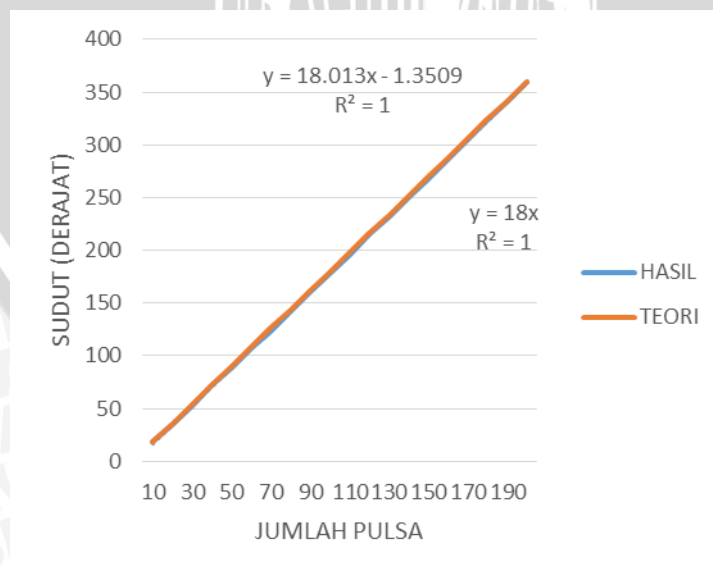


Gambar 5. 16 Blok Pengujian Sudut Motor .

5.2.3.1 Pengujian Sudut Motor Dengan Mode *Full Step*

Pengujian sudut motor dengan mode *Full Step* dilakukan dengan cara mengatur logika H/F dengan logika 0 pada IC L297. *Stepper Motor driver* akan memberikan pulsa untuk menggerakkan *stepper motor*. Pergerakan sudut *stepper motor* di baca melauai penggaris busur. Tabel pengujian sudut dengan mode *full step* dapat dilihat dari Tabel 5.4. Berdasarkan Tabel 5.4 dapat dijadikan grafik seperti Gambar 5.17.

Berdasarkan Tabel 5.4 didapatkan hasil pengujian pemberian pulsa terhadap sudut *stepper motor* pada mode *full step* menunjukkan bahwa hasil pada 3 kali pengujian memiliki kesalahan paling besar 3 %. Grafik pada Gambar 5.17 menunjukkan bahwa antara grafik hasil dan teori saling berhimpit secara linear dengan $R^2=1$ terhadap trendline linear. Didapatkan resolusi pengujian sudut *stepper motor* pada mode *full step* yaitu 1,8°/pulsa.



Gambar 5. 17 Grafik Pengujian Sudut *Stepper Motor* Pada Mode *Full Step*

Tabel 5. 4 Pengujian Sudut Dengan Mode *Full Step*

Jumlah Pulsa	Sudut (Derajat)			Rata-Rata	Teori	Kesalahan (%)
	Hasil					
	1	2	3			
10	18	18	18	18	18	0
20	36	35	35	35.33	36	2
30	52	53	52	52.33	54	3
40	72	71	71	71.33	72	1
50	89	89	89	89.00	90	1
60	107	107	106	106.67	108	1
70	123	124	123	123.33	126	2
80	142	142	142	142.00	144	1
90	161	160	160	160.33	162	1
100	179	179	179	179.00	180	1
110	195	196	196	195.67	198	1
120	215	215	215	215.00	216	0
130	233	232	233	232.67	234	1
140	250	250	251	250.33	252	1
150	268	268	268	268.00	270	1
160	286	287	287	286.67	288	0
170	305	305	305	305.00	306	0
180	323	324	324	323.67	324	0
190	341	342	341	341.33	342	0
200	360	360	360	360.00	360	0

5.2.3.1.1 Pengujian Resolusi Sudut Motor Dengan Mode *Full Step*

Pengujian Resolusi Bertujuan untuk mengetahui apakah resolusi yang didapatkan sesuai dengan hasil pengujian sebelumnya. Pengujian Resolusi sudut motor dengan mode *full step* dilakukan dengan memberikan sinyal step dengan range pemberian 1 pulsa sampai 10 pulsa. Perputaran sudut motor di ukur menggunakan busur 360°. Hasil Pengujian ditunjukkan pada Tabel 5.5 berikut.

Berdasarkan Tabel 5.5 di dapatkan hasil pengujian resolusi sudut motor dengan mode *full step*. resolusi didapatkan dengan membagi sudut dengan pulsa. Didapatkan hasil 1.8⁰/ *step*.



Tabel 5. 5 Pengujian Resolusi Sudut Motor Dengan Mode *Full Step*

PULSA	SUDUT (DERAJAT)			
	P.1	P.2	P.3	RATA-RATA
1	1.7	1.8	1.9	1.8
2	3.6	3.4	3.5	3.5
3	5.6	5.5	5.4	5.5
4	7.2	7.2	7.2	7.2
5	9	9	9	9
6	10.8	11	10.6	10.8
7	12.6	12.6	12.6	12.6
8	14.4	14.4	14.4	14.4
9	16.2	16.2	16.2	16.2
10	18	18	18	18

5.2.3.2 Pengujian Sudut Motor Dengan Mode *Half Step*

Pengujian sudut motor dengan mode *Half Step* dilakukan dengan cara mengatur logika H/F dengan logika 1 pada IC L297. *Stepper Motor driver* akan memberikan pulsa untuk menggerakkan *stepper motor*. Pergerakan sudut *stepper motor* di baca melauai penggaris busur. Tabel pengujian sudut dengan mode *half step* dapat dilihat dari Tabel 5.6. Berdasarkan Tabel 5.6 dapat dijadikan grafik seperti Gambar 5.19.

Berdasarkan Tabel 5.6 didapatkan hasil pengujian pemberian pulsa terhadap sudut *stepper motor* pada mode *half step* menunjukkan bahwa hasil pada 3 kali pengujian memiliki kesalahan paling besar 3 %. Grafik pada Gambar 5.19 menunjukkan bahwa antara grafik hasil dan teori saling berhimpit secara linear dengan R^2 mendekati 1 terhadap trendline linear. Didapatkan resolusi pengujian sudut *stepper motor* pada mode *half step* yaitu $0,9^{\circ}$ /pulsa. Gambar pengujian *stepper motor* seperti pada Gambar 5.18. Dalam perhitungan teori perhitungan putaran sudut motor stepper dapat dijelaskan dalam persamaan (5-1).

$$\text{Jumlah pulsa} = \frac{\text{Sudut putaran}}{x} \quad (5-1)$$

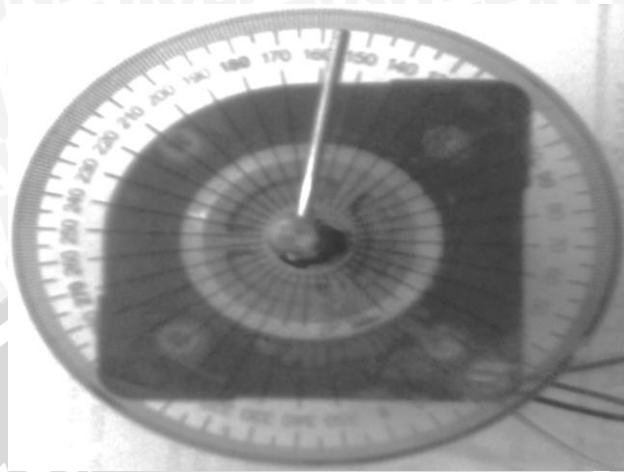
Dimana $x = 1,8^{\circ}$: sudut per langkah *stepper motor* untuk mode *full step*

$x = 0,9^{\circ}$: sudut per langkah *stepper motor* untuk mode *half step*

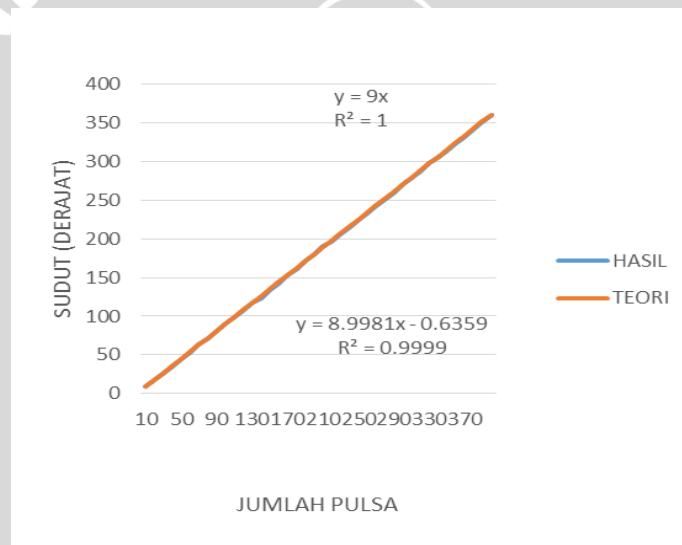
Tabel 5. 6 Pengujian Sudut Dengan Mode *Half Step*

jumlah pulsa	sudut (derajat)			rata-rata	teori	kesalahan (%)
	hasil					
	1	2	3			
10	9	9	9	9	9	0
20	18	18	18	18.00	18	0
30	27	26	27	26.67	27	1
40	36	35	34	35.00	36	3
50	44	44	44	44.00	45	2
60	52	53	52	52.33	54	3
70	63	63	63	63.00	63	0
80	72	71	71	71.33	72	1
90	81	81	81	81.00	81	0
100	90	90	90	90.00	90	0
110	99	99	98	98.67	99	0
120	107	107	106	106.67	108	1
130	117	118	117	117.33	117	0
140	123	124	123	123.33	126	2
150	134	134	134	134.00	135	1
160	142	142	142	142.00	144	1
170	153	153	153	153.00	153	0
180	161	160	160	160.33	162	1
190	170	172	172	171.33	171	0
200	180	180	180	180.00	180	0
210	190	190	190	190.00	189	1
220	195	196	196	195.67	198	1
230	206	206	206	206.00	207	0
240	215	215	215	215.00	216	0
250	224	225	225	224.67	225	0
260	233	232	233	232.67	234	1
270	242	242	242	242.00	243	0
280	250	250	251	250.33	252	1
290	260	260	260	260.00	261	0
300	270	270	270	270.00	270	0
310	279	279	279	279.00	279	0
320	286	287	287	286.67	288	0
330	298	298	298	298.00	297	0
340	305	305	305	305.00	306	0
350	314	314	314	314.00	315	0
360	323	324	324	323.67	324	0
370	332	332	332	332.00	333	0
380	341	342	341	341.33	342	0
390	350	350	350	350.00	351	0
400	360	360	360	360.00	360	0





Gambar 5. 18 Pengujian Sudut *Stepper Motor*.



