

***TRAINER KIT* INDUSTRIAL MOTOR SERVO SEBAGAI  
MEDIA PEMBELAJARAN ROBOTIKA DAN MESIN  
OTOMATIS 2 (RMO 2)**

**PROYEK AKHIR**

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan  
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

Krisna Monica	(NIRM)	0031613
M Dedy Deva Sanjaya Hata Babelin	(NIRM)	0031617

**POLITEKNIK MANUFAKTUR NEGERI  
BANGKA BELITUNG  
TAHUN 2019**

**LEMBAR PENGESAHAN**

**TRAINER KIT INDUSTRIAL MOTOR SERVO SEBAGAI MEDIA  
PEMBELAJARAN ROBOTIKA DAN MESIN OTOMATIS 2 (RMO 2)**

Oleh :

Krisna Monica (NIRM) 0031613

M Dedy Deva Sanjaya Hata Babelin (NIRM) 0031617

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan  
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui

Pembimbing 1

Pembimbing 2



I Made Andik Setiawan, Ph.D

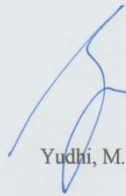


Ocsirendi, M.T.

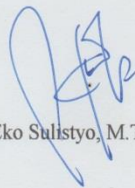
Penguji 1

Penguji 2

Penguji 3



Yudhi, M.T.



Eko Sulisty, M.T.



Nofriyani, M.T.

## PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini:

Nama Mahasiswa 1 : Krisna Monica NIRM 0031613

Nama Mahasiswa 2 : M Dedy Deva Sanjaya Hata Babelin NIRM 0031617

Dengan Judul : *Trainer Kit* Industrial Motor Servo Sebagai Media  
Pembelajaran Robotika dan Mesin Otomatis 2 (RMO 2)

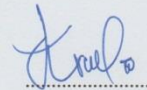
Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku

Sungailiat, 12 September 2019

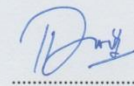
Nama Mahasiswa

Tanda Tangan

1. Krisna Monica



2. M Dedy Deva Sanjaya Hata Babelin



## ABSTRAK

*Materi pembelajaran di Laboratorium Robotika dan Mesin Otomatis 2 (RMO 2) mahasiswa Teknik Elektronika mempelajari cara mengontrol servo motor. Pada materi Robotika dan Mesin Otomatis 2 (RMO 2) ini mempelajari cara mengontrol posisi dan kecepatan motor menggunakan sensor posisi dan kecepatan. Namun di Politeknik Manufaktur Negeri Bangka Belitung media pembelajaran untuk mengontrol motor servo sudah tersedia, tetapi banyak mengalami kerusakan. Untuk itu perlu dibuat Trainer Kit Industrial Motor Servo sebagai Media Pembelajaran Robotika dan Mesin Otomatis 2 (RMO 2) beserta hardware praktikum yang disesuaikan dengan modul praktikum yang telah ada yang akan memandu dalam melakukan praktikum. Pada trainer kit ini terdapat 4 box modul yang memiliki fungsi yang berbeda. Box modul tersebut terdiri dari driver motor, tachogenerator dan motor dc, increment encoder, absolute encoder dan potensiometer. Box modul ini dilengkapi dengan arduino uno dan dapat diprogram menggunakan software arduino IDE serta LCD sebagai penampil nilai. Serta dilengkapi juga dengan PID controller yang berfungsi untuk mengontrol kecepatan motor. Dari hasil pengujian yang dilakukan maka diketahui bahwa rata-rata persentase error tiap modul berbeda seperti modul potensiometer memiliki error sebesar 7,4%, absolute dan increment encoder memiliki error 2%. Tachogenerator memiliki ketelitian 87%.*

*Kata kunci: servo, kecepatan, trainer kit, PID*

## **ABSTRACT**

*Learning material in the Robotics and Automatic Engineering Laboratory 2 (RMO 2) Electronic Engineering students learn how to control a servo motor. In Robotics and Automatic Machine 2 (RMO 2), this study learns how to control the position and speed of the motor using position and speed sensors. But in the Bangka Belitung State Manufacturing Polytechnic the learning media to control the servo motor is already available, but many have suffered damage. For this reason, it is necessary to make an Industrial Servo Motor Trainer Kit as Learning Media for Robotics and Automatic Machines 2 (RMO 2) along with practicum hardware that is adapted to the existing practicum modules that will guide in conducting the practice. In this trainer kit there are 4 module boxes that have different functions. The module box consists of motor driver, tachogenerator and dc motor, increment encoder, absolute encoder and potentiometer. This module box is equipped with Arduino Uno and can be programmed using Arduino IDE software and LCD as a value viewer. And also equipped with a PID controller that functions to control the speed of the motor. From the results of tests carried out it is known that the average percentage error of each different module such as the potentiometer module has an error of 7.4%, absolute and increment encoder has an error of 2%. Tachogenerator has 87% accuracy.*

*Key words: servo, industry, speed, trainer kit*

## KATA PENGHANTAR

Segala puji bagi Tuhan Yang Maha Esa yang telah memberikan karunia-Nya yang tiada terkira kepada kita semua dan khususnya kepada penulis. Hanya berkat pertolongan-Nya akhirnya penulis bisa menyelesaikan laporan proyek akhir ini. Laporan proyek akhir ini disusun dalam rangka memenuhi salah satu syarat kelulusan Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung.

Dalam penulisan laporan akhir ini tentunya tidak lepas dari kekurangan dan keterbatasan yang penulis miliki. Penulis menyadari bahwa laporan akhir ini jauh dari sempurna sehingga penulis membutuhkan kritik dan saran yang bersifat membangun untuk kemajuan pendidikan di masa yang akan datang. Dalam penyusunan dan penulisan laporan akhir ini tidak lepas juga dari bantuan, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis ingin menyampaikan terimakasih kepada yang terhormat:

1. Tuhan Yang Maha Esa atas segala rezeki, nikmat, rahmat dan hidayah yang telah diberikan kepada penulis.
2. Orang tua dan keluarga penulis yang selalu memberikan doa dan dukungan baik moril maupun materi serta semangat yang menghibur penulis dikala jenuh.
3. Bapak Sugeng Ariyono, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak I Made Andik S., Ph.D, selaku pembimbing I yang telah memberikan bimbingan, masukan, kritikan, saran serta waktunya selama pembuatan proyek akhir ini.
5. Bapak Ocsirendi, M.T., selaku pembimbing II yang telah memberikan bimbingan, masukan, kritikan, saran serta waktunya selama pembuatan proyek akhir ini.

6. Seluruh staf pengajar di Politeknik Manufaktur Negeri Bangka Belitung. Teman-teman dan rekan kerja penulis yang telah banyak membantu dalam proses penyelesaian laporan ini.
7. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan Proyek Akhir ini yang tidak dapat disebutkan satu per satu.
8. Serta teman-teman seperjuangan dalam mengerjakan Proyek Akhir 2019 yang telah membantu dalam memberikan semangat, masukan, saran dan yang membantu dalam pembuatan Proyek Akhir ini.

Semoga proyek akhir ini dapat berguna untuk menambah wawasan dan wacana bagi rekan-rekan mahasiswa/mahasiswi.

Sungailiat, 12 September 2019

Penulis

## DAFTAR ISI

HALAMAN SAMPUL .....	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN BUKAN PLAGIAT .....	ii
ABSTRAK .....	iii
<i>ABSTRACT</i> .....	v
KATA PENGHANTAR .....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR .....	xi
DAFTAR LAMPIRAN.....	xiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Proyek Akhir .....	3
BAB II DASAR TEORI .....	4
2.1 Motor Servo.....	4
2.2 <i>Pulse Width Modulation (PWM)</i> .....	5
2.3 <i>Rotary Encoder</i> .....	6
2.3.1 <i>Absolute Encoder</i> .....	6
2.3.2 <i>Increment encoder</i> .....	7
2.4 Potensiometer .....	8
2.5 Tachogenerator .....	9
2.6 PID.....	10
BAB III METODE PELAKSANAAN .....	12
3.1 Komponen yang digunakan.....	12



BAB IV PEMBAHASAN.....	14
4.1    Sensor Posisi Menggunakan Potensiometer .....	14
4.2    Sensor Posisi Menggunakan <i>Increment Encoder</i> .....	20
4.3    Sensor Posisi Menggunakan <i>Absolute Encoder</i> .....	27
4.4    Pengaturan Kecepatan Menggunakan PWM.....	31
4.5    Pengaturan Kecepatan Menggunakan Tachogenerator .....	38
4.6    Pengaturan Kecepatan Menggunakan Kontroller PID .....	43
BAB V KESIMPULAN DAN SARAN.....	49
5.1    Kesimpulan.....	49
5.2    Saran .....	50
DAFTAR PUSTAKA .....	51

