

**APLIKASI KONTROL PID UNTUK PENGATURAN POSISI MOTOR
DC PADA SISTEM *ROTARY PARKING***

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan Untuk Memenuhi Persyaratan

Memperoleh Gelar Sarjana Teknik



Disusun oleh:

KHARISMA DARMAWAN PUTRA

NIM. 0810633058

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2012

LEMBAR PERSETUJUAN

**APLIKASI KONTROL PID UNTUK PENGATURAN POSISI MOTOR DC PADA
SISTEM *ROTARY PARKING***

SKRIPSI

KONSENTRASI TEKNIK KONTROL

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

KHARISMA DARMAWAN PUTRA

NIM. 0810633058

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Goegoes Dwi Nusantoro, ST., MT.,
NIP. 19711013 200604 1 001

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



LEMBAR PENGESAHAN
APLIKASI KONTROL PID UNTUK PENGATURAN POSISI MOTOR
DC PADA SISTEM *ROTARY PARKING*

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan Untuk Memenuhi Persyaratan

Memperoleh Gelar Sarjana Teknik

Disusun oleh:

KHARISMA DARMAWAN PUTRA

NIM. 0810633058

Skripsi ini telah diuji dan dinyatakan lulus pada

Tanggal **10 Agustus 2012**

DOSEN PENGUJI

Penguji I

Ir. Purwanto, M.Sc
NIP.19540424 198601 1 001

Penguji II

Fitriana Suhartati S.T.,M.T.
NIP.19741017 199802 2 001

Penguji III

Ir. Retnowati
NIP.19511224 198203 2 001

Mengetahui
Ketua Jurusan Teknik Elektro

DR. Ir. Sholeh Hadi Pramono, MS.
NIP. 19580728 198701 1 001



KATA PENGANTAR

Segala puji syukur penulis panjatkan hanya bagi Allah SWT, Pemelihara seluruh alam raya, yang atas limpahan rahmat, taufik dan hidayah-Nya, penulis mampu menyelesaikan Tugas Akhir ini. Tugas akhir ini dikerjakan demi memenuhi salah satu syarat guna memperoleh gelar sarjana Teknik di Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Penulis menyadari bahwa tugas akhir ini bukanlah tujuan akhir dari belajar karena belajar adalah sesuatu yang tidak terbatas. terselesaikannya skripsi ini tentunya tak lepas dari dorongan dan uluran tangan berbagai pihak. Oleh karena itu, tak salah kiranya bila penulis mengungkapkan rasa terima kasih dan penghargaan kepada:

1. Bapak Sudarmanto dan Ibu Annekewati selaku orang tua saya yang tidak pernah lelah mendukung saya.
2. Bapak Goegoes dan Bapak M. Azis Muslim selaku pembimbing I dan II.
3. Adik-adiku, Nanda, Wawan, dan Kaka yang selalu memberi semangat.
4. Candrika Citra Sari untuk celoteh dan perhatiannya yang selalu memberi semangat.
5. Teman-teman saya, Mas Aldo, Mas Bima, Rahma, Risa dan Wahyu yang selalu membantu dalam pengerjaan Tugas Akhir ini.
6. Segenap anggota asisten Lab. Sistem Kontrol, Maho, Rio, Arif, Seif, Risa dan adek-adek 2009 yang selalu membantu pengerjaan tugas akhir ini.
7. Segenap anggota tim robot, Bagus, Irfan dkk.

Semua pihak yang telah banyak membantu saya dan tidak saya sebutkan. Semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan skripsi ini dengan melimpahkan rahmat dan karunia-Nya.

Semoga karya penelitian tugas akhir ini dapat memberikan manfaat dan kebaikan bagi banyak pihak demi kemaslahatan bersama serta bernilai ibadah di hadapan Allah SWT. Amien.

Malang, 27 Juli 2012

Penulis



DAFTAR ISI**BAB I PENDAHULUAN**

1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Sistematika Pembahasan.....	3

BAB II TINJAUAN PUSTAKA

2.1 Kontroler PID.....	4
2.1.1 Kontroler Proporsional	4
2.1.2 Kontroler Integral.....	5
2.1.3 Kontroler Diferensial.....	6
2.1.4 Kontroler Proporsional Integral Diferensial (PID).....	6
2.1.5 Metode Perancangan Kontroler Integral Diferensial (PID).....	7
2.2 Motor DC Gearbox.....	10
2.3 PWM (Pulse Width Modulation).....	12
2.4 Modul Pengendali Motor DC EMS 5A H-Bridge	12
2.5 Mikrokontroler Atmega 16.....	13
2.5.1 Arsitektur AVR.....	14
2.5.2 Struktur dan Operasi Port.....	16
2.5.3 Timer/Counter	18

BAB III METODOLOGI PENELITIAN

3.1 Studi Literatur	19
3.2 Penentuan Spesifikasi Alat	19
3.3 Perancangan Sistem.....	20
3.4 Realisasi Pembuatan Sistem	20



3.4.1 Perangkat Keras.....	20
3.4.2 Perangkat Lunak.....	20
3.5 Pengujian Alat.....	21
3.6 Pengambilan Kesimpulan.....	21
BAB IV PERANCANGAN DAN PEMBUATAN ALAT	
4.1 Perancangan Sistem.....	22
4.1.1 Diagram Blok.....	22
4.1.2 Prinsip Kerja Alat.....	23
4.2 Perancangan Perangkat Keras (hardware).....	24
4.2.1 Spesifikasi Alat.....	24
4.2.2 Potensiometer Linear.....	25
4.2.3 Modul Mikrokontroler AVR Atmega 16.....	26
4.2.4 Perancangan Driver.....	28
4.3 Perancangan Kontroler.....	29
4.3.1 Tuning pada <i>Rotary Parking</i>	30
4.4 Perancangan Pergerakan <i>Rotary Parking</i>	31
4.5 Perancangan Perangkat Lunak (software).....	41
BAB V PENGUJIAN DAN ANALISIS SISTEM	
5.1 Pengujian Potensiometer.....	42
5.2 Pengujian Driver Motor DC.....	46
5.3 Pengujian Kecepatan yang Dihasilkan.....	47
5.4 Pengukuran Letak Posisi Akhir.....	48
5.5 Pengujian Keseluruhan.....	61
5.6 Pengujian Motor DC.....	62
5.7 Hasil Pengujian Ketepatan.....	65
BAB IV PENUTUP	
6.1 Kesimpulan.....	68

6.2 Saran 68

DAFTAR PUSTAKA 69

LAMPIRAN 70



DAFTAR TABEL

Tabel 2.1 Aturan Penalaran *Ziegler-Nichols* berdasarkan Respon *Unit Step* dari Plan.....9

Tabel 2.2 Aturan dasar *Ziegler-Nichols* berdasarkan *Critical Gain* Kcr dan *Critical Period* Pcr..... 10

Tabel 2.3 Fungsi Khusus Port B Atmega 16 17

Tabel 2.4 Fungsi Khusus Port C Atmega 16 17

Tabel 4.1 Fungsi Pin Mikrokontroler.....27

Tabel 4.2 Rencana Pergerakan Sistem..... 32

Tabel 5.1 Hasil Pengujian Potensiometer..... 43

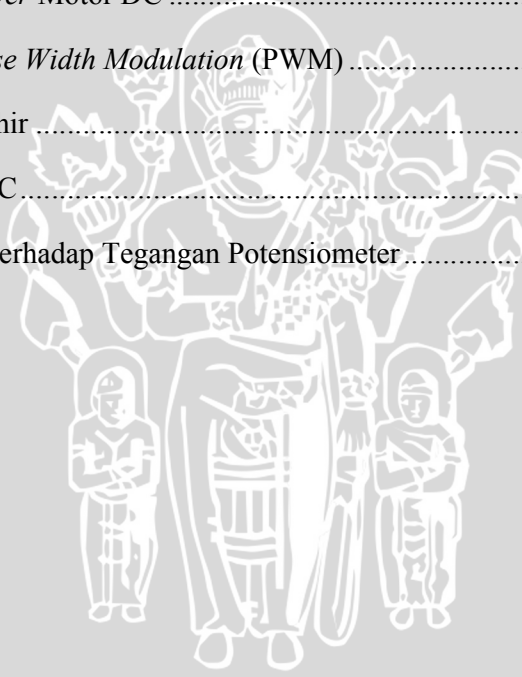
Tabel 5.2 Hasil Pengujian *Driver Motor DC* 46

Tabel 5.3 Hasil Pengujian *Pulse Width Modulation (PWM)* 48

Tabel 5.4 Pengujian Posisi Akhir 49

Tabel 5.5 Data Sudut Motor DC..... 61

Tabel 5.6 Posisi Sudut Motor terhadap Tegangan Potensiometer..... 66



DAFTAR GAMBAR

Gambar 2.1 Diagram Blok Kontroler Proporsional.....	5
Gambar 2.2 Diagram Blok Kontroler Integral.....	5
Gambar 2.3 Diagram Blok Kontroler Differensial	6
Gambar 2.4 Diagram Blok Kontroler PID	7
Gambar 2.5 Kurva Respon Unit Step Menunjukkan 25% <i>Maximum Overshoot</i>	7
Gambar 2.6 Respon Plan terhadap Masukan Berupa Unit Step	8
Gambar 2.7 Respon Plan berbentuk S.....	8
Gambar 2.8 Sistem Loop Tertutup dengan Kontroler Proporsional.....	9
Gambar 2.9 Osilasi Berkesinambungan dengan Periode <i>Pcr</i>	10
Gambar 2.10 Ilustrasi Motor DC <i>Brushed</i>	11
Gambar 2.11 Sinyal PWM Secara Umum.....	12
Gambar 2.12 Koneksi Modul EMS 5A H-Bridge	13
Gambar 2.13 Pengendali Motor DC Menggunakan EMS 5A H-Bridge	13
Gambar 2.14 Arsitektur AVR.....	15
Gambar 2.16 Konfigurasi PIN Atmega 16.....	16
Gambar 4.1 Diagram Blok Sistem <i>Rotary Parking</i>	22
Gambar 4.2 Skema Sistem Rotary Parking Tampak Depan	24
Gambar 4.3 Skema Sistem Rotary Parking Tampak Samping.....	25
Gambar 4.4 Skema Sistem Rotary Parking Tampak Atas	25
Gambar 4.5 Rangkaian Potensiometer.....	26
Gambar 4.6 Potensiometer pada Sistem.....	26
Gambar 4.7 Minimum Sistem Mikrokontroler Atmega 16.....	27
Gambar 4.8 Modul Mikrokontroler Atmega 16	28
Gambar 4.9 Koneksi Modul EMS 5A H-Bridge	28

Gambar 4.10 Pengendali Motor DC menggunakan EMS 5A H-Bridge.....	29
Gambar 4.11 Grafik <i>Tunning</i> pada Rotary Parking.....	30
Gambar 4.12 Pengambilan Data ADC pada Rotary Parking	30
Gambar 4.13 <i>Flowchart</i> Program	41
Gambar 5.1 Pengujian Potensiometer	42
Gambar 5.2 Grafik Hubungan Antara Sudut dengan Resistansi	44
Gambar 5.3 Grafik Hubungan Antara Sudut dengan Tegangan.....	44
Gambar 5.4 Grafik Antara Tegangan Perhitungan dengan Tegangan Pengukuran terhadap Sudut	45
Gambar 5.5 Grafik Antara Tegangan Perhitungan dengan Tegangan Pengukuran terhadap Resistansi.....	45
Gambar 5.6 Diagram Blok Pengujian <i>Driver</i> Motor DC.....	46
Gambar 5.7 Hubungan Nilai PWM dan RPM.....	47
Gambar 5.8 Pergerakan rotary Parking dengan Acuan Garasi 1	61
Gambar 5.9 Respon Sistem dengan Tujuan 60.....	62
Gambar 5.10 Respons Sistem dengan tujuan 120.....	62
Gambar 5.11 Respons Sistem dengan Tujuan 180	63
Gambar 5.12 Respons Sistem dengan Tujuan 240	63
Gambar 5.13 Respons Sistem dengan Tujuan 300	63
Gambar 5.13 Respons Sistem dengan Tujuan Acak.....	64



ABSTRAK

Kharisma Darmawan Putra, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Agustus 2012, Aplikasi Kontrol PID untuk Pengaturan Posisi Motor DC pada Sistem *Rotary Parking*, Dosen Pembimbing : Goegoes Dwi Nusantoro, ST., MT., M. Azis Muslim, ST., MT., Ph.D

Saat ini, pengguna mobil semakin meningkat. Sedangkan luas lahan parkir tidak bertambah. Salah satu solusi untuk mengatasi permasalahan tersebut dengan menggunakan sistem *rotary parking*, yaitu sistem parkir otomatis dengan menggunakan lahan yang relatif sempit namun dapat menampung mobil dengan jumlah yang lebih banyak. Metode kontrol yang dapat digunakan pada sisten *rotary parking* adalah metode control PID. Salah satu keuntungan kontrol PID adalah memiliki respons yang halus dan cepat. Pada skripsi ini digunakan metode *Zieger Nichlos* untuk proses *tunning*. Dalam perencanaan pembuatannya digunakan mikrokontroler Atmega16, potensiometer linier, motor DC, komunikasi serial RS-232. Dari hasil pengujian terhadap aplikasi kontroler PID ini didapat $K_p = 4.8$, untuk $K_i = 8.9$, $K_d = 0.6$ dan menunjukkan bahwa respons sistem untuk pergerakan garasi 1,2,3,4,5,6 mempunyai *error steady state* sebesar 0-0.416%. Sedangkan T_s untuk sudut $60^\circ = 1.169s$, sudut $120^\circ = 2.2415s$, sudut $180^\circ = 3.507s$, sudut $240^\circ = 4.342s$, dan sudut $300^\circ = 5.331s$. Dari pengujian juga didapatkan toleransi kesalahan sebesar 2%-10%.

Kata kunci: Rotary Parking, Mikrokontroler Atmega16, PID, Motor.

BAB I PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan zaman, perkembangan alat transportasi pribadi semakin berkembang pesat. Alat transportasi yang cukup berkembang salah satunya adalah mobil. Karena bentuk dan modelnya yang beraneka ragam, serta harga yang semakin terjangkau sehingga membuat alat transportasi ini semakin banyak digunakan. Akan tetapi saat ini penambahan jumlah pengguna mobil tidak disertai dengan penambahan luas lahan parkir yang tersedia terutama pada daerah kota besar, sehingga sering menimbulkan masalah bagi pengguna mobil.

Salah satu solusi untuk mengatasi permasalahan di atas adalah pembuatan sistem *rotary parking*, yaitu sistem parkir otomatis dengan menggunakan lahan yang relatif kecil namun dapat menampung mobil dengan jumlah yang lebih banyak. Sistem ini terbukti mampu mengatasi keterbatasan jumlah lahan parkir yang relatif sempit. Prinsip kerja dari sistem ini adalah mobil diparkir ke dalam garasi lewat sebuah gerbang dan digerakkan ke atas sementara garasi yang kosong diarahkan ke gerbang agar pengguna parkir yang lain dapat menggunakan garasi yang kosong dengan cepat. Sedangkan untuk pengambilan mobil yang telah di parkir, telah disediakan *push button* untuk masing-masing garasi, hal ini memudahkan pengguna karena pengguna dapat menekan tombol tersebut sesuai dengan nomor garasi parkir mobinya.

Pada penelitian ini akan dirancang bangun sebuah sistem parkir menggunakan kontroler PID. Karena respons sistem yang halus sehingga sangat berpengaruh saat awal motor berjalan dan saat terjadi pengereman disaat akan berhenti.

Pada penelitian ini akan dibuat miniatur *rotary parking* yang mendekati kondisi aslinya. Sistem *rotary parking* ini membutuhkan alat yang dapat menggerakkan posisi penyimpanan mobil. Salah satu alat yang dapat digunakan sebagai penggerak adalah motor DC.

1.2 Rumusan Masalah

Mengacu pada permasalahan yang disebutkan pada latar belakang, maka ditetapkan rumusan masalah sebagai berikut :

- 1) Merancang sistem *rotary parking* dengan system elektriknya menggunakan modul mikrokontroller Atmega 16 dengan bahasa pemrograman CV-AVR.
- 2) Mengatur posisi motor DC dan garasi berdasarkan kontrol PID agar sesuai dengan posisi sistem *rotary parking*.

1.3 Batasan Masalah

Karena luasnya objek kajian maka perlu dilakukan pembatasan masalah agar pembahasan lebih terfokus pada rumusan masalah. Adapun batasan masalah dalam skripsi ini adalah:

- 1) Sistem parkir otomatis yang dibuat merupakan sebuah miniatur, yang dirancang dan dibuat seperti aslinya.
- 2) Pembahasan mekanik serta model matematis tidak dijabarkan, pembahasan ditekankan pada penggunaan kontrol PID pada sistem.
- 3) Sistem elektrik menggunakan modul mikrokontroler ATmega 16.
- 4) Dimensi dan berat mobil ditentukan menggunakan replika.
- 5) Dalam satu sistem *rotary parking* dapat menampung 6 mobil.
- 6) Menggunakan sensor *limit switch* sebagai pendeteksi ada tidaknya mobil.
- 7) Menggunakan motor DC yang difungsikan sebagai servo
- 8) Dalam proses parkir mobil, sistem ini mulai bekerja ketika seseorang sudah keluar dari garasi.
- 9) Jika dalam proses pengambilan mobil terdapat dua orang yang akan mengambil mobilnya, maka orang tersebut harus bergantian dengan menunggu orang pertama telah mengambil mobilnya.

1.4 Tujuan

Tujuan dari sistem ini adalah merancang miniatur sistem *rotary parking* yang dapat menyesuaikan kecepatan saat motor mulai bekerja hingga motor berhenti, dengan tetap mempertahankan keluaran sistem sesuai *setpoint* dengan kontroler PID.

1.5 Sistematika Pembahasan

Agar penyusunan laporan skripsi ini dapat mencapai sasaran dan tidak menyimpang dari judul yang telah ditentukan, maka diperlukan sistematika pembahasan yang jelas. Pembahasan dalam skripsi ini sebagai berikut :

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah dan sistematika pembahasan.

BAB II Tinjauan Pustaka

Menjelaskan dasar teori penunjang penelitian yang ada pada alat ini, yang terdiri dari teori Mikrokontroller ATMEGA 16 sebagai sistem otomasi, sensor *limit switch*, sensor *potensiometer*, *led*, *push button*, saklar, motor dc, serta kontrol PID.

BAB III Metodologi

Berisi tentang metode penelitian dan perencanaan alat serta pengujian.

BAB IV Perancangan Sistem

Membahas perancangan alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja dan pembuatan alat. Setelah itu, bagaimana penerapannya dalam sistem secara keseluruhan.

BAB V Pengujian Alat

Memuat hasil pengujian terhadap alat yang telah dibuat.

BAB VI Kesimpulan dan Saran

Memuat kesimpulan dan saran-saran

BAB II
TINJAUAN PUSTAKA

2.1 Kontroler PID

Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Disamping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolok ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata.K.,1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespons *input*, dan bagaimana peredaman terhadap adanya lonjakan (*overshoot*).

Stabilitas umum sistem dinamis tanpa input dapat digambarkan dengan kriteria stabilitas Lyapunov. Sebuah sistem linier yang mengambil sebuah input disebut stabilitas *bounded-input bounded-output* (BIBO) jika outputnya akan tetap dalam batas untuk setiap input yang dibatasi. (Ogata.K.,1997).

Dalam sistem pengendalian terdapat 2 macam *loop*:

1. Pengendalian dengan *loop* terbuka

Sistem kontrol *loop* terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolan. Jadi pada sistem kontrol *loop* terbuka, keluaran tidak diukur atau diumpkan balikan untuk dibandingkan dengan masukan.

2. Pengendalian dengan *loop* tertutup

Sistem kontrol *loop* tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolan. Disebut juga sistem kontrol yang menggunakan umpan balik untuk memperkecil kesalahan sistem.

2.1.1 Kontroler Proporsional

Untuk kontroler dengan aksi kontrol proporsional, hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah:

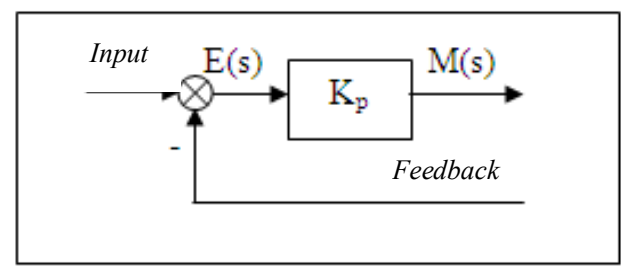
$$m(t) = K_p e(t) \dots\dots\dots(2-1)$$

atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = \dots\dots\dots(2-2)$$

Dimana K_p adalah kepekaan proporsional atau penguatan.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggerak, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata.K.,1997). Diagram blok kontroler proporsional ditunjukkan pada Gambar 2.1.



Gambar 2.1 Diagram Blok Kontroler Proporsional

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

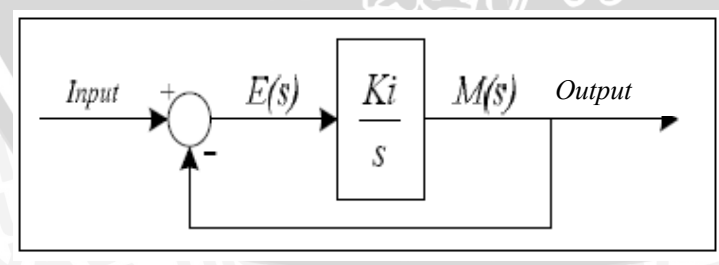
2.1.2 Kontroler Integral

Pada kontroler dengan aksi integral, harga keluaran kontroler $m(t)$ diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak $e(t)$.

