

**PENGATURAN POSISI MOTOR SERVO
PADA MINIATUR *ROTARY PARKING***

**SKRIPSI
JURUSAN TEKNIK ELEKTRO**

*Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik*



Disusun Oleh:

NOVANDRA PUSPAWARDHANA

NIM. 0710633012-63

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2014**

LEMBAR PERSETUJUAN

**PENGATURAN POSISI MOTOR SERVO
PADA MINIATUR *ROTARY PARKING***

SKRIPSI

JURUSAN TEKNIK ELEKTRO

*Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik*



Disusun oleh:

NOVANDRA PUSPAWARDHANA

NIM. 0710633012-63

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

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

Tri Nurwati, S.T., M.T.
NIP. 197906152 200812 2 003

LEMBAR PENGESAHAN

PENGATURAN POSISI MOTOR SERVO PADA MINIATUR *ROTARY PARKING*

Disusun oleh:

NOVANDRA PUSPAWARDHANA

NIM. 0710633012-63

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal **11 Agustus 2014**

Majelis Penguji :

Ir. Moch. Rusli, Dipl.Ing.
NIP. 19630104 198701 1 001

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

Dr. Ir. Erni Yudaningtyas, M.T.
NIP. 19650913 199002 2 001

Mengetahui :
Ketua Jurusan Teknik Elektro

M. Aziz Muslim, S.T., M.T., Ph.D.
NIP. 19741203 200012 1 001

PENGANTAR

Alhamdulillah, segenap puji dan syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Pengaturan Posisi Motor Servo Pada *Rotary Parking*” yang diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik.

Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada berbagai pihak yang telah membantu dan mendukung dalam penyelesaian skripsi ini, yaitu :

1. Keluarga di rumah : Bosbesar, Ibu, Mbak Ririn, Dek Metta.
2. Seluruh jajaran dosen dan staff di TEUB.
3. Pembimbing skripsi Ibu Fitri dan Ibu Tri.
4. Seluruh karyawan fakultas Teknik di Elektro
5. Angkatan ‘CORE’ 2007 (Kahim, Galih, Nizam, Adi, Gitok, Eres, Gosong, Maul, Yuli, Puput dan yang lainnya).
6. Angkatan 2006, 2008, 2009, 2010 yang telah membantu saya baik akademis maupun non akademis.

Dalam penulisan skripsi ini, penulis menyadari adanya kekurangan dan ketidaksempurnaan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun untuk kelengkapan dan kesempurnaan skripsi ini. Penulis berharap semoga skripsi ini dapat bermanfaat khususnya bagi rekan-rekan mahasiswa.

Malang, Agustus 2014

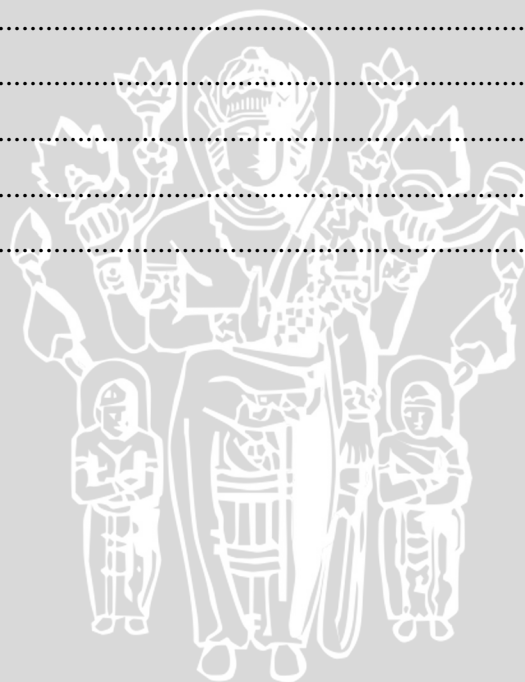
Penulis

DAFTAR ISI

PENGANTAR	i
DAFTAR ISI	ii
DAFTAR TABEL	iv
DAFTAR GAMBAR	v
DAFTAR LAMPIRAN	vii
ABSTRAK	vii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Sistematika Pembahasan	2
BAB II TINJAUAN PUSTAKA	4
2.1 Motor Servo	4
2.2 Kontroller	6
2.3 Kontroller <i>On-Off</i>	7
2.4 Mikrokontroller Arduino-Uno	9
2.5 Teori Dasar Sensor	9
2.6 Sensor Potensiometer Linier	9
2.7 Sensor <i>Limit Switch</i>	9
2.8 <i>Push Button</i>	10
BAB III METODOLOGI	12
3.1 Perancangan Sistem	12
3.2 Spesifikasi Alat	12
3.3 Perancangan dan Realisasi Pembuatan Alat	12
3.4 Pengujian dan Analisa Data	13
3.5 Pengambilan Kesimpulan.....	13
BAB IV PERANCANGAN DAN PEMBUATAN ALAT	14
4.1. Perancangan Sistem	14
4.1.1 Diagram Blok Sistem.....	14
4.1.2 Prinsip Kerja Alat	15



4.2.	Perancangan Perangkat Keras	15
4.2.1	Spesifikasi Alat	15
4.2.2	Potensiometer Linier	17
4.2.3	Modul Arduino Uno.....	18
4.3.	Perancangan Kontroller <i>On-Off</i>	19
4.4.	Perancangan Pergerakan <i>Rotary Parking</i>	20
4.5.	Perancangan Perangkat Lunak	22
BAB V PENGUJIAN ALAT		23
5.1	Pengujian Potensiometer Linier	23
5.2	Pengujian Motor Servo	27
5.3	Pengujian <i>Limit Switch</i> dan <i>Push Button</i>	29
5.4	Pengujian Keseluruhan Sistem.....	34
BAB VI PENUTUP		38
6.1.	Kesimpulan	38
6.2.	Saran.....	41
DAFTAR PUSTAKA		42
LAMPIRAN		43



DAFTAR TABEL

Lampiran	Judul	Halaman
Tabel 4. 1 :	Fungsi Pin Arduino Uno	18
Tabel 5. 1 :	Pengujian Potensiometer.....	24
Tabel 5. 2 :	Pengujian Sinyal PWM dari Motor Servo	28
Tabel 5. 3 :	<i>Set Point Rotary Parking</i>	32
Tabel 5. 4 :	Data Pengujian Posisi <i>Rotary Parkir</i> Keadaan Parkir Dengan Beban 200 Gram	32
Tabel 5. 5 :	Data Pengujian Posisi <i>Rotary Parkir</i> Keadaan Parkir Dengan Beban 400 Gram	33
Tabel 5. 6 :	Data Pengujian Posisi <i>Rotary Parkir</i> Keadaan Ambil Parkir Dengan Beban 200 Gram	33
Tabel 5. 7 :	Data Pengujian Posisi <i>Rotary Parkir</i> Keadaan Parkir Ambil Dengan Beban 400 Gram	34
Tabel 5. 8 :	Data Pengujian antara Setpoint Motor Servo dan Sudut Baca Potensiometer Pada Setpoint 0 ⁰	34
Tabel 5. 9 :	Data Perbandingan antara Setpoint Motor Servo dan Sudut Baca Potensiometer Pada Setpoint 60 ⁰	35
Tabel 5. 10 :	Data Perbandingan antara Setpoint Motor Servo dan Sudut Baca Potensiometer Pada Setpoint 120 ⁰	35
Tabel 5. 11 :	Data Perbandingan antara Setpoint Motor Servo dan Sudut Baca Potensiometer Pada Setpoint 180 ⁰	35
Tabel 5. 12 :	Data Pengujian antara Setpoint Motor Servo dan Sudut Baca Potensiometer Pada Setpoint 0 ⁰	36
Tabel 5. 13 :	Data Perbandingan antara Setpoint Motor Servo dan Sudut Baca Potensiometer Pada Setpoint 60 ⁰	36
Tabel 5. 14 :	Data Perbandingan antara Setpoint Motor Servo dan Sudut Baca Potensiometer Pada Setpoint 120 ⁰	37
Tabel 5. 15 :	Data Perbandingan antara Setpoint Motor Servo dan Sudut Baca Potensiometer Pada Setpoint 180 ⁰	37



DAFTAR GAMBAR

BAB II

Gambar 2.1 : Motor Servo	4
Gambar 2.2 : Konfigurasi Pin Pengkabelan Motor Servo	5
Gambar 2.3 : Pengaturan Sudut Motor Servo	5
Gambar 2.4 : Ilustrasi dari Kontroller <i>On-Off</i>	7
Gambar 2.5 : Ilustrasi Band Pada Setpoint Kontroller <i>On-Off</i>	8
Gambar 2.6 : Skema Arduino Uno	8
Gambar 2.7 : Potensiometer Linier 10K Ω	9
Gambar 2.8 : Simbol dan Bentuk <i>Limit Switch</i>	10
Gambar 2.9 : Konstruksi dan Simbol <i>Limit Switch</i>	10
Gambar 2.10 : Simbol dan macam <i>bentuk Push Button</i>	11

BAB IV

Gambar 4.1 : Diagram Blok Sistem	14
Gambar 4.2 : Skema Model <i>Rotary Parking</i> Tampak Samping	16
Gambar 4.3 : Skema Model <i>Rotary Parking</i> Tampak Depan	16
Gambar 4.4 : Skema Model <i>Rotary Parking</i> Tampak Atas	17
Gambar 4.5 : Rangkaian Potensiometer	17
Gambar 4.6 : Potensiometer pada Sistem	18
Gambar 4.7 : Modul Arduino Uno	19
Gambar 4.8 : Desain Sistem Arduino Uno	19
Gambar 4.9 : Diagram Blok Kontroller <i>On-Off</i>	20
Gambar 4.10 : Persamaan Kontrol <i>On-Off</i>	20
Gambar 4.11 : Pergerakan Ketika <i>Push Buton</i> dan <i>Limit Switch</i> yang Dipilih Bernomor 1 Saat Garasi 1 Sebagai Acuan	21
Gambar 4.12 : Pergerakan Ketika <i>Push button</i> dan <i>Limit switch</i> yang Dipilih Bernomor 2 Saat Garasi 1 Sebagai Acuan	21
Gambar 4.13 : Pergerakan Ketika <i>Push button</i> dan <i>Limit switch</i> yang Dipilih Bernomor 1 Saat Garasi 2 Sebagai Acuan	22
Gambar 4.14 : Flowchart Perancangan Perangkat Lunak	22

BAB V

Gambar 5.1 : Pengujian Potensiometer	23
Gambar 5.2 : Grafik Hubungan Antara Sudut Dengan Resistansi	25
Gambar 5.3 : Grafik Hubungan Antara Sudut Dengan Tegangan Potensiometer	25
Gambar 5.4 : Grafik Perbandingan Antara Tegangan Perhitungan dan Tegangan Pengukuran Pada Potensiometer	26
Gambar 5.5 : Grafik Perbandingan Antara Tegangan Perhitungan dan Tegangan Pengukuran Terhadap Resistansi Potensiometer	26
Gambar 5. 6 : Skema Pengujian Motor Servo	27
Gambar 5. 7 : Grafik Hubungan PWM Terhadap Sudut Motor	29
Gambar 5. 8 : Skema Pengujian <i>Limit Switch</i>	30
Gambar 5. 9 : Skema Pengujian <i>Push Button</i>	30
Gambar 5.10 : Grafik Respon <i>Limit Switch</i>	30
Gambar 5. 11 : Grafik Respon <i>Push Button</i>	31
Gambar 5. 12 : Grafik Perbandingan Sudut <i>Setpoint</i> 0 ⁰ dan Pembacaan Potensiometer Linier.	38
Gambar 5. 13 : Grafik Perbandingan Sudut <i>Setpoint</i> 60 ⁰ dan Pembacaan Potensiometer Linier	38
Gambar 5. 14 : Grafik Perbandingan Sudut <i>Setpoint</i> 120 ⁰ dan Pembacaan Potensiometer Linier	39
Gambar 5. 15 : Grafik Perbandingan Sudut <i>Setpoint</i> 180 ⁰ dan embacaan Potensiometer Linier	39
Gambar 5. 16 : Grafik Perbandingan Sudut <i>Setpoint</i> 0 ⁰ dan Pembacaan Potensiometer Linier	40
Gambar 5. 17 : Grafik Perbandingan Sudut <i>Setpoint</i> 60 ⁰ dan Pembacaan Potensiometer Linier	40
Gambar 5. 18 : Grafik Perbandingan Sudut <i>Setpoint</i> 120 ⁰ dan Pembacaan Potensiometer Linier	41
Gambar 5. 19 : Grafik Perbandingan Sudut <i>Setpoint</i> 180 ⁰ dan Pembacaan Potensiometer Linier	41



DAFTAR LAMPIRAN

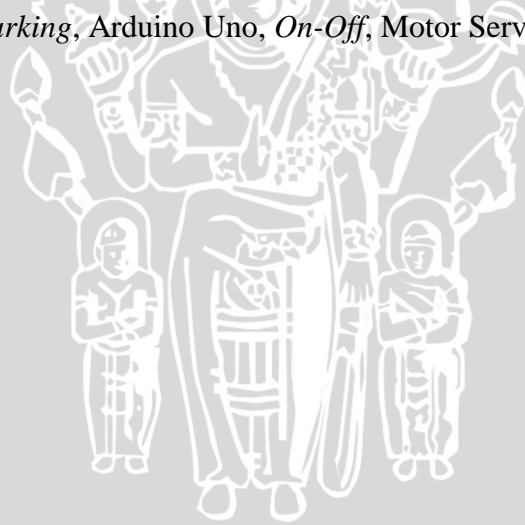
Lampiran	Judul	Halaman
Lampiran 1	: Foto Alat	43
Lampiran 2	: Listing Program	45
Lampiran 3	: Datasheet	56



ABSTRAK

Saat ini, peningkatan jumlah mobil tidak berbanding lurus dengan bertambahnya jumlah lahan parkir. Salah satu solusi mengatasi permasalahan tersebut menggunakan sistem *rotary parking*, yaitu sistem parkir otomatis menggunakan lahan yang terbatas untuk menampung mobil dengan jumlah lebih banyak daripada sistem parkir konvensional. Metode kontrol yang digunakan pada sistem *rotary parking* ini adalah controller *On-Off*. Salah satu keuntungan metode ini adalah sederhana dalam perancangan dan instalasi. Dalam perancangannya digunakan Arduino Uno, potensiometer linier, motor servo dan komunikasi USB. Dari hasil pengujian terhadap aplikasi kontroler *On-Off* menunjukkan bahwa pada percobaan menggunakan beban 200 gram *error* rata-rata pada sudut 0° sebesar 0,348%, pada sudut 60° sebesar 3,333%, pada sudut 120° sebesar 3,375% dan pada sudut 180° sebesar 3,889%. Pada percobaan menggunakan beban 400 gram *error* rata-rata pada sudut 0° sebesar 0,635%, pada sudut 60° sebesar 6,665%, pada sudut 120° sebesar 8,15% dan pada sudut 180° sebesar 8,055%.