Gambar 5. 19 Grafik Pengujian Sudut *Stepper Motor* Pada Mode *Half Step*.

5.2.3.2.1 Pengujian Resolusi Sudut Motor Dengan Mode *Half Step*

Pengujian Resolusi Bertujuan untuk mengetahui apakah resolusi yang didapatkan sesuai dengan hasil pengujian sebelumnya. Pengujian Resolusi sudut motor dengan mode *half step* dilakukan dengan memberikan sinyal step dengan range pemberian 1 pulsa sampai 10 pulsa. Perputaran sudut motor di ukur menggunakan busur 360°. Hasil Pengujian ditunjukkan pada Tabel 5.7 berikut.

Berdasarkan Tabel 5.7 di dapatkan hasil pengujian resolusi sudut motor dengan mode *half step*. Resolusi didapatkan dengan membagi sudut dengan pulsa. Didapatkan hasil 0.9⁰/ *step*.

Tabel 5. 7 Pengujian Resolusi Sudut Motor Dengan Mode *Half Step*

PULSA	SUDUT (DERAJAT)			
	P.1	P.2	P.3	RATA-RATA
1	1	0.9	0.8	0.9
2	2	1.9	1.8	1.9
3	2.7	2.7	2.7	2.7
4	3.6	3.6	3.6	3.6
5	4.5	4.5	4.5	4.5
6	5.4	5.4	5.4	5.4
7	6.3	6.3	6.3	6.3
8	7.2	7.2	7.2	7.2
9	8.1	8.1	8.1	8.1
10	9	9	9	9

5.2.4 Pengujian Jarak Pergeseran *Slider* Saat Motor Bergerak

Pengujian jarak pergeseran *slider* saat motor bergerak di bagi menjadi dua yaitu pada mode *full step* dan *half step*. Pengujian pergeseran dilakukan dengan menggunakan *sticker* penggaris yang ditempelkan pada alumunium. Diagram blok pengujian ditunjukkan oleh Gambar 5.20.



Gambar 5. 20 Blok Pengujian Jarak Pergeseran *Slider* Saat Motor Bergerak.

5.2.4.1 Pengujian Jarak Pergeseran *Slider* Saat Motor Bergerak Dengan Mode *Full Step*

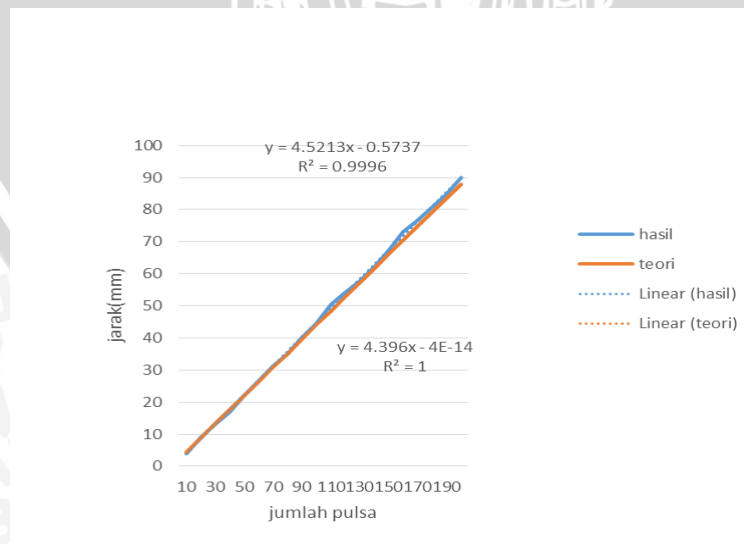
Pengujian jarak pergeseran *slider* saat motor bergerak dengan dilakukan dengan mengatur logika H/F pada IC L297 dengan logika 0. *Driver* akan memberikan sinyal untuk menggerakkan *stepper motor* dan *stepper motor* menggerakkan *slider*. Jarak pergeseran *slider* di baca dengan *sticker* penggaris yang dipasang pada alumunium. Tabel pengujian jarak pergeseran *slider* dengan mode *full step* dapat dilihat dari Tabel 5.8. Berdasarkan data dari Tabel 5.8 di dapatkan Gambar grafik seperti Gambar 5.21.

Berdasarkan Tabel 5.8 di dapatkan hasil pengujian jarak pergeseran *slider* dengan mode *full step* dengan kesalahan terbesar adalah 9 %. Kesalahan tersebut didapatkan karena pada saat pemberian pulsa sebesar 10 pulsa rantai yang

mengkopel antara motor dan *slider* sedikit longgar. Gambar 5.21 menunjukkan grafik hasil pengujian jarak *slider* dimana secara teori didapatkan grafik yang linear dengan $R^2= 1$ terhadap *trendline linear* dan hasil pengujian di dapatkan hasil yang mendekati *linear* dengan $R^2=0,9996$ terhadap *trendline linear*. Didapatkan resolusi pengujian jarak *slider* yaitu 0,44 mm/pulsa.

Tabel 5. 8 Pengujian Jarak Pergeseran *Slider* Dengan Mode *Full Step*

JUMLAH PULSA	jarak(mm) hasil			rata-rata(mm)	teori	kesalahan (%)
	1	2	3			
10	4	4	4	4.0	4.4	9
20	9	9	9	9.0	8.8	2
30	13	13	13	13.0	13.2	1
40	17	17	17	17.0	17.6	3
50	22	22	22	22.0	22.0	0
60	27	27	26	26.7	26.4	1
70	31	31	31	31.0	30.8	1
80	35	35	35	35.0	35.2	0
90	40	40	40	40.0	39.6	1
100	44	44	45	44.3	44.0	1
110	50	51	50	50.3	48.4	4
120	54	54	54	54.0	52.8	2
130	58	57	58	57.7	57.1	1
140	62	62	62	62.0	61.5	1
150	67	67	67	67.0	65.9	2
160	73	73	73	73.0	70.3	4
170	76	78	76	76.7	74.7	3
180	81	81	80	80.7	79.1	2
190	85	85	84	84.7	83.5	1
200	90	90	90	90.0	87.9	2



Gambar 5. 21 Grafik Pengujian Jarak Pergeseran *Slider* Dengan Mode *Full Step*.

5.2.4.1.1 Pengujian Resolusi Jarak Pergeseran Slider Saat Motor Bergerak Dengan Mode Full Step

Pengujian Resolusi Jarak pergeseran *slider* saat motor bergerak dengan mode *full step* bertujuan untuk mengetahui apakah hasil pengujian resolusi sama dengan pengujian sebelumnya. Pengujian dilakukan dengan cara memberikan pulsa pada stepper motor dengan range 1- 5 pulsa dan dihitung jaraknya dengan menggunakan jangka sorong dengan ketelitian 0.1 mm. Tabel 5.9 menunjukkan hasil pengujian resolusi.

Berdasarkan Tabel 5.9 di dapatkan hasil pengujian Resolusi pergeseran *slider* saat motor bergerak dengan mode *full step*. Didapatkan Resolusi sebesar 0.43 mm/pulsa dengan perbedaan terhadap percobaan sebelumnya yaitu 0.44 mm/pulsa. Hal ini dikarenakan penggunaan alat ukur dengan ketelitian 0.1 mm sehingga hasil tidak optimal.