Jadi,

$$\frac{M(s)}{E(s)} = \frac{K_i}{s} \quad (2-3)$$

Jika harga $e(t)$ diduakalikan, maka harga $m(t)$ berubah dengan laju perubahan menjadi dua kali semula. Jika kesalahan penggerak nol, maka harga $m(t)$ tetap stasioner. Aksi kontrol integral seringkali disebut kontrol *reset* (Ogata.K.,1997). Gambar 2.2 menunjukkan diagram blok kontroler integral.



Gambar 2.2 Diagram Blok Kontroler Integral

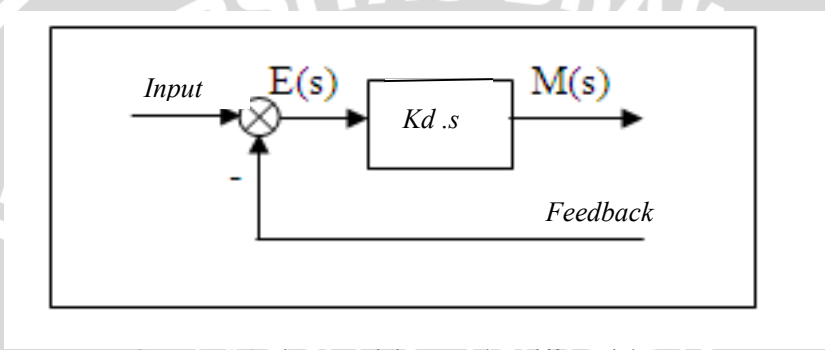
Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

2.1.3 Kontroler Differensial

Kontroler ini digunakan untuk memperbaiki atau mempercepat respons transien sebuah sistem kontrol dengan cara memperbesar *phase lead* terhadap penguatan kontrol dan mengurangi *phase lag* penguatan tersebut (Ogata.K.,1997). Kontroler differensial tidak dapat mengeluarkan *output* bila tidak ada perubahan *input*, selain itu kontroler differensial tidak dapat digunakan untuk proses yang mengandung *noise*. Hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan penggerak $e(t)$ adalah :

$$\dots\dots\dots(2-4)$$

Gambar 2.3 menunjukkan diagram blok kontroler differensial.



Gambar 2.3 Diagram Blok Kontroler Differensial
 Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

2.1.4 Kontroler Proporsional Integral Differensial (PID)

Gabungan aksi kontrol proporsional, integral, dan differensial mempunyai keunggulan dibandingkan dengan masing-masing dari tiga aksi kontrol tersebut. Persamaan kontroler PID ini dapat dinyatakan sebagai berikut :

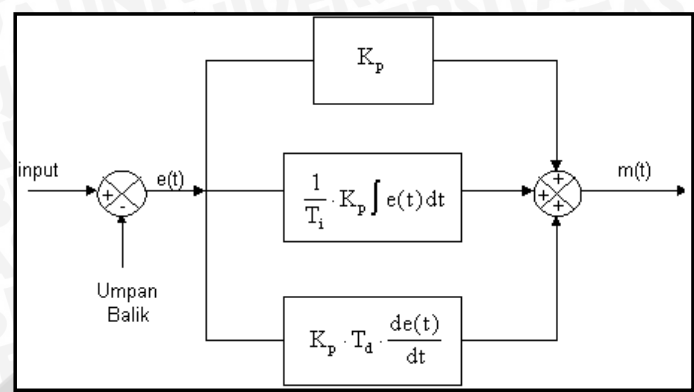
$$\dots\dots\dots(2-5)$$

Dalam transformasi Laplace dinyatakan sebagai berikut :

$$\dots\dots\dots(2-6)$$

(Ogata.K.,1997)

Gambar 2.4 menunjukkan diagram blok kontroler PID.



Gambar 2.4 Diagram Blok Kontroler PID

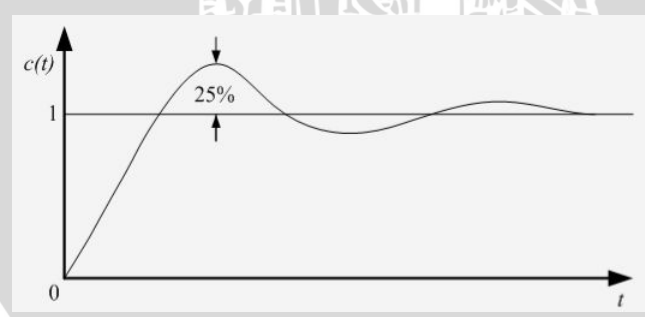
Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

2.1.5 Metode Perancangan Kontroler Proporsional Integral Diferensial (PID)

Menggunakan Metode Ziegler-Nichols.

Ziegler dan Nichols mengemukakan aturan-aturan untuk menentukan nilai dari *gain* proporsional K_p , waktu integral T_i , dan waktu derivatif T_d berdasarkan karakteristik respon transien dari plan yang diberikan. Penentuan parameter kontroler PID atau penalaan kontroler PID tersebut dapat dilakukan dengan bereksperimen dengan plan. (Ogata, K., 1997)

Terdapat dua metode yang disebut dengan aturan penalaan *Ziegler-Nichols*, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai *25% maximum overshoot* pada respon unit step, seperti ditunjukkan dalam Gambar 2.5



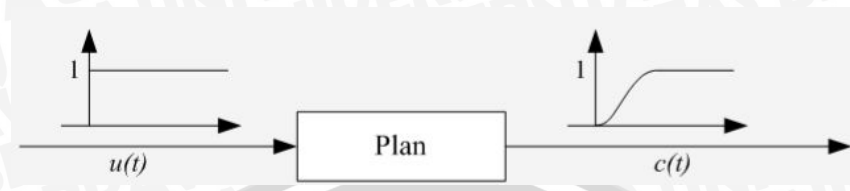
Gambar 2.5 Kurva Respons Unit Step yang Menunjukkan 25% Maximum Overshoot

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

- a) Metode Pertama

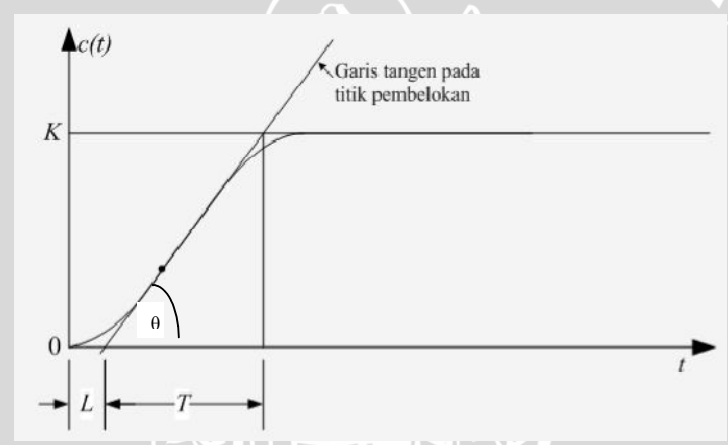
Metode pertama atau sering disebut metode kurva reaksi, respon dari plan dapat dapat

diperoleh secara eksperimental dengan masukan berupa unit *step*, seperti yang ditunjukkan dalam Gambar 2.6



Gambar 2.6 Respons Plan Terhadap Masukan Berupa Unit Step
 Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

Jika dalam plan tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respons unit *step* berbentuk seperti huruf S, seperti dalam Gambar 2.7. Jika respons tidak memberikan bentuk kurva S, maka metode ini tidak berlaku.(Ogata, K., 1997).



Gambar 2.7 Respons Plan berbentuk S
 Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu $c(t) = K$, seperti yang telah ditunjukkan dalam Gambar 2.7.

Fungsi alih $C(s)/U(s)$ dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\dots\dots\dots(2-7)$$

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari K_p , T_i dan T_d berdasarkan pada formula yang ditunjukkan dalam Tabel 2.1 (Ogata, K., 1997)

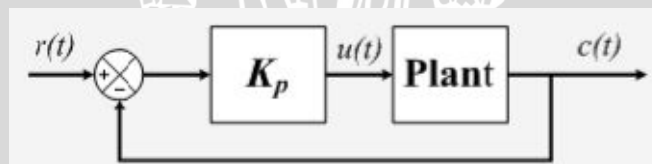
Tabel 2.1 Aturan Penalaan *Ziegler-Nichols* Berdasarkan Respon Unit Step Dari Plan

Tipe Kontroler			
P	–	∞	0
PI	0,9–	$\frac{1}{0,3}$	0
PID	1,2 –	$2L$	$0,5 L$

Sumber: Ogata, K. 1997

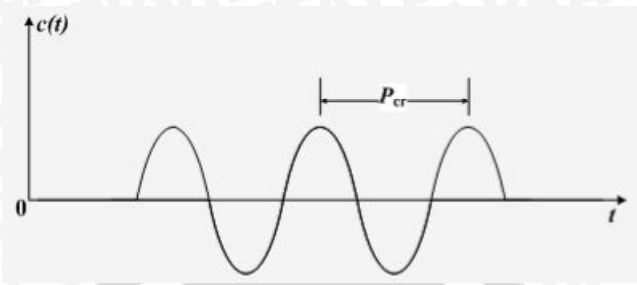
b) Metode Kedua

Dalam metode kedua *Ziegler-Nichols*, mula-mula yang dilakukan adalah membuat $T_i = 0$ dan $T_d = 0$. Kemudian hanya dengan menggunakan tindakan kontrol proporsional, harga ditingkatkan dari nol ke suatu nilai kritis K_{cr} , disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika keluaran tidak memiliki osilasi berkesinambungan untuk nilai K_p manapun yang telah diambil, maka metode ini tidak berlaku). Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis K_{cr} dan periode P_{cr} dapat ditentukan. Diagram blok sistem loop tertutup dengan kontroler proporsional dapat dilihat dalam Gambar 2.8 dan untuk osilasi berkesinambungan dengan periode P_{cr} dapat dilihat dalam gambar 2.9. Ziegler dan Nichols menyarankan penyetelan nilai parameter K_p, T_i, T_d dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.2. (Ogata, K., 1997)



Gambar 2.8 Sistem Loop Tertutup dengan Kontroler Proporsional

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997



Gambar 2.9 Osilasi Berkesinambungan dengan periode P_{cr}
 Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

Tabel 2.2 Aturan Dasar Ziegler-Nichols Berdasarkan Critical Gain K_{cr} dan Critical Period P_{cr}

Tipe Kontroler			
P	$0.5 K_{cr}$	∞	0
PI	$0.45 K_{cr}$	$\frac{1}{2} P_{cr}$	0
PID	$0.60 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

Sumber: Teknik Kontrol Automatik, Katsuhiko Ogata, 1997

2.2 Motor DC Gearbox

Prinsip kerja motor DC *brushed* sesuai dengan hukum kemagnetan Lorenz, yaitu membangkitkan fluksi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul ggl induksi. Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri.

Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah gaya putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dengan :

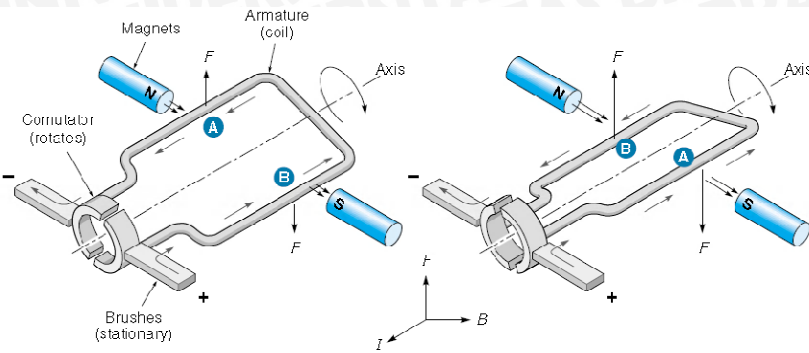
$$F=B.I.L \text{ (Newton)} \dots\dots\dots(2.8)$$

Dimana :

B = kerapatan fluks magnet (weber)

L = panjang konduktor (meter)

I = arus listrik (ampere)



Gambar 2.10. Ilustrasi Motor DC Brushed

Sumber : Kilian, 2002

Gambar 2.9 adalah ilustrasi cara kerja motor DC yang mempunyai satu lilit kawat a–b berada di dalam medan magnet. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (*armature*).

Pada jangkar diberikan arus yang berasal dari sumber yang terhubung dengan sikat (*brushes*). Sikat-sikat ini terpasang pada sebuah cincin yang terbelah dua, yang disebut cincin belah (*comutator*). Adapun tujuan dari konstruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatinya.

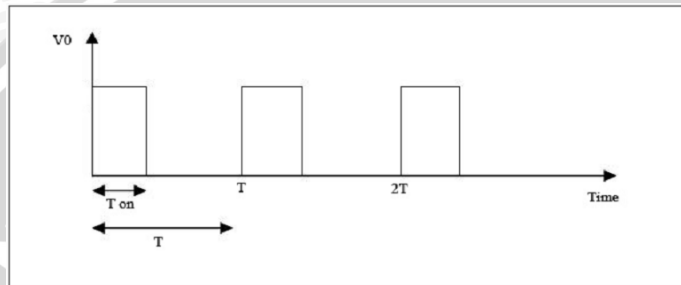
Pada kawat yang berada di kanan arus mengalir dari depan ke belakang. Pada kawat yang berada di bagian kiri, arus mengalir dari belakang ke depan kawat a dan b secara bergantian berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu bersifat bolak-balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi posisi vertikal.

Bagian *comutator* berfungsi sebagai penyearah mekanik. Fluksi magnet yang ditimbulkan magnet permanen disebut medan magnet motor. Dalam gambar 2.4 arah fluk magnetik adalah dari kiri ke kanan. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah . Gaya-gaya yang bekerja sama kuatnya, sehingga terdapat kopel yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar 180° arah arus berbalik, pada saat itu penghantar a dan penghantar b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.

2.3 PWM (Pulse Width Modulation)

PWM (Pulse Width Modulation) digunakan untuk mengatur kecepatan dari motor DC. Dimana kecepatan motor DC tergantung pada besarnya *Duty cycle* yang diberikan pada motor DC tersebut.

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *Duty cycle* bervariasi dari 0%-100%. Dengan mengatur *Duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM (Pulse width Modulation) secara umum dapat dilihat dalam Gambar 2.10



Gambar 2.11 Sinyal PWM Secara Umum
 Sumber: www.electronics-scheme.com

— ... (%)(2-9)

Dengan :

- T on = Periode logika tinggi
- T = Periode keseluruhan

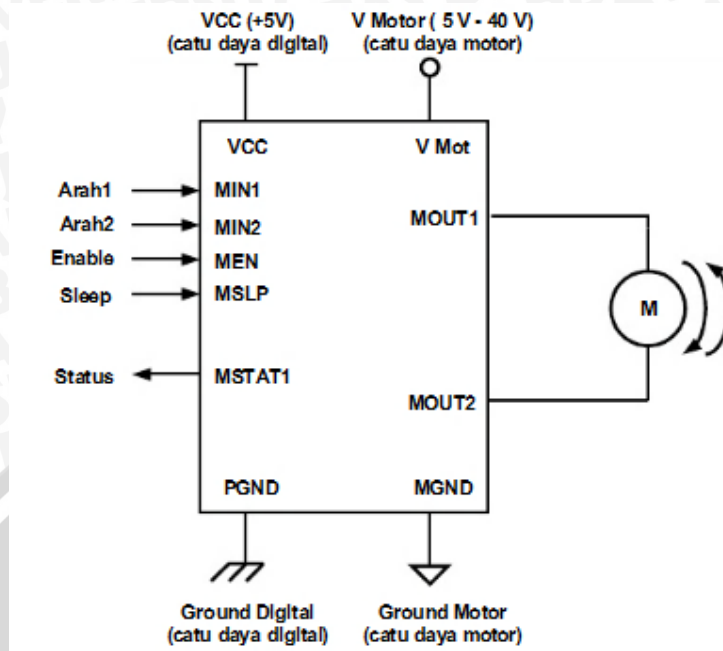
...(V)(2-10)

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut :

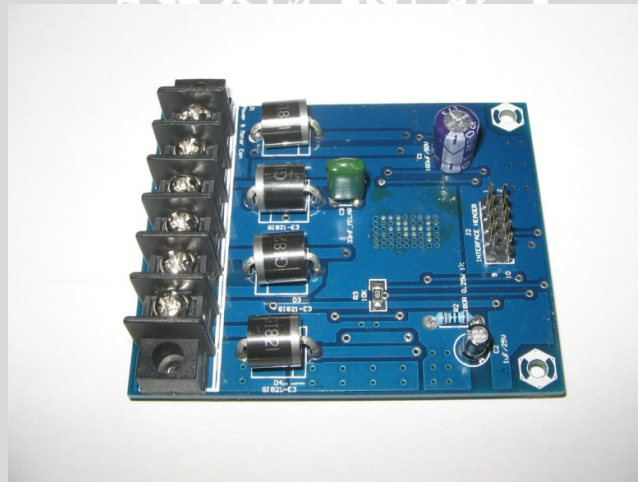
— ... (Hz)(2-11)

2.4 Modul Pengendali Motor DC EMS 5A H-Bridge

Modul pengendali motor DC yang digunakan adalah modul EMS 5A H-Bridge. Secara garis besar, fungsi modul pengendali motor ini adalah untuk mengendalikan arah dan kecepatan putaran motor DC sesuai instruksi kendali dari mikrokontroler pengguna. Gambar koneksi dari modul pengendali motor DC EMS 5A H-Bridge ditunjukkan pada Gambar 2.11



Gambar 2.12 Koneksi Modul EMS 5A H-Bridge



Gambar 2.13 Pengendali Motor DC Menggunakan EMS 5A H-Bridge

2.5 Mikrokontroler Atmega16

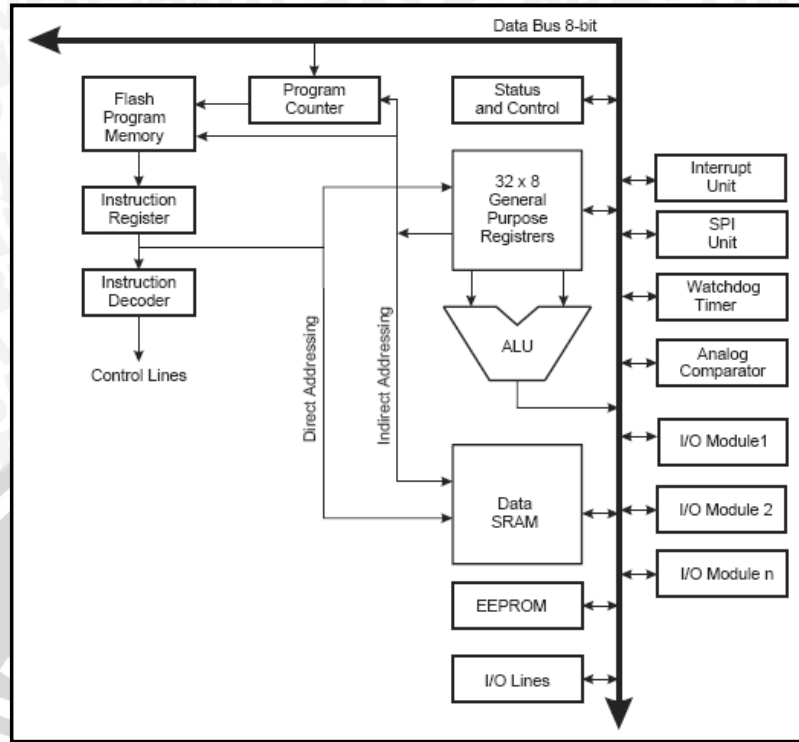
Mikrokontroler Atmega 16 merupakan mikrokontroler CMOS 8-bit buatan Atmel keluarga AVR. AVR mempunyai 32 register general-purpose, timer/counter dengan metode compare, interrupt eksternal dan internal, serial USART, programmable Watchdog Timer, ADC dan PWM internal. Karakteristik utama yang dimiliki oleh mikrokontroler Atmega 16 adalah:

1. Saluran Input/Output (I/O) ada 32 buah, yaitu PORTA, PORTB, PORTC, PORTD
2. ADC / Analog to Digital Converter 10 bit sebanyak 8 channel pada PORTA
3. 2 buah timer/counter 8-bit dan 1 buah timer/counter 16-bit dengan prescalers dan kemampuan pembanding
4. Watchdog timer dengan osilator internal
5. Tegangan operasi 2,75 - 5,5 V pada ATmega16L dan 4,5 - 5,5 V pada ATmega16
6. EEPROM sebesar 512 byte yang dapat diprogram saat operasi
7. Antarmuka komparator analog
8. 4 channel PWM
9. kecepatan nilai (speed grades) 0 - 8 MHz untuk ATmega16L dan 0 - 16 MHz untuk ATmega16

2.5.1 Arsitektur AVR

Mikrokontroler AVR menggunakan arsitektur Harvard yang memisahkan memori dan bus untuk program dan data sehingga memaksimalkan performa. Instruksi pada memori program dieksekusi secara *pipeline*. Ketika satu instruksi masih dieksekusi, instruksi selanjutnya sudah disiapkan untuk eksekusi tanpa menunggu eksekusi pertama selesai. Konsep ini memungkinkan instruksi untuk dieksekusi setiap satu siklus waktu.