## DAFTAR TABEL

Tabel 3.1 Komponen yang digunakan .....	13
Tabel 4.1 Hasil Pengukuran Potensiometer .....	16
Tabel 4.2 <i>Connector</i> Pin pada <i>Increment Encoder</i> ( KY-040) .....	20
Tabel 4.3 Hasil Percobaan pada <i>Channel A</i> dan <i>B</i> ( <i>clockwise</i> ).....	22
Tabel 4.4 Hasil Percobaan pada <i>Channel A</i> dan <i>B</i> ( <i>Counter Clockwise</i> ).....	22
Tabel 4.5 <i>Connector</i> pada <i>absolute encoder</i> ( Hohner SCS10).....	27
Tabel 4.6 Hasil Pengujian Arah Putaran Motor .....	34
Tabel 4.7 Hasil Pengukuran Tegangan <i>Output</i> dan Kecepatan Motor.....	35
Tabel 4.8 Hasil Pengukuran Tegangan Tachogenerator dan Motor DC.....	40

## DAFTAR GAMBAR

Gambar 2.1 Contoh <i>dutycycle</i> 25%, 50% dan 75% [10].....	5
Gambar 2.2 <i>Absolute Encoder</i> [11].....	6
Gambar 2.3 <i>Increment encoder</i> [11] .....	7
Gambar 2.4 Struktur Fisik Potensiometer [13] .....	8
Gambar 2.5 Rangkaian Potensiometer [13] .....	8
Gambar 2.6 Rangkaian Pembagi Tegangan .....	9
Gambar 2.7 Blok Diagram PID Kontroller .....	10
Gambar 4.1 (a) Desain <i>Layout</i> Tampak Atas Potensiometer (b) Desain <i>Layout</i> Tampak Bawah Potensiometer (c) Potensiometer pada <i>Trainer Kit</i> .....	14
Gambar 4.2 Koneksi Perkabelan Potensiometer .....	15
Gambar 4.3 Grafik Tegangan <i>Output</i> Arduino Terhadap Nilai ADC.....	17
Gambar 4.4 Grafik Tegangan <i>Output</i> Multitester Terhadap Nilai ADC .....	18
Gambar 4.5 Tampilan LCD Nilai ADC dan Tegangan.....	18
Gambar 4.6 Grafik Nilai Tahanan Terhadap Nilai ADC .....	19
Gambar 4.7 Hasil Pengukuran Nilai Tahanan Potensiometer.....	19
Gambar 4.8 (a) Desain <i>Layout</i> Tampak Atas <i>Increment Encoder</i> (b) Desain <i>Layout</i> Tampak Bawah (c) <i>Increment Encoder</i> pada <i>Trainer Kit</i> .....	21
Gambar 4.9 Koneksi Perkabelan <i>Increment Encoder</i> pada <i>Trainer Kit</i> .....	21
Gambar 4.10 Sinyal Keluaran <i>Increment Encoder</i> Searah Jarum Jam .....	23
Gambar 4.11 Sinyal Keluaran <i>Increment Encoder</i> Berlawanan Jarum Jam.....	23
Gambar 4.12 Hasil Pengukuran Tegangan <i>Increment</i> pada Logika Rendah .....	24
Gambar 4.13 Hasil Pengukuran Tegangan <i>Increment</i> Pada Logika Tinggi.....	24
Gambar 4.14 (a) <i>Output</i> LED A dan B padam (b) <i>Output</i> A dan B <i>high</i> .....	25
Gambar 4.15 (a) <i>Output</i> LED A dan B nyala (b) <i>Output</i> A dan B <i>low</i> .....	25
Gambar 4.16 (a) <i>Ouput</i> LED A nyala B padam (b) <i>Output</i> A <i>low</i> B <i>high</i> .....	25
Gambar 4.17 (a) <i>Output</i> LED A padam B nyala (b) <i>Output</i> A <i>high</i> B <i>low</i> .....	26
Gambar 4.18 (a) Desain <i>Layout</i> Tampak Atas <i>Absolute Encoder</i> (b) Desain <i>Layout</i> Tampak Bawah <i>Absolute Encoder</i>	

(c) <i>Absolute Encoder</i> pada <i>Trainer Kit</i> .....	28
Gambar 4.19 Koneksi Perkabelan <i>Absolute Encoder</i> pada <i>Trainer Kit</i> .....	29
Gambar 4.20 Hasil Pengukuran Tegangan <i>Absolute</i> Pada Logika Rendah .....	29
Gambar 4.21 Hasil Pengukuran Tegangan <i>Absolute</i> Pada Logika Tinggi.....	30
Gambar 4.22 Hasil Pengujian <i>Absolute Encoder</i> .....	30
Gambar 4.23 (a) Desain <i>Layout</i> Tampak Atas <i>Driver Motor</i> (b) <i>Driver Layout</i> Tampak Bawah <i>Driver Motor</i> (c) <i>Driver Motor</i> pada <i>Trainer Kit</i> .....	32
Gambar 4.24 Koneksi Perkabelan <i>Driver Motor</i> Pada <i>Trainer Kit</i> .....	33
Gambar 4.25 Blok Diagram Pengukuran <i>Driver Motor</i> L298N .....	33
Gambar 4.26 Sinyal PWM dengan <i>dutycycle</i> 30,2% .....	35
Gambar 4.27 Motor Servo Tanpa Gear.....	36
Gambar 4.28 Grafik hubungan PWM dengan tegangan <i>output</i> beban motor.....	36
Gambar 4.29 Grafik hubungan antara PWM dengan kecepatan.....	37
Gambar 4.30 Hasil Pengujian Kecepatan pada PWM 30 % 50% 100% .....	37
Gambar 4.31 (a) Desain <i>Layout</i> Tampak Atas Tachogenerator dan Motor DC (b) Desain <i>Layout</i> Tampak Atas Tachogenerator dan Motor DC (c) Tachogenerator dan Motor DC pada <i>Trainer Kit</i> .....	39
Gambar 4.32 Koneksi perkabelan tachogenerator dan motor DC pada <i>trainer kit</i> 40	
Gambar 4.33 Grafik hubungan antara tegangan <i>output</i> dengan <i>dutycycle</i> .....	42
Gambar 4.34 Grafik hubungan antara kecepatan dengan <i>dutycycle</i> .....	42
Gambar 4.35 Hasil Pengujian Kecepatan Motor dengan <i>dutycycle</i> 150, 200 dan 255 .....	43
Gambar 4.36 Grafik Respon $K_p = 1$ .....	44
Gambar 4.37 Grafik Respon $K_p = 2$ .....	45
Gambar 4.38 Grafik Respon $K_p = 1$ dan $K_i = 0,5$ .....	46
Gambar 4.39 Grafik Respon $K_p = 2$ dan $K_i = 1$ .....	46
Gambar 4.40 Grafik Respon $K_p=1$ , $K_i = 0,5$ dan $K_d = 0,5$ .....	47
Gambar 4.41 Grafik Respon $K_p=1$ , $K_i = 0,5$ dan $K_d = 0,5$ .....	47

## **DAFTAR LAMPIRAN**

LAMPIRAN I	Data Riwayat Hidup
LAMPIRAN 2	Modul Pratikum

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Industri manufaktur sekarang ini telah banyak menggunakan robot sebagai pengganti manusia dalam mengerjakan suatu pekerjaan yang bersifat terus-menerus yang membutuhkan konsentrasi, kecepatan serta akurasi yang tinggi. Manusia memiliki keterbatasan dalam efisiensi kerja, yaitu seperti tidak dapat berkerja secara terus-menerus tanpa istirahat dan saat kelelahan biasanya manusia melakukan kesalahan dalam bekerja. Maka dari itu, pabrik-pabrik besar menggunakan robot atau sistem robot sebagai pengganti tenaga manusia. [1] [2]

Menurut kamus Meriam-Webster, Robot adalah sebuah alat yang dapat melakukan tindakan seperti manusia yang dapat bergerak otomatis dan dapat dikontrol. Robot juga merupakan gabungan peralatan yang menghasilkan gerakan yang otomatis berdasarkan gerakan yang diperintah. Ada beberapa jenis robot pada saat ini seperti robot jenis humanoid, robot beroda, robot berkaki, robot yang memiliki kecerdasan manusia serta robot yang dapat dikendalikan secara manual maupun otomatis. [3] [4]

Robot dapat melakukan gerakan sesuai dengan kontrol yang diberikan. Disamping itu robot bergerak karena adanya aktuator atau penggerak. Aktuator adalah sebuah peralatan mekanis untuk menggerakkan atau mengontrol sebuah sistem kerja robot. Aktuator merupakan penghasil gerakan mekanik. [5]

Aktuator atau penggerak robot merupakan komponen penting pada robot. Aktuator yang biasa digunakan seperti motor DC, motor stepper dan motor servo. Motor DC biasanya digunakan pada robot line follower. Hal ini karena motor DC memiliki kecepatan tinggi dengan torsi besar. Contoh lain penggerak robot ialah motor stepper. Motor Stepper sering digunakan sebagai penggerak lengan, tuas penjepit pada robot yang tidak memerlukan kecepatan dan torsi besar namun diperlukan ketelitian. Motor servo banyak digunakan pada robot lengan, kaki serta

bagian lainnya yang memiliki gerakan terbatas. Robot lengan merupakan salah satu robot yang sering kita jumpai di industri. Pada robot lengan, sendi-sendinya digerakkan menggunakan motor servo. Motor servo banyak digunakan sebagai penggerak robot karena mudah untuk dikendalikan, memiliki ketelitian sudut dan sebagai penggerak yang memiliki kontrol posisi yang akurat. [6]

Di Politeknik Manufaktur Negeri Bangka Belitung salah satu materi pembelajaran yang diberikan kepada mahasiswa ialah Robotika dan Mesin Otomatis 2 (RMO 2). Pada praktikum mata kuliah ini, salah satu yang diajarkan adalah mengenai motor servo. Oleh karena itu, untuk menyelesaikan kompetensi pembelajaran Robotika dan Mesin Otomatis 2 (RMO 2) sangat diperlukan media pembelajaran yang berguna untuk mempermudah mahasiswa dalam memahami kompetensi baik secara teori maupun praktik.