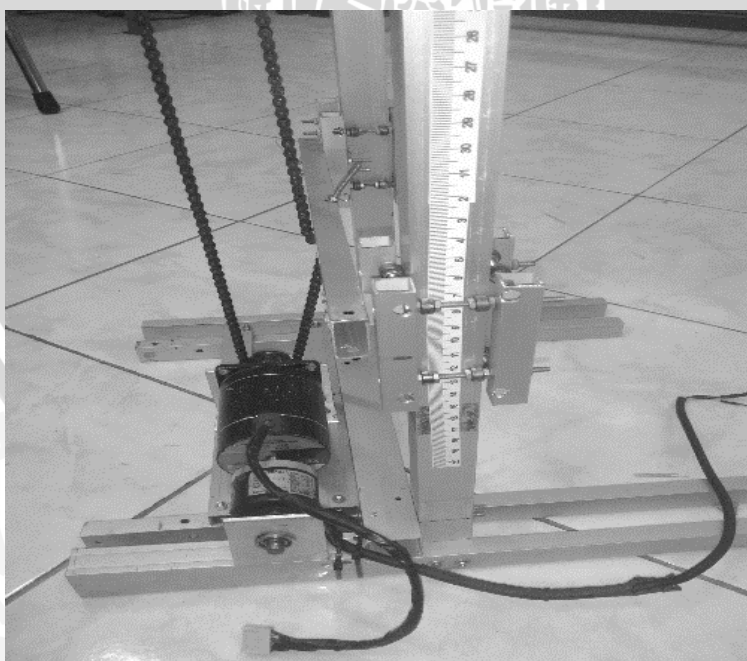
Tabel 5. 9 Pengujian Resolusi Pergeseran *Slider* Saat Motor Bergerak Dengan Mode *Full Step*

Pulsa	JARAK (mm)			
	P.1	P.2	P.3	RATA-RATA
1	0.4	0.5	0.4	0.43
2	1	0.9	1.1	1.00
3	1.2	1.1	1.3	1.20
4	1.8	1.6	1.5	1.63
5	2.2	2.5	2.2	2.30

5.2.4.2 Pengujian Jarak Pergeseran *Slider* Saat Motor Bergerak Dengan Mode *Half Step*

Pengujian jarak pergeseran *slider* saat motor bergerak dengan dilakukan dengan mengatur logika H/F pada IC L297 dengan logika 1. *Driver* akan memberikan sinyal untuk menggerakkan *stepper motor* dan *stepper motor* menggerakkan *slider*. Jarak pergeseran *slider* di baca dengan *sticker* penggaris yang dipasang pada alumunium. Tabel pengujian jarak pergeseran *slider* dengan mode *half step* dapat dilihat dari Tabel 5.10. Berdasarkan data dari Tabel 5.10 di dapatkan Gambar grafik seperti Gambar 5.22.

Berdasarkan Tabel 5.10 di dapatkan hasil pengujian jarak pergeseran *slider* dengan mode *full step* dengan kesalahan terbesar adalah 6 %. Kesalahan tersebut didapatkan karena pada saat pemberian pulsa sebesar 10 pulsa rantai yang mengkopel antara motor dan *slider* sedikit longgar. Gambar 5.22 menunjukkan grafik hasil pengujian jarak *slider* dimana secara teori didapatkan grafik yang linear dengan $R^2= 1$ terhadap *trendline linear* dan hasil pengujian di dapatkan hasil yang mendekati *linear* dengan $R^2=0,9998$ terhadap *trendline linear*. Didapatkan resolusi pengujian jarak *slider* yaitu 0,21875 mm/pulsa. Gambar pengujian jarak pergeseran *slider* dengan mode *half step* seperti pada Gambar 5.23.



Gambar 5. 22 Pengujian Jarak Pergeseran *Slider* Dengan Mode *Half Step*.

Tabel 5. 10 Pengujian Jarak Pergeseran *Slider* Dengan Mode *Half Step*

JUMLAH PULSA	jarak(mm)			rata-rata(mm)	teori	kesalahan (%)
	hasil					
	1	2	3			
10	2	3	2	2.3	2.2	6
20	5	4	4	4.3	4.4	1
30	7	7	6	6.7	6.6	1
40	8	9	9	8.7	8.8	1
50	11	11	11	11.0	11.0	0
60	13	13	13	13.0	13.2	1
70	15	15	15	15.0	15.4	3
80	17	17	17	17.0	17.6	3
90	19	20	19	19.3	19.8	2
100	22	22	22	22.0	22.0	0
110	24	24	24	24.0	24.2	1
120	26	26	26	26.0	26.4	1
130	28	28	28	28.0	28.6	2
140	30	30	30	30.0	30.8	3
150	34	33	33	33.3	33.0	1
160	35	36	34	35.0	35.2	0
170	37	38	36	37.0	37.4	1
180	40	41	40	40.3	39.6	2
190	41	41	42	41.3	41.8	1
200	45	44	43	44.0	44.0	0
210	46	48	45	46.3	46.2	0
220	48	48	50	48.7	48.4	1
230	51	50	50	50.3	50.6	0
240	52	52	52	52.0	52.8	1
250	56	54	55	55.0	55.0	0
260	57	56	56	56.3	57.1	1
270	58	58	59	58.3	59.3	2
280	62	61	63	62.0	61.5	1
290	63	63	63	63.0	63.7	1
300	65	66	65	65.3	65.9	1
310	68	68	68	68.0	68.1	0
320	70	70	70	70.0	70.3	0
330	72	72	72	72.0	72.5	1
340	74	74	74	74.0	74.7	1
350	77	77	77	77.0	76.9	0
360	79	79	79	79.0	79.1	0
370	81	81	81	81.0	81.3	0
380	83	83	83	83.0	83.5	1
390	85	86	86	85.7	85.7	0
400	88	89	87	88.0	87.9	0



5.2.4.2.1 Pengujian Resolusi Jarak Pergeseran *Slider* Saat Motor Bergerak Dengan Mode *Half Step*

Pengujian Resolusi Jarak pergeseran *slider* saat motor bergerak dengan mode *half step* bertujuan untuk mengetahui apakah hasil pengujian resolusi sama dengan pengujian sebelumnya. Pengujian dilakukan dengan cara memberikan pulsa pada stepper motor dengan range 1- 5 pulsa dan dihitung jaraknya dengan menggunakan jangka sorong dengan ketelitian 0.1 mm. Tabel 5.11 menunjukkan hasil pengujian resolusi.

Berdasarkan Tabel 5.11 di dapatkan hasil pengujian Resolusi pergeseran *slider* saat motor bergerak dengan mode *half step*. Didapatkan Resolusi sebesar 0,21875 mm/pulsa dengan perbedaan terhadap percobaan sebelumnya yaitu 0.233 mm/pulsa. Hal ini dikarenakan penggunaan alat ukur dengan ketelitian 0.1 mm sehingga hasil tidak optimal.

Tabel 5. 11 Pengujian Resolusi Pergeseran *Slider* Saat Motor Bergerak Dengan Mode *Half Step*

PULSA	JARAK			
	P.1	P.2	P.3	RATA-RATA
1	0.3	0.2	0.2	0.233
2	0.5	0.6	0.6	0.567
3	0.7	0.6	0.7	0.667
4	0.8	0.9	0.9	0.867
5	1	1	1	1.000

5. 3 Pengujian Waktu Pergerakan *Slider*

Pengujian ini dilakukan dengan memberikan beban mulai dari 0 sampai dengan 400 gram. Bertujuan untuk Mengetahui respon kinerja pergerakan dari *stepper motor* terhadap pemberian beban. *Slider* digerakkan sejauh 5-40 cm dan di hitung waktu tempuhnya. Seperti yang ditunjukkan Tabel 5.12 berikut.

Tabel 5. 12 Pengujian Waktu Pergerakan *Slider*

JARAK	WAKTU (detik)				
	TANPA BEBAN	100 gr	200 gr	300 gr	400 gr
5	2,58	2,23	2,35	2,41	2,29
10	4,53	4,74	4,70	4,66	4,61
15	6,74	6,92	6,76	6,9	7,26
20	9,02	9,11	9,21	9,16	9,11
25	11,28	11,43	11,33	11,52	11,42
30	13,62	13,64	13,68	13,69	13,58
35	15,76	15,89	15,95	15,78	15,96
40	18,16	18,18	18,22	18,14	18,19

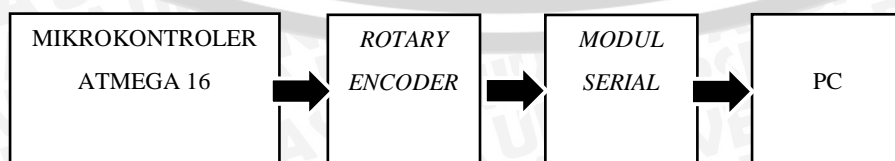
Berdasarkan Tabel 5.12 didapatkan waktu tempuh pada saat *slider* diberikan beban. Tabel menunjukkan bahwa semakin berat beban maka waktu tempuh akan semakin lama.

5.4 Pengujian Rangkaian *Rotary Encoder*

5.4.1 Pengujian Sudut *Rotary Encoder*

Pengujian sudut *rotary encoder* bertujuan untuk mengetahui resolusi pulsa pembacaan sudut. *Rotary encoder* dipasang penggaris busur untuk mengukur sudutnya. Pengujian sudut *rotary encoder* dilakukan dengan memutar *rotary encoder* secara manual dengan *range* sudut 10° sampai 360°. Diagram blok pengujian ditunjukkan oleh Gambar 5.24. Berdasarkan Tabel 5.8 di dapatkan grafik pada Gambar 5.25. Tabel pengujian sudut *rotary encoder* dapat dilihat dari Tabel 5.13.