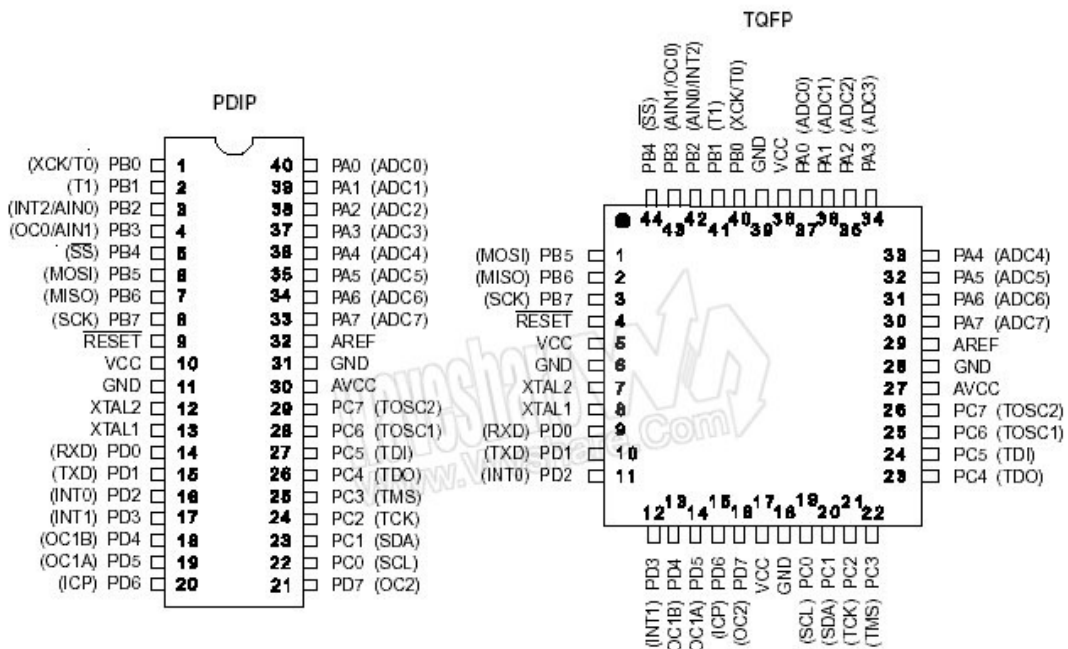
Register file terdiri atas 32 x 8 bit *general purpose working register* dengan waktu akses satu siklus waktu, yang memungkinkan operasi ALU (*Arithmetic Logic Unit*) dijalankan dalam satu siklus waktu. Dua *operand* diambil dari register, operasi ALU dijalankan, dan hasilnya disimpan kembali dalam *register file*. Operasi aritmetika dan logika dapat dijalankan oleh ALU, baik antar register atau antara register dengan konstanta. Hasil dari operasi aritmetika disimpan dalam *register status (Status Register)*, menggantikan isi yang sebelumnya. Arsitektur AVR ditunjukkan dalam Gambar 2.13.



Gambar 2.14 Arsitektur AVR

Sumber : Atmel, 2004 : 7

Program *Flash memory* dibagi dalam dua bagian, *Boot Program* dan *Application Program*. Kedua bagian memiliki *Lock bits* untuk mengunci operasi tulis (*write*) dan baca/tulis (*read/write*). Konfigurasi pin mikrokontroler Atmega16 dapat dilihat dalam Gambar 2.14.



Gambar 2.15 Konfigurasi Pin Atmega16

Sumber : Atmel, 2004 : 2

2.5.2 Struktur dan Operasi Port

Mikrokontroler Atmega 16 ini mempunyai 4 buah port, yang memiliki 8 buah jalur I/O. Beberapa karakteristik port mikrokontroler Atmega 16 dijelaskan secara singkat:

- 1). Unit I/O dapat dialamati perjalur atau per port
- 2). Setiap jalur I/O memiliki *buffer*, penahan (*latch*), kemudi input dan kemudi output.
- 3). Setiap jalur I/O terdapat register pengatur apakah dijadikan input atau dijadikan output.
- 4). Port B adalah I/O *bi-directional* 16 bit dengan resistor *pull-up* internal. Sebagai masukan, pin port B yang diberi *pull-low* secara eksternal akan mengalirkan arus bila resistor *pull-up* diaktifkan. Port B juga memiliki fungsi khusus, seperti yang terlihat dalam Tabel 2.3.

Tabel 2.3 Fungsi Khusus Port B Atmega 16

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1)
PB5	SCK (SPI Bus Master clock Input)
PB4	MISO (SPI Bus Master Input/Slave Output)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2 (Timer/Counter2 Output Compare Match Output)
PB2	\overline{SS} (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output)
PB0	ICP1 (Timer/Counter1 Input Capture Pin)

Sumber : Atmel, 2004 : 56

- 5). Port C adalah I/O *bi-directional* 8 bit dengan resistor *pull-up* internal. Sebagai masukan, pin port C yang diberi *pull-low* secara eksternal akan mengalirkan arus bila resistor *pull-up* diaktifkan. Port C juga memiliki fungsi khusus, seperti yang terlihat dalam Tabel 2.4.

Tabel 2.4 Fungsi Khusus Port C Atmega 16

Port Pin	Alternate Function
PC6	RESET (Reset pin)
PC5	ADC5 (ADC Input Channel 5) SCL (Two-wire Serial Bus Clock Line)
PC4	ADC4 (ADC Input Channel 4) SDA (Two-wire Serial Bus Data Input/Output Line)
PC3	ADC3 (ADC Input Channel 3)
PC2	ADC2 (ADC Input Channel 2)
PC1	ADC1 (ADC Input Channel 1)
PC0	ADC0 (ADC Input Channel 0)

Sumber : Atmel, 2004 : 59

- 6). Port D adalah I/O *bi-directional* 8 bit dengan resistor *pull-up* internal. Sebagai masukan, pin port D yang diberi *pull-low* secara eksternal akan mengalirkan arus bila resistor *pull-up* diaktifkan. Port D juga memiliki fungsi khusus, seperti yang terlihat dalam Tabel 2.5.

Tabel 2.5 Fungsi Khusus Port D Atmega 16

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input)
PD6	AIN0 (Analog Comparator Positive Input)
PD5	T1 (Timer/Counter 1 External Counter Input)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

Sumber : Atmel, 2004 : 61

2.5.3 Timer/Counter

Mikrokontroler Atmega 16 memiliki 3 buah *timer/counter* yang terdiri atas 2 buah *timer/counter* 8 bit dan 1 buah *timer/counter* 16 bit. Ketiga *timer/counter* ini dapat diatur dalam mode yang berbeda. Selain itu semua *timer/counter* dapat difungsikan sebagai sumber interupsi. *Timer/counter* dapat digunakan dalam 4 mode operasi, yaitu :

- 1). Mode pertama (mode 0) adalah mode normal, timer digunakan sebagai pencacah tunggal yang dapat mencacah dari 0x00 sampai dengan 0xFF. Setelah mencapai nilai 0xFF maka register *counter* akan reset atau kembali ke 0x00.
- 2). Mode kedua (mode 1) adalah *Phase Correct PWM* (PCP). Mode ini digunakan untuk menghasilkan sinyal PWM dimana nilai register *counter* yang mencacah naik dan turun secara terus menerus akan selalu dibandingkan dengan register pembanding OCRn. Hasil perbandingan register *counter* dan OCRn digunakan untuk membangkitkan sinyal PWM yang dikeluarkan pada pin OCn.
- 3). Mode ketiga (mode 2) adalah *clear timer on compare match* (CTC). Register *counter* akan mencacah naik kemudian akan direset atau kembali menjadi 0x00 pada saat nilai TCNT sama dengan OCRn.
- 4). Mode keempat (mode 3) adalah *fast PWM*. Mode ini hampir sama dengan mode *phase correct PWM*, hanya perbedaannya adalah register *counter* mencacah naik saja dan tidak mencacah turun.

BAB III

METODOLOGI PENELITIAN

Kajian dalam skripsi ini merupakan penelitian yang bersifat aplikatif, yaitu merancang suatu sistem pengendalian menggunakan kontroler PID yang bertujuan agar dapat menampilkan performansi sistem sesuai dengan yang direncanakan. Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut:

3.1 Studi Literatur

Studi literatur mengacu pada prinsip kerja alat yang dirancang meliputi sistem *rotary parking* menggunakan kontroler PID, karakteristik dari komponen yang digunakan serta studi tentang pengendalinya yaitu menggunakan mikrokontroler Atmega 16 sebagai pusat pengontrol dengan kendali kontroler PID. Adapun hal-hal yang berhubungan dengan hal tersebut adalah :

- a. Studi tentang proses pengendalian Motor DC
 - Karakteristik motor DC
- b. Studi tentang rangkaian sensor
 - Karakteristik spesifikasi sensor Potensiometer.
 - Karakteristik spesifikasi sensor *Limit Switch*.
- c. Kontroler PID
 - Karakteristik umum kontroler PID.
 - Teknik pemrograman *CV-AVR* Atmega 16 sebagai pusat pengontrol kendali PID.

3.2 Penentuan Spesifikasi Alat

Spesifikasi alat secara global ditentukan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Dengan perencanaan sebagai berikut :

1. Motor DC sebagai penggerak *rotary parking*.
2. Sensor *Limit Switch* sebagai pendeteksi ada tidaknya mobil.
3. Sensor Potensiometer sebagai pendeteksi garasi.
4. Mikrokontroler Atmega 16 sebagai tempat proses data sistem.
5. *Push button* sebagai memilih garasi yang dikehendaki dan sebagai indikator posisi penumpang.
6. *LED* sebagai indikator garasi telah terisi atau belum.

3.3 Perancangan Sistem

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem pada *rotary parking* tersebut dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan.

Perancangan sistem yang dilakukan meliputi:

- Penentuan motor DC yang digunakan dalam model *rotary parking*.
- Penentuan spesifikasi sistem yang akan dibuat, meliputi :
 - a. Penentuan deskripsi kerja sistem secara keseluruhan.
 - b. Kontroler PID dan rangkaian elektronik pendukung.
- Perancangan program pada *CodeVisionAVR* Atmega 16 sebagai pusat pengontrol dengan kontrol PID.

3.4 Realisasi Pembuatan Sistem

3.4.1 Perangkat Keras

Komponen utama perangkat keras adalah motor DC, sensor posisi mobil menggunakan *limit switch* dan *potensiometer linier*, *push button*, dan mikrokontroler Atmega16.

3.4.2 Perangkat Lunak

Pembuatan perangkat lunak (*software*) dilakukan dengan pembuatan *flowchart* terlebih dahulu, kemudian dilakukan perancangan program pada *CodeVisionAVR* ATmega 16 sebagai pusat pengontrol dengan kendali PID.

3.5 Pengujian Alat

Setelah semua komponen pada alat sudah terhubung sesuai dengan diagram blok sistem yang telah dirancang dan perangkat lunak untuk mendukung sistem telah dibuat, maka diadakan pengujian dan analisa alat. Metode pengujian alat adalah sebagai berikut

- Menguji pada tiap-tiap blok rangkaian.
- Menguji sensor yang digunakan.
- Menggabungkan beberapa blok rangkaian menjadi sebuah sistem.
- Mengadakan pengujian sistem secara keseluruhan.
- Mengevaluasi hasil pengujian sistem secara keseluruhan

3.6 Pengambilan Kesimpulan

Kesimpulan didapat berdasarkan dari hasil perealisasiian alat miniatur *rotary parking*. Kesimpulan diambil berdasarkan data yang diperoleh dari pengujian sistem secara keseluruhan. Jika hasil yang didapatkan telah sesuai dengan yang direncanakan sebelumnya, maka sistem kendali tersebut telah berhasil memenuhi harapan dan tentunya memerlukan pengembangan lebih lanjut untuk penyempurnaan.



BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

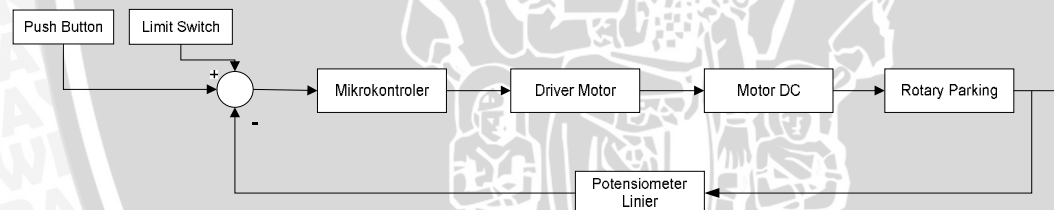
4.1 Perancangan Sistem

Perancangan alat ini dilakukan secara bertahap dalam bentuk blok sehingga akan memudahkan dalam analisis pada setiap bloknya maupun secara keseluruhan. Perancangan ini terdiri dari:

- Cara kerja alat.
- Spesifikasi alat.
- Perancangan perangkat keras (perancangan sensor potensiometer, modul pengendali motor DC, rangkaian mikrokontroler pengatur motor DC).
- Perancangan perangkat lunak (perancangan algoritma kontrol PID pada *CodeVisionAVR* Atmega 16).

4.1.1 Diagram blok

Perancangan algoritma Kontrol PID pada mikrokontroler. Blok sistem yang dirancang ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram Blok Sistem *Rotary Parking*

Keterangan dari blok diagram diatas adalah sebagai berikut :

- Potensiometer berfungsi untuk mengukur besarnya posisi sudut motor DC saat itu. Potensiometer yang digunakan adalah potensiometer linier 10 K Ω yang diberi tegangan sebesar 5 V dan akan memberikan perubahan tegangan yang dibaca oleh pin ADC *digital signal processor*.
- Pusat pengendalian sistem menggunakan Atmega 16 yang memberikan keluaran berupa *duty cycle* PWM kepada *driver* motor.
- Motor yang digunakan adalah motor DC, yang berfungsi sebagai aktuator. *Driver* motor menggunakan modul EMS 5A H-Bridge.

4.1.2 Prinsip kerja alat

Cara kerja alat adalah sebagai berikut :

- Menggunakan catu daya sebesar 12 volt
- Potensiometer sebagai sensor posisi motor DC. Potensiometer akan memberikan keluaran berupa level tegangan yang berubah-ubah sesuai dengan posisi garasi pada *rotary parking*.
- Masukan sistem berasal dari *limit switch* dan *push button*. *Limit switch* dipasang di setiap garasi, berfungsi sebagai sensor pendeteksi ada tidaknya mobil pada suatu garasi, sehingga sebagai penentu *set point* pada saat proses memposisikan garasi kosong ke bawah. *Push button* berfungsi sebagai penentu *set point* pada proses pengambilan mobil. Seorang pengguna harus menekan *push button* yang bernomor sama dengan garasi tempat parkir mobilnya, sehingga garasi yang bernomor sesuai akan diposisikan di dasar.
- Ketika *limit switch* atau *push button* aktif maka mikrokontroler akan memproses *limit switch* atau *push button* yang aktif, kemudian menentukan *set point* dari sistem. Mikrokontroler akan membandingkan *set point* tersebut dengan besarnya tegangan yg diperoleh dari potensiometer linier, kemudian diproses dan diolah untuk menghasilkan data yang diinginkan. Selanjutnya mikrokontroler akan memerintahkan *driver* motor DC untuk menggerakkan motor ke kanan atau ke kiri dengan kecepatan sesuai dengan kondisi dan masukan dari *limit switch* ataupun *push button* dan potensiometer.
- Jika terdapat lebih dari satu *limit switch* yang mendeteksi bahwa tidak ada mobil didalam garasi, maka *limit switch* yang dipilih adalah *limit switch* yang bernomor paling kecil.
- Proses tersebut berjalan terus sampai didapat posisi sudut putar motor DC yang diinginkan.
- Pada saat mobil sudah diparkir kedalam garasi, maka pengguna harus keluar dari garasi dan menekan tombol *start* untuk memulai kerja sistem. Hal ini dilakukan untuk memastikan bahwa tidak ada orang yang masih ada didalam mobil, selain itu juga untuk memulai menggerakkan sistem megarahkan garasi yang kosong ke posisi dasar.
- Jika dalam penggunaannya terdapat 2 orang atau lebih secara bersamaan ingin mengambil mobilnya, maka kedua orang tersebut harus bergantian. Orang pertama terlebih dahulu menekan *push button* kemudian mengambil mobilnya, setelah itu orang berikutnya dapat melakukan langkah yang sama seperti orang pertama.

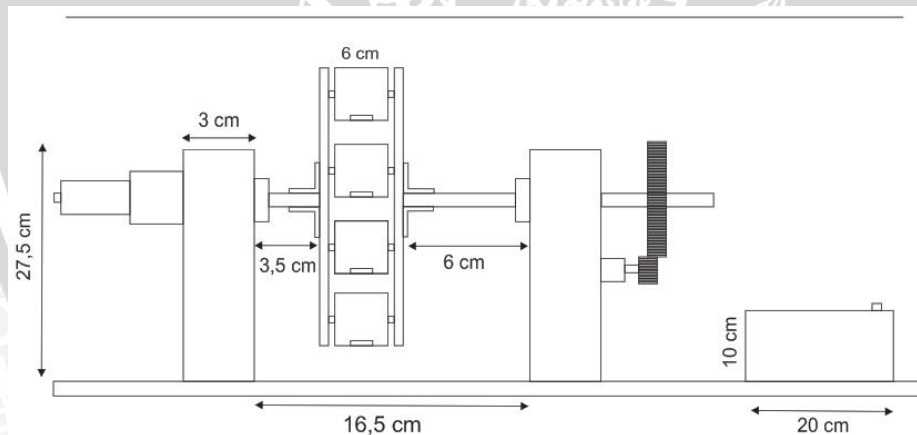
4.2 Perancangan perangkat keras (*hardware*)

Berdasarkan diagram blok perancangan alat yang telah disusun, perancangan perangkat keras meliputi perancangan mekanik, sensor potensiometer dan *driver* motor serta penggunaan modul Atmega 16. Di bawah ini adalah penjelasan masing-masing rangkaian penyusun keseluruhan alat.

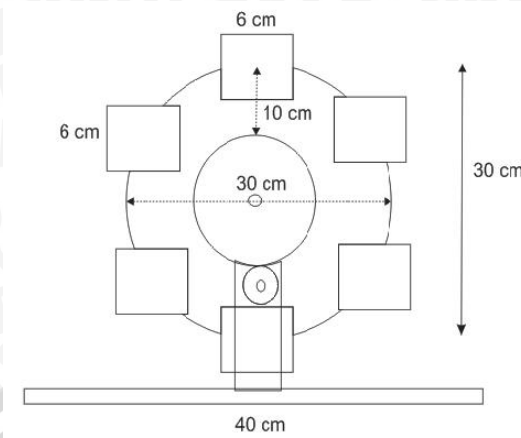
4.2.1 Spesifikasi Alat

Spesifikasi model sistem *rotary parking* adalah sebagai berikut :

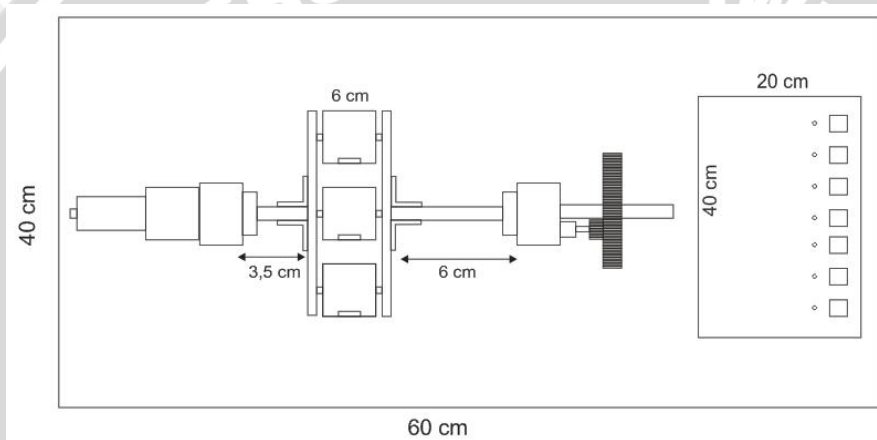
- Model sistem rotary parking yang dibuat terdiri 6 buah garasi yang dapat menampung 6 buah mobil.
- Pergerakan model rotary parking menggunakan motor DC dan menggunakan pipa aluminium sebagai poros penghubung antara model *rotary parking* dan motor DC.
- Potensiometer yang digunakan adalah potensiometer linier 10 K Ω yang dipasang sejajar dengan motor DC.
- Ukuran dari model *rotary parking* dapat dilihat pada gambar 4.2 dan 4.3 dengan satuan ukuran sentimeter.