Kata kunci: *Rotary Parking*, Arduino Uno, *On-Off*, Motor Servo.



BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan zaman menuntut manusia untuk melakukan mobilitas yang tinggi, sehingga kebutuhan akan alat transportasi pribadi menjadi hal yang umum bagi masyarakat. Salah satu alat transportasi pribadi yang digunakan adalah mobil. Dengan berbagai kelebihan dan harga yang semakin terjangkau 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 telah diterapkan di negara-negara maju seperti Jepang dan Jerman serta terbukti mampu mengatasi keterbatasan jumlah lahan parkir yang relatif sempit. 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.

Oleh karena itu pada skripsi ini akan dibuat model *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 servo. Motor servo adalah sebuah motor dengan sistem umpan balik tertutup di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor servo, serangkaian gear, potensiometer dan rangkaian kontrol. Kelebihan motor servo yaitu bisa dikendalikan derajat putarannya dan dapat dipilih keperluan sesuai dengan kekuatannya. Pengaturan sudut putaran pada motor servo ini menggunakan Arduino Uno.

Diharapkan model *rotary parking* ini dapat dimanfaatkan sebagai acuan agar dapat dikembangkan lebih lanjut.

1.2 Rumusan Masalah

1. Bagaimana merancang sistem *rotary parking* dengan pembagian sudut menggunakan modul Arduino Uno dengan bahasa pemrograman Arduino 1.0.5 ?
2. Bagaimana mengimplementasikan kontrol *On-Off* berbasis arduino sebagai penggerak motor servo dalam pengaturan posisi sudut putar motor servo pada sistem *rotary parking* ?

1.3 Batasan Masalah

Untuk menekankan pada objek pembahasan yang ada maka pada penelitian ini diberikan batasan masalah sebagai berikut:

1. Sistem parkir otomatis yang dibuat merupakan sebuah model, dengan tidak menitikberatkan pada mekanisme keamanan dan kenyamanan pengendara.
2. Pembahasan modul Arduino Uno sebagai tempat proses data pada sistem.
3. Dimensi dan berat mobil ditentukan menggunakan replika.
4. Motor servo *Towerpro MG995* dengan sudut putar maksimal 180° .
5. Ptensiometer linier sebagai pendeteksi sudut putar motor servo.
6. *Limit switch* sebagai sensor pendeteksi ada tidaknya mobil.

1.4 Tujuan

Tujuan dari sistem ini adalah merancang model sistem *rotary parking* menggunakan motor servo sebagai penggerak dengan kontroler *On-Off*.

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 secara garis besar adalah sebagai berikut:

BAB I

Pendahuluan

Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II Tinjauan Pustaka

Menjelaskan dasar teori penunjang penelitian yang ada pada alat ini, yang terdiri dari modul Arduino Uno sebagai sistem otomasi, motor servo, sensor *limit switch*, sensor potensiometer linier, *push button*, *led*, serta kontrol *On-Off*.

BAB III Metode Penelitian

Menjelaskan tentang metodologi penelitian yang terdiri dari perancangan sistem, spesifikasi alat, perancangan dan realisasi pembuatan alat, pengujian dan analisa data dan pengambilan kesimpulan

BAB IV Perancangan dan Pembuatan Alat

Menjelaskan tentang perancangan dan pembuatan alat yang meliputi prinsip kerja alat, perancangan perangkat keras dan perangkat lunak.

BAB V Pengujian Alat

Menjelaskan tentang pengujian alat dan analisa yang meliputi pengujian bagian blok sistem dan pengujian sistem secara keseluruhan.

BAB VI Penutup

Menjelaskan tentang pengambilan kesimpulan sesuai dengan hasil perealisasiian dan pengujian alat sesuai dengan tujuan dan rumusan masalah, serta pemberian saran untuk pengembangan.

BAB II

TINJAUAN PUSTAKA

2.1 Motor Servo

Motor servo adalah sebuah motor dengan sistem *closed feedback* di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri atas sebuah motor, serangkaian *internal gear*, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut putaran servo. Sedangkan sudut sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.

Motor servo mampu bekerja dua arah (*CW* dan *CCW*) di mana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal *PWM* pada bagian pin kontrolnya. Gambar 2.1. menunjukkan gambar motor servo.

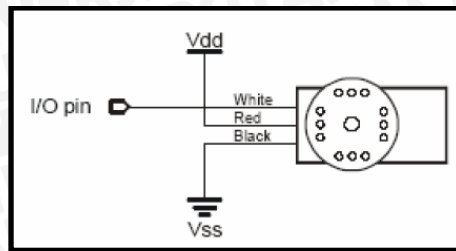


Gambar 2.1. Motor Servo

Motor servo merupakan motor yang berputar lambat, di mana biasanya ditunjukkan oleh *rate* putarannya yang lambat, namun demikian memiliki torsi yang kuat karena *internal gear*-nya. Karakteristik motor servo adalah sebagai berikut :

1. Memiliki 3 jalur kabel : *power*, *ground*, dan *control* seperti ditunjukkan dalam Gambar 2.2.
2. Pin *Control* untuk mengendalikan posisi.
3. Operasional motor servo dikendalikan oleh sebuah pulsa selebar ± 20 ms, di mana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari *range* sudut maksimum.
4. Konstruksi didalamnya meliputi *internal gear*, *potensiometer*, dan *feedback control*.

Gambar 2.2. menunjukkan konfigurasi pin pengkabelan motor servo.



Gambar 2.2. Konfigurasi Pin Pengkabelan Motor Servo

Sumber : Parallax, Inc.

Secara umum terdapat dua jenis motor servo, yaitu :

1. Motor Servo Standar 180°

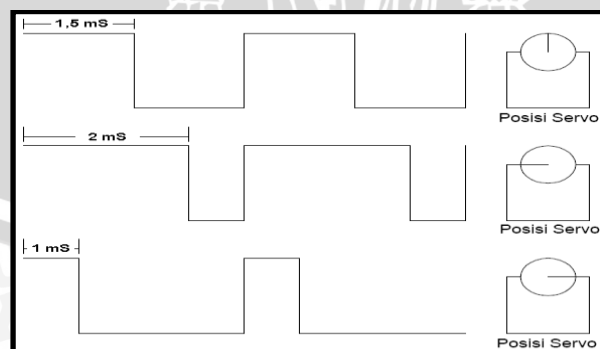
Motor servo jenis ini merupakan motor yang hanya mampu bergerak dua arah (*CW* dan *CCW*) dan mempunyai defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180°.

2. Motor Servo *Continuous*

Motor servo jenis ini mampu bergerak dua arah (*CW* dan *CCW*) dan tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu) sehingga motor ini berputar 360°.

Pengaturan sudut motor servo diperlukan untuk mengetahui gerakan dari motor servo dan pulsa yang harus diberikan untuk bergerak ke kanan atau bergerak ke kiri.

Gambar 2.3. menunjukkan teknik PWM untuk mengatur sudut motor servo.



Gambar 2.3. Pengaturan Sudut Motor Servo

Sumber : Parallax, Inc.

Dalam Gambar 2.3. diasumsikan bahwa saat diberikan sinyal periodik dengan lebar 1 ms maka motor servo akan bergerak dengan sudut 0° , jika diberi sinyal 1,5 ms maka motor servo akan bergerak dengan sudut 90° , dan jika diberi sinyal 2 ms maka motor servo akan bergerak dengan sudut 180° . Dari kondisi tersebut bisa ditarik kesimpulan bahwa untuk mengatur sudut putaran motor servo diperlukan sinyal dengan lebar pulsa seperti pada persamaan berikut:

$$S = D \left(\frac{1000}{180} \right) + 1000 \mu s \text{ atau } \dots\dots\dots (2.1)$$

$$S = (0,18)D + 1000 \mu s$$

dimana

- $S =$ Lebar Pulsa dalam μs
- $D =$ Sudut putar servo dalam derajat

2.2 Kontroler

Keberadaan kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem. Pada prinsipnya hal itu disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Artinya, karakteristik *plant* harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu subsistem, yaitu kontroler.

Salah satu fungsi komponen kontroler adalah mengurangi sinyal kesalahan, yaitu perbedaan antara nilai referensi/nilai yang diinginkan dan nilai aktual. Hal ini sesuai dengan tujuan sistem control, yaitu mendapat nilai sinyal keluaran sama dengan nilai yang diinginkan/referensi. Semakin kecil kesalahan yang terjadi, semakin baik kinerja sistem kontrol yang diterapkan.

Apabila perbedaan antara nilai referensi dengan nilai keluaran relatif besar, maka kontroler yang baik seharusnya mampu mengatasi perbedaan ini dengan segera menghasilkan sinyal keluaran untuk mengubah perilaku *plant*. Dengan demikian sistem secara cepat mengubah keluaran *plant* sampai diperoleh selisih dengan nilai referensi sekecil mungkin.

Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran *plant* dengan nilai referensi, kemudian menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata, 1995).

2.3 Kontroler *On-Off*

Pada sistem kontrol dua posisi, elemen aktuasi hanya mempunyai dua posisi yang tetap. Kontrol *On-Off* ini banyak digunakan di industri karena murah dan sederhana. Sinyal kontrol akan tetap pada satu keadaan dan akan berubah ke keadaan lainnya bergantung pada nilai *error* positif atau negatif.

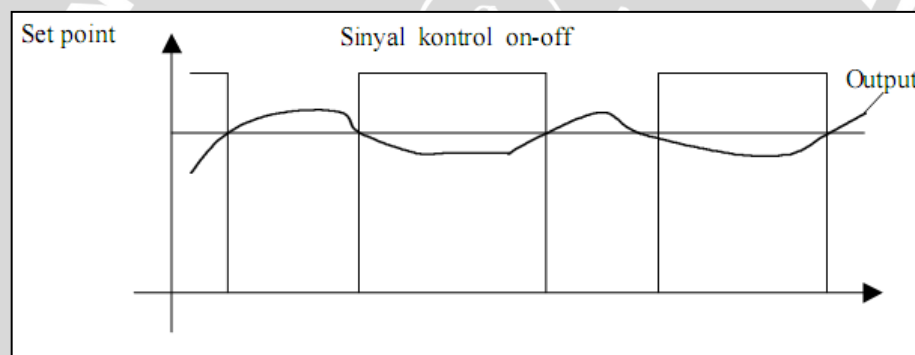
$u(t)$ = sinyal kontrol

$e(t)$ = sinyal *error*

$u(t) = U1$, $e(t) > 0$

$= U2$, $e(t) < 0$

Kontroler dua posisi pada umumnya dijumpai pada komponen elektrik (*relay*) dan komponen pneumatik (katup dan silinder). Ilustrasi dari kontroler *On-Off* seperti pada gambar 2.4.



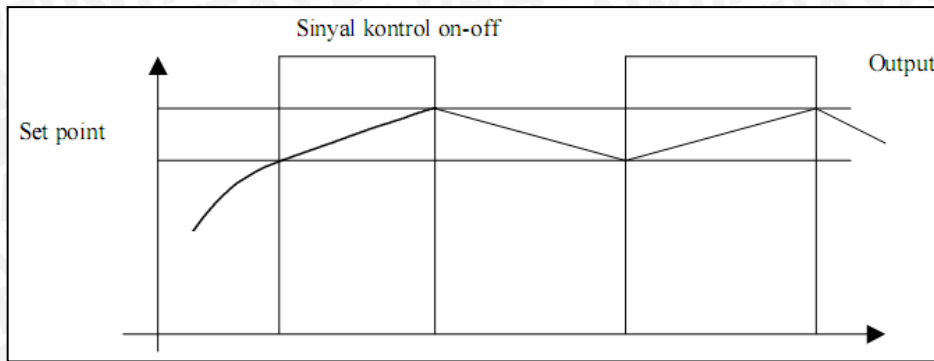
Gambar 2.4. Ilustrasi dari Kontroler *On-Off*

Sumber : www.geocities.com/al_dodi/kerja/kp3.pdf

Dari gambar 2.4 dapat diamati bahwa jika *output* lebih besar dari *set point*, aktuatur akan *off*. *Output* akan turun dengan sendirinya sehingga menyentuh *set point* lagi. Pada saat itu, sinyal kontrol akan kembali *on* (aktuatur *on*) dan mengembalikan *output* pada *set point*nya. Demikian seterusnya sinyal kontrol dan aktuatur akan *On-Off* terus menerus.

Kelemahan dari kontroler *On-Off* ini adalah jika *output* beresilasi di sekitar *set point* (keadaan yang memang diinginkan) akan menyebabkan aktuatur bekerja keras untuk *On-Off* dengan frekuensi yang tinggi. Hal ini akan menyebabkan kontroler akan cepat aus dan memakan energi yang banyak (boros).

Untuk sedikit mengatasi hal ini maka dibuat suatu *band* pada *set point* sehingga mengurangi frekuensi *On-Off* dari kontroler. Ilustrasinya diperlihatkan pada gambar 2.5.



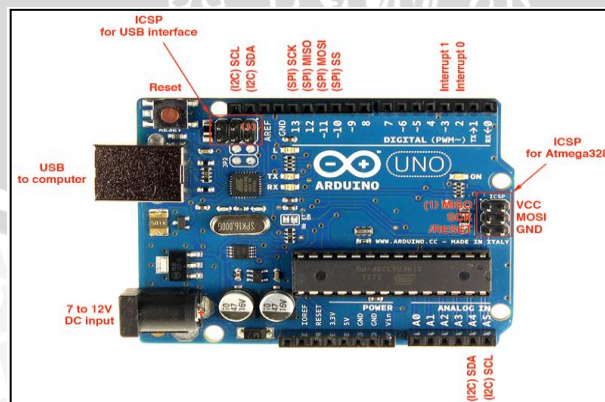
Gambar 2.5. Ilustrasi Band Pada *Setpoint* Kontroler *On-Off*

Sumber : www.geocities.com/al_dodi/kerja/kp3.pdf

Sinyal kontrol akan *off* ketika *output* menyentuh batas atas dan baru *on* kembali ketika menyentuh batas bawah. *Band* dari *set point* ini disebut juga *diferensial gap*. Dengan keadaan seperti ini serta mengatur besarnya *diferensial gap* maka frekuensi *On-Off* dapat dikurangi tetapi harus dibayar dengan penurunan akurasi terhadap *set point*.