Untuk mengatasi masalah tersebut, dibuatlah proyek akhir yang berjudul “Trainer Kit Industrial Motor Servo Sebagai Media Pembelajaran Robotika dan Mesin Otomatis 2 (RMO 2)” untuk membantu praktikum program studi teknik elektronika. Pada trainer kit ini dilengkapi dengan modul beberapa modul antara lain *increment encoder*, *absolute encoder*, potensiometer, dan tachogenerator. Serta penggunaan modul ini dilengkapi dengan memakai arduino uno dan diprogram menggunakan *software* arduino IDE serta LCD sebagai penampil nilai. Selain itu, kontrol PID adalah subsistem yang akan ditambahkan untuk mengontrol kecepatan motor servo. Dengan adanya peralatan ini diharapkan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung dapat lebih memahami pembelajaran Robotika dan Mesin Otomatis 2 (RMO 2).

## **1.2 Perumusan Masalah**

Berdasarkan uraian latar belakang diatas, adapun permasalahan yang ingin diselesaikan pada proyek akhir ini adalah:

1. Bagaimana merancang dan membuat *Trainer Kit Industrial Motor Servo* yang dapat memudahkan mahasiswa dalam pembelajaran Robotika dan Mesin Otomatis 2 (RMO 2)

2. Bagaimana membuat modul ajar praktikum dari *Trainer Kit Industrial Motor Servo* yang sesuai dengan kompetensi pembelajaran Robotika dan Mesin Otomatis 2 (RMO 2) di Politeknik Manufaktur Negeri Bangka Belitung

### **1.3 Batasan Masalah**

Berdasarkan perumusan masalah diatas, perlu adanya batasan-batasan permasalahan guna memperjelas ruang lingkup proyek akhir ini.

1. *Trainer kit* ini menggunakan LCD hanya sebagai penampil nilai.
2. PID hanya dapat mengontrol kecepatan menggunakan tegangan.

### **1.4 Tujuan Proyek Akhir**

Tujuan Proyek Akhir yang ingin dicapai dengan dibuatnya alat dengan judul *Trainer Kit Industrial Motor Servo* sebagai Media Pembelajaran Robotika dan Mesin Otomatis 2 (RMO 2) ini adalah sebagai berikut ini:

1. Membuat *Trainer Kit Industrial Motor Servo* yang dapat memudahkan mahasiswa dalam pembelajaran Robotika dan Mesin Otomatis 2 (RMO 2).
2. Membuat modul ajar praktikum dari *Trainer Kit Industrial Motor Servo* yang sesuai dengan kompetensi pembelajaran Robotika dan Mesin Otomatis 2 (RMO 2) di Politeknik Manufaktur Negeri Bangka Belitung.



## **BAB II**

### **DASAR TEORI**

Pada bab ini akan menjelaskan tentang teori-teori yang digunakan dalam menyelesaikan proyek akhir, antara lain sebagai berikut.

#### **2.1 Motor Servo**

Motor Servo merupakan motor yang digunakan sebagai sumber bergerak dalam sistem servo, dengan umpan balik (*feedback*) berupa posisi dan kecepatan untuk setiap pengontrolannya. Motor servo menerima masukan berupa pulsa. Kecepatan motor servo tergantung lebar pulsa yang dikirimkan [7].

Motor servo terdiri dari sebuah motor DC, serangkaian *gear*, potensiometer, dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Keunggulan dari penggunaan motor servo adalah sebagai berikut [8].

- Tidak bergetar saat beroperasi.
  - Daya yang dihasilkan sebanding dengan ukuran dan berat motor.
  - Penggunaan arus listrik sebanding dengan beban yang diberikan.
  - Resolusi dan akurasi dapat diubah dengan hanya mengganti encoder yang dipakai.
  - Tidak berisik saat beroperasi dengan kecepatan tinggi.
- Selain itu, motor servo juga memiliki beberapa kelemahan, yaitu:
- Memerlukan pengaturan yang tepat untuk menstabilkan umpan balik.
  - Motor menjadi tidak terkendali jika *encoder* tidak memberikan umpan balik.
  - Beban berlebih dalam waktu yang lama dapat merusak motor. [8]

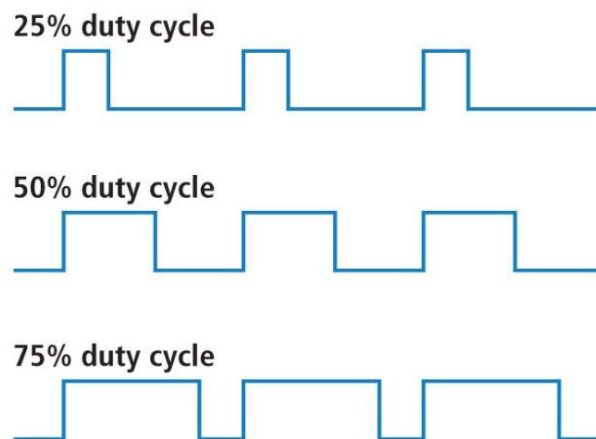
Motor servo adalah motor yang mampu bekerja dua arah (*clockwise* dan *counter clockwise*) dimana arah dan sudut pergerakan rotornya dapat dikendalikan

hanya dengan memberikan pengaturan *dutycycle* sinyal PWM pada bagian pin kontrolnya [8].

## 2.2 Pulse Width Modulation (PWM)

*Pulse width modulation* (PWM) adalah teknik modulasi dengan cara mengatur lebar pulsa *high/low* (*dutycycle*) dengan nilai amplitudo dan frekuensi yang tetap. *Dutycycle* adalah bentuk persentase (%) dari lebar pulsa *high* dan *low* dalam suatu periode sinyal pulsa kotak-kotak [9].

Jika diketahui sinyal pulsa kotak-kotak dengan suatu periode tertentu dan frekuensi tetap, memiliki pulsa kondisi *high* sama besar dengan *low* maka *dutycycle* sebesar 50% (*high = low*). Jika pulsa kondisi *high* terus menerus tanpa *low* artinya *dutycycle* bernilai 100%. Dan jika pulsa kondisi *high* lebih besar dari pada keadaan *low* maka *dutycycle* sebesar lebih dari dari 50% (*high > low*) dan sebaliknya jika siklus *high* kurang dari *low* maka *dutycyclenya* kurang dari 50% (*high < low*). Untuk lebih memahami penjelasan diatas dapat dilihat pada Gambar 2.3 [9] [10].



Gambar 2.1 Contoh *dutycycle* 25%, 50% dan 75% [10]

Penggunaan PWM pada pengaturan kecepatan motor servo dapat dilakukan dengan mengatur *dutycycle* (kondisi *high* dan *low*) sinyal pulsa. Semakin besar *dutycycle* (semakin lebar kondisi *high*) yang diberikan maka kecepatan motor akan berputar semakin cepat dan semakin kecil *dutycycle*

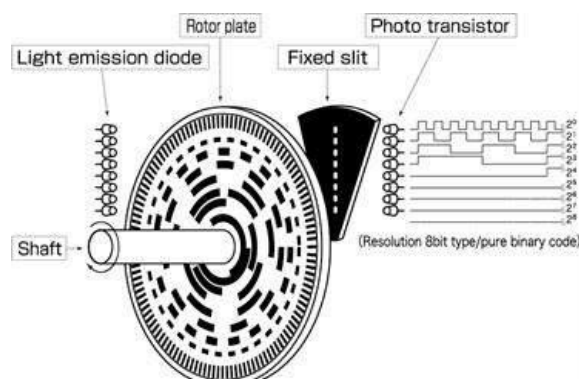
(semakin kecil kondisi *high*) yang diberikan maka motor akan berputar semakin lambat [9].

### 2.3 Rotary Encoder

*Rotary encoder* merupakan sensor penyandi, yang tersusun dari piringan tipis yang memiliki lubang pada bagian lingkaran piringannya. Yang mana diberi kode pada piringan/lempengan dan dilengkapi dengan sensor optik yang berguna untuk menghasilkan *serial* pulsa yang diterjemahkan menjadi gerakan, posisi, dan arah. Informasi yang dihasilkan berupa kode *digital*. Kode digital dapat berupa sama/tetap pada setiap posisi atau lebih dikenal dengan *absolute encoder*. Ada juga kode digital yang berupa kenaikan dari posisi sebelumnya atau disebut dengan *increment encoder* [11].