Gambar 4.2 Skema sistem *rotary parking* Tampak Depan



Gambar 4.3 Skema sistem rotary parking Tampak Samping

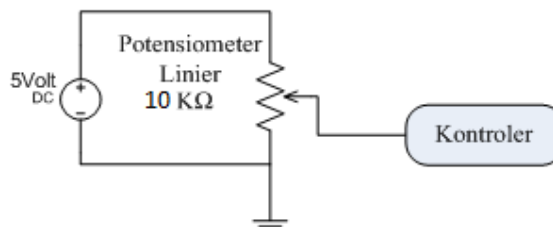


Gambar 4.4 Skema sistem rotary parking Tampak Atas

4.2.2 Potensiometer Linier

Potensiometer linier digunakan sebagai pengukur sudut atau posisi dari garasi sistem rotary parking, dengan meletakkan poros motor pada poros potensiometer, maka setiap pergeseran sudut dari rotary parking akan ikut mempengaruhi perubahan resistansi pada kaki potensiometer linier. Potensiometer yang digunakan adalah potensiometer linier 10 K Ω .

Sistem perancangan dan hasil perancangan potensiometer linier ditunjukkan pada gambar 4.5 dan 4.6 di bawah ini.



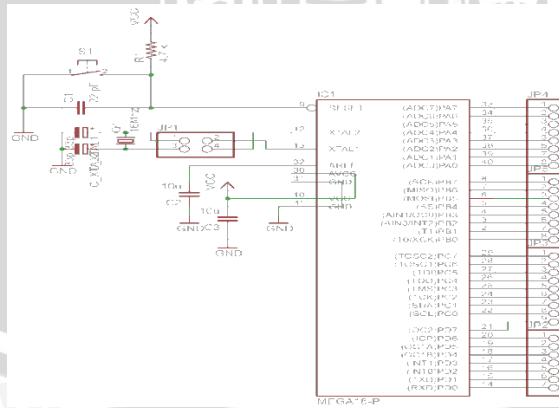
Gambar 4.5 Rangkaian Potensiometer



Gambar 4.6 Potensiometer pada Sistem

4.2.3. Modul Mikrokontroler AVR Atmega 16

Pada model *rotary parking* ini digunakan mikrokontroler Atmega16 sebagai pengolah dalam proses pengaturan posisi motor DC dalam menggerakkan garasi. Konfigurasi kaki I/O dari mikrokontroler Atmega16 ditunjukkan dalam Gambar 4.7.



Gambar 4.7 Minimum Sistem Mikrokontroler Atmega16

Mikrokontroler Atmega 16 mempunyai 4 port, 32 jalur yang dapat diprogram menjadi masukan atau keluaran. Pin masukan dan keluaran mikrokontroler pada perancangan ini akan difungsikan sesuai tabel 4.1

Tabel 4.1 Fungsi Pin Mikrokontroler

No	Pin	Fungsi
1	PA0	Jalur masukan potensiometer <i>rotary parking</i>
2	PB0	Jalur keluaran pertama <i>driver motor</i>
3	PB1	Jalur keluaran kedua <i>driver motor</i>
4	PB2	Jalur keluaran ketiga <i>driver motor</i>
5	PC0	Jalur masukan <i>limit switch</i> pada garasi 1
6	PC1	Jalur masukan <i>limit switch</i> pada garasi 2
7	PC2	Jalur masukan <i>limit switch</i> pada garasi 3
8	PC3	Jalur masukan <i>limit switch</i> pada garasi 4
9	PC4	Jalur masukan <i>limit switch</i> pada garasi 5
10	PC5	Jalur masukan <i>limit switch</i> pada garasi 6
11	PD0	Jalur masukan <i>push button 1</i>
12	PD1	Jalur masukan <i>push button 2</i>
13	PD2	Jalur masukan <i>push button 3</i>
14	PD3	Jalur masukan <i>push button 4</i>
15	PD4	Jalur masukan <i>push button 5</i>
16	PD5	Jalur masukan <i>push button 6</i>

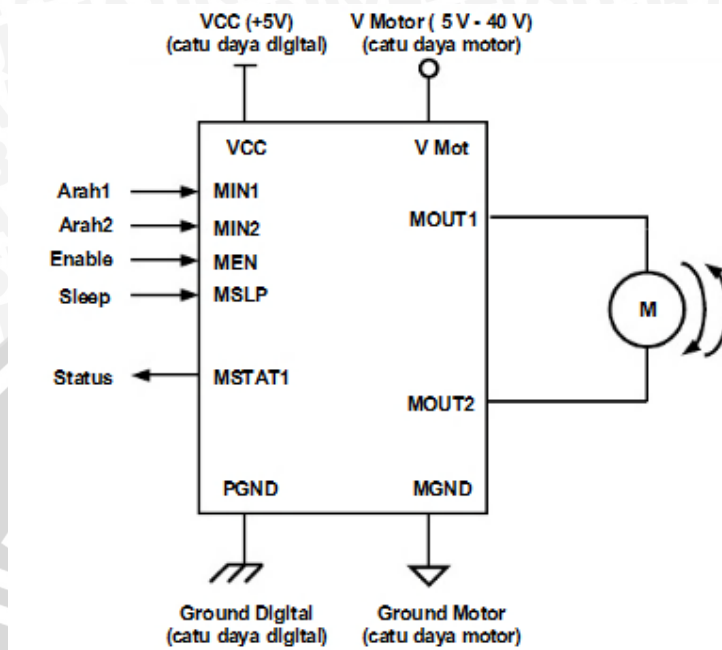


Gambar 4.8 Modul Mikrokontroler Atmega16

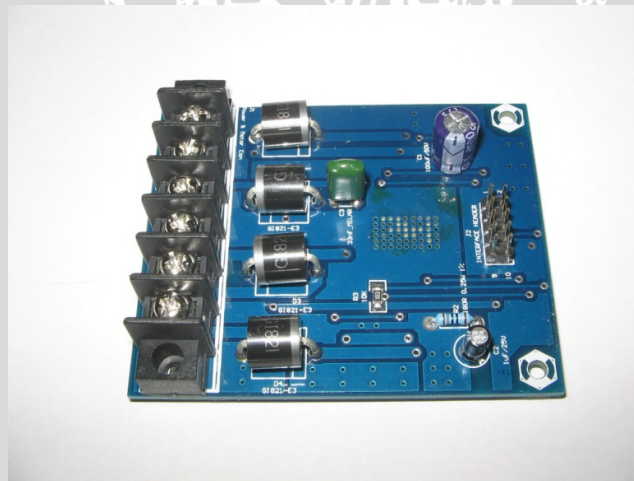
4.2.4 Perancangan *Driver*

Modul pengendali motor DC yang digunakan adalah modul EMS 5A H-Bridge. Secara garis besar, fungsi modul pengendali motor ini adalah untuk mengendalikan arah dan

kecepatan putaran motor DC sesuai instruksi kendali dari mikrokontroler pengguna. Gambar koneksi dari modul pengendali motor DC EMS 5A H-Bridge ditunjukkan pada Gambar 4.8



Gambar 4.9 Koneksi Modul EMS 5A H-Bridge



Gambar 4.10 Pengendali Motor DC Menggunakan EMS 5A H-Bridge

4.3 Perancangan Kontroler

Dalam perancangan kontroler, terlebih dahulu harus diketahui bentuk respons sistem. Plan (objek yang dikontrol) pada sistem adalah *rotary parking*. Aktuator plan adalah motor DC. Langkah-langkah yang dilakukan untuk mengetahui respons sistem adalah:

- Motor dihubungkan dengan catu daya dengan tegangan 12 volt DC.
- Memberikan penguatan proporsional dari nol hingga alat dapat memberikan respons berupa osilasi berkesinambungan, dengan penguatan integral nol dan penguatan diferensial nol.
- Mengukur waktu antara tidak ada perubahan tegangan sampai terjadi kenaikan tegangan.
- Mengamati perubahan nilai kecepatan motor yang terjadi pada tiap waktu.
- Mencatat perubahan nilai kecepatan motor hasil pengamatan.

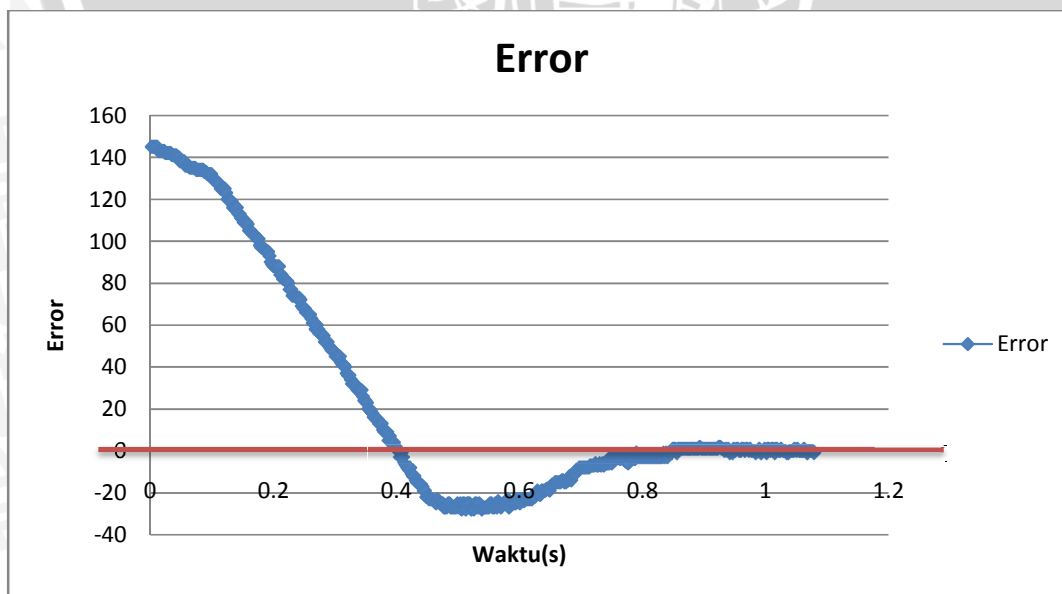
Setelah selesai melakukan pengukuran nilai kecepatan motor, maka langkah selanjutnya adalah menentukan identifikasi sistem, sebagai berikut :

- Membuat grafik yang menyatakan hubungan antara sudut keluaran dengan waktu.
- Menentukan nilai K_{cr} dan P_{cr} seperti dalam Gambar 4.11, K_{cr} merupakan nilai penguatan kritis dan P_{cr} merupakan nilai periode (T).
- Menentukan parameter kontroler PID berdasarkan metode kedua *Ziegler-Nichols* (metode loop tertutup).

Metode perancangan kontroler PID (*tunning*) ini dilakukan sebanyak 3 kali.

4.3.1 Tuning pada *Rotary Parking*

Tunning kontroler dengan K_p sebesar 4,8, K_i sebesar 8,9 dan K_d sebesar 0,6 mendapatkan respons seperti Gambar 8.



Gambar 4.11 Grafik Tuning pada rotary parking

145	0.0022
145	0.0044
145	0.0066
145	0.0088
145	0.011
143	0.0132
143	0.0154
143	0.0176
143	0.0198
143	0.022
142	0.0242
142	0.0264
142	0.0286
142	0.0308
142	0.033
141	0.0352
141	0.0374
141	0.0396
141	0.0418
140	0.044
140	0.0462
138	0.0484
138	0.0506
138	0.0528
138	0.055
136	0.0572
136	0.0594

Gambar 4.12 Pengambilan Data ADC pada rotary parking

Respons sistem menampilkan data setiap 0,0022 s, maka dari grafik diketahui bahwa nilai P_{cr} = jumlah data x waktu antar data

$$= 490 \times 0,0022$$

$$= 1,078$$

dan $K_{cr} = 8$

Dengan menggunakan metode kedua dari teori Ziegler-Nichols ,maka akan didapatkan nilai sebagai berikut :

$$K_p = 0,60 \times K_{cr} = 4,8$$

$$T_i = 0,5 \times P_{cr} = 0,539$$

$$T_d = 0,125 \times P_{cr} = 0,13475$$

Nilai – nilai pengutan K_i dan K_d dapat dicari dengan :

$$\frac{K_i}{K_p} = 8,9$$

$$\frac{K_d}{K_p} = 0,6$$

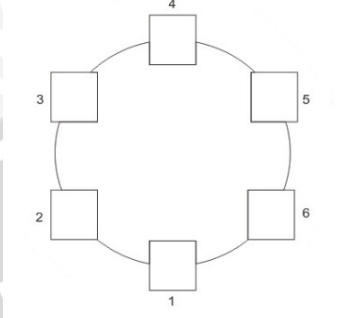
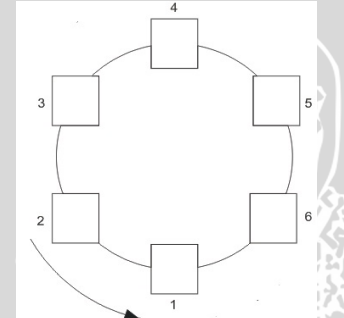
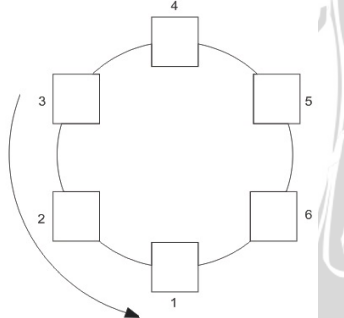
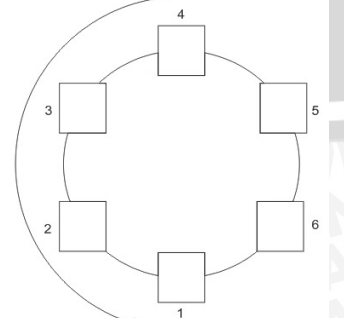


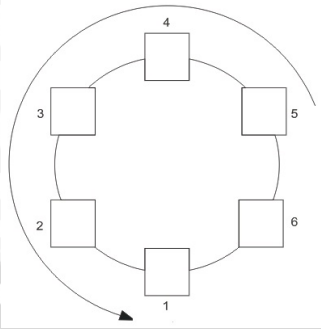
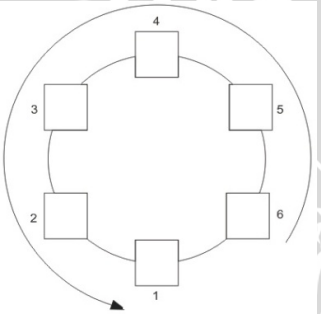
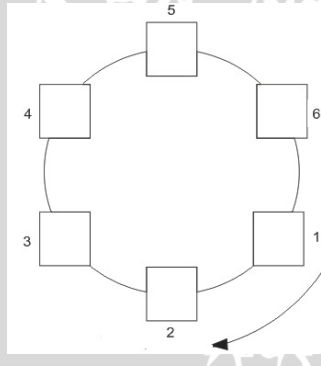
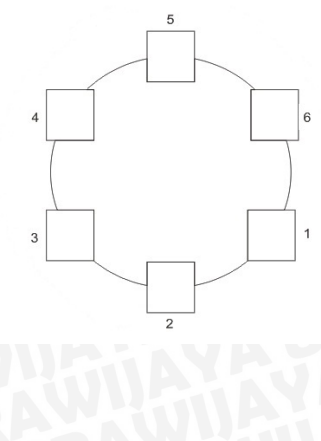
4.4 Perancangan Pergerakan *Rotary Parking*

Pergerakan model sistem *rotary parking* ditentukan dari posisi garasi 1, 2, 3, 4, 5, dan 6 menuju posisi dasar pada model *rotary parking*. Penentuan *set point* atau besarnya sudut yang harus ditempuh berdasarkan pengukuran manual terhadap titik-titik tujuan. Pengukuran yang diambil berupa besar sudut motor DC sebagai posisi garasi yang akan dituju. Pergerakan tiap posisi ditunjukkan pada Tabel 4.2.

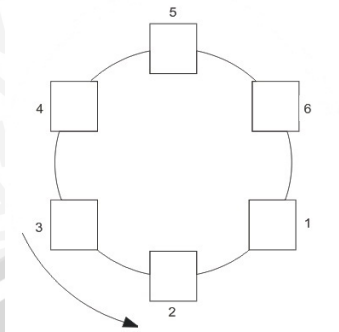
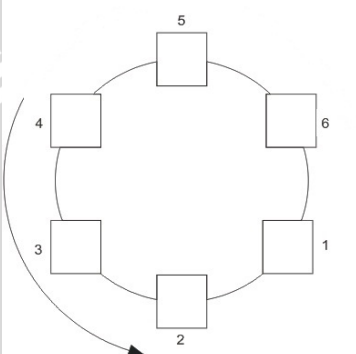
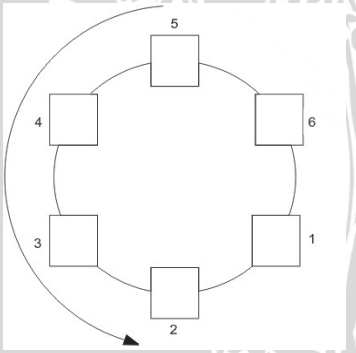
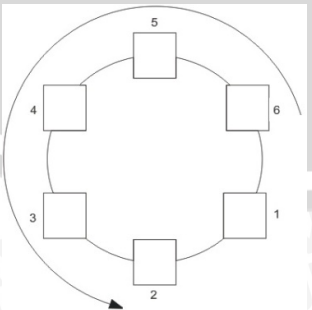


Tabel 4.2 Rencana Pergerakan Sistem

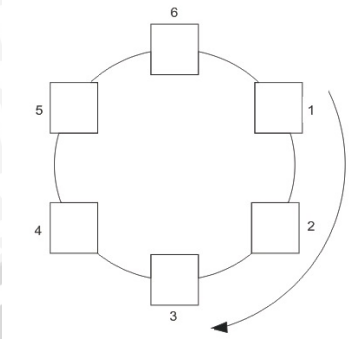
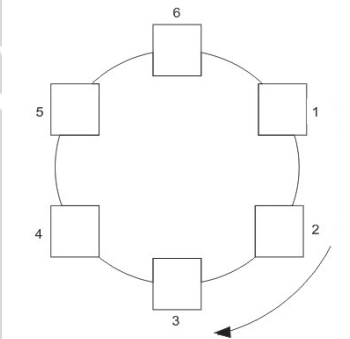
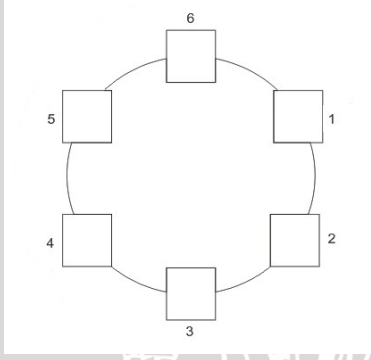
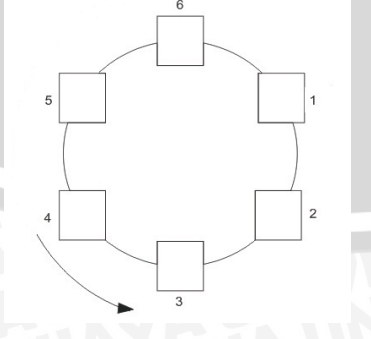
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
1	1		Stop
	2		Counter Clock Wise (CCW)
	3		Counter Clock Wise (CCW)
	4		Counter Clock Wise (CCW)

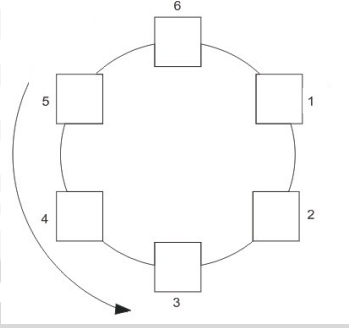
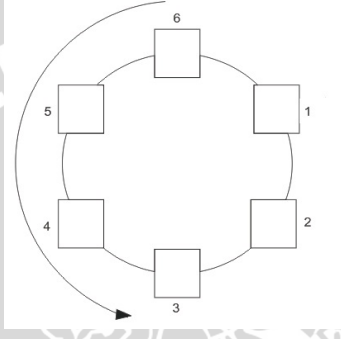
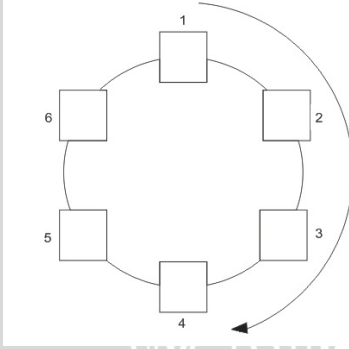
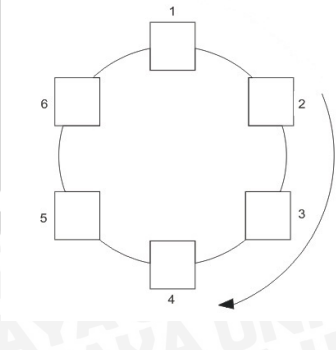
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
1	5		Counter Clock Wise (CCW)
	6		Counter Clock Wise (CCW)
2	1		Clock Wise (CW)
	2		Stop