2.4 Mikrokontroler Arduino Uno

Arduino UNO adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin digital input/output (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya sebesar 5 volt dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.



Gambar 2.6. Skema Arduino Uno

Sumber : electroschematics.com

2.5 Teori Dasar Sensor

Sensor adalah jenis transduser yang digunakan untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi tegangan dan arus listrik. Sensor merupakan salah satu komponen yang sangat penting dalam mendukung terjadinya kontrol proses yang berfungsi sebagai berikut

1. Menyediakan masukan dari proses dan dari lingkungan eksternal
2. Mengubah informasi fisik misalnya suhu, tekanan, laju aliran dan posisi untuk sinyal listrik.
3. Terkait dengan variabel fisik pada cara yang diketahui sehingga sinyal listriknya dapat digunakan untuk memonitor dan mengontrol proses.

Sensor sering digunakan untuk pendeteksian pada saat melakukan pengukuran atau pengendalian. Beberapa jenis sensor yang banyak digunakan dalam rangkaian elektronik antara lain sensor cahaya, sensor suhu, dan sensor tekanan.

2.6 Sensor Potensiometer Linier

Prinsip kerja potensiometer linier sama dengan potensiometer pada umumnya, yang menghasilkan perubahan resistansi yang terjadi antara tiga kakinya apabila terdapat pergeseran pada poros potensiometer tersebut. Potensiometer linier memiliki tingkat presisi dan perubahan resistansi yang linier antara tiga kakinya bila terjadi perderajat pergeseran pada poros potensiometer linier tersebut.

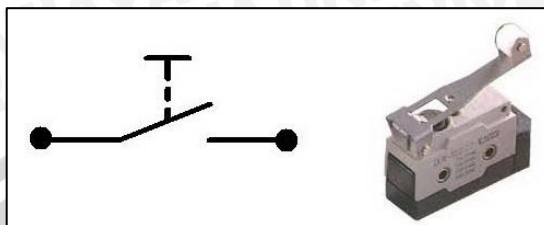


Gambar 2.7. Potensiometer Linier 10K Ω

2.7 Sensor Limit Switch

Limit switch merupakan jenis saklar yang dilengkapi dengan katup yang berfungsi menggantikan tombol. Prinsip kerja *limit switch* sama seperti saklar *Push ON* yaitu hanya akan terhubung pada saat katupnya ditekan pada batas penekanan tertentu

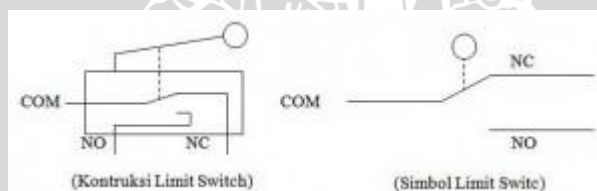
yang telah ditentukan dan akan memutuskan saat saat katup tidak ditekan. *Limit switch* termasuk dalam kategori sensor mekanis yaitu sensor yang akan memberikan perubahan elektrik saat terjadi perubahan mekanik pada sensor tersebut. Penerapan dari *limit switch* adalah sebagai sensor posisi suatu benda (objek) yang bergerak. Simbol *limit switch* ditunjukkan pada Gambar 2.8.



Gambar 2.8. Simbol dan Bentuk *Limit Switch*

Sumber : elektronika-dasar.web.id

Limit switch umumnya digunakan untuk memutuskan dan menghubungkan rangkaian menggunakan objek atau benda lain. Menghidupkan daya yang besar, dengan sarana yang kecil. Sebagai sensor posisi atau kondisi suatu objek. Prinsip kerja *limit switch* diaktifkan dengan penekanan pada tombolnya pada batas/daerah yang telah ditentukan sebelumnya sehingga terjadi pemutusan atau penghubungan rangkaian dari rangkaian tersebut. *Limit switch* memiliki 2 kontak yaitu NO (*Normally Open*) dan kontak NC (*Normally Close*) di mana salah satu kontak akan aktif jika tombolnya tertekan. Konstruksi dan simbol limit switch dapat dilihat seperti Gambar 2.9.



Gambar 2.9. Konstruksi dan Simbol *Limit Switch*

Sumber : elektronika-dasar.web.id

2.8 *Push Button*

Saklar merupakan komponen elektronika yang berfungsi untuk menghubungkan dan memutuskan dua titik atau lebih dalam suatu rangkaian elektronika. Salah satu jenis saklar adalah *push button* yaitu saklar yang hanya akan menghubungkan dua titik atau lebih pada saat tombolnya ditekan dan pada saat tombolnya tidak ditekan maka akan memutuskan dua titik atau lebih dalam suatu rangkaian elektronika. Simbol saklar *push button* ditunjukkan pada Gambar 2.10.



Gambar 2.10. Simbol dan macam bentuk Push Button
Sumber : elektronika-dasar.web.id



BAB III

METODE PENELITIAN

Kajian dalam skripsi ini merupakan penelitian yang bersifat aplikatif, yaitu pengaturan posisi motor servo pada sistem *rotary parking* yang bertujuan agar dapat menampilkan unjuk kerja sesuai dengan yang direncanakan. Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat adalah sebagai berikut:

3.1 Perancangan Sistem

Perencanaan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem pengidentifikasi dapat berjalan sesuai dengan yang telah direncanakan.

Perancangan sistem yang dilakukan meliputi:

1. Penentuan motor servo 180° sebagai penggerak *rotary parking*.
2. Penentuan spesifikasi sistem yang akan dibuat, meliputi :
 - a. Penentuan kerja sistem secara keseluruhan.
 - b. Kontrol *On-Off* dan rangkaian elektronik.
3. Perancangan program pada Arduino 1.0.5 sebagai pengontrol pada *rotary parking*.

3.2 Spesifikasi Alat

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

1. Motor servo 180° sebagai penggerak *rotary parking*.
2. Sensor *limit switch* dan *potensiometer* sebagai pendeteksi ada tidaknya mobil serta posisi garasi.
3. Modul Arduino Uno sebagai tempat proses data sistem.
4. *Push button* sebagai pemilih garasi yang dikehendaki.

3.3 Perancangan dan Realisasi Pembuatan Alat

Untuk merealisasikan alat atau sistem yang telah dirancang sebelumnya maka perlu diperhatikan beberapa tahapan dalam pembuatan alat tersebut, diantaranya yaitu :

- Penyempurnaan mekanik model sistem *rotary parking* secara keseluruhan yang meliputi penentuan batasan sudut putar motor servo serta pemasangan sensor dan motor.
- Pembuatan rangkaian elektronika meliputi rangkaian potensiometer dan penguat tegangan.
- Pengisian program yang telah dirancang sebelumnya pada Arduino 1.0.5 dengan kendali kontrol *On-Off*.

3.4 Pengujian dan Analisa Data

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

- Pengujian pada sensor yang digunakan.
- Menguji pada tiap-tiap blok rangkaian.
- Mengadakan pengujian sistem secara keseluruhan.
- Mengevaluasi hasil pengujian sistem secara keseluruhan.

3.5 Pengambilan Kesimpulan

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

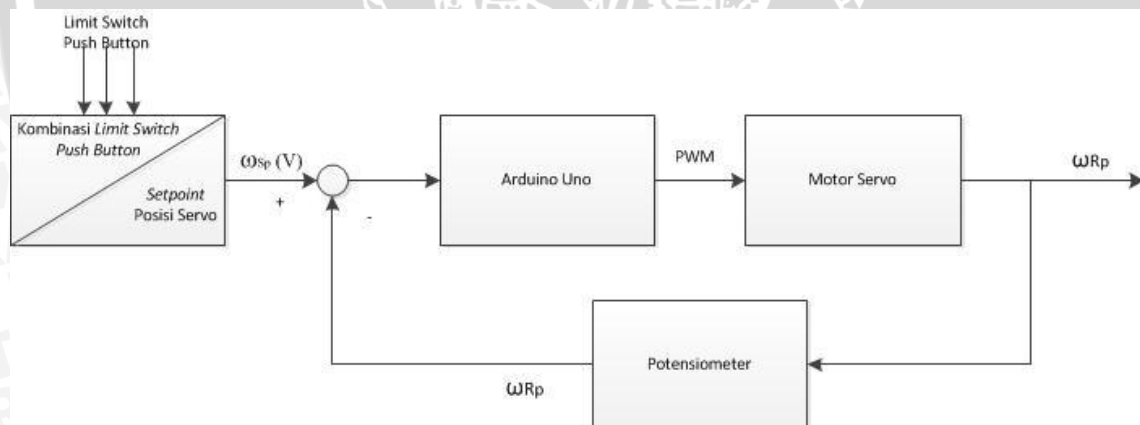
Perancangan alat ini dilakukan secara bertahap dalam bentuk blok sehingga akan memudahkan dalam analisis pada setiap bloknya maupun secara keseluruhan.

Perancangan ini terdiri dari:

- Cara kerja alat.
- Spesifikasi Alat.
- Perancangan perangkat keras (perancangan sensor potensiometer, perancangan sensor *limit switch*, perancangan *push button*, modul Arduino Uno).
- Perancangan perangkat lunak (perancangan algoritma kontrol *On-Off* dengan bahasa pemrograman Arduino 1.0.5 pada Arduino Uno)

4.1.1 Diagram Blok Sistem

Diagram blok sistem yang dirancang dtunjukkan pada Gambar 4.1



Gambar 4.1. Diagram Blok Sistem

Keterangan dari blok diagram diatas adalah sebagai berikut :

- *Set point* sistem berasal dari *push button*, dan *limit switch*. *Set point* dari *push button* digunakan ketika proses pengambilan mobil, *set point* dari *limit switch* digunakan saat proses mengarahkan garasi kosong ke bagian dasar.
- Pusat pengendalian sistem menggunakan Arduino Uno yang memberikan keluaran berupa *duty cycle* PWM kepada motor servo.
- Motor yang digunakan adalah motor servo yang berfungsi sebagai aktuator.

4.1.2 Prinsip Kerja Alat

Cara kerja alat adalah sebagai berikut :

- Menggunakan catu daya sebesar 5 volt.
- Potensiometer sebagai sensor posisi motor servo. Potensiometer terpasang sejajar dengan motor servo dan memberikan keluaran berupa level tegangan yang berubah-ubah sesuai dengan posisi motor servo saat itu.
- 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 *setpoint* pada saat proses memposisikan garasi kosong ke bawah. *Push button* berfungsi sebagai penentu *setpoint* 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 Arduino akan memproses *limit switch* mana atau *push button* yang aktif, kemudian menentukan *setpoint* dari sistem. Arduino akan membandingkan *setpoint* tersebut dengan besarnya tegangan yang diperoleh dari potensiometer linier, kemudian diproses dan diolah untuk menghasilkan data yang diinginkan. Selanjutnya Arduino akan memerintahkan motor servo untuk menggerakkan motor ke kanan atau ke kiri dengan sudut yang sesuai dengan kondisi dan masukan dari *limit switch* ataupun *push button* dan potensiometer.
- Proses tersebut berjalan terus sampai didapatkan posisi sudut putar motor servo yang diinginkan.

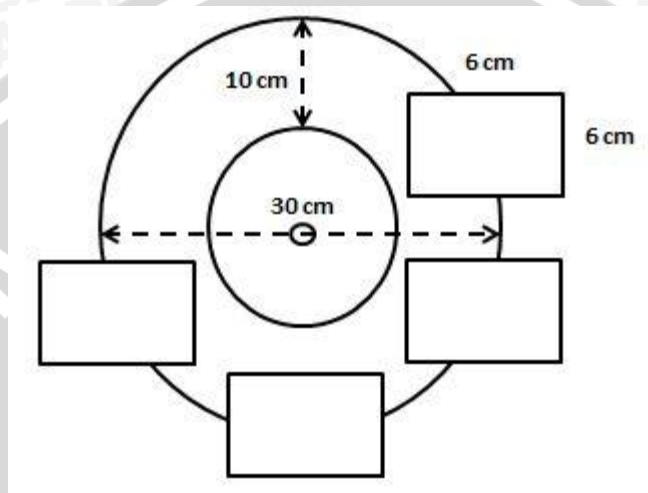
4.2 Perancangan Perangkat Keras

Perancangan perangkat keras meliputi perancangan mekanik, sensor potensiometer dan penggunaan modul Arduino Uno. Di bawah ini adalah penjelasan masing-masing rangkaian penyusun keseluruhan alat.

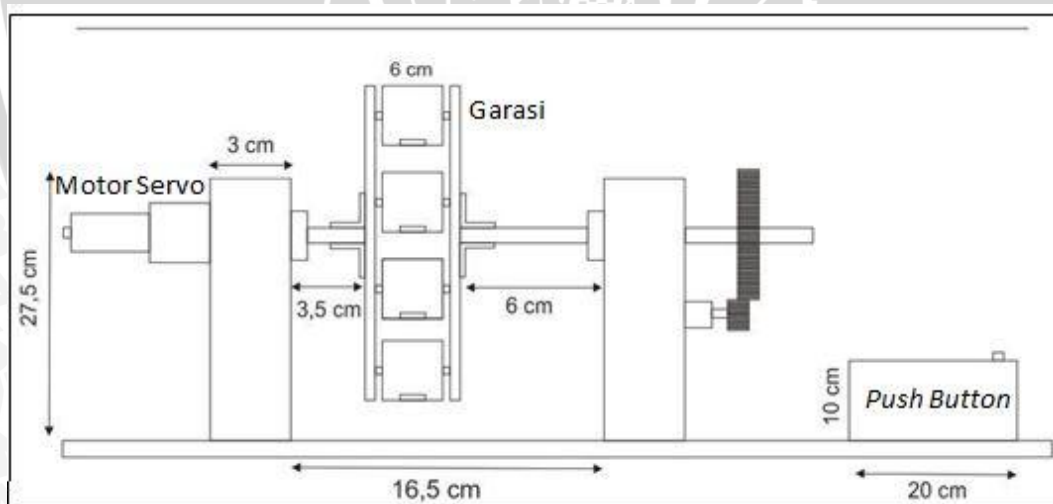
4.2.1 Spesifikasi Alat

Model sistem *rotary parking* yang dibuat terdiri dari 4 buah garasi yang dapat menampung 4 buah mobil.