#### 2.3.1 Absolute Encoder

*Absolute encoder* merupakan sensor yang memiliki piringan dengan pola unik serta sinyal optik yang tersusun dalam jalur *concentric*. Piringan ini membentuk segmen-segmen cincin *concentric* yang jumlah segmennya dua kali jumlah segmen cincin sebelumnya. *Absolute encoder* menghasilkan kode *digital* yang menyatakan sejumlah posisi poros yang berbeda. Misalnya, jika ada 8-bit, *encoder* mampu menghasilkan 256 posisi berbeda atau resolusi sudut 1,406 (360/256) derajat. Dan jika ada 10-bit *encoder* menghasilkan 1024 posisi berbeda dengan resolusi 0.351 (360/1024) derajat. Jenis pengkodean numerik yang digunakan dalam *absolute encoder* adalah *gray code* dan kode biner [11].

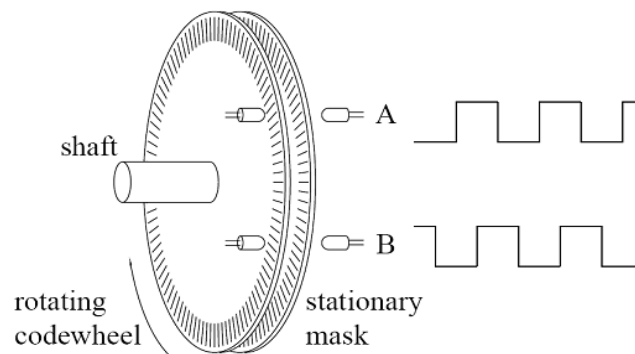


Gambar 2.2 Absolute Encoder [11]

Kelipatan dua dari tiap segmen membentuk sistem biner yang dihasilkan dari LED dan *photo* transistor sebanyak jumlah cincin yang ada. Dengan membaca *output* biner yang dihasilkan maka posisi dari poros yang kita ukur dapat kita ketahui untuk diteruskan ke rangkaian pengendali. Sebagai sensor posisi, *absolute encoder* mengkombinasikan data *output* yang diterbaca menjadi berapa derajat perpindahan posisinya [11].

### 2.3.2 Increment encoder

*Increment encoder* disebut *encoder* yang lebih sederhana daripada *absolute encoder*. *Increment encoder* terdiri dari dua bit dan dua sensor yang *outputnya* disebut *channel A* dan B [11].

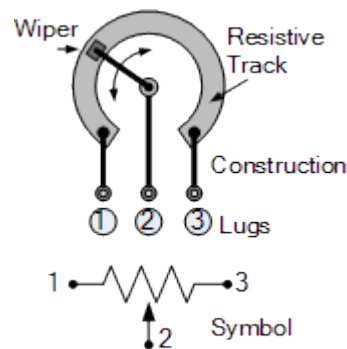


Gambar 2.3 Increment encoder [11]

*Increment encoder* memiliki tuas poros yang dapat diputar yang akan menghasilkan pulsa. Saat poros diputar, pulsa pada *channel* yang frekuensinya sebanding dengan kecepatan putar poros dengan hubungan fase antara sinyal menghasilkan arah rotasi (putaran). *Channel A* dan B digunakan untuk menentukan arah rotasi (putaran) dengan menilai *channel* mana yang memimpin *channel* yang lain. Sinyal dari dua *channel* adalah 1/4 siklus dari fase satu sama lain dan dikenal sebagai sinyal *quadrature*. Sebagai pengatur kecepatan, sensor *increment encoder* ini menghitung *clock* frekuensi dari sebuah pulsa. Sedangkan sebagai sensor posisi, *increment encoder* dilakukan dengan menghitung jumlah pulsa data *output* [11].

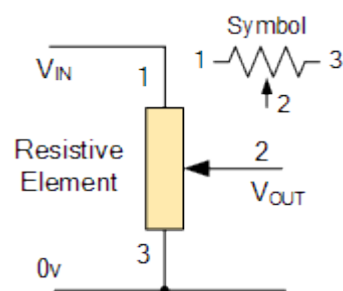
## 2.4 Potensiometer

Potensiometer adalah sensor analog yang berguna untuk mendeteksi posisi. Potensiometer merupakan jenis resistor yang nilai tahanannya bisa diatur sesuai yang diinginkan. Lihat Gambar 2.4 yang menggambarkan bahwa potensiometer terdiri dari tiga kaki terminal dan memiliki tuas putar sebagai pengaturnya. Potensiometer merupakan potensiometer yang perubahan tahanannya berbanding lurus dengan tegangan *output* yang dihasilkan. Yang mana semakin besar perubahan tahanan maka tegangan *output* yang dihasilkan semakin besar begitu juga sebaliknya [12] [13].

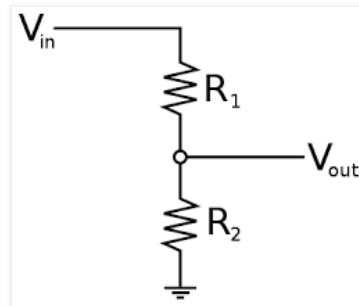


Gambar 2.4 Struktur Fisik Potensiometer [13]

Potensiometer putar merupakan hambatan *variable* yang nilai tahanannya dapat berubah. Nilai hambatan akan bertambah seiring diputarnya poros searah jarum jam (CW) [12].



Gambar 2.5 Rangkaian Potensiometer [13]



Gambar 2.6 Rangkaian Pembagi Tegangan

Potensiometer sebagai sensor posisi haruslah dalam satuan arus atau tegangan, sehingga dapat menggunakan rangkaian pembagi tegangan. Berikut ini merupakan rumus pembagi tegangan.

$$I = \frac{V_{in}}{R_1 + R_2} \dots\dots\dots(2.1)$$

$$V_{out} = I \cdot R \dots\dots\dots(2.2)$$

$$V_{out} = \frac{V_{in}}{R_1 + R_2} \cdot R_2 \dots\dots\dots(2.3)$$

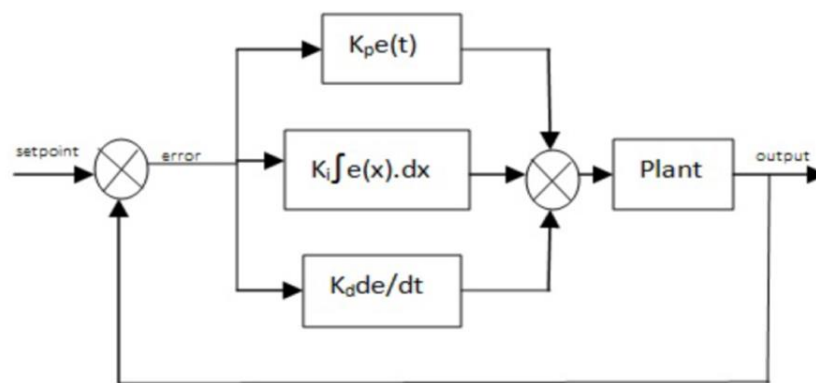
### 2.5 Tachogenerator

Tachogenerator adalah memiliki prinsip kerja sama seperti motor DC. Tachogenerator adalah generator listrik yang kuat untuk mendeteksi kecepatan berputar peralatan listrik. Tachogenerator memberikan tegangan sebanding dengan kecepatan rotasi tapi kemampuan deteksi kecepatan dari tachogenerator tergantung pada kekuatan magnet permanennya. Tegangan keluaran tachogenerator bisa DC atau AC [14].

Tachogenerator mempunyai perbandingan rotasi terhadap tegangan keluaran berbeda-beda umumnya adalah 1000 rpm berbanding 1 volt DC, artinya 1000 rpm kecepatan motor yang terkopel pada tachogenerator, tachogenerator akan menghasilkan tegangan 1 volt. Perbandingan antara kecepatan yang dideteksi oleh tachogenerator dengan tegangan keluarannya adalah berbanding lurus [14].

## 2.6 PID

Kontroler PID (*Proportional-Integral-Derivative controller*) merupakan kontroler dengan umpan balik dengan parameter berupa P, I, D yang berguna untuk memberi respon ke sistem yang diinginkan. Kontroler PID berfungsi untuk meminimalisir *error* dengan cara menghitung nilai *error* yang terjadi secara berulang-ulang. Kontroler mencoba untuk mengurangi nilai *error* dengan mengatur *variabel* kontrolnya. PID terdiri dari parameter kontroler P, I, dan D yang mana ketiga parameter bisa dipakai bersamaan ataupun hanya memakai salah satu nya tergantung dari sistem pemakaian yang diinginkan [15].



Gambar 2.7 Blok Diagram PID Kontroller

Keterangan :

$k_p$  = Konstanta Proporsional

$k_i$  = Konstanta Integral

$k_d$  = Konstanta Derivatif

$e(t)$  = *Error*

Setiap kontroler baik P (Proporsional), I (Integral) maupun D (Derivatif) kontroler memiliki kelebihan dan kekurangan. *Proportional Controller* merupakan perkalian antara konstanta *proportional* dengan *setpoint* yang diinginkan dikurangi dengan *error*. *P Controller* dapat mengurangi *rise time* sehingga *rise time* menjadi cepat, menambah *overshoot* dan mengurangi *steady state*. Sedangkan *Integral Controller* memiliki karakteristik mengurangi *rise time*, menambah *overshoot* dan *setling time*, serta berbeda dengan *controller* lain,

*Integral Controller* dapat menghilangkan *steady state error*. *Derivatif Controller* dapat mengurangi *overshoot* dan *setling time*, tetapi tidak memberikan dampak apapun terhadap *steady state error* [16].