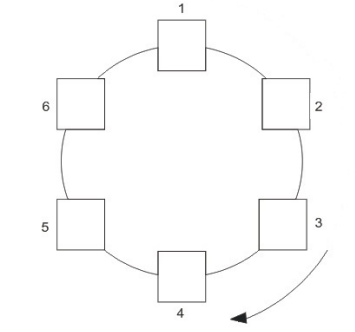
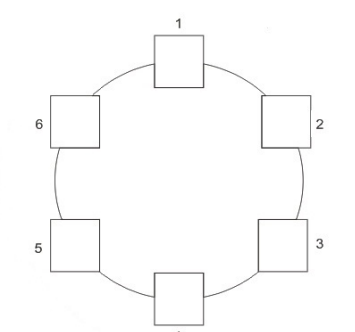
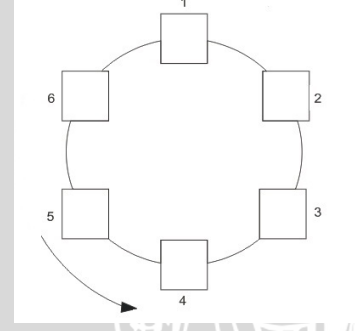
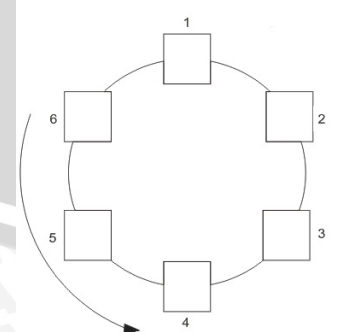
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
2	3		Counter Clock Wise (CCW)
	4		Counter Clock Wise (CCW)
	5		Counter Clock Wise (CCW)
	6		Counter Clock Wise (CCW)



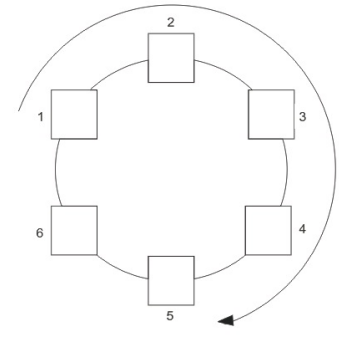
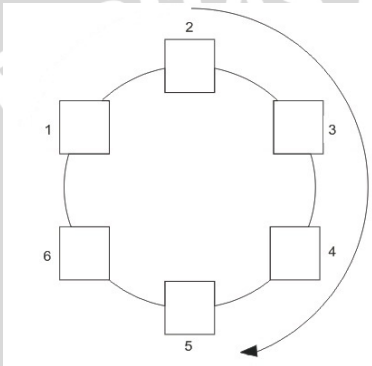
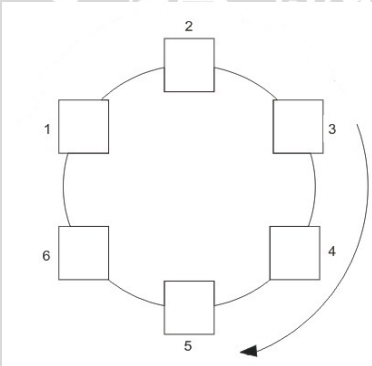
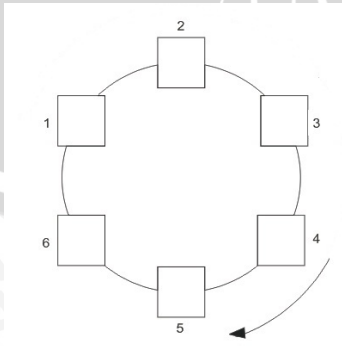
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
3	1		Clock Wise (CW)
	2		Clock Wise (CW)
	3		Stop
	4		Counter Clock Wise (CCW)

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
3	5		Counter Clock Wise (CCW)
	6		Counter Clock Wise (CCW)
4	1		Clock Wise (CW)
	2		Clock Wise (CW)

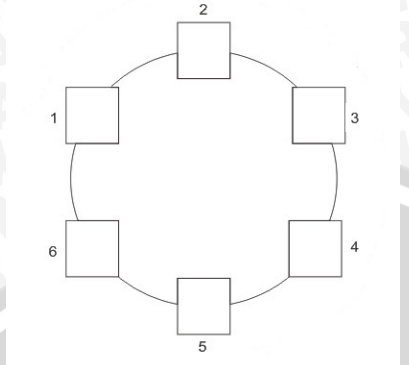
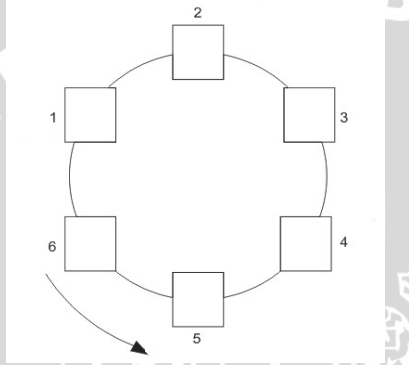
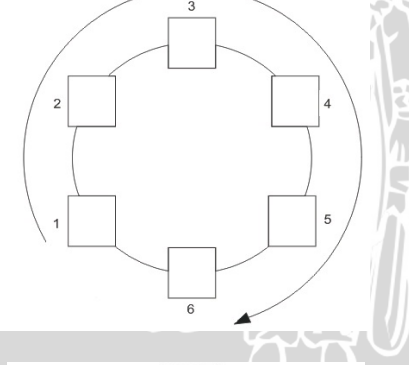
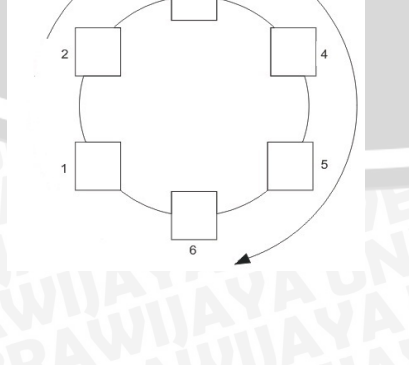


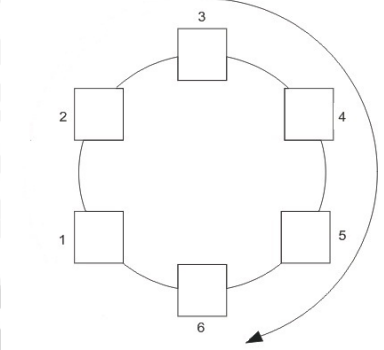
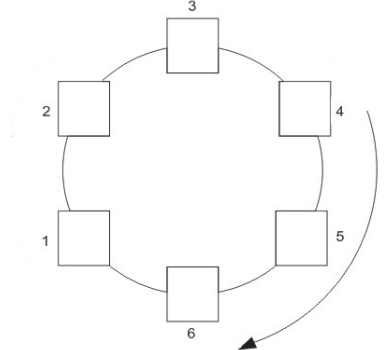
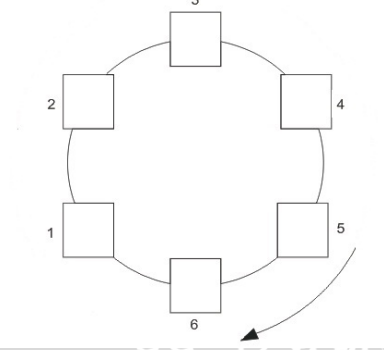
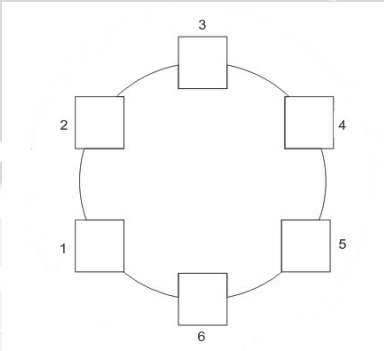
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
4	3		Clock Wise (CW)
	4		Stop
	5		Counter Clock Wise (CCW)
	6		Counter Clock Wise (CCW)



Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
5	1		Clock Wise (CW)
	2		Clock Wise (CW)
	3		Clock Wise (CW)
	4		Clock Wise (CW)



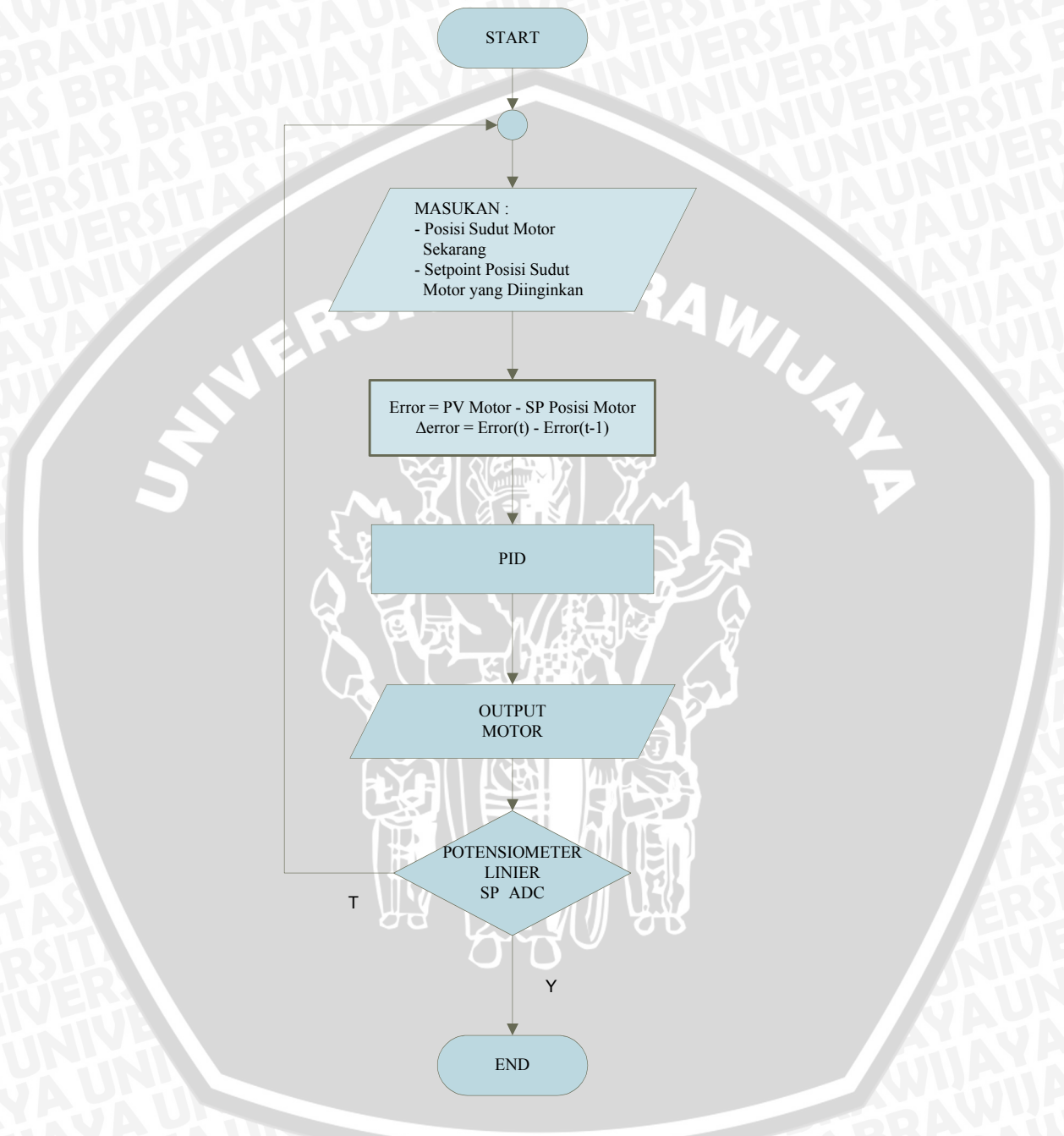
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
5	5		Stop
6	6		Counter Clock Wise (CCW)
6	1		Clock Wise (CW)
6	2		Clock Wise (CW)

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Keterangan Pergerakan
6	3		Clock Wise (CW)
	4		Clock Wise (CW)
	5		Clock Wise (CW)
	6		Stop



4.5 Perancangan perangkat lunak (*software*)

Flowchart perancangan perangkat lunak ditunjukkan pada gambar 4.49 berikut:



Gambar 4.13 Flowchart Program

BAB V

PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan.

5.1 Pengujian Potensiometer

a. Tujuan

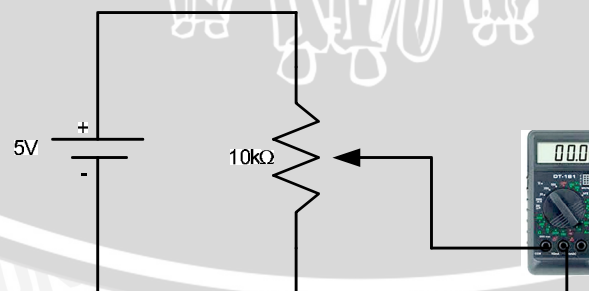
Mengetahui tingkat kelinieran dari potensiometer dalam membaca perubahan sudut putar motor dc.

b. Peralatan yang digunakan

- Potensiometer.
- Multimeter.
- Penggaris busur.
- Catu daya 5 volt.

c. Langkah pengujian

1. Merangkai peralatan seperti pada gambar 5.1
2. Potensiometer diputar dari 0° sampai 360°
3. Mencatat dan menggambar data tegangan keluaran tiap 15° dari potensiometer untuk melihat kelinierannya.



Gambar 5.1. Pengujian Potensiometer

d. Hasil Pengujian

Secara teori, tegangan keluaran dari potensiometer ditulis dalam persamaan

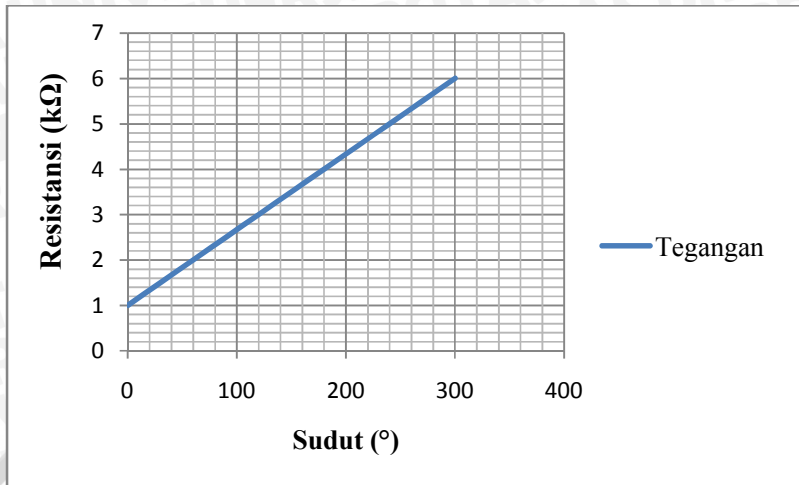
(5.1)

data hasil pengujian potensiometer ditunjukkan pada tabel 5.1 berikut

Tabel 5.1. Hasil Pengujian Potensiometer

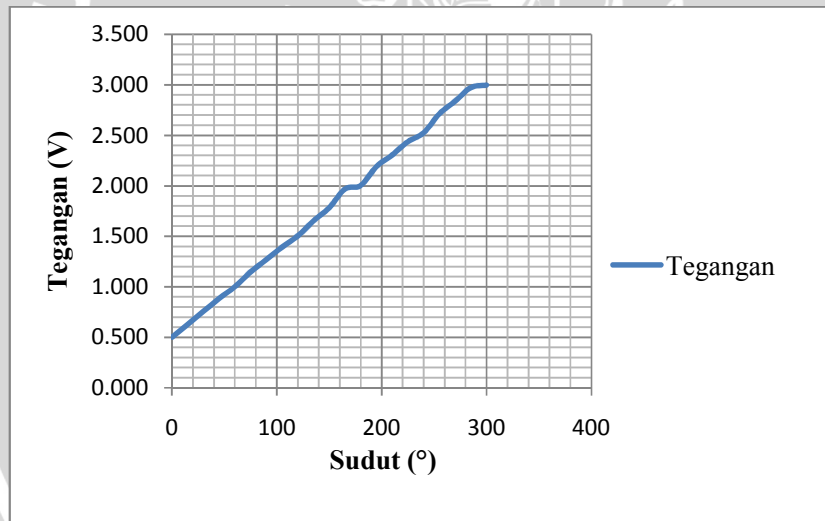
no	sudut (°)	Resistansi (k Ω)	Tegangan Perhitungan (V)	Tegangan Pengukuran (V)	Error (%)
1	0	1	0,498	0,50000	0,39900
2	15	1,25	0,626	0,62499	0,16100
3	30	1,5	0,754	0,74999	0,57874
4	45	1,75	0,883	0,87499	0,91529
5	60	2	0,999	0,99999	0,13230
6	75	2,25	1,145	1,12499	1,77880
7	90	2,5	1,267	1,24999	1,38765
8	105	2,75	1,391	1,37499	1,16465
9	120	3	1,502	1,49999	0,13433
10	135	3,25	1,651	1,62498	1,60102
11	150	3,5	1,782	1,74998	1,81056
12	165	3,75	1,967	1,87498	4,90772
13	180	4	2,002	1,99998	0,10100
14	195	4,25	2,189	2,12498	3,01279
15	210	4,5	2,304	2,24998	2,38622
16	225	4,75	2,436	2,37498	2,56945
17	240	5	2,524	2,49998	0,94769
18	255	5,25	2,712	2,62497	3,31532
19	270	5,5	2,837	2,74997	3,15256
20	270	5,5	2,837	2,87497	3,34016
21	285	5,75	2,971	2,99997	0,11010

Grafik hubungan antara sudut dengan resistansi digambarkan pada gambar 5.2



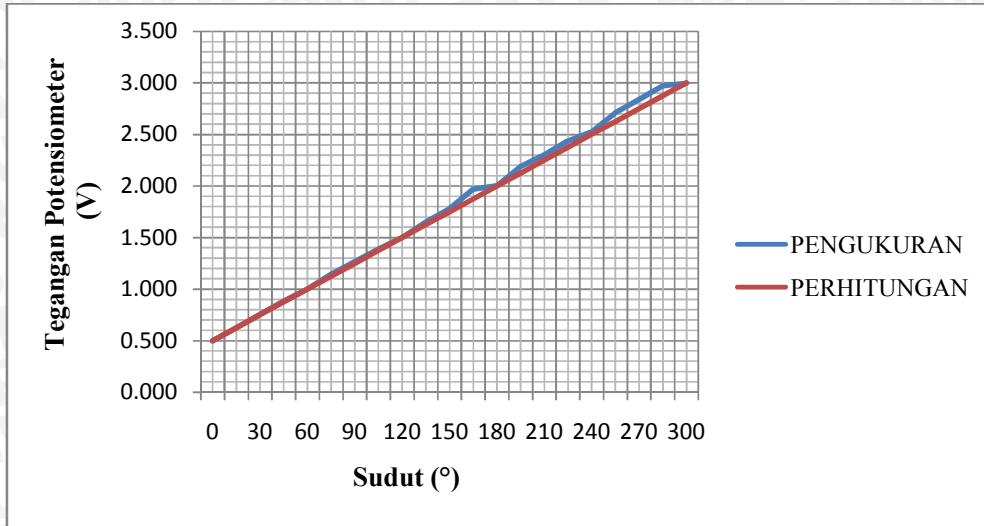
Gambar 5.2. Grafik Hubungan Antara Sudut dengan Resistansi

Hubungan antara sudut dengan tegangan potensiometer digambarkan pada gambar 5.3



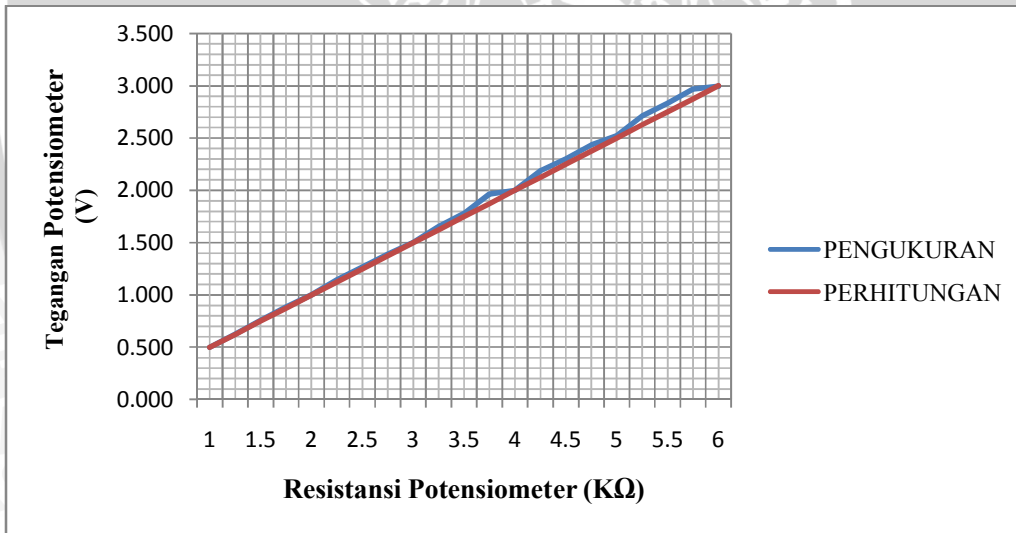
Gambar 5.3. Grafik Hubungan Antara Sudut dengan Tegangan

Grafik antara tegangan perhitungan dan tegangan pengukuran terhadap sudut pada potensiometer digambarkan pada gambar 5.4



Gambar 5.4. Grafik Antara Tegangan Perhitungan dengan Tegangan Pengukuran terhadap sudut

Grafik antara tegangan perhitungan dan tegangan pengukuran terhadap sudut pada potensiometer digambarkan pada gambar 5.5



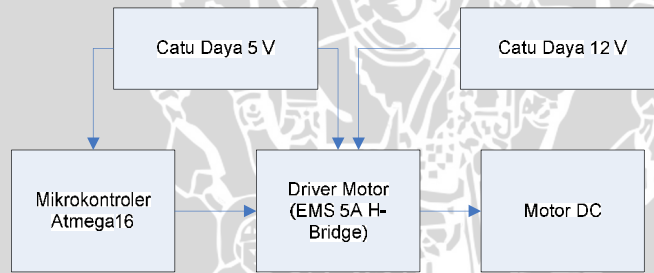
Gambar 5.5. Grafik Antara Tegangan Perhitungan dengan Tegangan Pengukuran terhadap resistansi

Dari hasil pengujian yang dilakukan pada potensiometer dapat dilihat bahwa potensiometer yang digunakan mempunyai kelinieran yang baik, sehingga ideal untuk digunakan sebagai pendeteksi posisi sudut putar motor dc dalam menggerakkan model *rotary parking* sampai pada posisi sudut putar yang sesuai dengan *set point*.