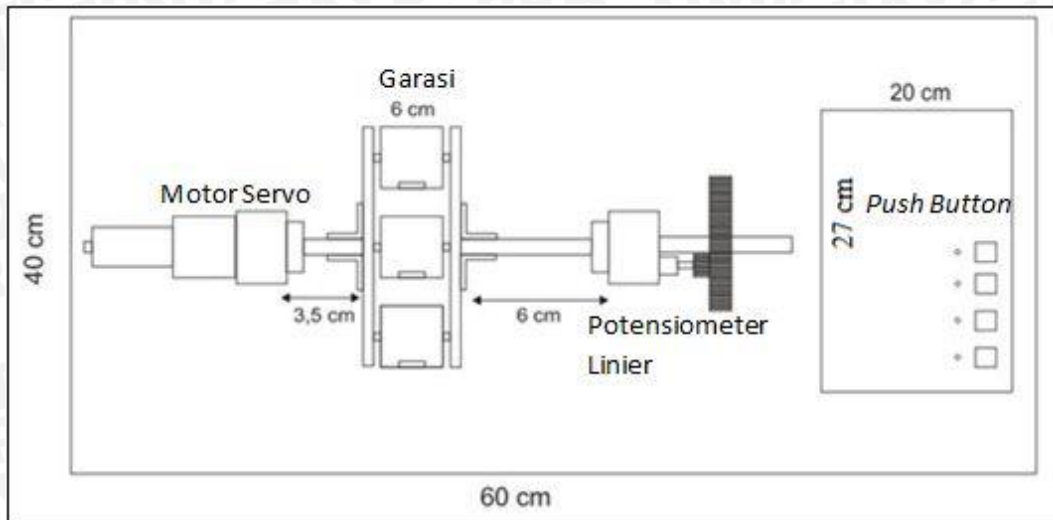
- Pergerakan model *rotary parking* menggunakan motor servo dan menggunakan pipa aluminium sebagai poros penghubung antara model *rotary parking* dan motor servo.
- Potensiometer yang digunakan adalah potensiometer linier 10 K Ω yang dipasang sejajar dengan motor servo.
- Ukuran dari model *rotary parking* dapat dilihat dalam Gambar 4.2, 4.3, dan 4.4.



Gambar 4.2. Skema Model *Rotary Parking* Tampak Samping



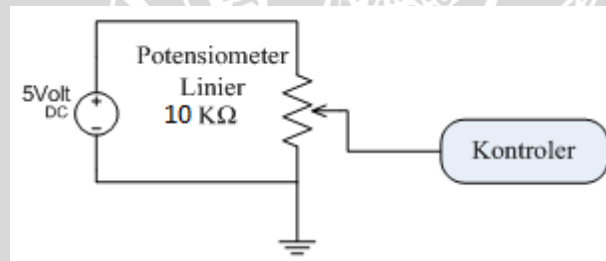
Gambar 4.3. Skema Model *Rotary Parking* Tampak Depan



Gambar 4.4. Skema Model Rotary Parking Tampak Atas

4.2.2 Potensiometer Linier

Potensiometer linier digunakan sebagai pengukur posisi sudut putaran motor DC. Penempatan sensor ini tepat sejajar dengan motor DC, jadi setiap perubahan sudut putar dari motor DC akan ikut mempengaruhi perubahan resistansi pada kaki potensiometer linier. Sistem perancangan dan hasil perancangan potensiometer linier ditunjukkan pada Gambar 4.5 dan 4.6.



Gambar 4.5. Rangkaian Potensiometer



Gambar 4.6. Potensiometer pada Sistem

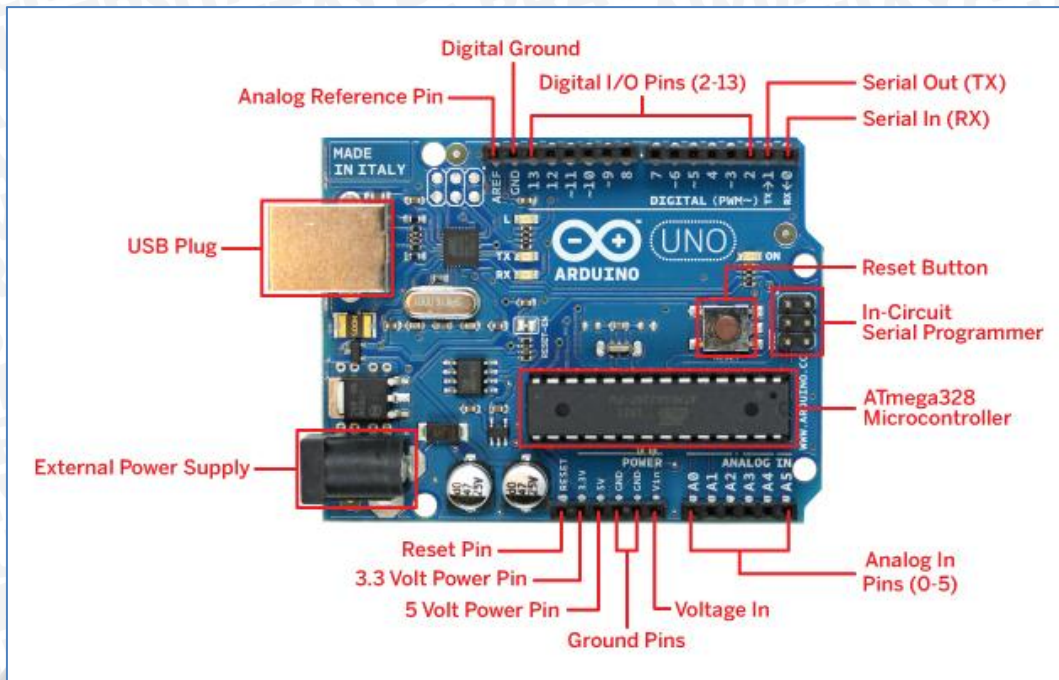
4.2.3 Modul Arduino Uno

Uno Arduino adalah *board* berbasis mikrokontroler pada ATmega32. *Board* ini memiliki 14 digital *input* / *output* pin (di mana 6 pin dapat digunakan sebagai *output PWM*), 6 *input analog*, 16 MHz *osilator* kristal, koneksi USB, *power jack* dan tombol reset.

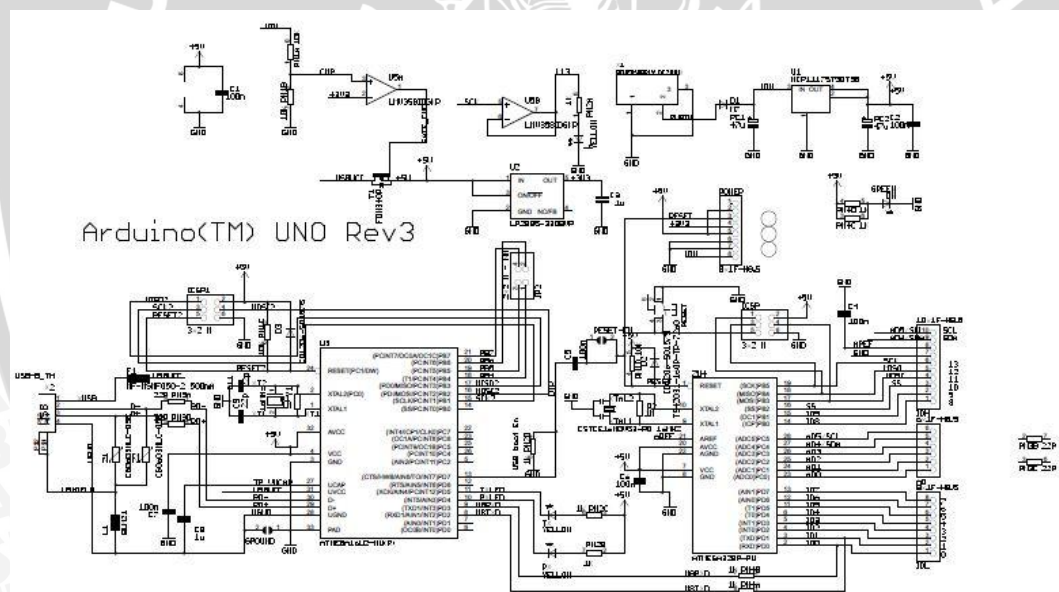
Pin masukan dan keluaran Arduino Uno pada perancangan ini akan disesuaikan sesuai Tabel 4.1.

4.1 Fungsi Pin Arduino Uno

No	Pin	Fungsi
1	2	Jalur masukan <i>push button</i> 1
2	3	Jalur masukan <i>push button</i> 2
3	4	Jalur masukan <i>push button</i> 3
4	5	Jalur masukan <i>push button</i> 4
5	6	Jalur masukan <i>limit switch</i> garasi 1
6	7	Jalur masukan <i>limit switch</i> garasi 2
7	8	Jalur masukan <i>limit switch</i> garasi 3
8	9	Jalur masukan <i>limit switch</i> garasi 4
9	10	Jalur masukan kontrol motor servo
10	11	Jalur masukan <i>potensiometer</i>
11	5V	Catu daya
12	GND	Ground



Gambar 4.7. Modul Arduino Uno

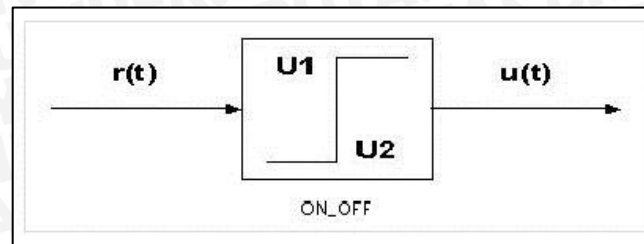


Gambar 4.8. Desain Sistem Arduino Uno

4.3 Perancangan Kontroler *On-Off*

Pada sistem kontrol *On-Off*, elemen pembangkit hanya memiliki dua posisi tertentu yaitu *On* dan *Off*. Kontroler *On-Off* memiliki karakteristik sinyal keluaran dari kontroler $u(t)$ tetap pada salah satu nilai maksimum atau minimum tergantung pada sinyal pembangkit kesalahan positif atau negatif.

Diagram blok controller *On-Off* yang memiliki masukan $e(t)$ dan keluaran $u(t)$, ditunjukkan pada



Gambar 4.9. Diagram Blok Controller *On-Off*

$$u(t) = \begin{cases} U1, & e(t) > 0 \\ U2, & e(t) < 0 \end{cases}$$

Gambar 4.10. Persamaan Controller *On-Off*

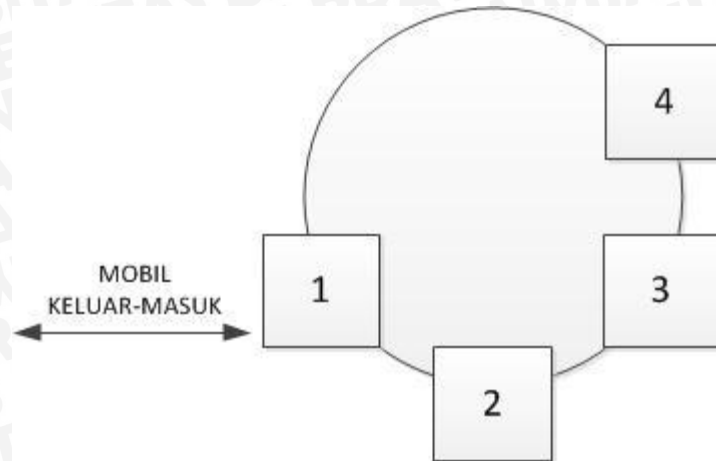
Persamaan pada Gambar 7 memiliki nilai $U1$ dan $U2$ yang konstan. Nilai minimum $U2$ dapat sebesar nol atau $-U1$. Pada sistem kontrol tertutup, sinyal $e(t)$ merupakan sinyal *error* sebesar selisih antara sinyal input dengan sinyal umpan balik.

4.4 Perancangan Pergerakan *Rotary Parking*

Pergerakan model sistem *rotary parking* ditentukan dari posisi garasi 1, 2, 3 dan 4 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 putar motor servo sebagai posisi garasi yang akan dituju.

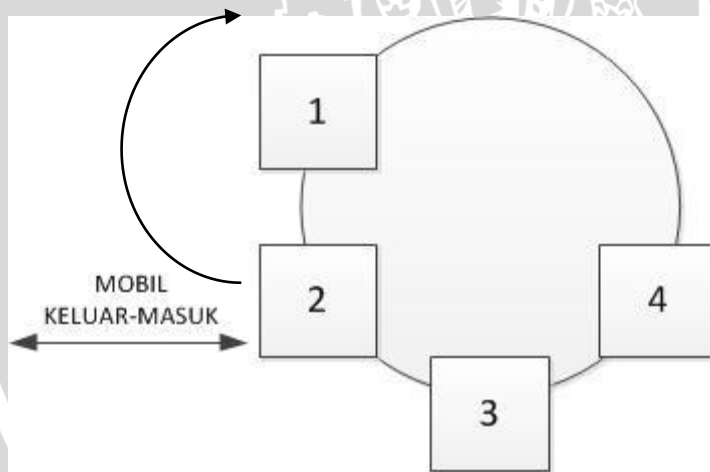
Ketika ingin mengarahkan garasi 1 ke bagian dasar maka *set point* sistem 0° atau dalam nilai ADC 535, jika ingin mengarahkan garasi 2 ke bagian dasar maka *set point* sistem 60° atau dalam nilai ADC 484, jika ingin mengarahkan garasi 3 ke bagian dasar maka *set point* sistem 120° atau dalam nilai ADC 433, jika ingin mengarahkan garasi 4 ke bagian dasar maka *set point* sistem 180° atau dalam nilai ADC 384.

Pergerakan model *rotary parking* akan ditunjukkan dalam Gambar 4.11, Gambar 4.12, dan Gambar 4.13.