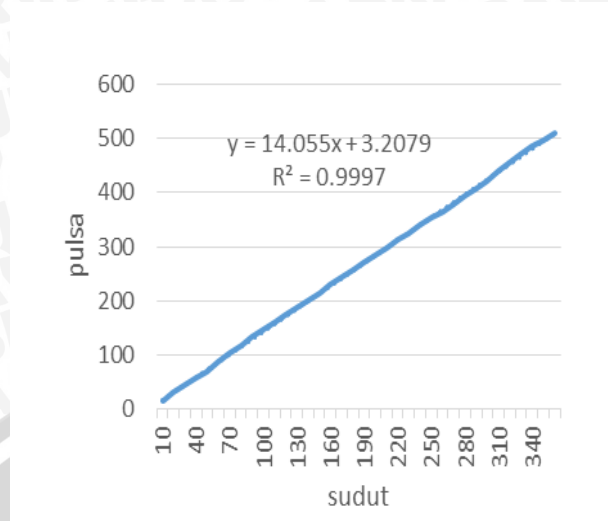
Berdasarkan Tabel 5.13 didapatkan hasil pembacaan *rotary encoder* yang konsisten di tiap pengujian. Hasil ini memiliki hasil yang mendekati linear dengan $R^2=0,9997$ terhadap *trendline linear* seperti yang ditunjukkan pada Gambar 5.25. Didapatkan resolusi pembacaan sudut *rotary* adalah 2 pulsa/derajat. Gambar pengujian sudut *rotary encoder* ditunjukkan pada Gambar 5.26.



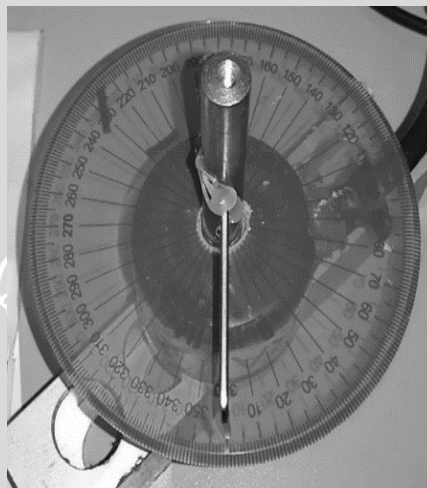
Gambar 5. 23 Blok Pengujian Sudut *Rotary Encoder*.

Tabel 5. 13 Pengujian Sudut *Rotary Encoder*

sudut (derajat)	pulsa		
	Pengujian 1	Pengujian 2	Pengujian 3
10	15	15	15
20	30	30	30
30	44	44	44
40	57	57	57
50	68	68	68
60	89	89	89
70	104	104	104
80	118	118	118
90	133	133	133
100	146	146	146
110	161	161	161
120	174	174	174
130	188	188	188
140	200	200	200
150	214	214	214
160	230	230	230
170	244	244	244
180	257	257	257
190	270	270	270
200	284	284	284
210	297	297	297
220	314	314	314
230	324	324	324
240	341	341	341
250	353	353	353
260	364	364	364
270	377	377	377
280	394	394	394
290	407	407	407
300	421	421	421
310	441	441	441
320	456	456	456
330	471	471	471
340	485	485	485
350	496	496	496
360	509	509	509



Gambar 5. 24 Grafik Pengujian Sudut *Rotary Encoder*.



Gambar 5. 25 Pengujian Sudut *Rotary Encoder*.

5.4.1.1 Pengujian Resolusi Sudut *Rotary Encoder*

Pengujian Resolusi sudut *rotary encoder* bertujuan untuk mengetahui apakah hasil resolusi sama dengan percobaan sebelumnya. Pengujian dilakukan dengan cara menggerakkan rotary tiap 1° sampa 10° kemudian data dari *rotary* ditampilkan ke PC menggunakan modul *serial*. Tabel 5.14 menunjukkan hasil pengujian resolusi sudut *rotary encoder*.

Berdasarkan Tabel 5.14 didapatkan hasil pengujian resolusi sudut *rotary encoder* yaitu 2.67 pulsa/derajat. Hal ini berbeda dengan percobaan sebelumnya yang dimana hasil yang didapat adalah 2 pulsa/derajat yang disebabkan mekanik yang kurang presisi.

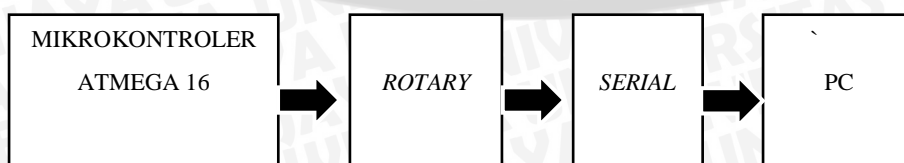
Tabel 5. 14 Pengujian Resolusi Sudut *Rotary Encoder*

SUDUT(°)	PULSA			
	P.1	P.2	P.3	RATA-RATA
1.00	2.00	3.00	3.00	2.67
2.00	3.00	5.00	5.00	4.33
3.00	4.00	4.00	6.00	4.67
4.00	6.00	5.00	8.00	6.33
5.00	7.00	6.00	10.00	7.67
6.00	6.00	9.00	12.00	9.00
7.00	7.00	10.00	13.00	10.00
8.00	7.00	13.00	15.00	11.67
9.00	9.00	13.00	16.00	12.67
10.00	13.00	14.00	19.00	15.33

5.4.2 Pengujian Jarak *Slider* dengan *Rotary Encoder*

Pengujian jarak *slider* dengan *rotary encoder* bertujuan untuk mengetahui resolusi pulsa *rotary encoder* terhadap jarak *slider*. Pengujian jarak *slider* dengan *rotary encoder* dilakukan dengan menggeser *slider* mulai dengan range 1 cm sampai 40 cm secara manual. Pergeseran *slider* akan membuat *rotary encoder* berputar. Pulsa dari *rotary encoder* akan di baca oleh modul *serial* dan di tampilkan ke PC. Diagram blok pengujian ditunjukkan oleh Gambar 5.27. Tabel pengujian jarak *slider* dengan *rotary encoder* dapat dilihat dari Tabel 15. Berdasarkan Tabel 5.15 di dapatkan grafik pada Gambar 5.28.

Berdasarkan Tabel 5.15 didapatkan resolusi pembacaan jarak *rotary* yaitu 57 pulsa/cm jika digunakan skala mm menjadi 5,7 pulsa/mm atau 6 pulsa/mm. Gambar grafik pembacaan *rotary* dapat dilihat pada Gambar 5.27 dimana grafik mendekati *linear* dengan $R^2=0,9999$ terhadap *trendline linear*. Pengujian jarak *slider* dengan *rotary encoder* dapat dilihat pada Gambar 5.29.

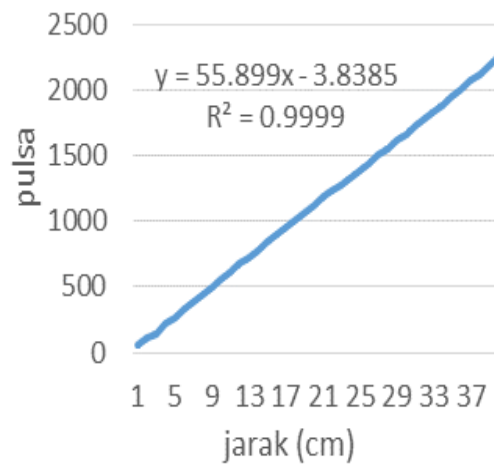


Gambar 5. 26 Pengujian Jarak *Slider* Dengan *Rotary Encoder*.

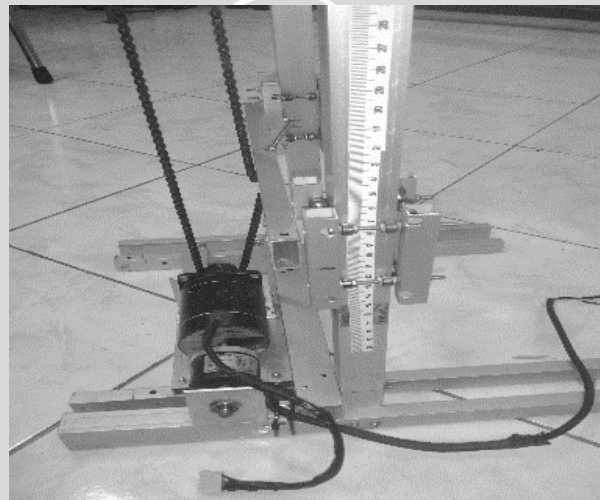
Tabel 5. 15 Pengujian Jarak Slider Dengan Rotary Encoder

Jarak (cm)	pulsa hasil		
	1	2	3
1	57	57	57
2	116	116	116
3	145	145	145
4	219	219	219
5	268	268	268
6	328	328	328
7	388	388	388
8	445	445	445
9	497	497	497
10	554	554	554
11	607	607	607
12	671	671	671
13	722	722	722
14	778	778	778
15	846	846	846
16	897	897	897
17	954	954	954
18	1014	1014	1014
19	1066	1066	1066
20	1116	1116	1116
21	1183	1183	1183
22	1234	1234	1234
23	1278	1278	1278
24	1329	1329	1329
25	1388	1388	1388
26	1444	1444	1444
27	1506	1506	1506
28	1555	1555	1555
29	1613	1613	1613
30	1657	1657	1657
31	1727	1727	1727
32	1787	1787	1787
33	1835	1835	1835
34	1886	1886	1886
35	1950	1950	1950
36	2005	2005	2005
37	2071	2071	2071
38	2117	2117	2117
39	2181	2181	2181
40	2250	2250	2250





Gambar 5. 27 Grafik Pengujian Jarak *Slider* Dengan *Rotary Encoder*.



Gambar 5. 28 Pengujian Jarak *Slider* Dengan *Rotary Encoder*.