5.2 Pengujian Driver Motor DC

- a. Tujuan

Mengetahui *output* dari *driver* motor apabila diberi *input* yang berbeda-beda.
- b. Peralatan yang digunakan
 - Minimum sistem mikrokontroler Atmega16
 - *Driver* motor
 - Motor DC
 - Catu daya 5V dan 12V
- c. Langkah pengujian
 1. Merangkai peralatan seperti pada gambar
 2. Mengisi mikrokontroler dengan program yang mengeluarkan data keluaran biner dengan keluaran 01 dan 10.
 3. Mengaktifkan catu daya 5 volt dan 12 volt
 4. Mencatat pergerakan motor DC.



Gambar 5.6. Diagram Blok Pengujian *Driver* Motor DC

d. Hasil pengujian

Tabel 5.2. Hasil Pengujian *Driver* Motor DC

No	Logika Keluaran		Arah Putaran Motor DC
	PORT B.1	PORT B.2	
1	1	0	Searah Jarum Jam
2	0	1	Berlawanan Arah Jarum jam

Berdasarkan hasil pengujian pada tabel 5.2 dapat diketahui bahwa motor DC akan berputar searah jarum jam ketika diberi logika 1-0, dan apabila diberi logika 0-1 maka motor akan berputar berlawanan arah jam. Dengan demikian dapat disimpulkan bahwa rangkaian *driver* motor DC ini dapat berjalan dengan baik pada sistem yang direncanakan.

5.3 Pengujian Kecepatan yang Dihasilkan (PWM)

a. Tujuan

Untuk mengetahui letak posisi akhir garasi model *rotary parking* yang diinginkan yang telah digerakkan oleh motor dc.

b. Peralatan yang digunakan

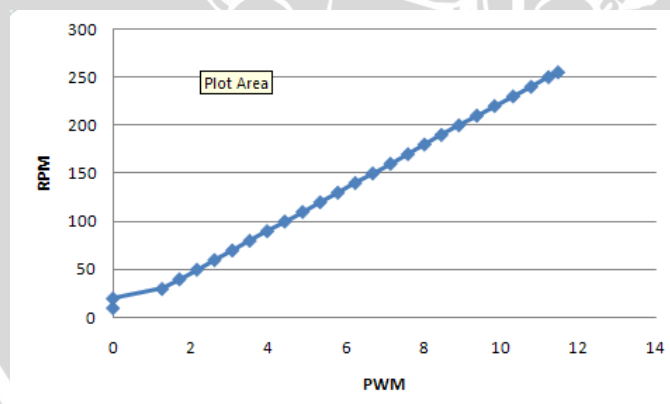
- Modul mikrokontroler Atmega16
- *Tachometer*.

c. Langkah pengujian

Mengukur kecepatan motor dengan nilai *pulse width modulation* (PWM) yang berbeda, sehingga didapatkan kecepatan yang diinginkan.

d. Hasil pengujian

Hasil pengujian kecepatan motor dengan nilai *pulse width modulation* (PWM) yang ditunjukkan pada Gambar 5.7 dan Tabel 5.3.



Gambar 5.7 Hubungan Nilai PWM dan RPM

Tabel 5.3. Hasil Pengujian *pulse width modulation* (PWM)

PWM	V MK (Volt)	V Motor (Volt)	RPM
10	0.215	0.28	0.000
20	0.410	0.61	0.000
30	0.410	0.61	1.260
40	0.795	1.29	1.710
50	0.984	1.63	2.160
60	1.176	2.08	2.610
70	1.351	2.76	3.070
80	1.542	2.87	3.520
90	1.752	3.45	3.970
100	1.944	4.60	4.420
110	2.136	5.50	4.880
120	2.329	6.20	5.330
130	2.522	6.70	5.785
140	2.715	7.40	6.235
150	2.907	7.77	6.685
160	3.102	8.20	7.140
170	3.292	8.54	7.590
180	3.488	8.94	8.015
190	3.660	9.20	8.450
200	3.870	9.53	8.900
210	4.050	9.90	9.365
220	4.250	9.95	9.820
230	4.420	10.10	10.295
240	4.600	10.38	10.760
250	4.780	10.54	11.210
255	4.900	10.67	11.455

5.4. Pengukuran Letak Posisi Akhir

a. Tujuan

Untuk mengetahui letak posisi akhir garasi model *rotary parking* yang diinginkan yang telah digerakkan oleh motor dc.

b. Peralatan yang digunakan

- Kotak pejal berukuran 3x3x3 dengan berat sama ringan sebanyak 6 buah.
- Busur.

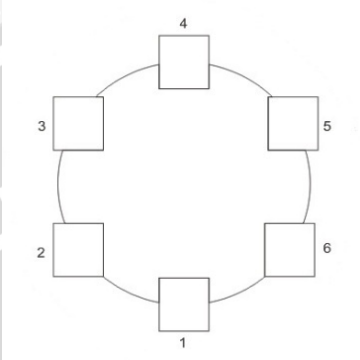
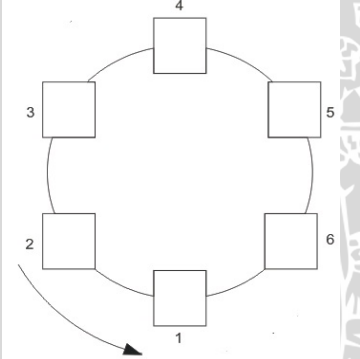
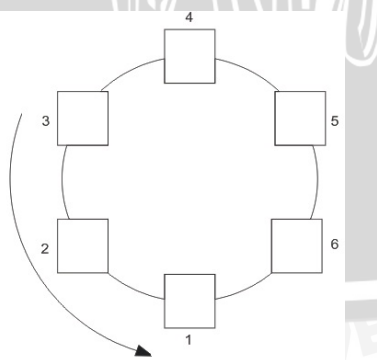
c. Langkah pengujian

Mengukur Posisi garasi 1,2,3,4,5,6 dari garasi acuan diukur menggunakan busur, sehingga didapatkan selisih tiap garasi memiliki beda sudut 30 derajat.

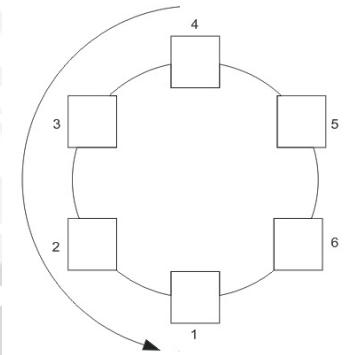
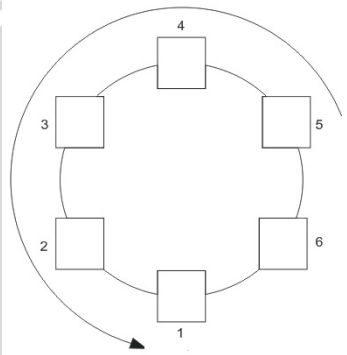
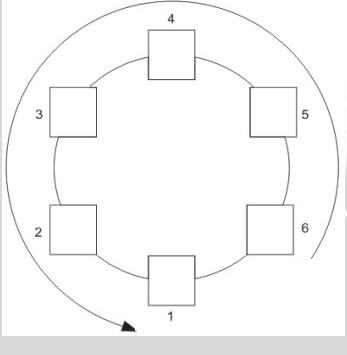
d. Hasil pengujian

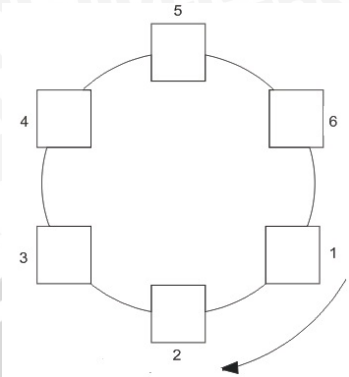
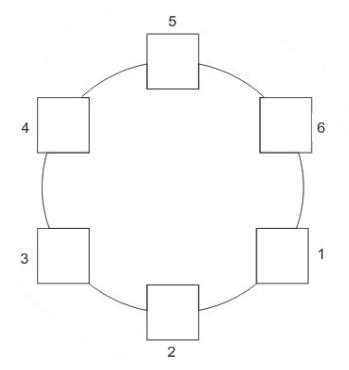
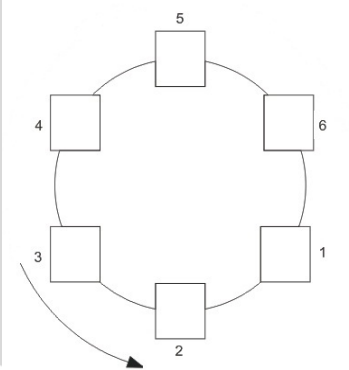
Hasil pengujian posisi garasi 1,2,3,4,5,6 dari titik acuan *rotary parking* dapat ditunjukkan pada Tabel 5.4.

Tabel 5.4 Pengujian Posisi Akhir

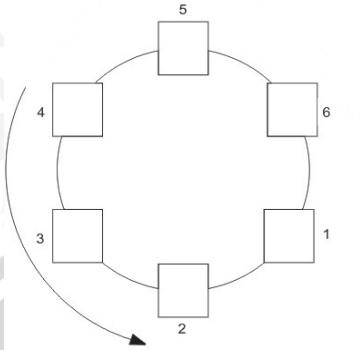
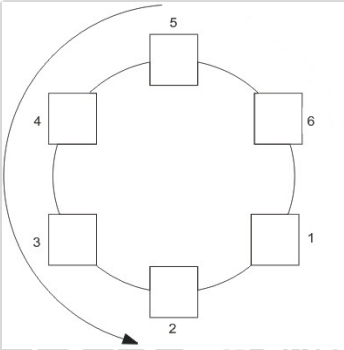
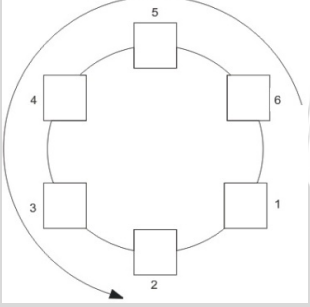
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
1	1		0
	2		60
	3		120



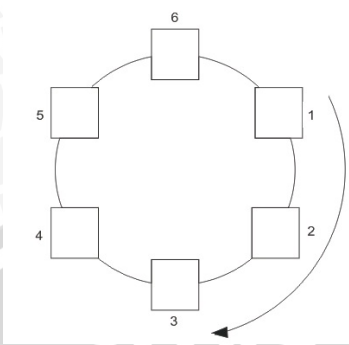
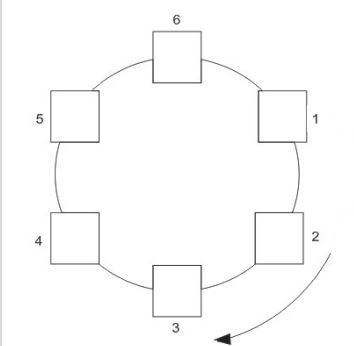
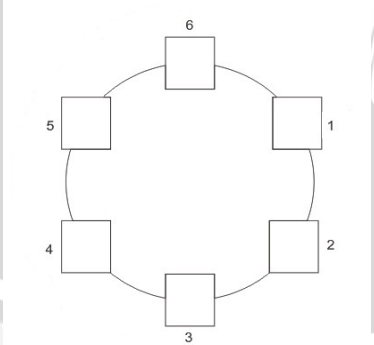
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
1	4		180
	5		240
	6		300

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
2	1		- 60
	2		0
	3		60

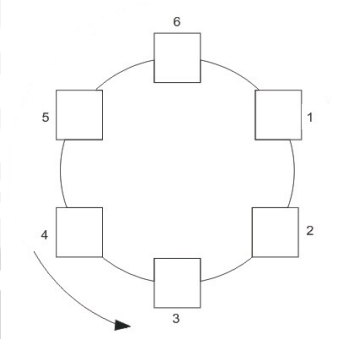
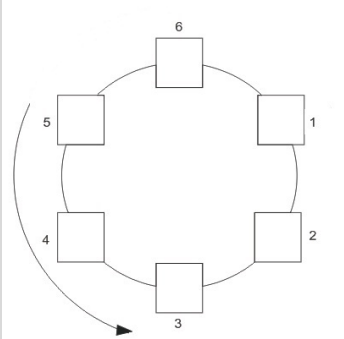
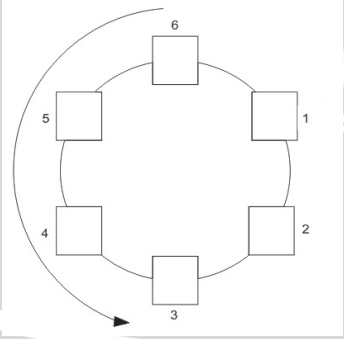


Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
2	4		120
	5		180
	6		240

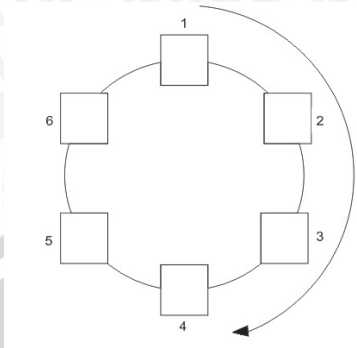
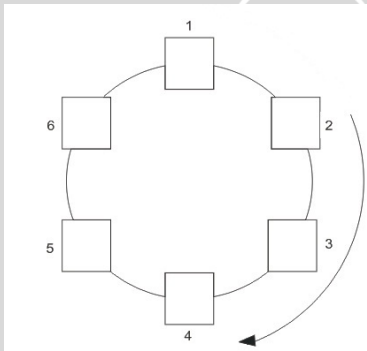
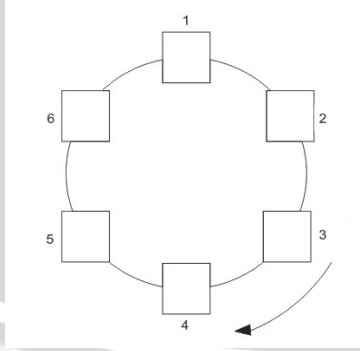


Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
3	1		- 120
	2		- 60
	3		0

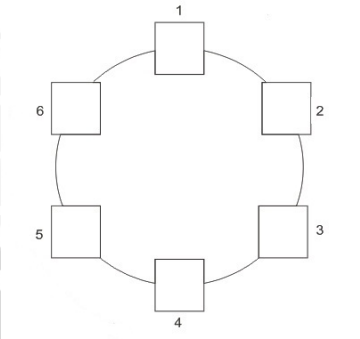
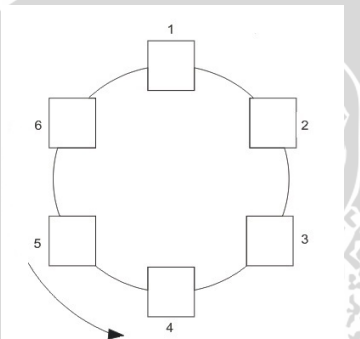
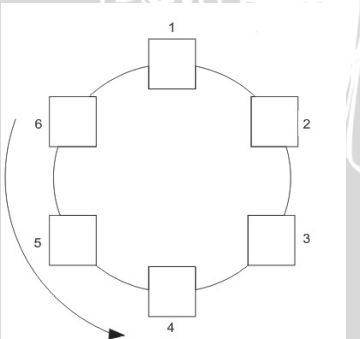


Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
3	4		60
	5		120
	6		180

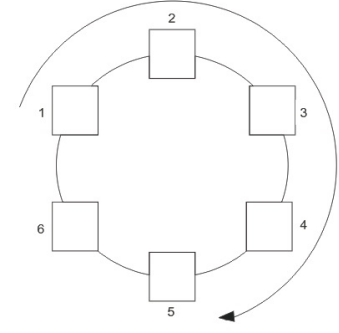
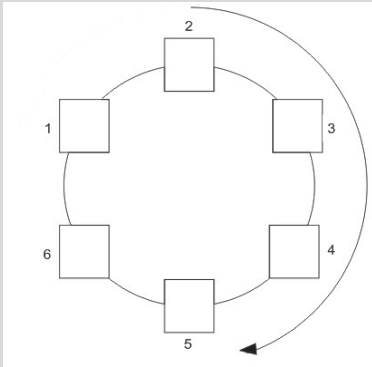
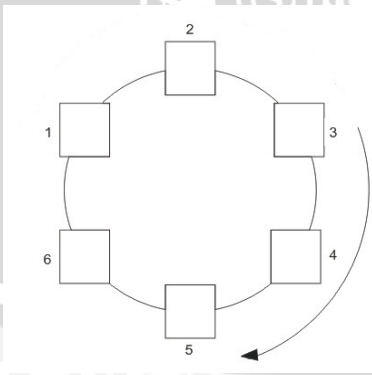


Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
4	1		- 180
	2		- 120
	3		- 60

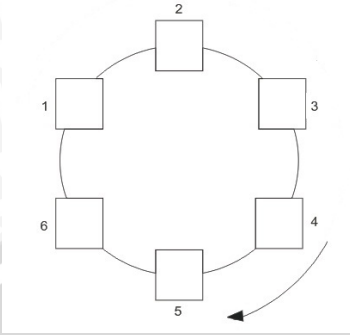
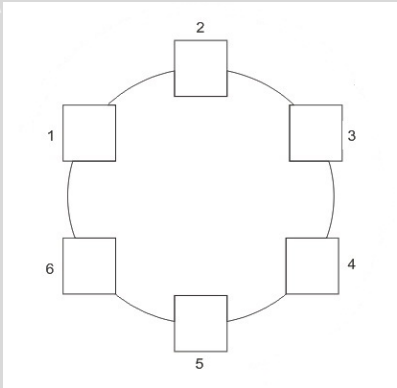
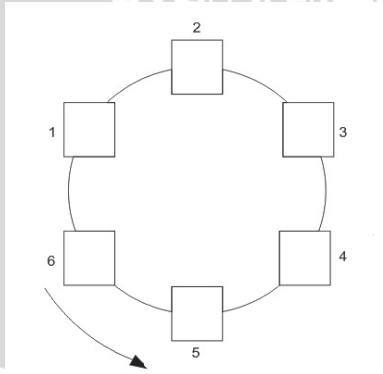


Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
4	4		0
	5		60
	6		120

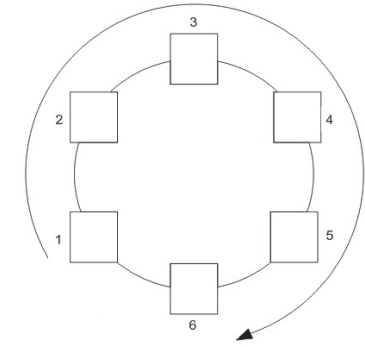
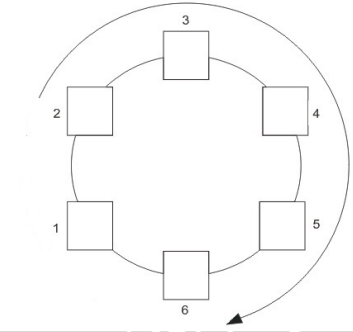
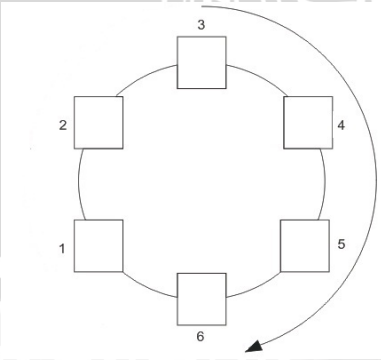


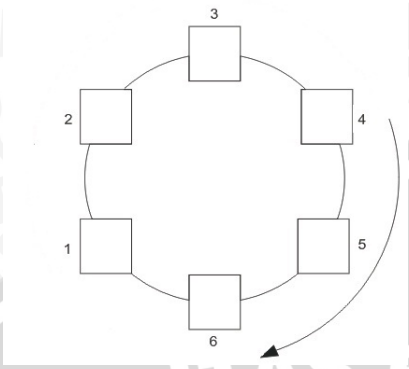
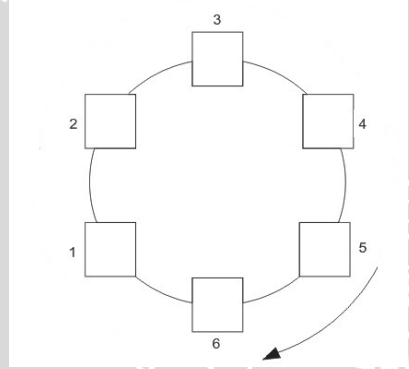
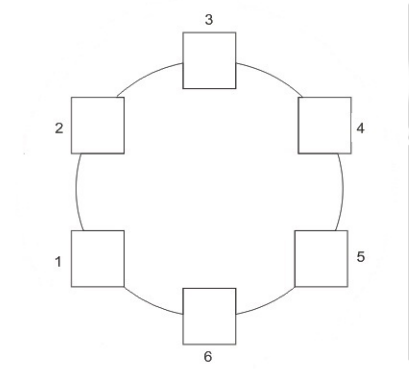
Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
5	1		- 240
	2		- 180
	3		- 120



Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
5	4		- 60
5	5		0
6	6		60



Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
6	1		- 300
	2		- 240
	3		- 180

Garasi Acuan	Garasi Tujuan	Gambar Pergerakan	Sudut (°)
6	4		- 120
	5		- 60
	6		0



5.5 Pengujian Keseluruhan

a. Tujuan

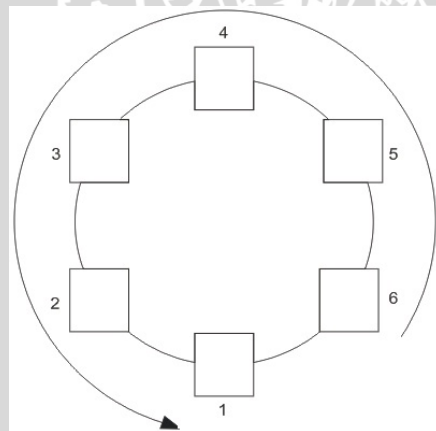
Untuk mengetahui kerja dari perangkat keras dan perangkat lunak setelah diintegrasikan dalam sebuah sistem terpadu

b. Peralatan yang digunakan

- Model *rotary parking* lengkap dengan potensiometer, motor DC, *limit switch*, serta *push button*.
- Catu daya 12 volt
- Sistem mikrokontroler Atmega 16
- *Driver* motor menggunakan modul EMS 5A H-Bridge
- Kabel serial

c. Hasil pengujian

Pengujian alat ini dilakukan dengan cara menjalankan *rotary parking* dengan cara memasukkan nilai *set point* pada setiap garasi yang dituju. Hasil pengujian ADC terhadap sudut dapat dilihat pada tabel 5.5.