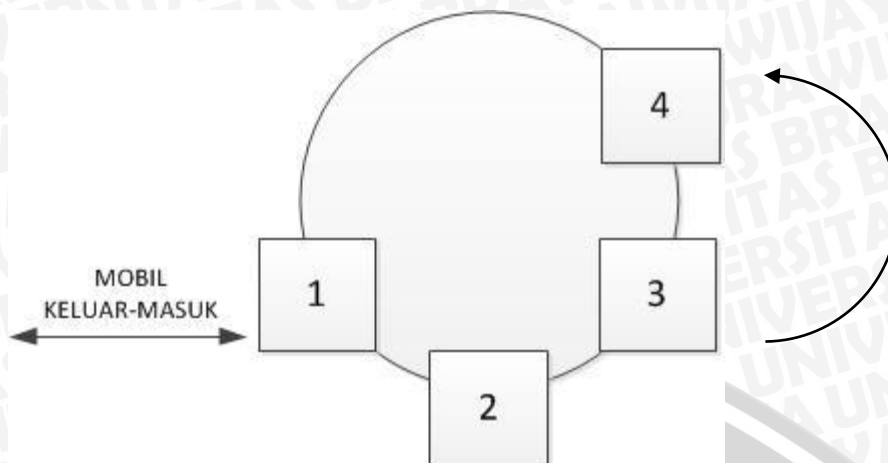
Gambar 4.11. Pergerakan Ketika *Push Button* dan *Limit Switch* yang Dipilih Bernomor 1 Saat Garasi 1 Sebagai Acuan

Tidak adanya arah panah pada Gambar 4.11 menunjukkan ketika garasi di dasar adalah garasi 1 dan *push button* yang dipilih bernomor 1 maka *rotary parking* tidak bergerak, karena nilai *present value* sama dengan nilai *set point*.



Gambar 4.12. Pergerakan Ketika *Push button* dan *Limit switch* yang Dipilih Bernomor 2 Saat Garasi 1 Sebagai Acuan

Arah panah pada Gambar 4.12 menunjukkan bahwa ketika garasi yang di dasar adalah garasi 1 dan akan mengarahkan garasi 2 ke dasar maka *rotary parking* akan bergerak *counter clock wise*, karena nilai *present value* lebih kecil daripada nilai *set point*.

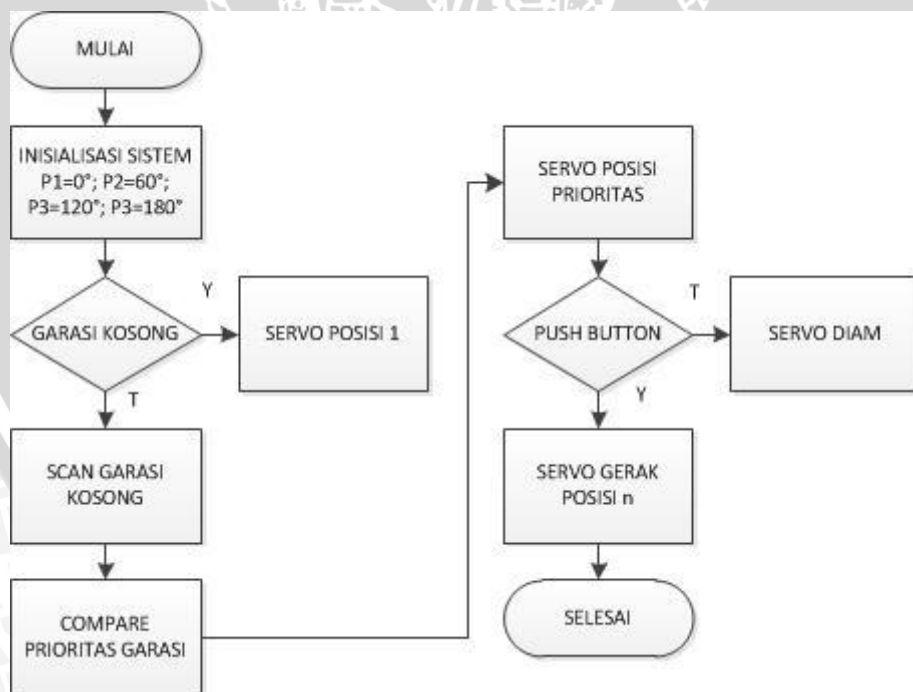


Gambar 4.13. Pergerakan Ketika *Push button* dan *Limit switch* yang Dipilih Bernomor 1 Saat Garasi 2 Sebagai Acuan

Arah panah pada Gambar 4.13 menunjukkan bahwa ketika garasi yang di dasar adalah garasi 2 dan akan mengarahkan garasi 1 ke dasar maka *rotary parking* akan bergerak *clock wise*.

4.5 Perancangan Perangkat Lunak

Flowchart perancangan perangkat lunak ditunjukkan pada Gambar 4.14



Gambar 4.14. Flowchart Perancangan Perangkat Lunak

BAB V

PENGUJIAN ALAT

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 Linier

a. Tujuan

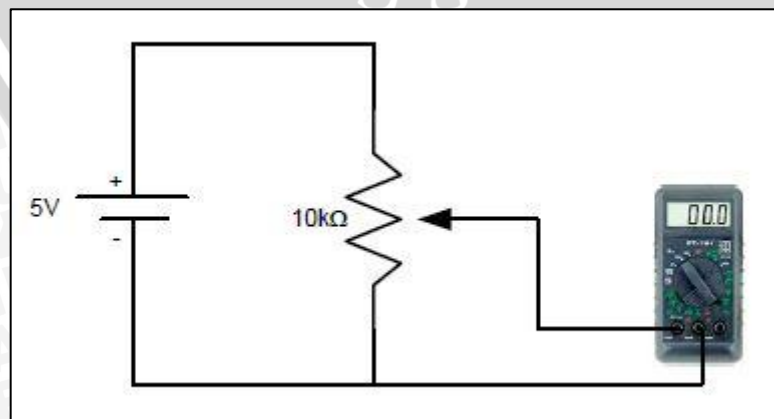
Mengetahui tingkat kelinieran dari potensiometer dalam membaca perubahan sudut putar motor servo.

b. Peralatan yang digunakan

- Potensiometer
- Multimeter
- Penggaris busur
- Catu daya 5V

c. Langkah pengujian

- Merangkai peralatan seperti Gambar 5.1.
- Potensiometer diputar dari 0° sampai 180°
- Mencatat dan menggambar data tegangan keluaran tiap 15° dari potensiometer untuk melihat kelinierannya.



Gambar 5.1. Pengujian Potensiometer

d. Hasil pengujian

Tegangan keluaran dari potensiometer ditulis dalam persamaan

$$V_{\text{output}} = \frac{R_{\text{potensiometer}}}{R_{\text{total}}} \times V_{\text{input}} \quad (5.1)$$

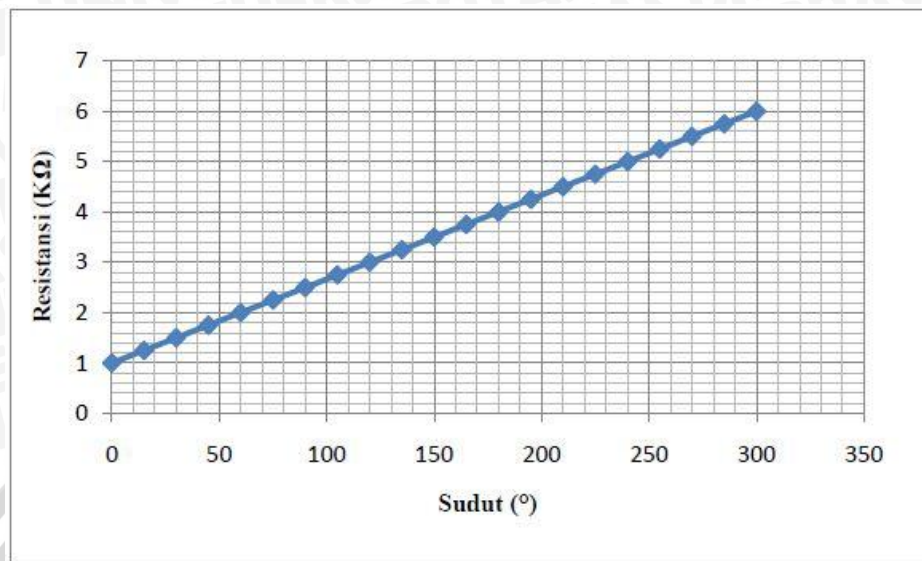
$$\text{Error} = \frac{\text{tegangan pengukuran} - \text{tegangan perhitungan}}{\text{tegangan pengukuran}} \times 100\% \quad (5.2)$$

Data hasil pengujian potensiometer ditunjukkan dalam tabel 5.1

Tabel 5.1 Pengujian Potensiometer

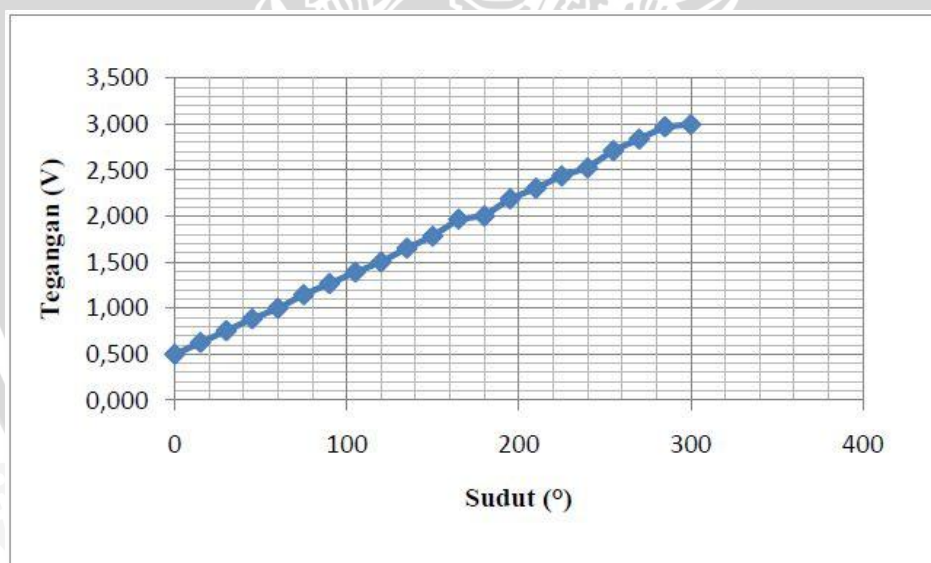
No	Sudut (°)	Resistansi (K Ω)	Tegangan Pengukuran (V)	Tegangan Perhitungan (V)	Error (%)
1	0	1	0,498	0,50000	0,39900
2	15	1,25	0,626	0,62499	0,16100
3	30	1,5	0,754	0,74999	0,57874
4	45	1,75	0,883	0,87499	0,91529
5	60	2	0,999	0,99999	0,13230
6	75	2,25	1,145	1,12499	1,77880
7	90	2,5	1,267	1,24999	1,38765
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	2,757	2,74997	3,15256
20	285	5,75	2,837	2,87497	3,34016
21	300	5,75	2,971	2,99997	0,11010

Grafik hubungan antara sudut dengan resistansi digambarkan pada Gambar 5.2.



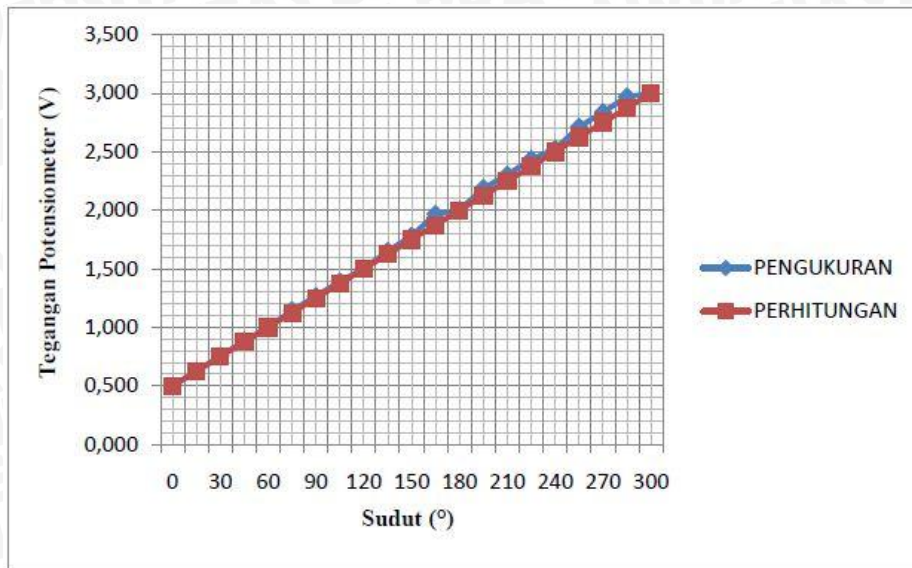
Gambar 5.2. Grafik Hubungan Antara Sudut Dengan Resistansi

Hubungan antara sudut dengan tegangan potensiometer digambarkan pada Gambar 5.3.



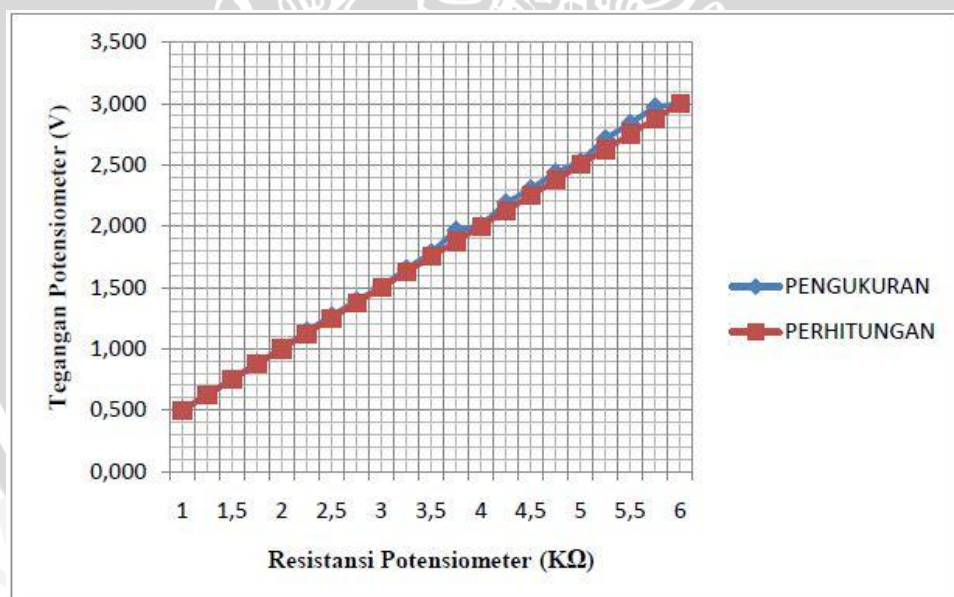
Gambar 5.3. Grafik Hubungan Antara Sudut Dengan Tegangan Potensiometer

Grafik perbandingan antara tegangan perhitungan dan tegangan pengukuran pada potensiometer digambarkan pada Gambar 5.4.