### **BAB III**

#### **METODE PELAKSANAAN**

Proyek akhir ini membuat atau mendesain sebuah *trainer kit* industrial motor servo yang disesuaikan dengan keperluan praktik laboratorium di Politeknik Manufaktur Negeri Bangka Belitung. Berdasarkan hasil diskusi dengan dosen pengajar, maka yang didapatkan:

1. *Trainer kit* ini menggunakan sensor-sensor yang sesuai dengan kebutuhan praktik laboratorium di Politeknik Manufaktur Bangka Belitung.
2. *Trainer kit* ini menggunakan *controller* berupa arduino uno yang dilengkapi dengan *software* Arduino IDE sehingga dapat memudahkan pengguna untuk membuat ulang program yang diinginkan.

Berdasarkan hal-hal di atas maka proyek akhir ini didesain sesuai dengan praktikum servo controller yang meliputi:

1. Modul *driver* motor
2. Modul *increment encoder*
3. Modul *absolute encoder*,
4. Modul potensiometer
5. Modul tachogenerator dan motor DC
6. *Controller* berupa arduino uno yang akan dieksekusi menggunakan *software* Arduino IDE dan LCD yang berfungsi sebagai penampil nilai.

#### **3.1 Komponen yang digunakan**

Sesuai dengan modul-modul yang diperlukan dalam desain dan pembuatan *trainer kit* ini, adapun alat-alat atau komponen yang digunakan diantaranya dapat dilihat pada Tabel 3.1.

Tabel 3.1 Komponen yang digunakan

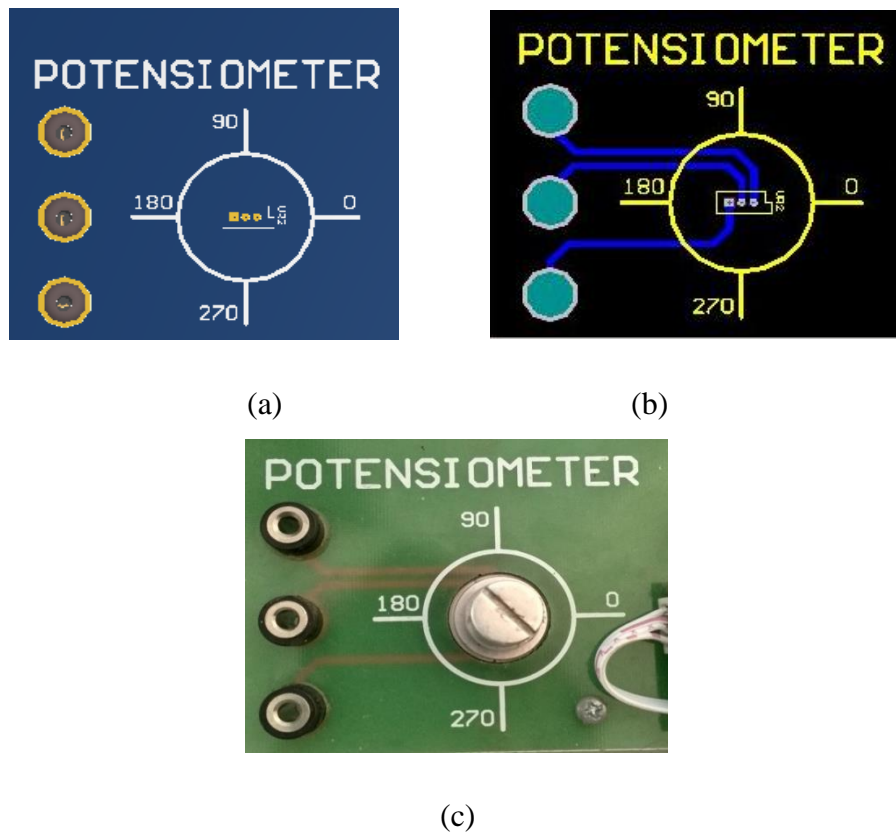
No	Nama	Spesifikasi
1.	Driver motor L298N	<ul style="list-style-type: none"> <li>a. Tegangan minimal masukan 5V~35V</li> <li>b. Arus masukan 0~36 mA</li> <li>c. Daya maksimal 25 W</li> </ul>
2	Arduino Uno	<ul style="list-style-type: none"> <li>a. Tegangan <i>input</i> (rekomendasi) 7~12V</li> <li>b. Tegangan <i>input</i> (limit) 6-20V</li> <li>c. Pin Digital I/O terdapat 14 pin( yang mana 6 pin dapat digunakan sebagai <i>output</i> PWM</li> <li>d. Pin Analog <i>Input</i> terdapat 6 pin</li> <li>e. Arus DC per pin I/O 20 mA</li> <li>f. Arus DC untuk 3.3 V 50 mA</li> </ul>
3	LCD	<ul style="list-style-type: none"> <li>a. Data I/O pin 7-14</li> <li>b. Vcc pin 15</li> <li>c. <i>Ground</i> pin 16</li> <li>d. <i>Number of Character</i> 16 char x 4 line</li> </ul>
6	<i>Rotary encoder</i> KY-040	<ul style="list-style-type: none"> <li>a. Tegangan kerja : 5V</li> <li>b. <i>Pulses/360° Rotation</i>: 20</li> <li>c. <i>Output</i> : 2-bit <i>gray code</i></li> <li>d. Angle Mekanik 360° <i>continuous</i></li> </ul>
7	Absolute encoder hohner Tipe SCS 10	<ul style="list-style-type: none"> <li>a. Tegangan masukkan : 11-24 V</li> <li>b. <i>Output</i> : 10-bit <i>gray code</i></li> </ul>
8	Tachogenerator FOGD 400	<ul style="list-style-type: none"> <li>a. Tegangan masukkan maksimal 20 V</li> <li>b. Tachogenerator dilengkapi sensor increment encoder dengan tegangan masukkan 11-26 V</li> </ul>
9	Sensor Tegangan	<ul style="list-style-type: none"> <li>a. Tegangan <i>Input</i> 0-25 VDC</li> <li>b. Tegangan deteksi 0,02445 VDC</li> <li>c. Ketelitian Pengukuran 0,00489 VDC</li> </ul>

## BAB IV PEMBAHASAN

Pada bab ini akan menjelaskan tentang uji coba modul-modul *trainer kit* yang telah dibuat. Modul-modul yang akan di uji coba antara lain modul *driver motor*, *increment encoder*, potensiometer, *absolute encoder*, tachogenerator dan motor DC. Dilengkapi dengan modul arduino uno dan di program melalui *software* Arduino IDE dan LCD sebagai penampil nilai.

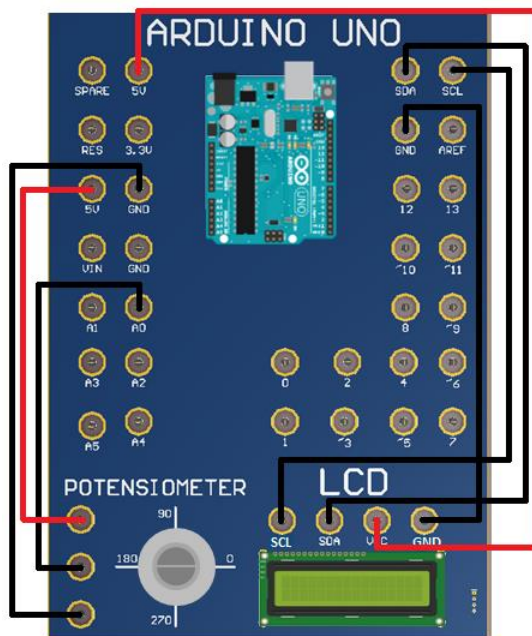
### 4.1 Sensor Posisi Menggunakan Potensiometer

Potensiometer menghasilkan nilai analog yang berfungsi sebagai pengatur PWM analog. Berikut tampilan potensiometer pada modul dan juga *layout PCB* dapat dilihat pada Gambar 4.1.



Gambar 4.1 (a) Desain *Layout* Tampak Atas Potensiometer (b) Desain *Layout* Tampak Bawah Potensiometer (c) Potensiometer pada *Trainer Kit*

Berikut Gambar 4.2 merupakan koneksi perkabelan potensiometer pada *trainer kit* saat proses uji coba. Yang mana mendapatkan *supply 5V – GND* dari Arduino, sedangkan *Vout* terhubung ke A0. Output yang terhubung ke A0 berfungsi untuk membaca nilai analog yang dihasilkan setiap potensiometer diputar searah jarum jam. Lihat gambar 4.2 untuk melihat koneksi perkabelan potensiometer ke arduino untuk menampilkan nilai analog dan tegangannya.



Gambar 4.2 Koneksi Perkabelan Potensiometer

Ketika potensiometer diputar maka nilai yang terukur berupa hambatan dan tegangan akan bertambah sampai nilai maksimal. Setelah mencapai nilai maksimum maka akan kembali ke nilai minimum. Untuk lebih paham, lihat Tabel 4.1 Hasil pengukuran potensiometer. Nilai yang terbaca pada arduino berupa nilai tegangan dan nilai analog *read* pada pin A0.