Gambar 5.8. Pergerakan rotary parking dari dengan acuan garasi 1

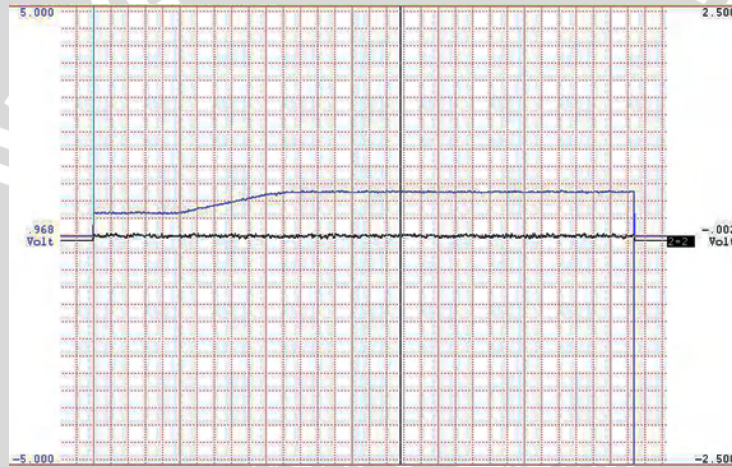
Tabel 5.5. Data Sudut Motor DC

Posisi garasi	Motor DC		
	adc	Vin(V)	Sudut(°)
1	102	0.498	0
2	204	0.999	60
3	306	1.423	120
4	408	2.002	180
5	510	2.524	240
6	612	2.999	300

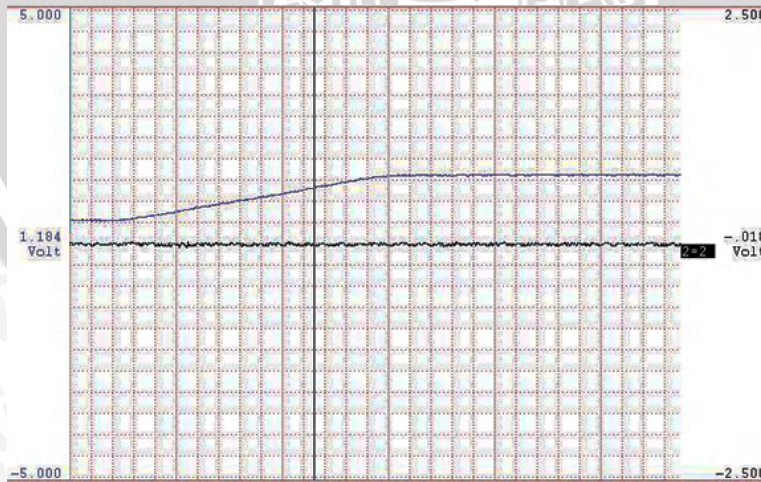
Penyajian grafik respons sistem tiap sendi dibutuhkan data sudut dan waktu, sedangkan dalam proses pengujian data yang didapat berupa data ADC dan urutan data ADC. Untuk itu diperlukan melakukan pengkonversian data hasil ADC menjadi sudut dan pengkonversian urutan data ADC tersebut menjadi satuan waktu. Karena penempatan dan data dari sensor potensiometer berbeda, maka pengkonversian dilakukan pada motor DC.

5.6 Pengujian Motor DC

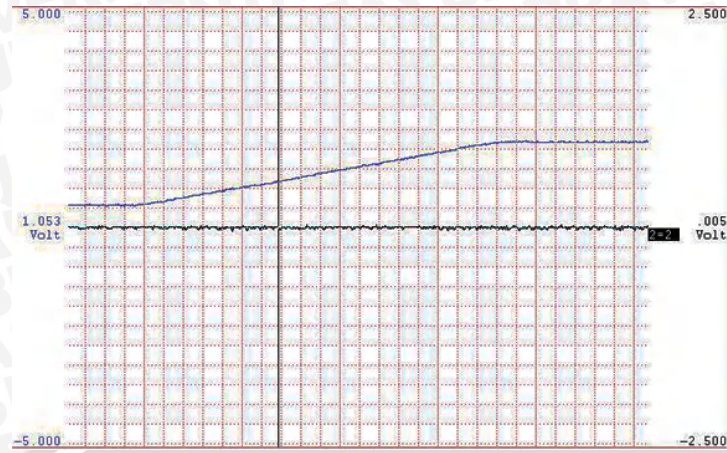
Pada pengujian ini, motor DC diberi *set point* berupa sudut, yaitu *set point* untuk menunjukkan posisi garasi yang akan dituju, maka respons sistem dengan kontroler PID ditunjukkan dalam gambar 5.43-5.48. Untuk garasi dengan banyak *setpoint* dengan urutan 1-3-2-6-5-2-4-2 dapat dilihat pada gambar 5.49.



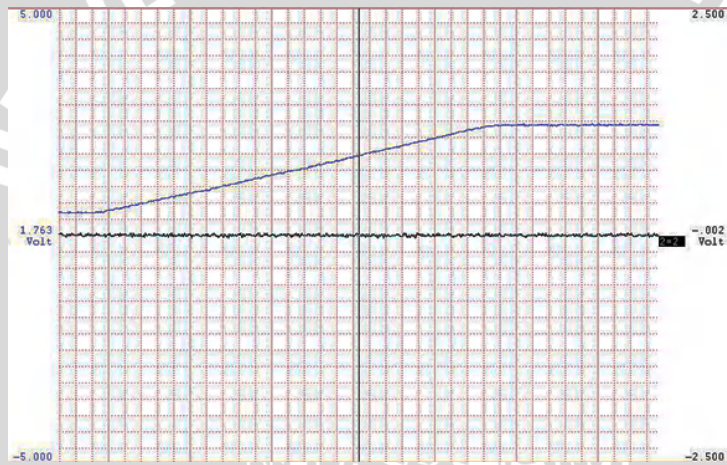
Gambar 5.9. Respons Sistem dengan tujuan 60°



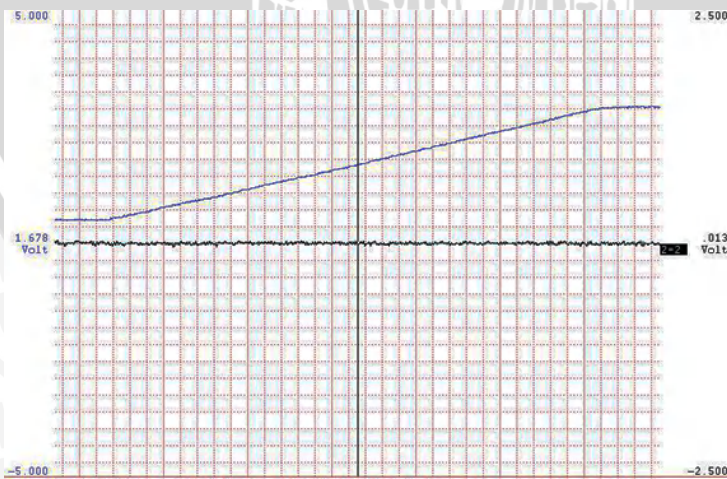
Gambar 5.10. Respons Sistem dengan tujuan 120°



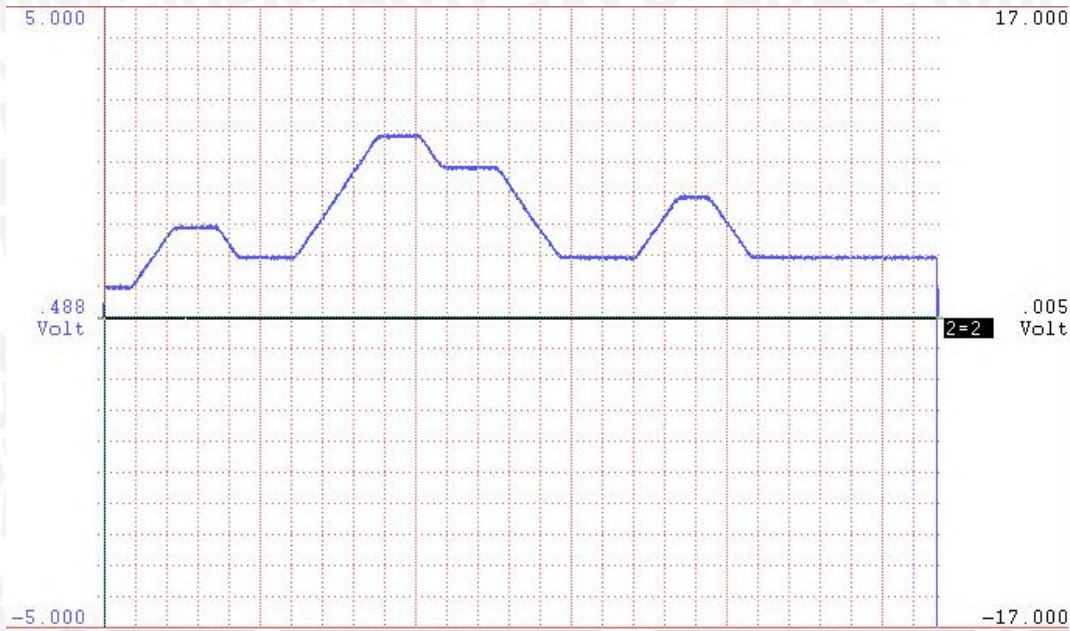
Gambar 5.11. Respons Sistem dengan tujuan 180°



Gambar 5.12. Respons Sistem dengan tujuan 240°



Gambar 5.13. Respons Sistem dengan tujuan 300°



Gambar 5.14. Respons Sistem dengan Tujuan Acak

Dari grafik respons sistem data sudut terhadap waktu, maka didapatkan data sebagai berikut :

a. Posisi acuan ke posisi tujuan

Time Settling atau T_s adalah waktu yang diperlukan kurva respons untuk mencapai dan menetap dalam daerah sekitar harga akhir yang nilainya ditentukan dengan presentase mutlak dari harga akhirnya (biasanya 5 % atau 2 %). T_s yang didapatkan dari pengujian ini adalah

T_s = waktu awal di posisi acuan - waktu akhir di posisi tujuan

- Untuk tujuan 300°

= 7,001-1,69

= 5,331 s

- Untuk tujuan 240°

= 5,01-0,668

= 4,342 s

- Untuk tujuan 180°

= 4,342-0,835

= 3,507 s

- Untuk tujuan 120°

= 2,839-0,4175

= 2,4215 s

- Untuk tujuan 60°

$$= 2,004-0,835$$

$$= 1,169 \text{ s}$$

- b. *Error steady state*, Ess adalah selisih antara nilai keluaran dengan nilai masukan pada saat kondisi *steady state*. Ess yang didapatkan dari pengujian ini adalah =

$$\frac{\text{°} - \text{°}}{\text{°}} \times 100 \% = 0.033\%$$

$$\frac{240.2^\circ - 240^\circ}{240^\circ} \times 100 \% = 0.083\%$$

$$\frac{180.3^\circ - 180^\circ}{180^\circ} \times 100 \% = 0.16\%$$

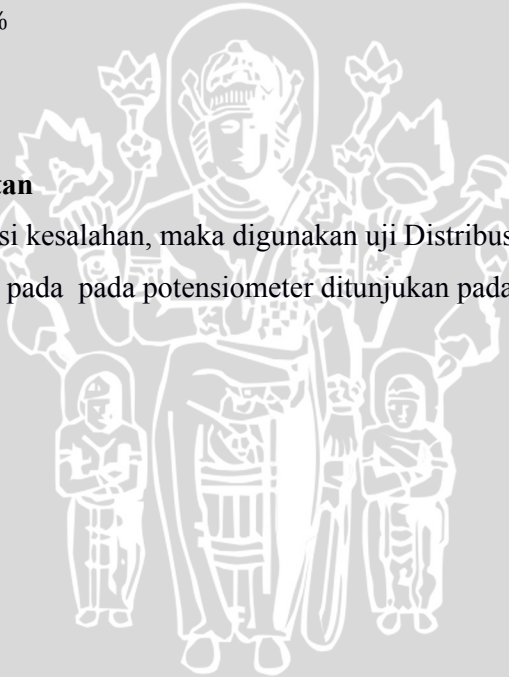
$$\frac{120.15^\circ - 120^\circ}{120^\circ} \times 100 \% = 0.125\%$$

$$\frac{60.25^\circ - 60^\circ}{60^\circ} \times 100 \% = 0.416\%$$

$$\frac{0^\circ - 0^\circ}{0^\circ} \times 100 \% = 0\%$$

5.7 Hasil Pengujian Ketepatan

Untuk mengetahui toleransi kesalahan, maka digunakan uji Distribusi T. Data perhitungan ketepatan tegangan pada pada potensiometer ditunjukkan pada Tabel 5.6 :



Tabel 5.6 Posisi Sudut Motor terhadap Tegangan Potensiometer

POSISI MOTOR (°)	RESISTANSI POTENSIMETER (kΩ)	TEGANGAN (V)	
		PENGUKURAN (x)	PERHITUNGAN (μ _c)
0	1	0.53000	0.50000
		0.60000	0.50000
		0.55000	0.50000
		0.60000	0.50000
		0.50000	0.50000
		0.50500	0.50000
		0.52000	0.50000
		0.54000	0.50000
		0.50000	0.50000
60	2	1.00000	0.99999
		1.20000	0.99999
		1.10000	0.99999
		1.00000	0.99999
		1.20000	0.99999
		1.20000	0.99999
		1.02000	0.99999
		1.00000	0.99999
		1.10000	0.99999
120	3	1.51500	1.49999
		1.51700	1.49999
		1.50000	1.49999
		1.50500	1.49999
		1.52100	1.49999
		1.53000	1.49999
		1.49900	1.49999
		1.51000	1.49999
		1.50000	1.49999
180	4	1.50100	1.49999
		2.32000	1.99998
		2.01000	1.99998
		2.10000	1.99998
		2.10000	1.99998
		2.00200	1.99998
		2.00000	1.99998
		1.99998	1.99998
		2.10000	1.99998
240	5	2.25000	1.99998
		2.30000	1.99998
		2.52367	2.49998
		2.51000	2.49998
		2.50100	2.49998
		2.49700	2.49998
		2.49900	2.49998
		2.50300	2.49998
		2.49800	2.49998
300	6	2.49900	2.49998
		2.51000	2.49998
		2.51500	2.49998
		2.99667	2.99997
		2.99900	2.99997
		3.10000	2.99997
		3.10500	2.99997
		2.99900	2.99997
		2.99600	2.99997
2.99900	2.99997		
3.10900	2.99997		
2.97700	2.99997		
2.99900	2.99997		

$$\frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

Keterangan:

- x = rata-rata data hasil perhitungan
- \bar{x} = rata-rata sampel hasil pengukuran
- s = standar deviasi sampel
- n = jumlah sampel
- v = derajat bebas = n-1
- α = tingkat signifikansi

Maka dengan n = 10, didapatkan :

Hipotesis data sudut 0°:

: =0,5

: ≠0,5

Statistik uji data sudut 0°:

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}}$$

$$= \frac{0,5345 - 0,5}{\sqrt{\dots}}$$

t = 2,814609

Untuk hasil perhitungan data lainnya dapat dilihat pada Tabel 5.7.

Tabel 5.7 Hasil Pengujian Distribusi T

POSISI MOTOR (°)	x rata-rata	μ ₀ rata-rata	s	t Hitung	alfa	1-alfa
0	0.5345	0.5	0.038762	2.814609	0.02	98
60	1.081	0.999	0.09122	2.808329	0.02	98
120	1.5098	1.499	0.01065	2.91187	0.02	98
180	2.118	1.998	0.12706	2.94224	0.02	98
240	2.50557	2.499	0.00882	2.00304	0.05	95
300	3.028	2.999	0.05338	1.65855	0.1	90

Dari Tabel 5.7 didapatkan tingkat kepercayaan (1 –) yang digunakan dalam pengujian adalah 90%-98%.

BAB VI PENUTUP

6.1 Kesimpulan

Dari perancangan, pengujian dan pengamatan yang telah dilakukan pada *rotary parking* maka dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan data respons sistem yang diperoleh dari pengujian dengan menggunakan metode kedua dari teori *Ziegler-Nicholes*, maka parameter kontroler PID dapat ditentukan pada motor DC mempunyai *gain* $K_p = 4.8$, $K_i = 2$, $K_d = 2.88$ dan toleransi error sebesar 2%-10%.
2. Hasil pengujian terhadap aplikasi kontroler PID ini menunjukkan bahwa respons sistem untuk tiap sudut mempunyai *error stedy state* sebesar 0-0.146%. Sedangkan T_s sudut 60=1,169 s, sudut 120=2,2415 s, sudut 180=3,507 s, sudut 240=4,342 s, dan sudut 300=5,331 s, Hasil pengujian ini menunjukkan bahwa kontroler PID menghasilkan respons sesuai yang direncanakan dan dapat diaplikasikan pada *rotary parking* untuk mencapai posisi yang dituju.

6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut disarankan:

1. Untuk pergerakan *rotary parking* disarankan tidak ada batasan sudut putar motor DC untuk memaksimalkan gerak dari *rotary parking*.
2. Dilakukan metode pengontrolan selain kontroler PID.