Gambar 5.4. Grafik Perbandingan Antara Tegangan Perhitungan dan Tegangan Pengukuran Pada Potensiometer

Grafik perbandingan antara tegangan perhitungan dan tegangan pengukuran terhadap resistansi potensiometer digambarkan pada Gambar 5.5



Gambar 5.5. Grafik Perbandingan Antara Tegangan Perhitungan dan Tegangan Pengukuran Terhadap Resistansi Potensiometer

Dari hasil pengujian yang dilakukan pada potensiometer dapat dilihat bahwa potensiometer yang digunakan mempunyai kelinieran yang baik karena hasil

pengukuran perubahan tegangan terhadap perubahan resistansi dapat dibuat persamaan sebagai berikut:

$$V = 0,49999 \times 10^3 R + 1.10^3 \quad (5.3)$$

Dimana :

V = Tegangan Keluaran Potensiometer

R = Resistansi Potensiometer

Sehingga ideal untuk digunakan sebagai pendeteksi posisi sudut putar motor servo dalam menggerakkan model *rotary parking* sampai pada posisi sudut putar yang sesuai dengan *set point*.

5.2 Pengujian Motor Servo

a. Tujuan

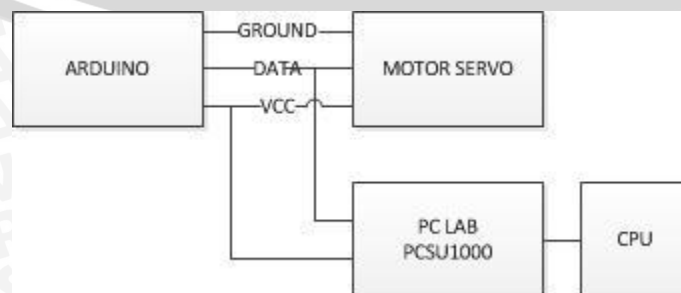
Mengetahui tingkat linearitas sudut pada motor servo *Towerpro MG995* 180% terhadap PWM.

b. Peralatan yang digunakan

- Motor Servo *Towerpro MG995*
- Catu daya 5V
- Arduino Uno
- Busur derajat
- Osiloskop Digital

c. Langkah pengujian

- Meangkai peralatan sesuai Gambar 5.6
- Menyalakan catu daya 5 V
- Beri masukan berupa sudut putar kepada motor servo melalui Arduino Uno
- Mencatat dan menggambar keluaran sinyal PWM dari motor servo melalui Arduino Uno untuk melihat kelinierannya



Gambar 5.6. Skema Pengujian Motor Servo

d. Hasil pengujian

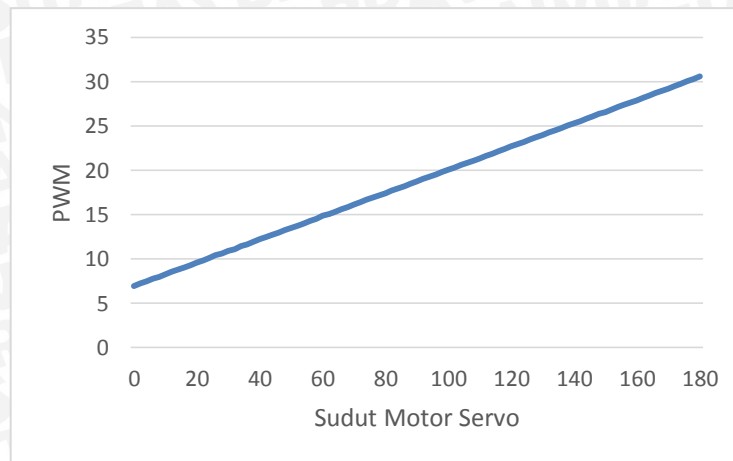
Data pengujian sinyal PWM dari motor servo ditunjukkan pada Tabel 5.2

Tabel 5.2 Pengujian Sinyal PWM dari Motor Servo

Sudut(⁰)	Duty Cycle (%)	PWM
0	2,72	6,936
10	3,24	8,262
20	3,76	9,588
30	4,28	10,914
40	4,79	12,2145
50	5,29	13,4895
60	5,83	14,8665
70	6,32	16,116
80	6,83	17,4165
90	7,35	18,7425
100	7,86	20,043
110	8,36	21,318
120	8,89	22,6695
130	9,4	23,97
140	9,92	25,296
150	10,4	26,52
160	10,94	27,897
170	11,46	29,223
180	12	30,6

Grafik hubungan PWM terhadap sudut motor servo digambarkan pada Gambar

5.7



Gambar 5.7 Grafik Hubungan PWM Terhadap Sudut Motor

Dari hasil percobaan, maka didapatkan motor servo dapat bekerja pada *Pulse Width Modulation* antara 6,936 – 30,6. Dari grafik hubungan PWM terhadap sudut motor servo diketahui bahwa perubahan PWM linier terhadap perubahan sudut motor.

5.3 Pengujian *Limit Switch* dan *Push Button*

a. Tujuan

Menguji *limit switch* dan *push button* yang digunakan pada sistem.

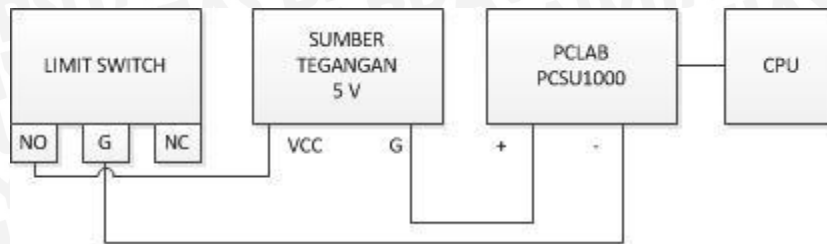
b. Peralatan yang digunakan

- *Limit switch* dan *push button*
- Catu daya 5V
- Arduino Uno
- Osiloskop Digital

c. Langkah pengujian

1) *Limit Switch*

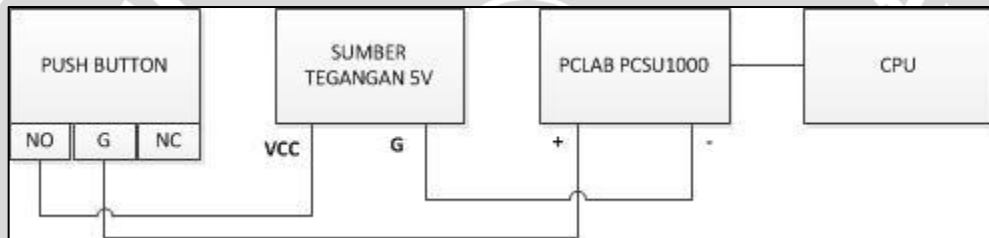
- Merangkai peralatan sesuai Gambar 5.8
- Meyalakan catu daya 5 V
- Mencatat keluaran yang tampil pada program PCLAB pada CPU.
- Ulangi langkah di atas dengan memberikan beban pada *limit switch*



Gambar 5.8 Skema Pengujian *Limit Switch*

2) *Push Button*

- Merangkai peralatan sesuai Gambar 5.9
- Meyalakan catu daya 5 V
- Mencatat keluaran yang tampil pada program PCLAB pada CPU.
- Ulangi langkah di atas dengan menekan *push button* satu kali

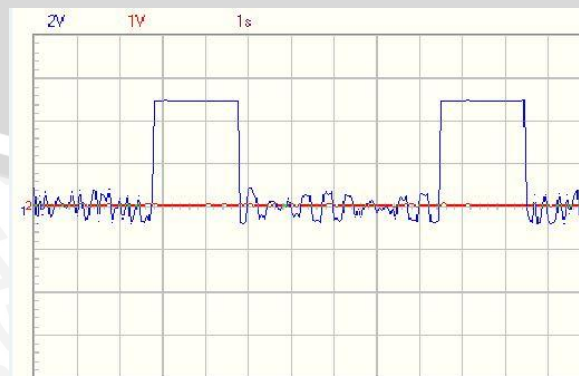


Gambar 5.9 Skema Pengujian *Push Button*

d. Hasil pengujian

1) *Limit Switch*

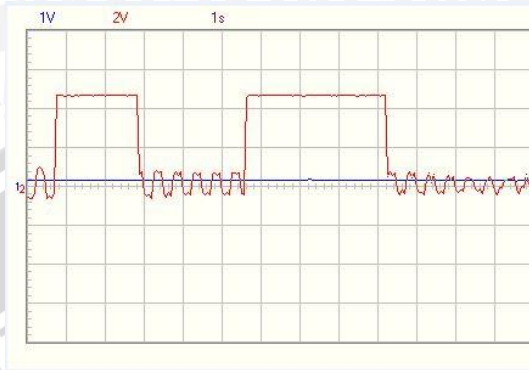
Pengujian *limit switch* menunjukkan bahwa *limit switch* dapat menghasilkan sinyal *high* saat kondisi diberi beban dan menghasilkan sinyal ambang saat kondisi tanpa beban, seperti ditunjukkan pada Gambar 5.10



Gambar 5.10 Grafik Respon *Limit Switch*

2) Push Button

Pengujian *push button* menunjukkan bahwa *push button* dapat menghasilkan sinyal *high* saat kondisi ditekan dan menghasilkan sinyal ambang saat kondisi tanpa tekanan, seperti ditunjukkan pada Gambar 5.11



Gambar 5.11 Grafik Respon Push Button

5.4 Pengujian Keseluruhan Sistem

a. Tujuan

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

b. Peralatan yang digunakan

- Rangkaian *rotary parking*
- Catu daya 5 V
- Modul Arduino
- Kabel serial
- Beban seberat 200 gram dan 400 gram.

c. Langkah pengujian

Pengujian alat ini dilakukan dengan cara menjalankan *rotary parking* dengan cara memasukkan nilai *set point* pada setiap garasi yang dituju. Nilai *set point* untuk masing-masing garasi berupa nilai ADC yang ditunjukkan dalam Tabel 5.3

Tabel 5.3 Set Point Rotary Parking

Garasi	ADC	Sudut ($^{\circ}$)
1	535	0
2	484	60
3	433	120
4	384	180

d. Hasil pengujian

1. Pengujian posisi parkir dengan beban 200 gram

Pengujian dilakukan dengan cara menjalankan *rotary parking* dengan memberi beban sebesar 200 gram kepada *limit switch* sehingga memberikan nilai untuk *set point* tiap tempat parkir.

Tabel 5.4 Data Pengujian Posisi *Rotary Parking* Keadaan Parkir Dengan Beban 200 Gram

No Parkir				Garasi Bawah
1	2	3	4	
0	0	0	0	1
1	0	0	0	2
0	1	0	0	1
0	0	1	0	1
0	0	0	1	1
1	1	0	0	3
1	1	1	0	4
1	0	1	0	2
1	0	0	1	2
0	1	1	0	1
0	1	0	1	1
0	0	1	1	1
1	1	0	1	3
1	0	1	1	2
1	1	1	1	1
0	1	1	1	1

2. Pengujian posisi parkir dengan beban 400 gram

Pengujian dilakukan dengan cara menjalankan *rotary parking* dengan memberi beban sebesar 400 gram kepada *limit switch* sehingga memberikan nilai untuk *set point* tiap tempat parkir.

Tabel 5.5 Data Pengujian Posisi *Rotary Parking* Keadaan Parkir Dengan Beban 400 Gram

No Parkir				Garasi
1	2	3	4	Bawah
0	0	0	0	1
1	0	0	0	2
0	1	0	0	1
0	0	1	0	1
0	0	0	1	1
1	1	0	0	3
1	1	1	0	4
1	0	1	0	2
1	0	0	1	2
0	1	1	0	1
0	1	0	1	1
0	0	1	1	1
1	1	0	1	3
1	0	1	1	2
1	1	1	1	1
0	1	1	1	1

3. Pengujian posisi ambil parkir dengan beban 200 gram

Pengujian dilakukan dengan cara menekan *push button* untuk memanggil garasi yang diinginkan.

Tabel 5.6 Data Pengujian Posisi *Rotary Parking* Keadaan Ambil Parkir Dengan Beban 200 Gram

<i>Push Button</i>				Posisi Garasi
1	2	3	4	
1	0	0	0	1
0	1	0	0	2
0	0	1	0	3
0	0	0	1	4

4. Pengujian posisi ambil parkir dengan beban 400 gram

Pengujian dilakukan dengan cara menekan *push button* untuk memanggil garasi yang diinginkan.

Tabel 5.7 Data Pengujian Posisi Rotary Parking Keadaan Ambil Parkir Dengan Beban 400 Gram

Push Button				Posisi Garasi
1	2	3	4	
1	0	0	0	1
0	1	0	0	2
0	0	1	0	3
0	0	0	1	4

5. Perhitungan *Error System* Pada Beban 200 Gram

Perhitungan dilakukan dengan membandingkan sudut *setpoint* motor servo yang dikontrol dengan Arduino Uno dengan sudut baca pada ADC potensiometer pada kondisi beban 200 gram untuk mencari nilai *error* sistem. Secara teori, *error* dapat dihitung dengan rumus :

$$Error = \left| \frac{\text{sudut potensiometer} - \text{setpoint}}{\text{setpoint}} \right| \times 100\%$$

Besar *error* pada pengaturan motor servo *rotary parking* ditunjukkan pada Tabel 5.8, tabel 5.9 tabel 5.10 dan tabel 5.11.

Tabel 5.8 Data Perbandingan antara *Setpoint* Motor Servo dan Sudut Baca Potensiometer Pada *Setpoint* 0°

Perobaan	<i>Setpoint</i> (derajat)	Sudut Potensiometer (derajat)	<i>Error</i> (persen)
1	0°	0°	0%
2	0°	2°	0,556%
3	0°	2°	0,556%
4	0°	1°	0,278%

Pada *setpoint* 0° diartikan sama dengan *setpoint* 360° agar dapat masuk ke dalam perhitungan *error*. Maka didapatkan rata-rata *error* pada sudut 0° sebesar 0,348 %.

Tabel 5.9 Data Perbandingan antara *Setpoint* Motor Servo dan Sudut Baca Potensiometer Pada *Setpoint* 60⁰

Percobaan	<i>Setpoint</i> (derajat)	Sudut Potensiometer (derajat)	<i>Error</i> (persen)
1	60 ⁰	58 ⁰	3,333%
2	60 ⁰	58 ⁰	3,333%
3	60 ⁰	57 ⁰	5%
4	60 ⁰	59 ⁰	1,667%

Maka didapatkan rata-rata *error* pada sudut 60⁰ sebesar 3,33 %.