Tabel 4.1 Hasil Pengukuran Potensiometer

No	Nilai Analog	Tegangan Arduino (V)	Tegangan Multitester (V)	Tahanan (K $\Omega$ )	Persentase Tegangan Error (%)
1	0	0	0	0	0
2	26	0,51	0,53	1,52	0,02
3	51	1,00	1,03	2,76	0,02
4	77	1,51	1,54	4,06	0,03
5	102	2,00	2,02	5,25	0,02
6	128	2,51	2,53	6,59	0,02
7	153	3,00	3,03	7,75	0,02
8	179	3,51	3,54	9,05	0,03
9	204	4,00	4,04	10,26	0,04
10	230	4,51	4,55	11,46	0,04
11	255	5	5,04	12,00	0,04

Perhitungan persentase *error* pada Tabel 4.1 dengan menggunakan beberapa sample maka yang didapat.

$$\text{persentase error} = \left| \frac{0,53 - 0,51}{100} \right| \times 100$$

$$\text{persentase error} = 0,02\%$$

$$\text{persentase error} = \left| \frac{2,53 - 2,51}{100} \right| \times 100$$

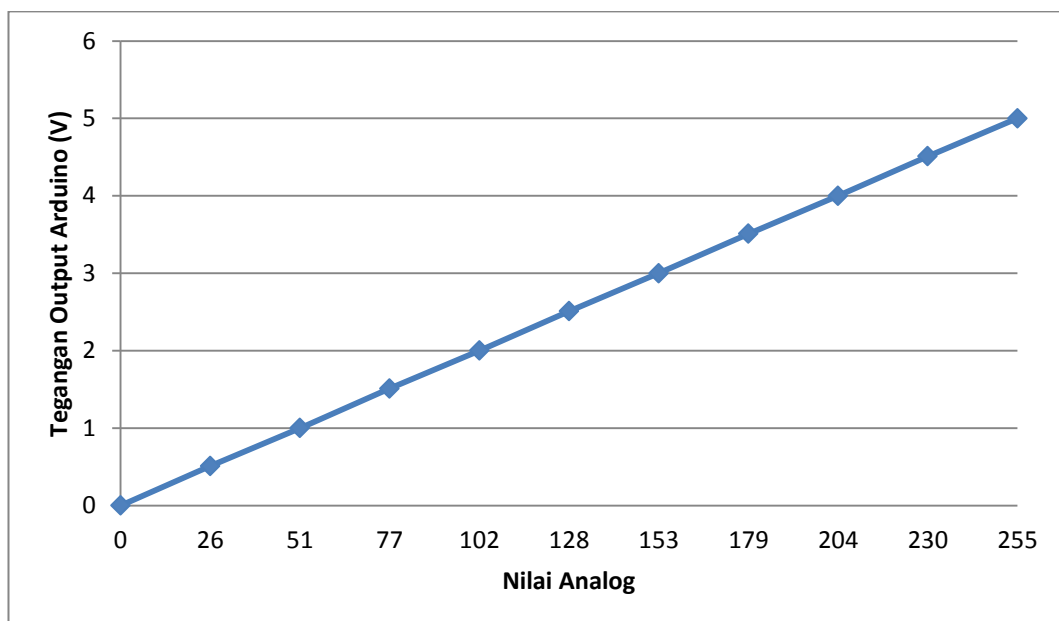
$$\text{persentase error} = 0,02\%$$

$$\text{persentase error} = \left| \frac{5,4 - 5}{100} \right| \times 100$$

$$\text{persentase error} = 0,04\%$$

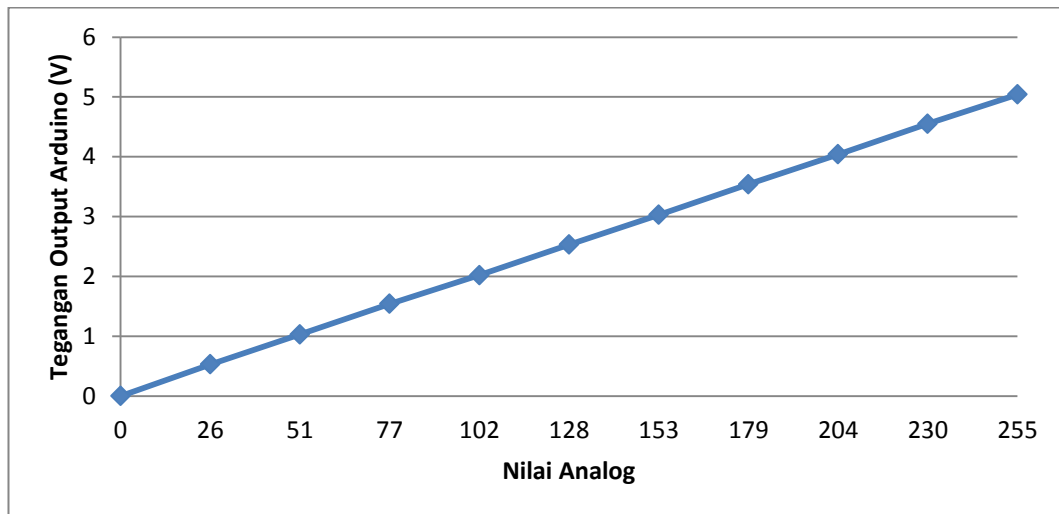
Berdasarkan hasil pada Tabel 4.1 maka dapat disimpulkan bahwa tegangan *output* berbanding lurus dengan bertambahnya nilai Analog arduino. Yang mana semakin besar nilai Analog maka nilai tegangan *output* yang dihasilkan semakin besar, sedangkan semakin kecil nilai Analog maka tegangan *output* yang dihasilkan akan semakin kecil.

Untuk lebih memahami Tabel 4.1 lihat Gambar 4.3 grafik hubungan antara tegangan *output* dengan nilai Analog.



Gambar 4.3 Grafik Tegangan *Output* Arduino Terhadap Nilai ADC

Untuk membandingkan nilai tegangan yang ditampilkan pada LCD maka diukur pula tegangan *output* menggunakan alat ukur multimeter. Untuk lebih jelas dapat dilihat pada Gambar 4.4.



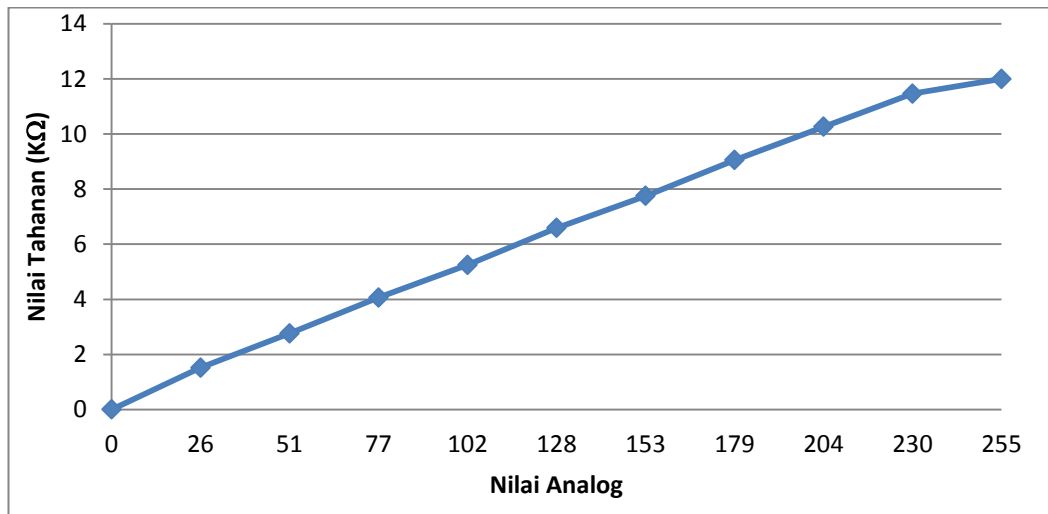
Gambar 4.4 Grafik Tegangan *Output* Multitester Terhadap Nilai ADC

Mengukur tegangan *output* dari potensiometer menggunakan multitester. Ditampilkan juga nilai Analog dan nilai tegangan *output* yang ditampilkan pada LCD. Berikut ini Gambar 4.5 merupakan tampilan dari nilai Analog dan tegangan *output arduino* dari potensiometer.



Gambar 4.5 Tampilan LCD Nilai ADC dan Tegangan

Hal ini berlaku juga untuk nilai tahanan yang terukur pada multimeter. Berdasarkan Tabel 4.1 dapat disimpulkan bahwa semakin kecil nilai Analog maka nilai tahanan yang terukur akan semakin kecil sedangkan semakin besar nilai Analog maka nilai tahanan yang terukur akan semakin besar. Untuk lebih memahami penjelasan diatas dapat dilihat Gambar 4.6 merupakan grafik hubungan antara perubahan nilai Analog terhadap nilai tahanan yang terukur.



Gambar 4.6 Grafik Nilai Tahanan Terhadap Nilai ADC

Nilai tahanan dari potensiometer diukur pada kaki *output* yang mana terhubung ke A0 dengan *ground* potensiometer. Nilai tahanan diukur tanpa adanya tegangan yang masuk ke potensiometer. Berikut Gambar 4.7 hasil pengukuran tahanan potensiometer.