DAFTAR PUSTAKA

- [1] Alfahrizi, Rosyam. 2011. *Implementasi Kontrol PID untuk Pergerakan Robot Lengan ROB2R pada Pemindahan Objek dengan Variasi Ketinggian*. Skripsi tidak dipublikasikan. Malang: Universitas Brawijaya.
- [2] ATMEL. 2007. *ATMEGA16/ATMEGA16L, 8-bit AVR Microcontroller with 16 Kbytes in System Programable Flash*
- [3] Maulana, Fakhri. 2010. *Desain Implementasi Kontroler PID pada lengan Robot dengan Tiga Derajat Kebebasan*, Malang. Fakultas Teknik Universitas Brawijaya.
- [4] Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid 1*. Jakarta. Penerbit Erlangga.
- [5] Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid 2*. Jakarta. Penerbit Erlangga

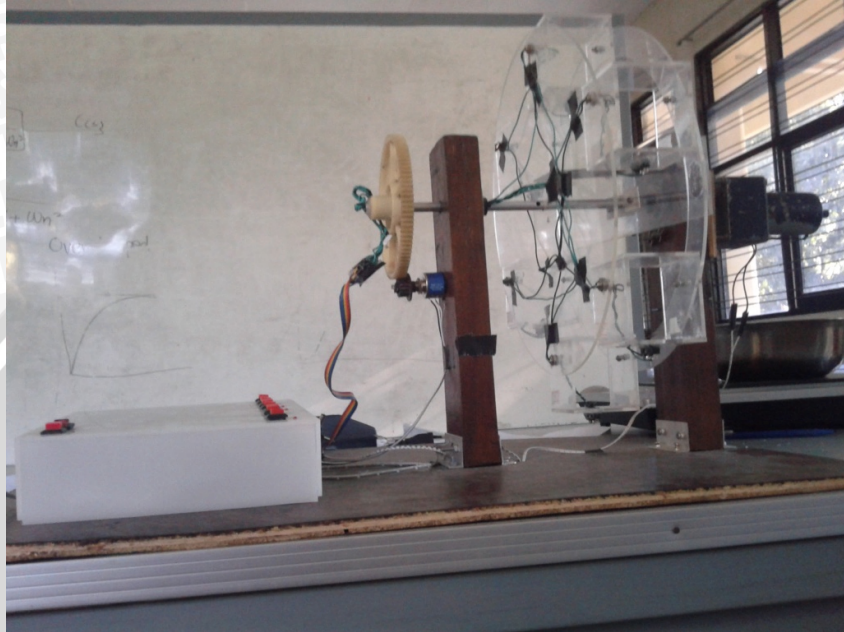


LAMPIRAN



LAMPIRAN I

FOTO ALAT



Gambar Alat Tampak Depan



Gambar Alat Tampak Samping



Gambar Alat Tampak Atas



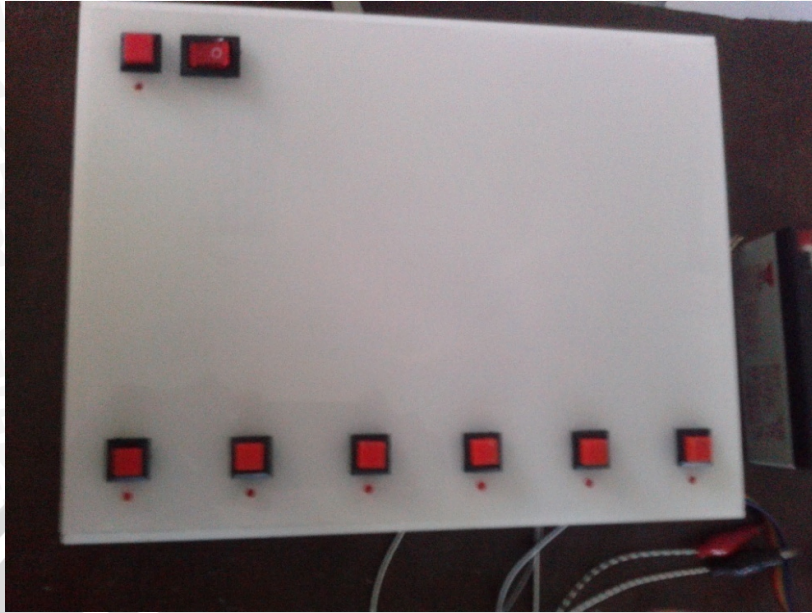
Gambar Motor DC pada Alat



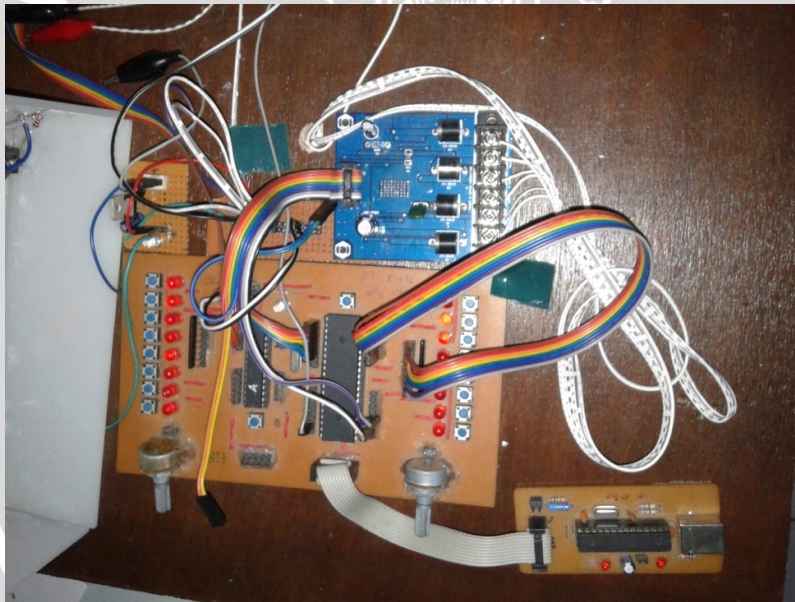
Gambar *Limit Switch* pada Alat



Gambar Sensor Potensiometer pada Alat



Gambar *Push Button* pada Alat



Gambar Rangkaian Elektrik pada Alat

LAMPIRAN II

LISTING PROGRAM

/*****

This program was produced by the
CodeWizardAVR V1.24.8d Professional
Automatic Program Generator

© Copyright 1998-2006 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project : Aplikasi Kontrol PID untuk Motor DC dalam sistem Rotary Parking

Version :

Date : 03/07/2012

Author : Kharisma Darmawan Putra

Company : Teknik Elektro

Comments:

Chip type : ATmega32
Program type : Application
Clock frequency : 16,000000 MHz
Memory model : Small
External SRAM size : 0
Data Stack size : 512



*****/

```
#include <mega32.h>
```

```
#include <math.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```



```
#define ADC_VREF_TYPE 0x00

#define maju  PORTB.1=0;PORTB.2=1;
#define mundur PORTB.1=1;PORTB.2=0;

#define stop  PORTB.1=1;PORTB.2=1;OCR0=0;

#define max_se 200
#define min_se -200
```

```
//deklarasi PIN untuk LS
```

```
#define LS0 PINC.0
#define LS1 PINC.1
#define LS2 PINC.2
#define LS3 PINC.3
#define LS4 PINC.4
#define LS5 PINC.5
```

```
//deklarasi PIN untuk PB
```

```
#define PB0 PIND.0
#define PB1 PIND.1
#define PB2 PIND.2
#define PB3 PIND.3
#define PB4 PIND.4
#define PB5 PIND.5
```

```
void LS();
```

```
void PB();
```

```
void MOTOR_MOVE_PID();
```

```
void KONTROLER(int i);
```

```
int SP,e=0,es=0,de=0,se=0;
```

```
float Kp= 4.8,Ki=8.9,Kd=0.6;
```




```
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE;

    // Start the AD conversion
    ADCSRA|=0x40;

    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

// Declare your global variables here

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

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

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

```
DDRB=0x0E;
```

```
// 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=0x00;
```

```
DDRC=0x00;
```

```
// 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: 15,625 kHz
```

```
// Mode: Fast PWM top=FFh
```

```
// OC0 output: Non-Inverted PWM
```

```
TCCR0=0x6D;
```

```
TCNT0=0x00;
```

```
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer 1 Stopped
```

```
// Mode: Normal top=FFFFh
```

```
// OC1A output: Discon.
```

```
// OC1B output: Discon.
```

```
// Noise Canceler: Off
```



```
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
```

```
TCCR1A=0x00;
```

```
TCCR1B=0x00;
```

```
TCNT1H=0x00;
```

```
TCNT1L=0x00;
```

```
ICR1H=0x00;
```

```
ICR1L=0x00;
```

```
OCR1AH=0x00;
```

```
OCR1AL=0x00;
```

```
OCR1BH=0x00;
```

```
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization
```

```
// Clock source: System Clock
```

```
// Clock value: Timer 2 Stopped
```

```
// Mode: Normal top=FFh
```

```
// OC2 output: Disconnected
```

```
ASSR=0x00;
```

```
TCCR2=0x00;
```

```
TCNT2=0x00;
```

```
OCR2=0x00;
```

```
// External Interrupt(s) initialization
```

```
// INT0: Off
```

```
// INT1: Off
```

```
// INT2: Off
```

UNIVERSITAS BRAWIJAYA



```
MCUCR=0x00;
```

```
MCUCSR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x00;
```

```
// USART initialization
```

```
// Communication Parameters: 8 Data, 1 Stop, No Parity
```

```
// USART Receiver: Off
```

```
// USART Transmitter: On
```

```
// USART Mode: Asynchronous
```

```
// USART Baud rate: 9600
```

```
UCSRA=0x00;
```

```
UCSRB=0x08;
```

```
UCSRC=0x86;
```

```
UBRRH=0x00;
```

```
UBRRL=0x67;
```

```
// Analog Comparator initialization
```

```
// Analog Comparator: Off
```

```
// Analog Comparator Input Capture by Timer/Counter 1: Off
```

```
ACSR=0x80;
```

```
SFIOR=0x00;
```

```
// ADC initialization
```

```
// ADC Clock frequency: 1000,000 kHz
```

```
// ADC Voltage Reference: AREF pin
```

```
ADMUX=ADC_VREF_TYPE;
```

```
ADCSRA=0x84;
```



```
while (1)
{
    LS0;
    PB0;
};
```

```
//cek LS (room kosong)
```

```
void LS0()
```

```
{
```

```
int flag=0;
```

```
e=0;
```

```
de=0;
```

```
se=0;
```

```
while(flag==0)
```

```
{
```

```
if(LS0==0)
```

```
{
```

```
if(LS1==0)
```

```
{
```

```
if(LS2==0)
```

```
{
```

```
if(LS3==0)
```

```
{
```

```
if(LS4==0)
```

```
{
```

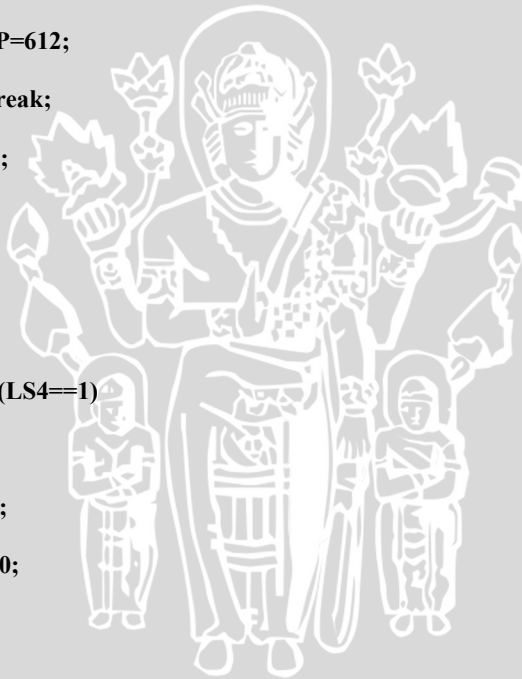
```
if(LS5==0)
```

```
{
```

UNIVERSITAS BRAWIJAYA



```
while(LS5==0)
{
break;
}
break;
}
else
{
while(LS1==1)
{
flag++;
SP=612;
break;
}};
break;
}
else
{while(LS4==1)
{
flag++;
SP=510;
break;
}};
break;
}
else
{while(LS3==1)
{
flag++;
SP=408;
```



```
break;
};
break;
}
else
{while(LS2==1)
{
flag++;
SP=306;
break;
}};
break;
}
else
{while(LS1==1)
{
flag++;
SP=204;
break;
}};
break;
}
else
{
while(LS0==1)
{
flag++;
SP=102;
break;
}};
```

UNIVERSITAS BRAWIJAYA



```
}  
while(flag==1)  
{  
    MOTOR_MOVE_PID();  
    while  
    (((SP==102)&(LS0==0))||((SP==204)&(LS1==0))||((SP==306)&(LS2==0))||((SP==408)&(LS3==0))||((SP==5  
    10)&(LS4==0))||((SP==612)&(LS5==0)))  
    {  
        stop;  
        flag=0;  
        break;  
    };  
};  
}
```

```
//cek PB
```

```
void PB()  
{
```

```
    if(PB0==0)
```

```
    {
```

```
        PB0=0;
```

```
        if (LS0==0)
```

```
            {SP=102;}
```

```
            MOTOR_MOVE_PID();
```

```
            while(read_adc(1)==SP)
```

```
            {
```

```
                stop;
```

```
                PB0=1;
```

```
                break;
```

```
            };
```

```
};
```

UNIVERSITAS BRAWIJAYA




```
if(PB1==0)
{
    PB1=0;
    if(LS1==0)
        {SP=204;}
    MOTOR_MOVE_PID();
    while(read_adc(1)==SP)
    {
        stop;
        PB1=1;
        break;
    };
};

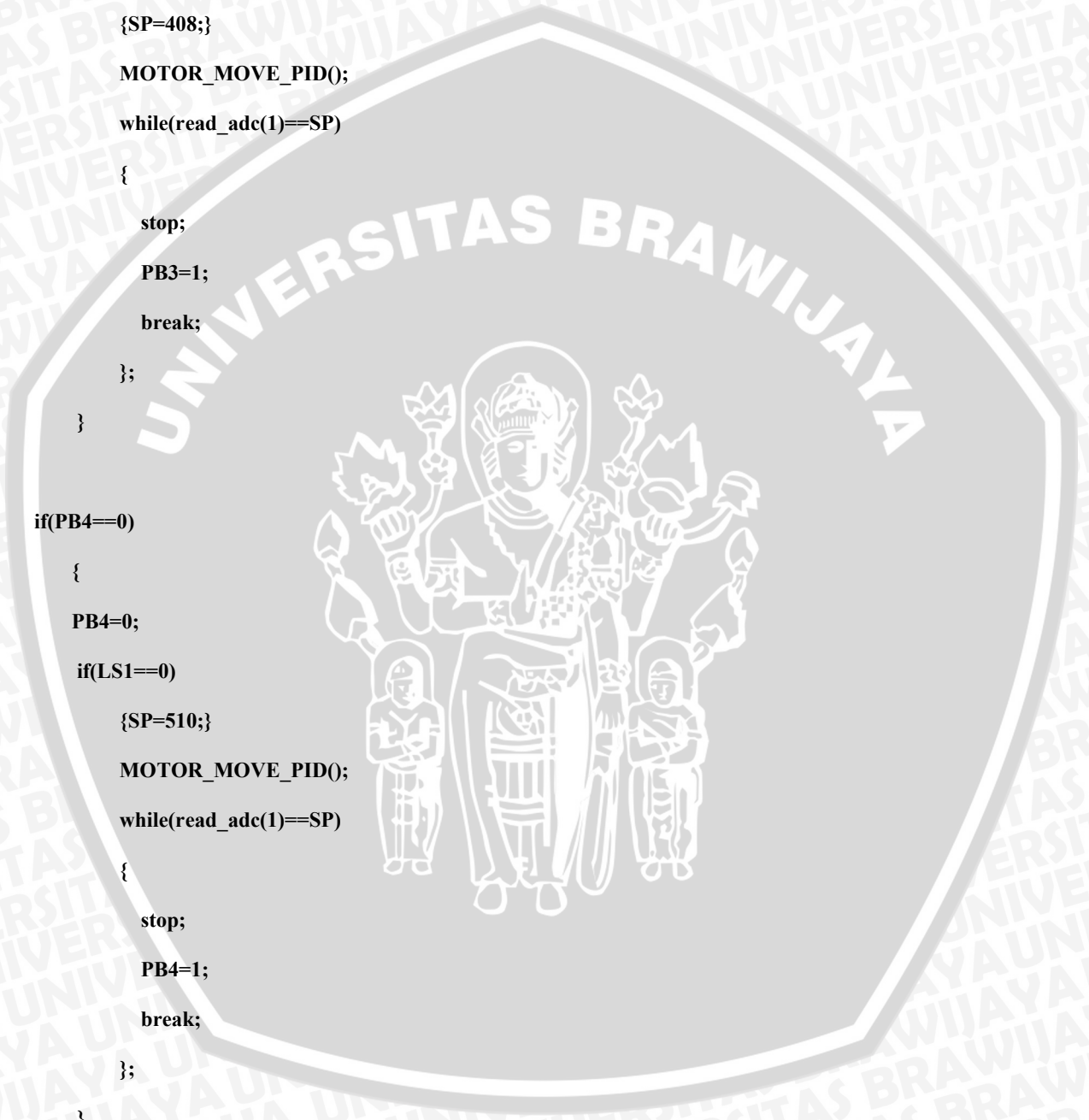
if(PB2==0)
{
    PB2=0;
    if(LS2==0)
        {SP=306;}
    MOTOR_MOVE_PID();
    while(read_adc(1)==SP)
    {
        stop;
        PB2=1;
        break;
    };
};
```



```
if(PB3==0)
{
  PB3=0;
  if(LS3==0)
  {SP=408;}
  MOTOR_MOVE_PID();
  while(read_adc(1)==SP)
  {
    stop;
    PB3=1;
    break;
  };
}

if(PB4==0)
{
  PB4=0;
  if(LS1==0)
  {SP=510;}
  MOTOR_MOVE_PID();
  while(read_adc(1)==SP)
  {
    stop;
    PB4=1;
    break;
  };
}
```

```
if(PB5==0)
```



```
{
  PB5=0;
  if(LS1==0)
    {SP=612;}
  MOTOR_MOVE_PID();
  while(read_adc(1)==SP)
  {
    stop;
    PB5=1;
    break;
  };
}
```

//Kontroler

```
void MOTOR_MOVE_PID()
```

```
{
```

```
int x=0;
```

```
float p=0,d=0,i=0,pid=0;
```

```
x= (int) read_adc(1);
```

```
e=(SP-x);
```

```
de=e-es;
```

```
se=se+e;
```

```
//windup prevention
```

```
if(se>max_se)
```

```
{
```

```
se=max_se;
```

```
};
```

UNIVERSITAS BRAWIJAYA



```
if(se<min_se)
{
se=min_se;
};

p=(float) e;
p= Kp*p;

i=(float) se;
i= Ki*i;

d=(float) de;
d= Kd*d;

pid=p+i+d;
x=(int) pid;

KONTROLER(x);

es=e;
}
```

```
void KONTROLER(int i)
{
int y=0;
y=i;

if (y>255)
{
y=255;
};

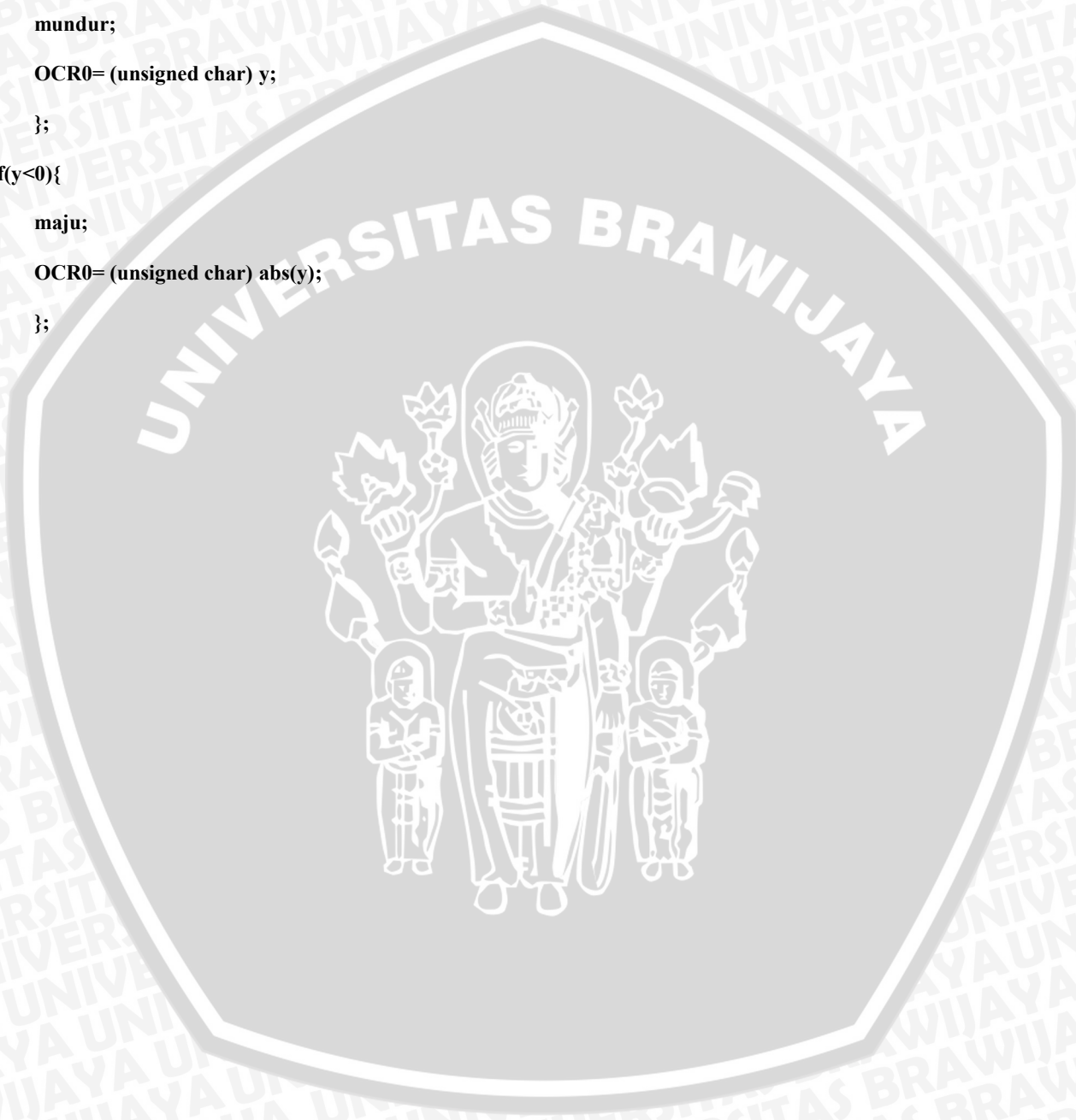
if (y<-255)
{
y=-255;
};

if(y==0)
```

UNIVERSITAS BRAWIJAYA



```
{  
  stop;  
};  
if(y>0){  
  mundur;  
  OCR0= (unsigned char) y;  
};  
if(y<0){  
  maju;  
  OCR0= (unsigned char) abs(y);  
};  
}
```



LAMPIRAN III
DATA SHEET



LAMPIRAN IV
TABEL UJI-T