Tabel 5.10 Data Perbandingan antara *Setpoint* Motor Servo dan Sudut Baca Potensiometer Pada *Setpoint* 120⁰

Percobaan	<i>Setpoint</i> (derajat)	Sudut Potensiometer (derajat)	<i>Error</i> (persen)
1	120 ⁰	115 ⁰	4,167%
2	120 ⁰	115 ⁰	4,167%
3	120 ⁰	116 ⁰	3,333%
4	120 ⁰	116 ⁰	3,333%

Maka didapatkan rata-rata *error* pada sudut 120⁰ sebesar 3,75 %.

Tabel 5.11 Data Perbandingan antara *Setpoint* Motor Servo dan Sudut Baca Potensiometer Pada *Setpoint* 180⁰

Percobaan	<i>Setpoint</i> (derajat)	Sudut Potensiometer (derajat)	<i>Error</i> (persen)
1	180 ⁰	171 ⁰	5%
2	180 ⁰	174 ⁰	3,333%
3	180 ⁰	172 ⁰	4,444%
4	180 ⁰	175 ⁰	2,778%

Maka didapatkan rata-rata *error* pada sudut 180⁰ sebesar 3,889%.

6. Perhitungan *Error System* Pada Beban 400 Gram

Perhitungan dilakukan dengan membandingkan sudut *setpoint* motor servo yang dikontrol dengan Arduino Uno dengan sudut baca pada ADC potensiometer pada kondisi beban 400 gram untuk mencari nilai *error* sistem. Secara teori, *error* dapat dihitung dengan rumus :

$$Error = \left| \frac{\text{sudut potensiometer} - \text{setpoint}}{\text{setpoint}} \right| \times 100\%$$

Besar *error* pada pengaturan motor servo *rotary parking* ditunjukkan pada Tabel 5.12, tabel 5.13 tabel 5.14 dan tabel 5.15.

Tabel 5.12 Data Perbandingan antara *Setpoint* Motor Servo dan Sudut Baca Potensiometer Pada

Setpoint 0°

Perobaan	<i>Setpoint</i> (derajat)	Sudut Potensiometer (derajat)	<i>Error</i> (persen)
1	0°	3°	0,833%
2	0°	2°	0,556%
3	0°	1°	0,278%
4	0°	3°	0,833%

Pada *setpoint* 0° diartikan sama dengan *setpoint* 360° agar dapat masuk ke dalam perhitungan *error*. Maka didapatkan rata-rata *error* pada sudut 0° sebesar 0,625 %.

Tabel 5.13 Data Perbandingan antara *Setpoint* Motor Servo dan Sudut Baca Potensiometer Pada

Setpoint 60°

Percobaan	<i>Setpoint</i> (derajat)	Sudut Potensiometer (derajat)	<i>Error</i> (persen)
1	60°	55°	8,333%
2	60°	55°	8,333%
3	60°	58°	3,333%
4	60°	56°	6,667%

Maka didapatkan rata-rata *error* pada sudut 60° sebesar 6,665 %.

Tabel 5.14 Data Perbandingan antara *Setpoint* Motor Servo dan Sudut Baca Potensiometer Pada *Setpoint* 120⁰

Percobaan	<i>Setpoint</i> (derajat)	Sudut Potensiometer (derajat)	<i>Error</i> (persen)
1	120 ⁰	113 ⁰	5,883%
2	120 ⁰	110 ⁰	8,333%
3	120 ⁰	113 ⁰	5,883%
4	120 ⁰	105 ⁰	12,5%

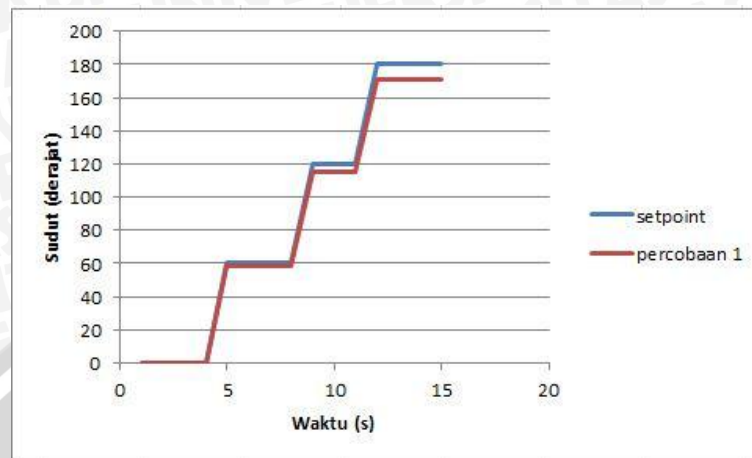
Maka didapatkan rata-rata *error* pada sudut 120⁰ sebesar 8,15 %.

Tabel 5.15 Data Perbandingan antara *Setpoint* Motor Servo dan Sudut Baca Potensiometer Pada *Setpoint* 180⁰

Percobaan	<i>Setpoint</i> (derajat)	Sudut Potensiometer (derajat)	<i>Error</i> (persen)
1	180 ⁰	165 ⁰	8,333%
2	180 ⁰	163 ⁰	9,444%
3	180 ⁰	167 ⁰	7,222%
4	180 ⁰	167 ⁰	7,222%

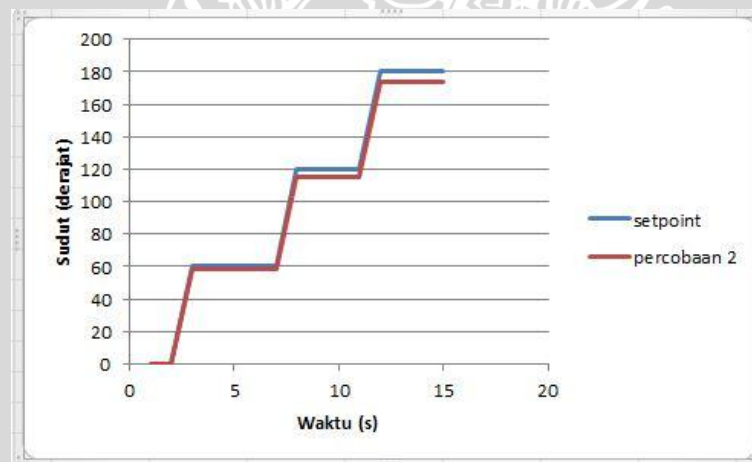
Maka didapatkan rata-rata *error* pada sudut 180⁰ sebesar 8,055%. Setelah dilakukan beberapa pengujian maka dapat ditampilkan grafik hubungan antara sudut *setpoint* dan pembacaan sudut potensiometer linier.

Grafik perbandingan antara sudut *setpoint* 0° dan pembacaan sudut potensiometer dengan beban 200 gram ditunjukkan pada gambar 5.12.



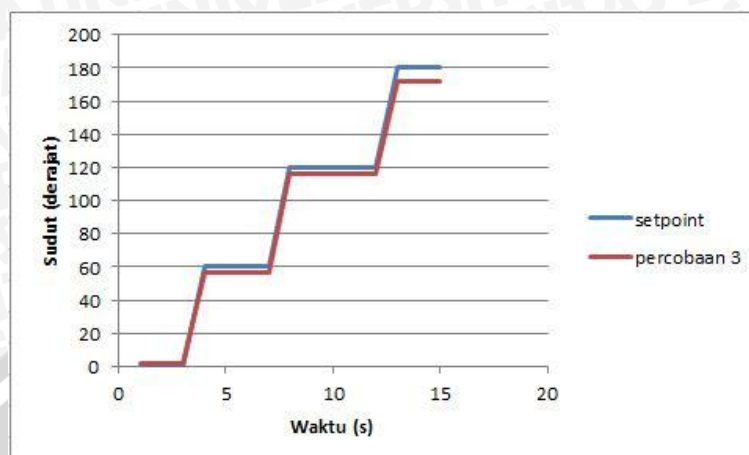
Gambar 5.12 Grafik Perbandingan Sudut *Setpoint* 0° dan Pembacaan Potensiometer Linier

Grafik perbandingan antara sudut *setpoint* 60° dan pembacaan sudut potensiometer dengan beban 200 gram ditunjukkan pada gambar 5.13.



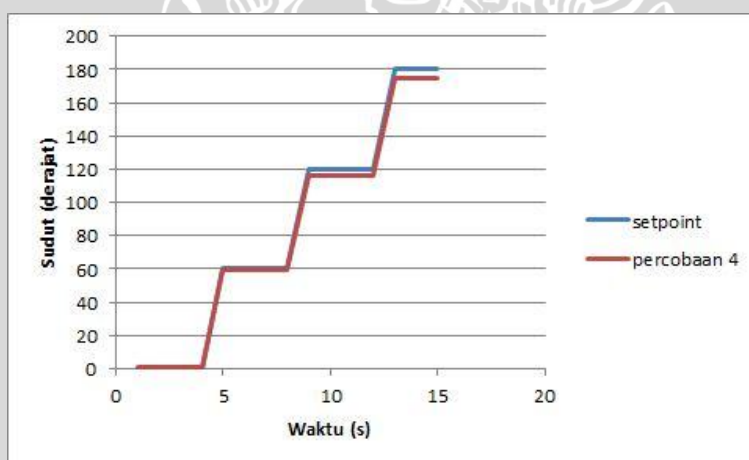
Gambar 5.13 Grafik Perbandingan Sudut *Setpoint* 60° dan Pembacaan Potensiometer Linier

Grafik perbandingan antara sudut *setpoint* 120° dan pembacaan sudut potensiometer dengan beban 200 gram ditunjukkan pada gambar 5.14.



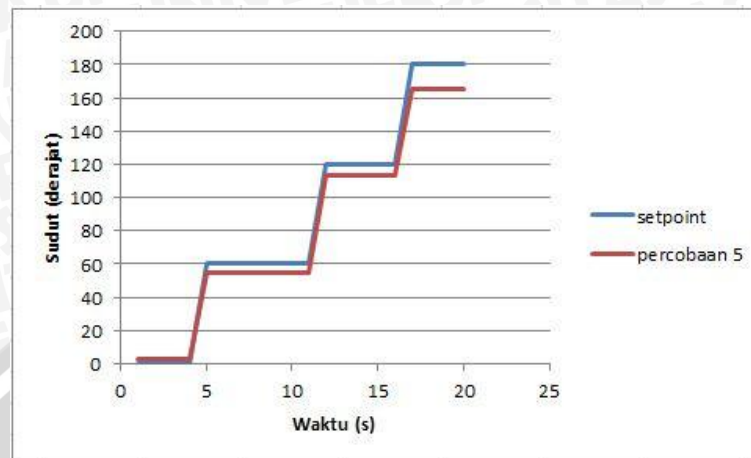
Gambar 5.14 Grafik Perbandingan Sudut *Setpoint* 120° dan Pembacaan Potensiometer Linier

Grafik perbandingan antara sudut *setpoint* 180° dan pembacaan sudut potensiometer dengan beban 200 gram ditunjukkan pada gambar 5.15.



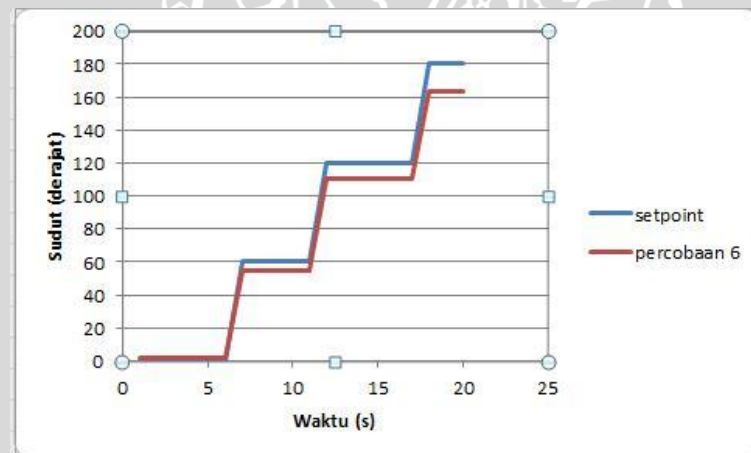
Gambar 5.15 Grafik Perbandingan Sudut *Setpoint* 180° dan embacaan Potensiometer Linier

Grafik perbandingan antara sudut *setpoint* 0° dan pembacaan sudut potensiometer dengan beban 400 gram ditunjukkan pada gambar 5.16.



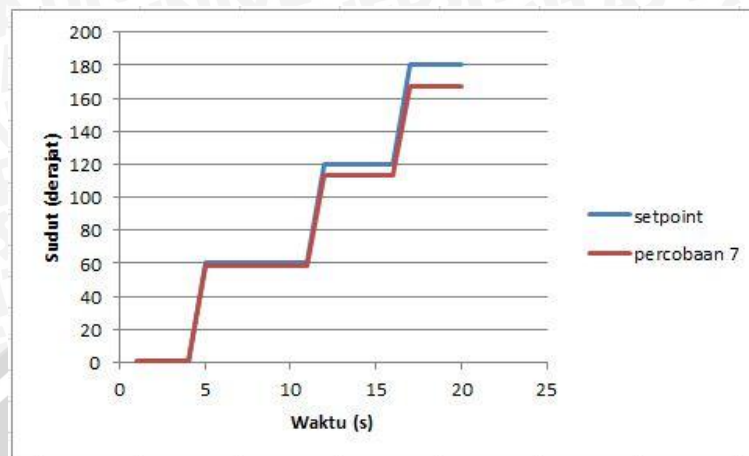
Gambar 5.16 Grafik Perbandingan Sudut *Setpoint* 0° dan Pembacaan Potensiometer Linier

Grafik perbandingan antara sudut *setpoint* 60° dan pembacaan sudut potensiometer dengan beban 400 gram ditunjukkan pada gambar 5.17.



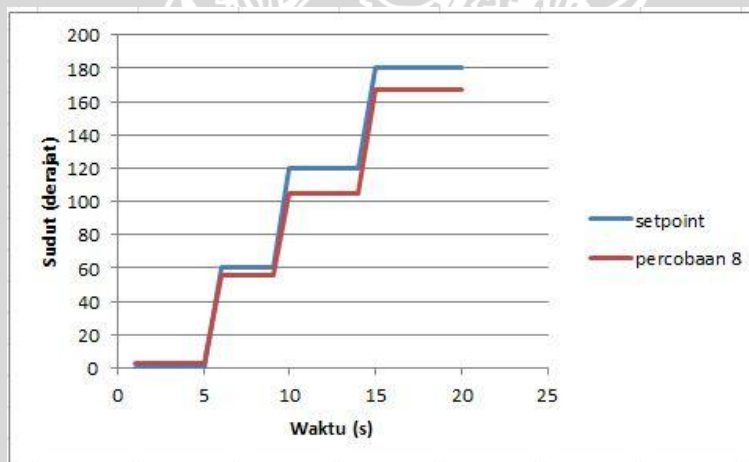
Gambar 5.17 Grafik Perbandingan Sudut *Setpoint* 60° dan Pembacaan Potensiometer Linier

Grafik perbandingan antara sudut *setpoint* 120° dan pembacaan sudut potensiometer dengan beban 400 gram ditunjukkan pada gambar 5.18.