5.4.2.1 Pengujian Resolusi Jarak *Slider* Dengan *Rotary Encoder*

Pengujian resolusi jarak *slider* dengan *rotary encoder* bertujuan untuk mengetahui apakah hasil resolusi sama dengan hasil percobaan sebelumnya. Pengujian dilakukan dengan cara menggeser *slider* tiap 1 mm sampai 10 mm kemudian data *rotary* dibaca menggunakan modul serial. Tabel 5.16 menunjukkan hasil pengujian resolusi jarak *slider* dengan *rotary encoder*.

Berdasarkan Tabel 5.16 didapatkan hasil pengujian jarak *slider* dengan *rotary encoder*. Didapatkan resolusi 6.33 pulsa/mm dimana berbeda dengan hasil pengujian sebelumnya yaitu 6 pulsa/mm. Hal ini disebabkan mekanik yang kurang presisi.

Tabel 5. 16 Pengujian Resolusi Jarak *Slider* Dengan *Rotary Encoder*

JARAK (mm)	PULSA			
	P.1	P.2	P.3	RATA-RATA
1.00	5.00	7.00	7.00	6.33
2.00	13.00	11.00	13.00	12.33
3.00	19.00	17.00	16.00	17.33
4.00	26.00	22.00	22.00	23.33
5.00	32.00	26.00	26.00	28.00
6.00	35.00	31.00	33.00	33.00
7.00	42.00	38.00	42.00	40.67
8.00	47.00	43.00	48.00	46.00
9.00	53.00	49.00	52.00	51.33
10.00	58.00	54.00	60.00	57.33

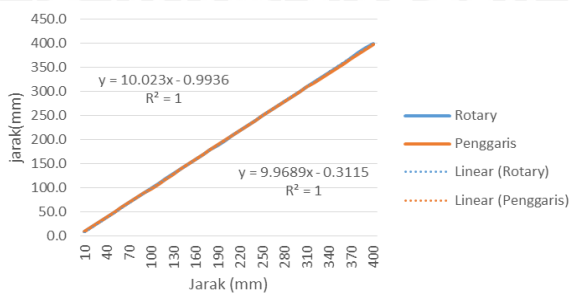
5.5 Pengujian Keseluruhan

Pengujian keseluruhan bertujuan untuk mengetahui performansi *slider* robot . Pengujian keseluruhan di lakukan dengan menggabungkan semua blok pengujian . Mulai dari *input keypad* sampai *slider* bergerak dan data pembacaan *rotary* di tampilkan ke LCD. Didapatkan data pembacaan nilai pergeseran *slider* terhadap *rotary encoder* dan penggaris ditunjukkan pada Tabel 5.17. Berdasarkan Tabel 5.17 grafik data pembacaan nilai pergeseran *slider* terhadap *rotary encoder* dan penggaris yang ditunjukkan pada Gambar 5.30.

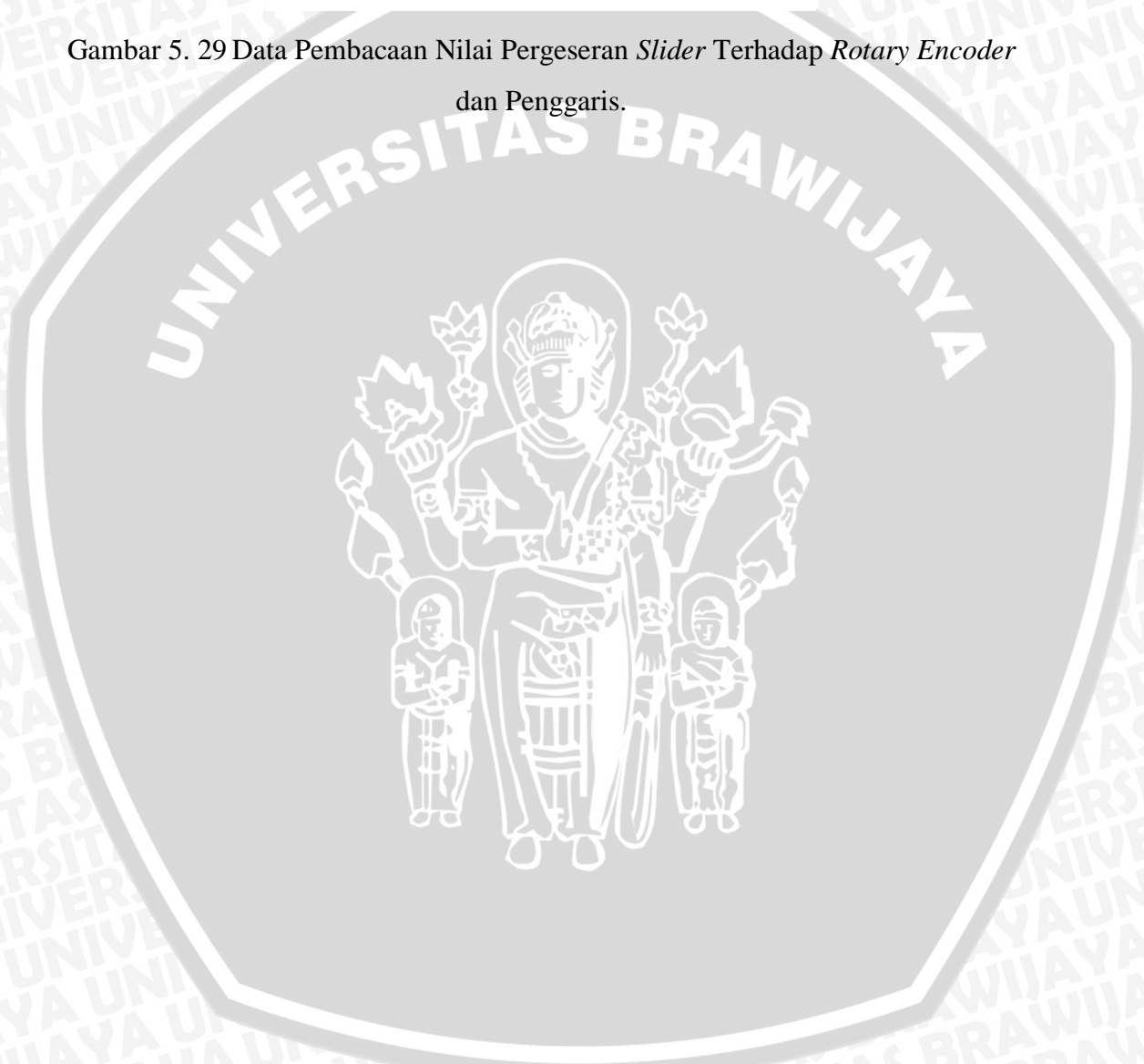
Berdasarkan Tabel 5.17 didapatkan hasil rata-rata kesalahan pembacaan *rotary encoder* terbesar adalah 3.3 % dan kesalahan pembacaan penggaris terbesar adalah 1,67 %. Hal ini terjadi karena dalam pembuatan mekaniknya kurang presisi sehingga terjadi *slip* pada rantai. Gambar 5.30 menunjukkan grafik hasil pembacaan jarak dari *rotary encoder* dan penggaris, dimana $R^2=1$ terhadap *trendline* linear.

Tabel 5. 17 Nilai Pergeseran *Slider Terhadap Rotary Encoder dan Penggaris*

INPUT KEYP AD (mm)	ROTARY ENCODER (mm)					PENGGARIS (mm)				
				RAT A- RAT A	ERRO R				RATA - RATA	KESALAHA N
	P.1	P.2	P.3			P.1	P.2	P.3		
10	9	10	10	9.7	3.3	10	10	10	10	0.00
20	19	20	20	19.7	1.7	20	20	20	20	0.00
30	29	29	30	29.3	2.2	29.5	29.5	29.5	29.5	1.67
40	39	40	40	39.7	0.8	39.5	39.5	39.5	39.5	1.25
50	49	49	49	49.0	2.0	49.5	49.5	49.5	49.5	1.00
60	59	59	59	59.0	1.7	59.5	59.5	59.5	59.5	0.83
70	68	70	69	69.0	1.4	69	69	69	69	1.43
80	78	80	79	79.0	1.3	78.5	78.5	78.5	78.5	1.88
90	88	90	90	89.3	0.7	89	89	89	89	1.11
100	98	99	99	98.7	1.3	98	98	98	98	2.00
110	108	110	109	109.0	0.9	108	108	108	108	1.82
120	119	120	120	119.7	0.3	118	118	118	118	1.67
130	129	130	130	129.7	0.3	128	128	128	128	1.54
140	138	140	140	139.3	0.5	139	139	139	139	0.71
150	149	150	149	149.3	0.4	150	150	150	150	0.00
160	158	160	160	159.3	0.4	160	160	160	160	0.00
170	168	170	170	169.3	0.4	170	170	170	170	0.00
180	178	179	180	179.0	0.6	179.5	179.5	179.5	179.5	0.28
190	188	189	188	188.3	0.9	190	190	190	190	0.00
200	198	198	198	198.0	1.0	200	200	200	200	0.00
210	209	210	210	209.7	0.2	210	210	210	210	0.00
220	219	219	220	219.3	0.3	220	220	220	220	0.00
230	230	230	230	230.0	0.0	230	230	230	230	0.00
240	240	240	240	240.0	0.0	240	240	240	240	0.00
250	249	249	249	249.0	0.4	250	250	250	250	0.00
260	259	260	260	259.7	0.1	260	260	260	260	0.00
270	271	270	270	270.3	-0.1	269	269	269	269	0.37
280	280	280	280	280.0	0.0	279	279	279	279	0.36
290	289	290	290	289.7	0.1	289	289	289	289	0.34
300	298	300	299	299.0	0.3	299	299	299	299	0.33
310	308	310	309	309.0	0.3	309	309	309	309	0.32
320	319	320	320	319.7	0.1	318	318	318	318	0.63
330	329	330	330	329.7	0.1	328	328	328	328	0.61
340	338	340	340	339.3	0.2	338	338	338	338	0.59
350	349	350	350	349.7	0.1	348	348	348	348	0.57
360	360	360	360	360.0	0.0	358	358	358	358	0.56
370	370	370	370	370.0	0.0	368	368	368	368	0.54
380	382	380	380	380.7	-0.2	378	378	378	378	0.53
390	393	390	390	391.0	-0.3	388	388	388	388	0.51
400	401	400	399	400.0	0.0	398	398	398	398	0.50



Gambar 5. 29 Data Pembacaan Nilai Pergeseran *Slider Terhadap Rotary Encoder* dan Penggaris.



BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

1. Dalam perancangan *slider* robot dibutuhkan pengujian sudut dan posisi dari *stepper motor* yaitu berupa sudut dan pergeseran *slider*. Didapatkan resolusi sudut $1,8^\circ/\text{step}$ untuk mode *full step* dan $0,9^\circ/\text{step}$ untuk mode *half step* dengan masing – masing kesalahan terbesar adalah 3 %. Didapatkan resolusi hasil pengujian jarak pergeseran *slider* sebesar 0,44 mm/pulsa untuk mode *full step* dengan kesalahan terbesar 9 % dan 0,21875 mm/pulsa untuk mode *half step* dengan kesalahan terbesar 6 %.
2. *Driver stepper motor unipolar* menggunakan L297 untuk mengatur fasa *stepper motor* dimana terdapat 4 sinyal keluaran (A, B, C, D) yang terhubung dengan transistor NPN TIP41C sebagai penguat arus *stepper motor*.
3. Sensor *rotary encoder* digunakan sebagai pembaca posisi dari *slider* dimana hasil pembacaan *rotary encoder* ditampilkan ke LCD. *Rotary encoder* dapat digunakan sebagai sensor sudut dan jarak. Didapatkan resolusi pembacaan sudut *rotary encoder* sebesar 2 pulsa/ derajat dan resolusi pembacaan jarak sebesar 57 pulsa/ cm.
4. Berdasarkan hasil pengujian keseluruhan didapatkan bahwa kesalahan pembacaan terbesar pada *rotary encoder* adalah 3,3 % dan kesalahan pembacaan penggaris sebesar 1,67 %.

6.2 Saran

Beberapa hal yang direkomendasikan untuk pengembangan lebih lanjut adalah.

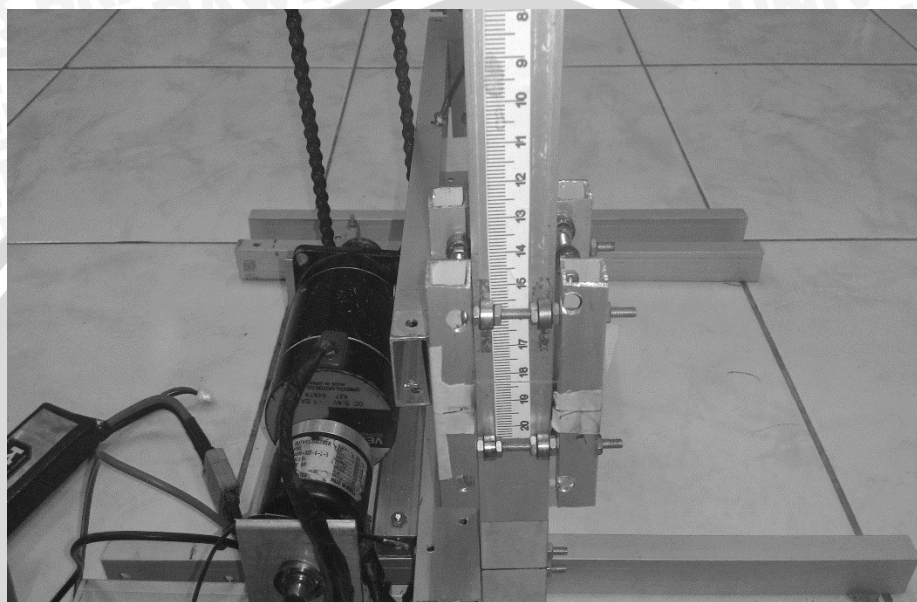
1. Perlu dilakukan pembuatan robot dengan mesin CNC agar ke presisian *slider* robot lebih tinggi.
2. Diperlukan digunakan aktuator *stepper motor* dengan spesifikasi *high performance* dengan torsi dan kecepatan yang lebih tinggi dan sudut lebih kecil dari $1,8^\circ$.

DAFTAR PUSTAKA

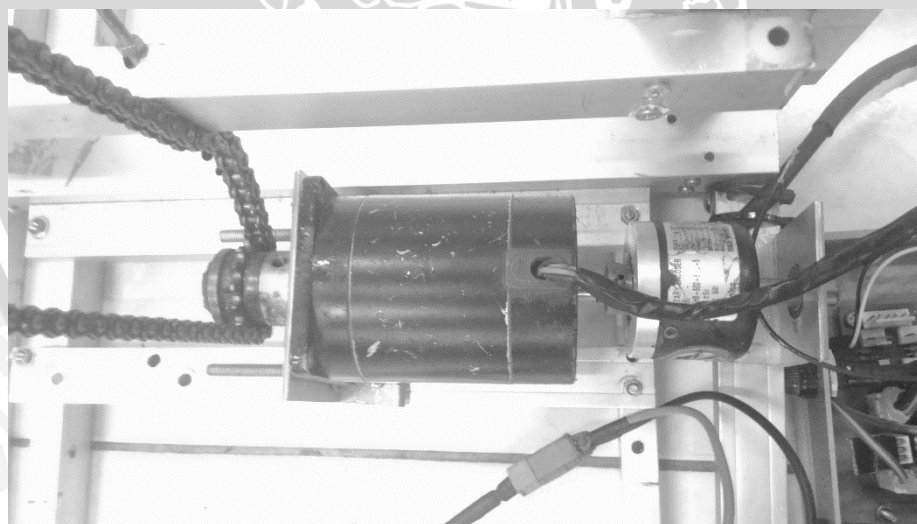
- Atmel. 2002. *Driving Unipolar Stepper Motors Using C51/C251*. San Jose: Atmel Corporation.
- Atmel. 2007. *ATMEGA16/ATMEGA16L, 8-bit AVR with 8 kbytes in System Programmable Flash*. San Jose : Atmel Corporation.
- Autonics. 2013. *E40 series Diameter 40mm Shaft Type Incremental Rotary Encoder*. Korea : Autonics.
- Bodson, Mark., NC, John., T.N Robert., B.R, Ronald. 1993. *High Performance Nonlinear Feedback Control Of A Permanent Magnet Stepper Motor*. New York : IEEE.
- Chaniotakis. Cory. 2006. *Transistors: Bipolar Junction Transistor (BJT)*. Cambridge : Institut Teknologi Massachusetts.
- Dikti 2013. *Panduan Kontes Robot Indonesia 2013*. Jakarta: DIKTI.
- Dikti 2014. *Panduan Kontes Robot Indonesia 2014*. Jakarta: DIKTI.
- Osbourne,Deo. 2011. *Keypad 3x4 Atmega 8535*. www.deo-ebp.blogspot. Diakses Tanggal: 18 Juni 2015.
- Hakim, Lukman., Dikairono, Rudy., Mujiono, Totok. 2013. *Implementasi perhitungan Posisi Robot Dengan FPGA Menggunakan Rotary Encoder*. Surabaya: Institut Teknologi Sepuluh November.
- Hendrayawan, Veri. 2014. *Implementasi Invers Kinematics Pada Sistem Pergerakan Mobile Robot Roda Mekanum*. Skripsi tidak dipublikasikan. Malang: Universitas Brawijaya.
- Irianto, Antonius. 2005. *Bipolar Junction Transistor*. Jakarta: Universitas Gunadarma.
- Jones, Douglas W. 2004. *Stepping Motors Fundamentals*. United State of America: Microchip Technology Incorporated.
- Kustija, Jaja. 2005. *Perancangan Pengontrol Motor Stepper Berbasis PC Dengan Bahasa Pascal*. Bandung: Universitas Pendidikan Indonesia.
- Parallax. 2011. *4X4 Matrix Membrane Keypad*. United State of America : Parallax.
- SGS-Thomson. 2001. *L297 Stepper Motor Controller*. Italy : SGS- Thomson.
- Suwito, Wahyu. 2014. *Pengaturan Posisi Silinder Pneumatic Pada Lengan Robot KRAI*. Skripsi tidak dipublikasikan. Malang: Universitas Brawijaya.
- Weinstar. 2009. *Weinstar Display 2x16 LCD*. www.weinstar.com.tw. Diakses tanggal: 20 Januari 2015.