Gambar 4.7 Hasil Pengukuran Nilai Tahanan Potensiometer

---

```

void loop() {
N_ADC = analogRead(potensio);           ] Membaca nilai Analog
                                           ] potensiometer
nilai_ADC = map(N_ADC, 0,1023,0,255);    ] mapping nilai Analog
Tegangan = nilai_ADC*(5.0/255.0);        ] baca nilai tegangan
delay(1000);                             ] delay 1 s
}

```

---



## 4.2 Sensor Posisi Menggunakan *Increment Encoder*

*Increment encoder* yang digunakan pada *trainer kit* ini adalah KY-040 yang memiliki karakteristik sebagai berikut ini.

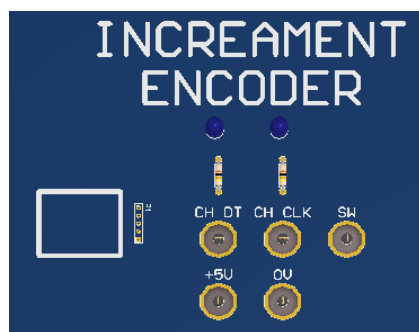
- *Increment encoder* ini memiliki dua buah sensor yaitu *channel A (DT)* dan *channel B (CLK)*.
- Tegangan masukan yang mengaktifkan sensor ini adalah 5V.

Berdasarkan datasheet, *increment encoder* ini memiliki lima buah pin *connector*. Untuk lebih jelas dapat dilihat pada Tabel 4.2.

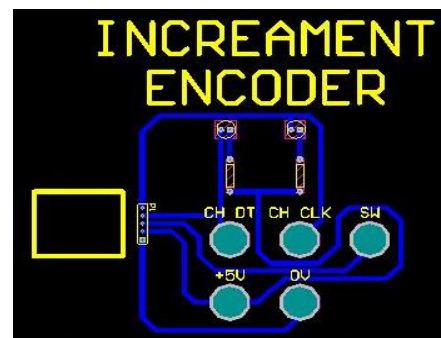
Tabel 4.2 *Connector Pin* pada *Increment Encoder* ( KY-040)

No	Pin Connector	Keterangan
1	GND	<i>Ground</i>
2	+	VCC
3	SW	<i>Switch</i>
4	DT	<i>Channel A</i>
5	CLK	<i>Channel B</i>

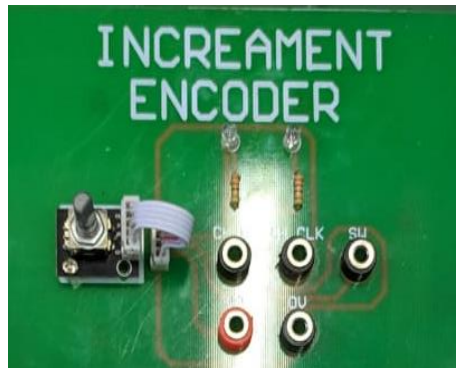
Berikut pada Gambar 4.8 merupakan desain *layout* dan gambar *trainer kit* pada *increment encoder* yang akan digunakan.



(a)



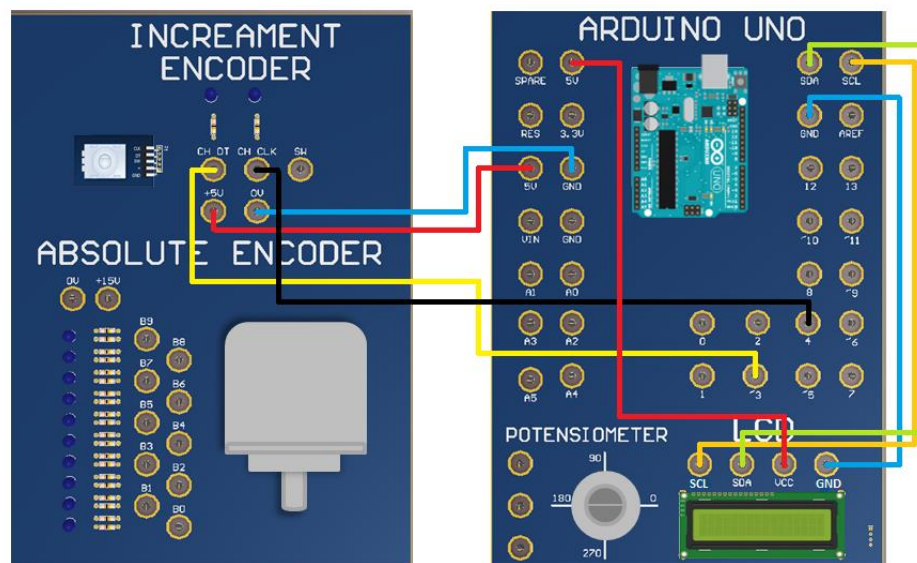
(b)



(c)

Gambar 4.8 (a) Desain *Layout* Tampak Atas *Increment Encoder* (b) Desain *Layout* Tampak Bawah (c) *Increment Encoder* pada *Trainer Kit*

Berikut Gambar 4.9 merupakan koneksi perkabelan *increment encoder* pada *trainer kit* saat proses uji coba.



Gambar 4.9 Koneksi Perkabelan *Increment Encoder* pada *Trainer Kit*

*Increment encoder* ini memiliki 4 *shaft*, yang mana akan terjadi pengulangan ketika diputar searah atau berlawanan arah jarum jam. *Output* dari kedua *channel* menentukan arah putar *increment encoder*. Saat *increment encoder* diputar maka kedua *output* yaitu *channel A* dan *B* bergantian mengeluarkan logika 1 (*high*) atau 0 (*low*). *Channel* mana terlebih dahulu yang berubah maka

akan menentukan kearah mana *increment encoder* di putar. *Increment encoder* diputar *clockwise* menyebabkan *channel A* berubah terlebih dahulu daripada *channel B*. Sedangkan saat *increment encoder* diputar *counter clockwise* menyebabkan nilai B berubah terlebih dahulu daripada A.

Untuk lebih jelas lihat pada Tabel 4.3 hasil percobaan yang menampilkan sinyal *output* ketika diputar searah jarum jam (*clockwise*).

Tabel 4.3 Hasil Percobaan pada *Channel A* dan B ( *clockwise* )

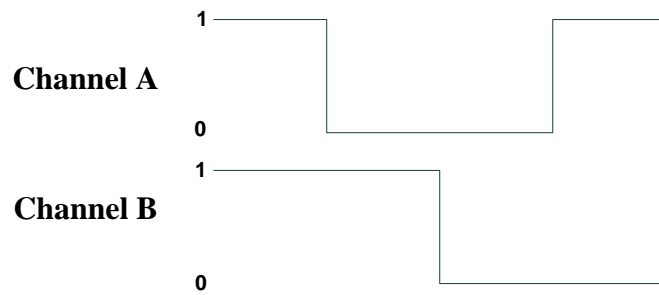
No	Channel A	Channel B
1	1	1
2	0	1
3	0	0
4	1	0

Sedangkan pada Tabel 4.4 menampilkan sinyal *output* ketika diputar searah jarum jam (*counter clockwise*).

Tabel 4.4 Hasil Percobaan pada *Channel A* dan B ( *Counter Clockwise* )

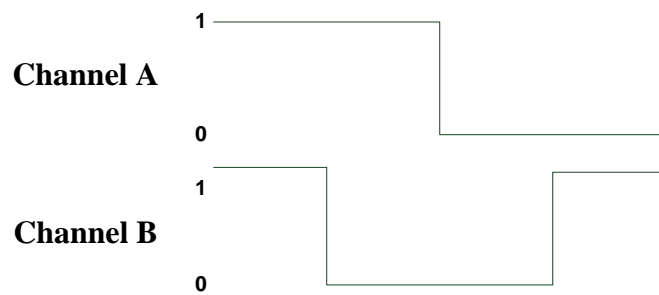
No	Channel A	Channel B
1	1	1
2	1	0
3	0	0
4	0	1

Untuk lebih memahami Tabel 4.3 dan Tabel 4.4 maka dibuatlah Gambar 4.10 dan Gambar 4.11 menampilkan sinyal keluaran *increment encoder* ketika diputar *clockwise* dan *counter clockwise*.



Gambar 4.10 Sinyal Keluaran *Increment Encoder* Searah Jarum Jam

Keterangan: 1 = LED padam  
0 = LED nyala



Gambar 4.11 Sinyal Keluaran *Increment Encoder* Berlawanan Jarum Jam

Keterangan: 1 = LED padam  
0 = LED nyala

Tegangan *input* untuk mengaktifkan sensor *increment encode* ini adalah +5 V. Lihat pada Gambar 4.12 dan Gambar 4.13, yang mana ketika sensor *increment encoder* mengeluarkan logika 1 maka tegangan *output* yang terukur adalah 3,3 mV (mendekati 0V) maka LED akan menyala artinya sensor berada pada kondisi *high* dan ketika sensor *increment encoder* mengeluarkan logika 0 maka tegangan *output* yang terukur adalah 5,03 V (mendekati 5V) maka LED akan padam yang artinya sensor berada pada kondisi *low* sensor padam. Dari penjelasan diatas dapat disimpulkan bahwa sensor *increment encoder* merupakan sensor yang aktif pada kondisi *low* (*active low*).