Gambar 5.18 Grafik Perbandingan Sudut *Setpoint* 120° dan Pembacaan Potensiometer Linier

Grafik perbandingan antara sudut *setpoint* 180° dan pembacaan sudut potensiometer dengan beban 400 gram ditunjukkan pada gambar 5.19.



Gambar 5.19 Grafik Perbandingan Sudut *Setpoint* 180° dan Pembacaan Potensiometer Linier

BAB VI PENUTUP

6.1 Kesimpulan

Kesimpulan yang dapat diambil dalam pembuatan penelitian ini adalah sebagai berikut :

1. Penggunaan Arduino Uno sebagai kontroller *On-Off* dapat digunakan sebagai pengatur sudut putaran motor servo dan garasi dalam sistem *rotary parking* dengan pembagian posisi pada sudut 0° , 60° , 120° , 180° dengan beban 200 gram dan 400 gram.
2. Hasil pengujian terhadap aplikasi kontroller *On-off* ini menunjukkan bahwa nilai *error* rata-rata pada percobaan dengan beban 200 gram pada sudut 0° sebesar 0,348%, pada sudut 60° sebesar 3,33%, pada sudut 120° sebesar 3,75% dan pada sudut 180° sebesar 3,889%. Pada percobaan menggunakan beban 400 gram *error* rata-rata pada sudut 0° sebesar 0,635%, pada sudut 60° sebesar 6,665%, pada sudut 120° sebesar 8,15% dan pada sudut 180° sebesar 8,055⁰. Hasil pengujian ini menunjukkan bahwa kontroler *On-Off* menghasilkan respon sesuai yang direncanakan dan dapat diaplikasikan pada *rotary parking* untuk mencapai posisi yang dituju.

6.2 Saran

Alat ini masih bisa dirancang ulang dan dikembangkan kembali, meliputi :

1. Memperbanyak jumlah tempat parkir dengan penggunaan motor servo *continues* 360 derajat dan pembagian sudut putar yang lebih kecil.
2. Merancang pengaturan perlambatan pergerakan sudut motor servo saat akan mencapai sudut yang dituju.
3. Memperbaiki desain mekanik untuk mengurangi besar *error*.

DAFTAR PUSTAKA

- [1] *Datasheet Arduino Uno.*
- [2] Ilmawan P., Aditya. 2013. *SISTEM PENGATURAN POSISI SUDUT PUTAR MOTOR DC PADA MODEL ROTARY PARKING MENGGUNAKAN KONTROLER PID BERBASIS ARDUINO MEGA 2560.* Malang. Fakultas Teknik Universitas Brawijaya.
- [3] Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid 1.* Jakarta. Penerbit Erlangga.
- [4] Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik Jilid 2.* Jakarta. Penerbit Erlangga



LAMPIRAN 1

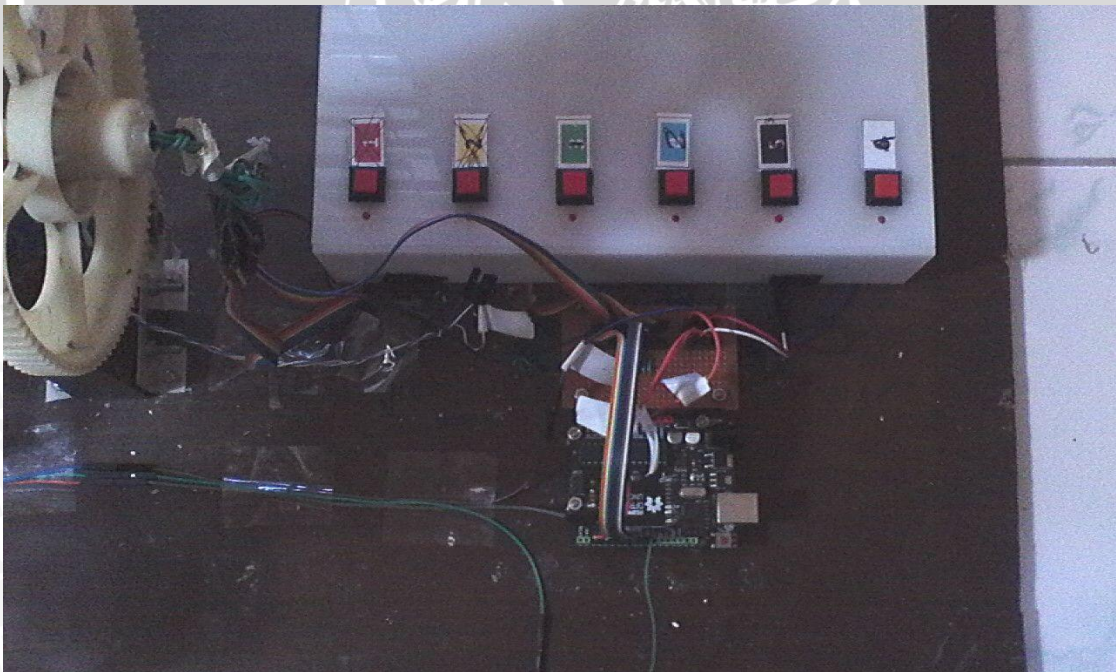
FOTO ALAT



TAMPAK DEPAN



TAMPAK ATAS



TAMPAK SAMPING



LAMPIRAN 2
LISTING PROGRAM




```
//=====SERVO=====
```

```
#include <Servo.h>
```

```
Servo myservo;
```

```
int potensio = A0;
```

```
int posisi;
```

```
int tujuan;
```

```
int buttonState1 = 0;
```

```
int buttonState2 = 0;
```

```
int buttonState3 = 0;
```

```
int buttonState4 = 0;
```

```
int limitState1 = 0;
```

```
int limitState2 = 0;
```

```
int limitState3 = 0;
```

```
int limitState4 = 0;
```

```
int pos = 0;
```

```
const int limit1Pin = 4; // limit switch 1
```

```
const int limit2Pin = 3; // limit switch 2
```

```
const int limit3Pin = 2; // limit switch 3
```

```
const int limit4Pin = 5; // limit switch 4
```

```
const int button1Pin = 9; //push button 1
```

```
const int button2Pin = 8; //push button 2
```

```
const int button3Pin = 7; //push button 3
```

```
const int button4Pin = 6; //push button 4
```




```
void setup()
{
  Serial.begin(9600);
  pinMode(limit1Pin, INPUT);
  pinMode(limit2Pin, INPUT);
  pinMode(limit3Pin, INPUT);
  pinMode(limit4Pin, INPUT);
  pinMode(limit4Pin, INPUT);
  pinMode(button1Pin, INPUT);
  pinMode(button2Pin, INPUT);
  pinMode(button3Pin, INPUT);
  pinMode(button4Pin, INPUT);
  pinMode(potensio, INPUT);
  myservo.attach(10);
}

void loop()
{
  bacaButton();

  while(buttonState1 == LOW && buttonState2 == LOW && buttonState3 ==
  LOW && buttonState4 == LOW)
  {
    switchLimit();
  }

  switchButton();
}
```

```
//===== Fungsi Membaca State  
Limit Switch=====
```

```
void bacaLimit()  
{  
    limitState1=digitalRead(limit1Pin);  
    limitState2=digitalRead(limit2Pin);  
    limitState3=digitalRead(limit3Pin);  
    limitState4=digitalRead(limit4Pin);  
    delay(50);  
}
```

```
//===== Fungsi Membaca State  
Push Button=====
```

```
void bacaButton()  
{  
    buttonState1 = digitalRead(button1Pin);  
    buttonState2 = digitalRead(button2Pin);  
    buttonState3 = digitalRead(button3Pin);  
    buttonState4 = digitalRead(button4Pin);  
    delay(50);  
}
```

```
//===== Fungsi Parkir  
Mobil=====
```

```
void switchLimit()  
{  
    bacaLimit();  
    bacaButton();  
    if (limitState1 == LOW)  
    {
```

```
for(pos = 0; pos<1; pos+=1)
{
myservo.write(pos);
delay(25);
}
Serial.println(pos);
pos=pos;
}
```

```
else if (limitState2 == LOW)
```

```
{
for(pos; pos<=60; pos+=1)
{
myservo.write(pos);
delay(25);
pos=pos;
}
Serial.println(pos);
pos=pos;
}
```

```
else if (limitState3 == LOW)
```

```
{
for(pos; pos<=130; pos+=1)
{
myservo.write(pos);
delay(25);
```



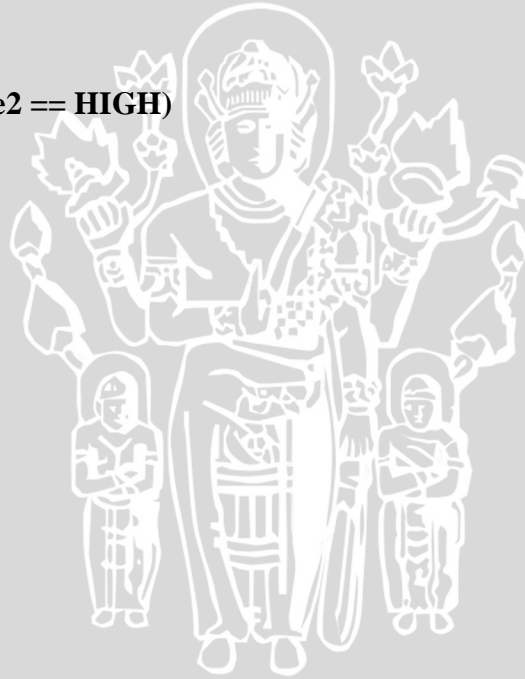

```
}  
pos=pos;  
}  
  
else if (limitState4 == LOW)  
{  
  for(pos; pos<=180; pos+=1)  
  {  
    myservo.write(pos);  
    delay(25);  
  }  
  Serial.println (pos);  
  pos=pos;  
}  
  
else if(limitState4 == HIGH)  
{  
  for(pos; pos>0; pos-=1)  
  {  
    myservo.write(pos);  
    delay(25);  
  }  
  Serial.println(pos);  
  pos=pos;  
}  
}
```



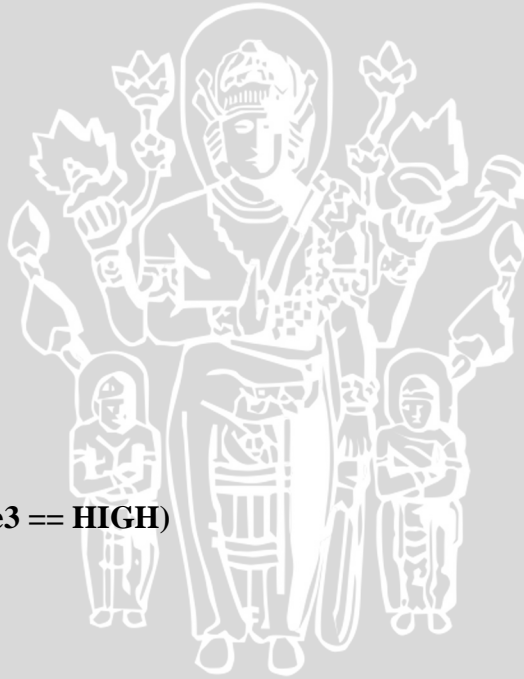
```
//=====Fungsi Ambil  
Mobil=====
```

```
void switchButton()  
{  
  bacaButton();  
  if (buttonState1 == HIGH)  
  {  
    tujuan=1;  
    if (pos>tujuan)  
    {  
      gerakCW();  
      bacaLimit();  
      while (limitState1 == HIGH)  
      {  
        tujuan=1;  
        gerakCW();  
      }  
    }  
  }  
  else  
  {  
    gerakCCW();  
    bacaLimit();  
    while (limitState1 == HIGH)  
    {  
      tujuan=1;  
      gerakCCW();  
    }  
  }  
}
```

```
    }  
    pos=pos;  
    // Serial.println(pos);  
    }  
    else if (buttonState2 == HIGH)  
    {  
        tujuan=62;  
        if (pos>tujuan)  
        {  
            gerakCW();  
            bacaLimit();  
            while (limitState2 == HIGH)  
            {  
                tujuan=62;  
                gerakCW();  
            }  
        }  
        else  
        {  
            gerakCCW();  
            while (limitState2 == HIGH)  
            {  
                bacaLimit();  
                gerakCCW();  
            }  
        }  
        pos=pos;
```




```
// Serial.println(pos);
}
else if (buttonState3 == HIGH)
{
    tujuan=132;
    if (pos>tujuan)
    {
        gerakCW();
        while (limitState3 == HIGH)
        {
            tujuan=132;
            gerakCW();
        }
    }
    else
    {
        gerakCCW();
        while (limitState3 == HIGH)
        {
            tujuan=132;
            gerakCCW();
        }
    }
    pos=pos;
}
// Serial.println(pos);
}
else if (buttonState4 == HIGH)
```



```

{
    tujuan=180;
    if (pos>tujuan)
    {
        gerakCW();
        while (limitState4 == HIGH)
        {
            tujuan=180;
            gerakCW();
        }
    }
    else
    {
        gerakCCW();
        while (limitState4 == HIGH)
        {
            tujuan=180;
            gerakCCW();
        }
    }
    pos=pos;
    // Serial.println(pos);
}
}

```

//=====Fungsi Gerak Motor
 "Clock Wise"=====



```
void gerakCW()
```

```
{  
  for(pos;pos>=tujuan;pos-=1)  
  {  
    myservo.write(pos);  
    delay(25);  
  }  
}
```

```
//===== Fungsi Gerak Motor  
"Counter Clock Wise"=====
```

```
void gerakCCW()
```

```
{  
  for(pos;pos<=tujuan;pos+=1)  
  {  
    myservo.write(pos);  
    delay(25);  
  }  
}
```

```
void bacaPotensio
```

```
{  
  posisi= analogRead (potensio);  
  Serial.println(posisi);  
}
```



LAMPIRAN 3
DATASHEET