LAMPIRAN I

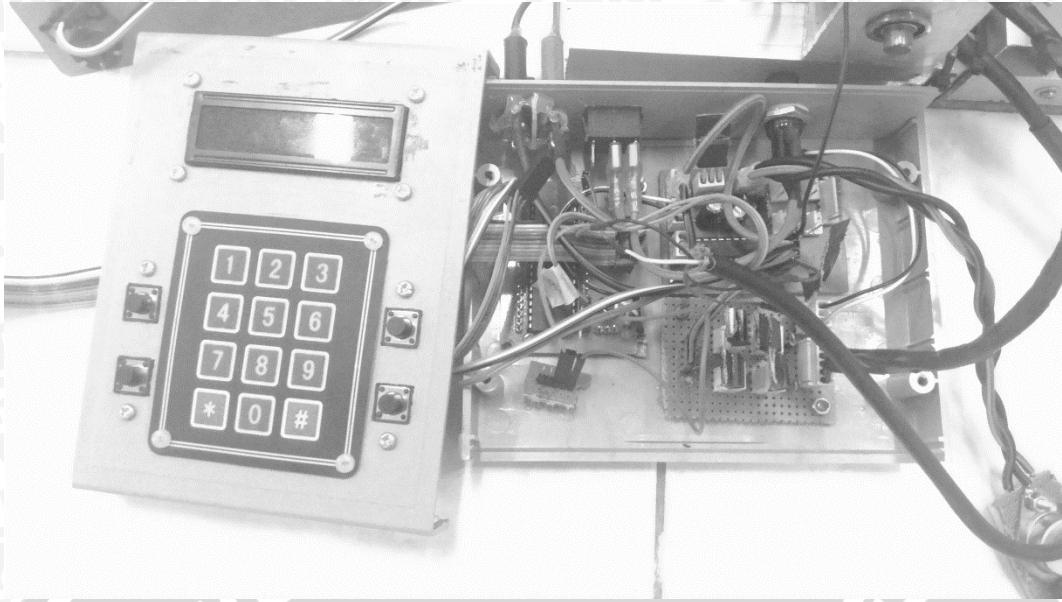
Dokumentasi *Slider* robot



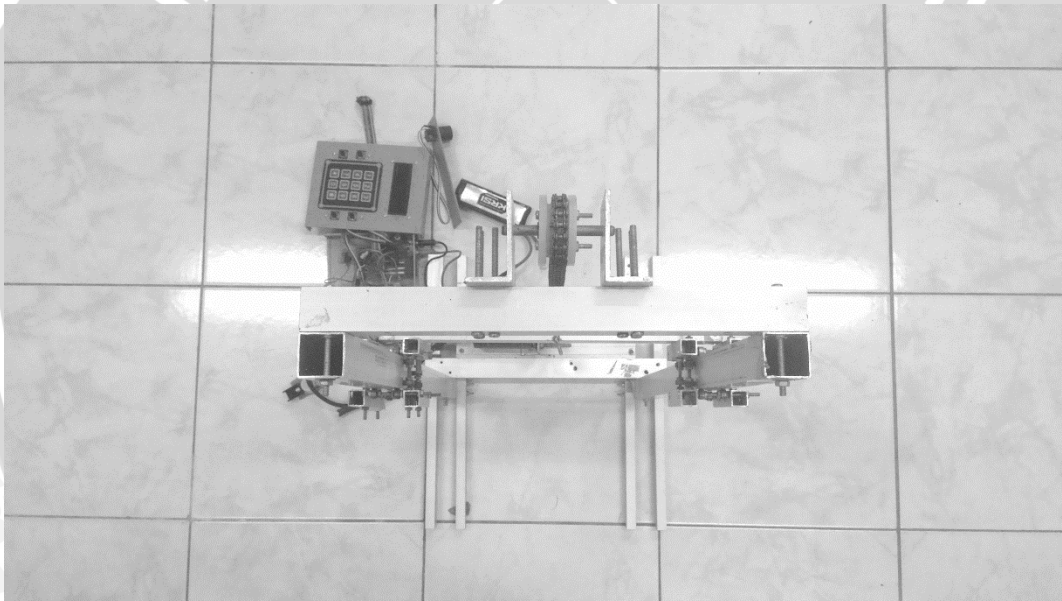
Gambar 1. *Slider Robot*



Gambar 2. *Stepper Motor dan Rotary Encoder*



Gambar 3. Elektrik Robot



Gambar 4. Robot Tampak Atas



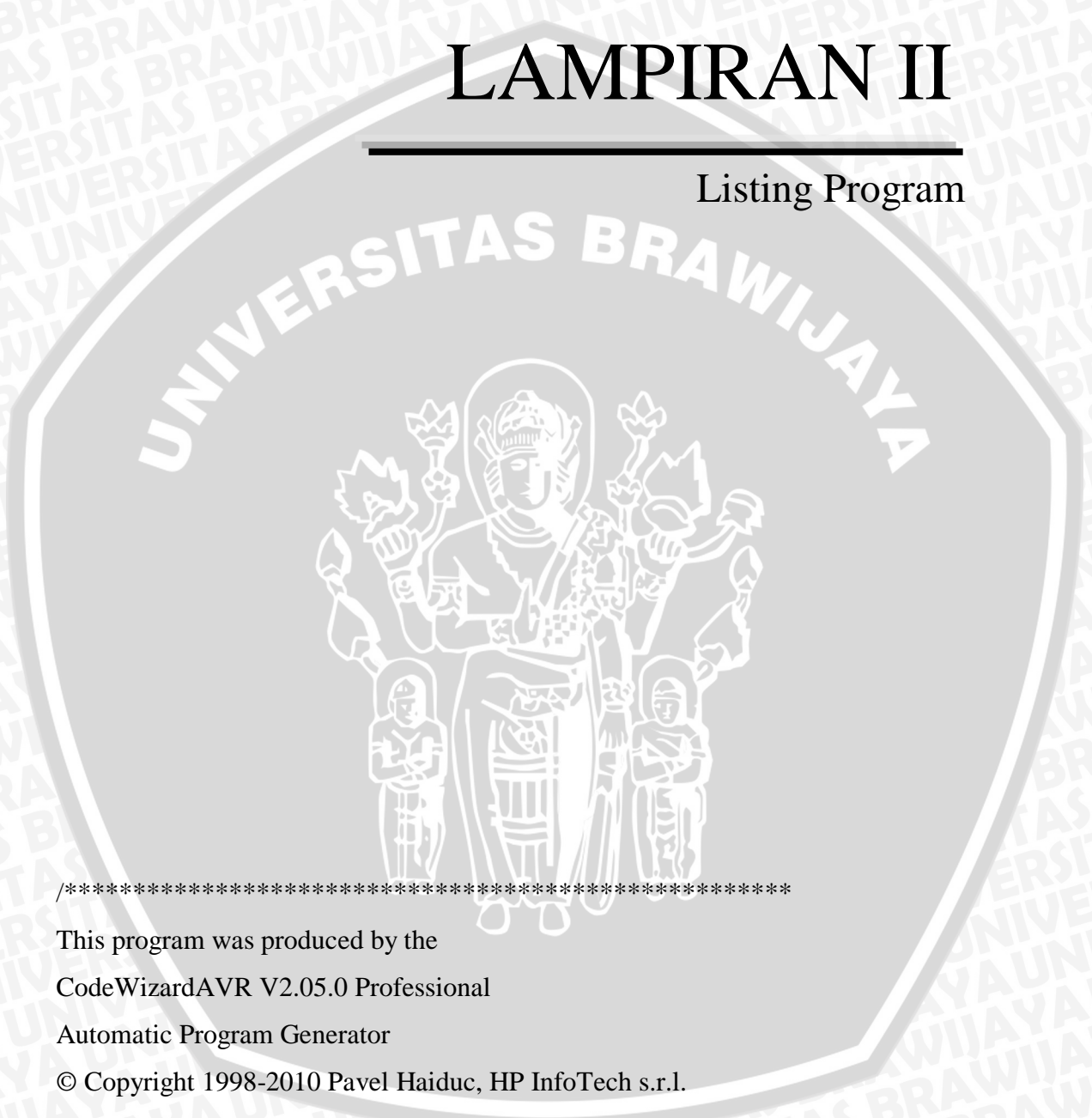
Gambar 5. Robot Tampak Depan



Gambar 6. Robot Tampak Perspektif

LAMPIRAN II

Listing Program



/******

This program was produced by the
CodeWizardAVR V2.05.0 Professional
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project : Skripsi muhamad jaka wimbang wicaksono

Version :

Date : 5/23/2015



Author : NeVaDa

Company :

Comments:

Chip type : ATmega16

Program type : Application

AVR Core Clock frequency: 16.000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

*****/

```
#include <mega16.h>
#include <stdlib.h>
#include <stdio.h>
// Alphanumeric LCD functions
#include <alcd.h>
#include <delay.h>
float jarak;
float keliling=90; //dalam milimeter
float pulsa_max=500;
float selisih;
int pulsa,jarak2,selisih2;
float jar_per_pul; //jarak per pulsa
float hasil,selisih,tmp_awal;
float Ton_off;
int frekuensi=0xC2F6;
float nilai_step2;
int nilai_step;
int x,y,z;
char lcd[16];
```



```

int clr;
unsigned char
sw_0,sw_1,sw_2,sw_3,sw_4,sw_5,sw_6,sw_7,sw_8,sw_9,sw_X,sw_H, penanda,
cntn, ta,hsl[4],aa;
unsigned int tmp;
char tampil[8];
int cnt=0,statmin=0, statplus=0;
//coba timer
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
if(PIND.7==0){
pulsa++;}
else pulsa--;
}

interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
PORTA.3=0; PORTA.4=1; PORTA.5=1; PORTA.6=1;
delay_us(10);
if(!(PINA&0x01))sw_H=1; else sw_H=0;
if(!(PINA&0x02))sw_0=1; else sw_0=0;
if(!(PINA&0x04))sw_X=1; else sw_X=0;

PORTA.3=1; PORTA.4=0; PORTA.5=1; PORTA.6=1;
delay_us(10);
if(!(PINA&0x01))sw_9=1; else sw_9=0;
if(!(PINA&0x02))sw_8=1; else sw_8=0;
if(!(PINA&0x04))sw_7=1; else sw_7=0;

PORTA.3=1; PORTA.4=1; PORTA.5=0; PORTA.6=1;

```

```

delay_us(10);
if(!(PINA&0x01))sw_6=1; else sw_6=0;
if(!(PINA&0x02))sw_5=1; else sw_5=0;
if(!(PINA&0x04))sw_4=1; else sw_4=0;

PORTA.3=1; PORTA.4=1; PORTA.5=1; PORTA.6=0;
delay_us(10);
if(!(PINA&0x01))sw_3=1; else sw_3=0;
if(!(PINA&0x02))sw_2=1; else sw_2=0;
if(!(PINA&0x04))sw_1=1; else sw_1=0;

PORTA.3=1; PORTA.4=1; PORTA.5=1; PORTA.6=1;
}

void keypad()
{
PORTA.3=0; PORTA.4=1; PORTA.5=1; PORTA.6=1;
delay_us(10);
if(!(PINA&0x01))sw_H=1; else sw_H=0;
if(!(PINA&0x02))sw_0=1; else sw_0=0;
if(!(PINA&0x04))sw_X=1; else sw_X=0;

PORTA.3=1; PORTA.4=0; PORTA.5=1; PORTA.6=1;
delay_us(10);
if(!(PINA&0x01))sw_9=1; else sw_9=0;
if(!(PINA&0x02))sw_8=1; else sw_8=0;
if(!(PINA&0x04))sw_7=1; else sw_7=0;

PORTA.3=1; PORTA.4=1; PORTA.5=0; PORTA.6=1;
delay_us(10);
if(!(PINA&0x01))sw_6=1; else sw_6=0;
if(!(PINA&0x02))sw_5=1; else sw_5=0;

```

```
if(!(PINA&0x04))sw_4=1; else sw_4=0;
```

```
PORTA.3=1; PORTA.4=1; PORTA.5=1; PORTA.6=0;
```

```
delay_us(10);
```

```
if(!(PINA&0x01))sw_3=1; else sw_3=0;
```

```
if(!(PINA&0x02))sw_2=1; else sw_2=0;
```

```
if(!(PINA&0x04))sw_1=1; else sw_1=0;
```

```
PORTA.3=1; PORTA.4=1; PORTA.5=1; PORTA.6=1;
```

```
if(sw_0) {ta=1;cntn=0;}
```

```
else if(sw_1) {ta=1;cntn=1;}
```

```
else if(sw_2) {ta=1;cntn=2;}
```

```
else if(sw_3) {ta=1;cntn=3;}
```

```
else if(sw_4) {ta=1;cntn=4;}
```

```
else if(sw_5) {ta=1;cntn=5;}
```

```
else if(sw_6) {ta=1;cntn=6;}
```

```
else if(sw_7) {ta=1;cntn=7;}
```

```
else if(sw_8) {ta=1;cntn=8;}
```

```
else if(sw_9) {ta=1;cntn=9;}
```

```
else if(sw_X)
```

```
{y=0;x=0;ta=0;cntn=0;hsl[0]=0;hsl[1]=0;hsl[2]=0;hsl[3]=0;aa=0;tmp=0;}
```

```
else if(sw_H) { y=1;}
```

```
else ta=0;
```

```
switch(penanda)
```

```
{
```

```
case 0:
```

```
if (ta)penanda=1;
```

```
else penanda=0;
```

```
break;
```

```
case 1:
```

```
if (ta){penanda=2;hsl[aa]=cntn;aa++;}
```

```
        else penanda = 0;
    case 2:
        if (ta){penanda=2;}
        else penanda = 0;
    }
    if(aa==1) tmp=hsl[0];
    else if(aa==2)tmp=hsl[0]*10+hsl[1];
    else if(aa==3)tmp=hsl[0]*100+hsl[1]*10+hsl[2];
    //tmp=hsl[0]*1000+hsl[1]*100+hsl[2]*10+hsl[3];

    if(clr++>5000){lcd_clear(); clr=0;}

    lcd_gotoxy(0,0);
    sprintf(lcd, "%4d",tmp);
    lcd_puts(lcd);
}
// Timer1 overflow interrupt service routine

interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    // Reinitialize Timer1 value
    TCNT1H=frekuensi >> 8;
    TCNT1L=frekuensi & 0xff;
    if(PINA.1==0)
        y=1;
    if(PINA.2==0)
    {
        x=0;
        y=0;
    }
}
```

```
if(y==1)
{
selisih2=tmp-(int)tmp_awal;
if(tmp>(int)tmp_awal) {selisih=(float)tmp-tmp_awal;PORTB.3=0;}
else {selisih=tmp_awal- (float)tmp;PORTB.3=1;}
nilai_step=(int)(2.23*selisih);
/*
if(selisih2>=0) PORTB.3=0; //arah
else if(selisih2<0) PORTB.3=1;*/ //arah
if(x<(nilai_step*2)){
PORTB.0=~PORTB.0;
x++;
}
else {PORTB=0x00;tmp_awal=(float)tmp;}
}
}

// Declare your global variables here

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

// Input/Output Ports initialization
// Port A 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
PORTA=0x07;
DDRA=0xF8;

// 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=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0xFF;
DDRC=0xFF;

// Port D 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
PORTD=0x00;
DDRD=0x00;

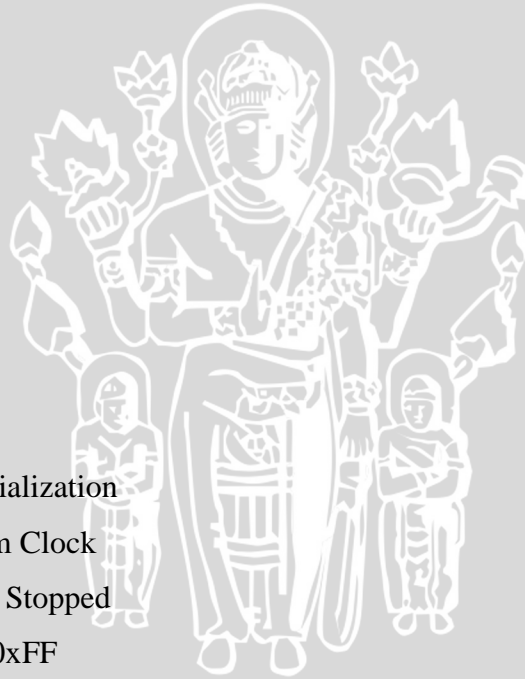
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 15.625 kHz
```

```
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x05;
TCNT1H=0xC2;
TCNT1L=0xF6;
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=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: On
```



```
// INT0 Mode: Falling Edge
// INT1: Off
// INT2: Off
GICR|=0x40;
MCUCR=0x02;
MCUCSR=0x00;
GIFR=0x40;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x04;

// USART initialization
// USART disabled
UCSRB=0x00;

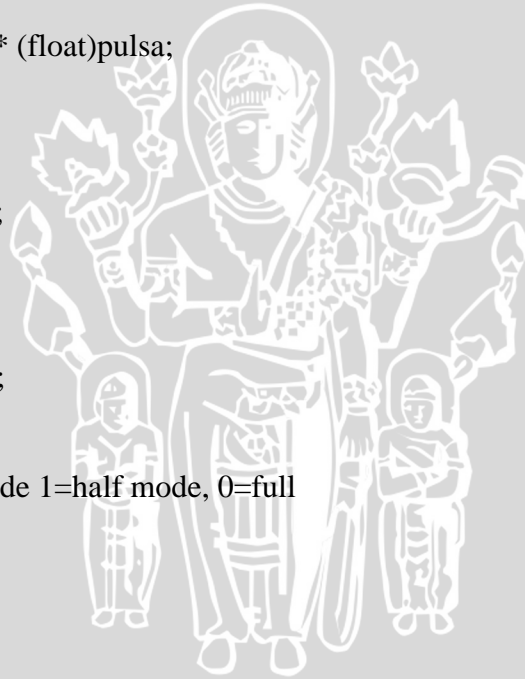
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;
```

```
// Global enable interrupts
#asm("sei")
lcd_init(16);
Ton_off=0.01;
hasil=65535-(15625*Ton_off);
frekuensi=(int)hasil;
nilai_step=200;
x=0;
jar_per_pul=keliling/pulsa_max; // jarak mm tiap pulsa
while (1)
{
    jarak=jar_per_pul * (float)pulsa;
    jarak2=(int)jarak;
    lcd_gotoxy(0,1);
    itoa(jarak2,tampil);
    lcd_puts(tampil);
    lcd_gotoxy(5,1);
    itoa(selisih,tampil);
    lcd_puts(tampil);
    PORTB.2=1; //mode 1=half mode, 0=full
}
}
```



UNIVERSITAS BRAWIJAYA

LAMPIRAN III

Datasheet