Gambar 4.12 Hasil Pengukuran Tegangan *Increment* pada Logika Rendah



Gambar 4.13 Hasil Pengukuran Tegangan *Increment* Pada Logika Tinggi

*Output increment encoder* dapat berupa nyala atau padamnya LED. *Output* sensor *increment* ini dihubungkan pada pin-pin digital arduino untuk melihat hasil berupa nyala atau padam dari LED yang ditampilkan pada LCD. LCD akan menampilkan arah putaran, kondisi ketika diputar searah jarum jam atau berlawanan arah jarum jam dan logika yang terbaca dari sensor *increment encoder*. Ketika *increment encoder* mengeluarkan logika 1 (*high*) maka LED akan padam dan ketika *increment encoder* mengeluarkan logika 0 (*low*) maka LED akan nyala. Untuk memahami penjelasan diatas, lihat Gambar 4.14 sampai 4.17.

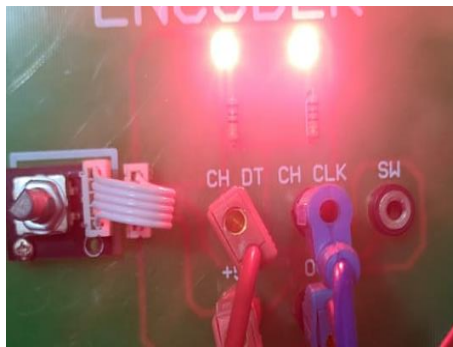


(a)



(b)

Gambar 4.14 (a) *Output* LED A dan B padam (b) *Output* A dan B high



(a)



(b)

Gambar 4.15 (a) *Output* LED A dan B nyala (b) *Output* A dan B low



(a)



(b)

Gambar 4.16 (a) *Output* LED A nyala B padam (b) *Output* A low B high



(a)



(b)

Gambar 4.17 (a) *Output* LED A padam B nyala (b) *Output* A high B low

Berikut ini merupakan program yang digunakan untuk membaca nilai dari sensor *increment encoder*.

---

```

void loop() {
    nilai_A = digitalRead(pinA);           ] baca nilai pin A
    nilai_B = digitalRead(pinB);           ] baca nilai pin B
    if (nilai_A != pinALast)                ] jika knob diputar, A tidak sama dengan nilai A
                                           ] yang lalu.
    {
        if (nilai_B != nilai_A)            ] Jika nilai B tidak sama dengan nilai B atau jika
                                           ] nilai A berubah terlebih dahulu
        {
            encoderPosCount ++;            ] maka increment nilai encoder (+1)
            bCW = true;                      ] kalau bCW true maka berputar CW
        }
        else                                 ] Jika Nilai B berubah dulu
        {
            bCW = false;                    ] maka kalau bCW false maka berputar CCW
            encoderPosCount--;              ] maka decrement nilai encoder (-1)
        }
    }
    pinALast = nilai_A;                     ] Nilai pin A terakhir = nilai A
}

```

---



### 4.3 Sensor Posisi Menggunakan *Absolute Encoder*

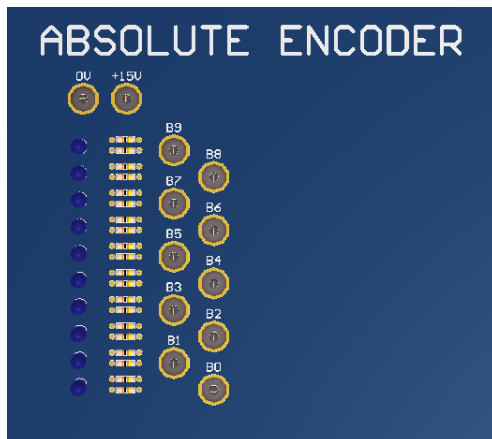
*Absolute encoder* yang digunakan adalah dari merek hohner tipe SCS 10. *Absolute encoder* ini mengeluarkan *output* berupa kode biner atau bisa juga *gray code*. Yang mana sensor ini memiliki 10 cincin *concentric* unik dengan 10 sensor optik. Hal ini menyebabkan sensor dapat menghasilkan 10 keluaran yang membentuk *gray code* sehingga sensor ini dapat mengeluarkan 1024 kondisi. Berdasarkan datasheet, *absolute encoder* ini memiliki 12 buah *connector*. Tabel 4.5 merupakan *connector* yang terdapat pada *absolute encoder*.

Tabel 4.5 *Connector* pada *absolute encoder* (Hohner SCS10)

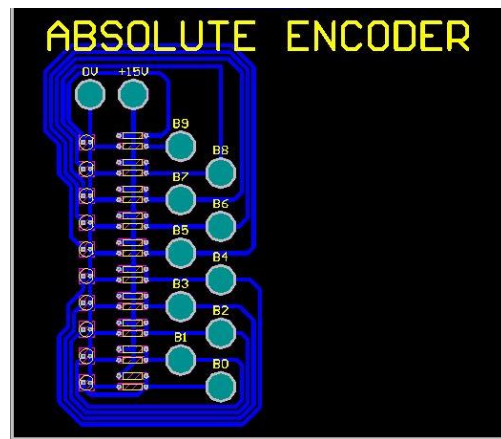
No	<i>Connectors</i>	<i>Plug</i>
1	VCC	<i>Brown</i>
2	GND	<i>White</i>
3	B0	<i>Green</i>
4	B1	<i>Yellow</i>
5	B2	<i>Grey</i>
6	B3	<i>Pink</i>
7	B4	<i>Blue</i>
8	B5	<i>Red</i>
9	B6	<i>Black</i>
10	B7	<i>Violet</i>
11	B8	<i>Pink-Gray</i>
12	B9	<i>Red-Blue</i>

Berikut tampilan *absolute encoder* pada modul dan juga *layout PCB* pada *trainer kit* dapat dilihat pada Gambar 4.18.





(a)



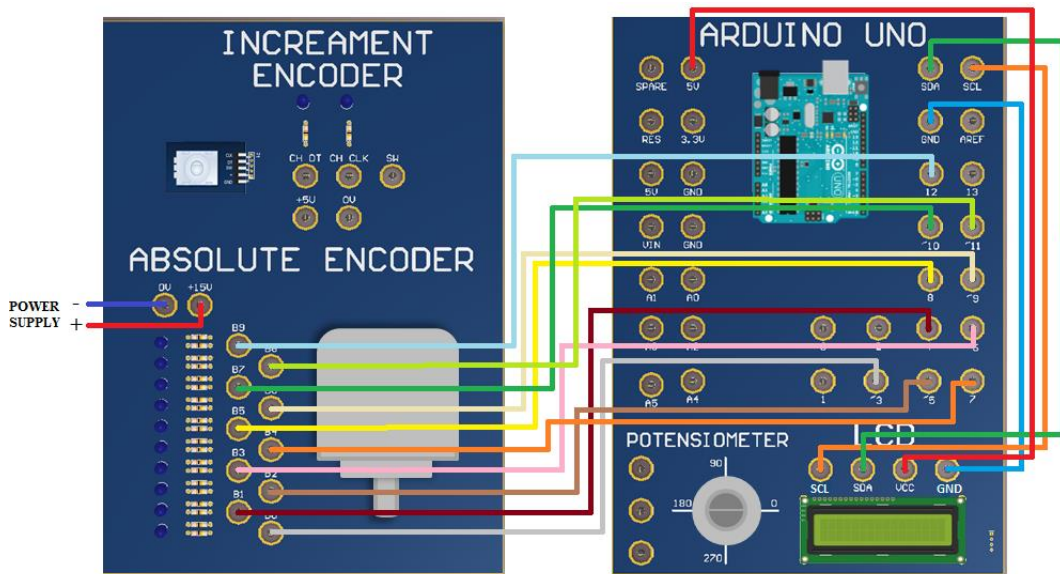
(b)



(c)

Gambar 4.18 (a) Desain *Layout* Tampak Atas *Absolute Encoder* (b) Desain *Layout* Tampak Bawah *Absolute Encoder* (c) *Absolute Encoder* pada *Trainer Kit*

Berikut Gambar 4.19 merupakan koneksi perkabelan *absolute encoder* pada *trainer kit* saat proses uji coba.

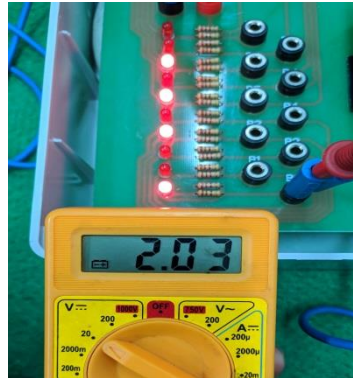


Gambar 4.19 Koneksi Perkabelan *Absolute Encoder* pada *Trainer Kit*

*Absolute encoder* memiliki 10 keluaran. Pada Gambar 4.19 *absolute encoder* mempunyai 10 *output* yang terhubung ke LED. Tegangan masukan untuk mengaktifkan sensor *absolute encoder* ini ialah +15V, tegangan ini melebihi tegangan *output* yang akan masuk ke arduino dan LED sehingga tegangan *output* akan diturunkan menggunakan rangkaian pembagi tegangan pada masing-masing *output*. Hasil pengukuran tegangan *output* pada masing-masing *output* sensor dapat dilihat pada Gambar 4.20 dan Gambar 4.21.

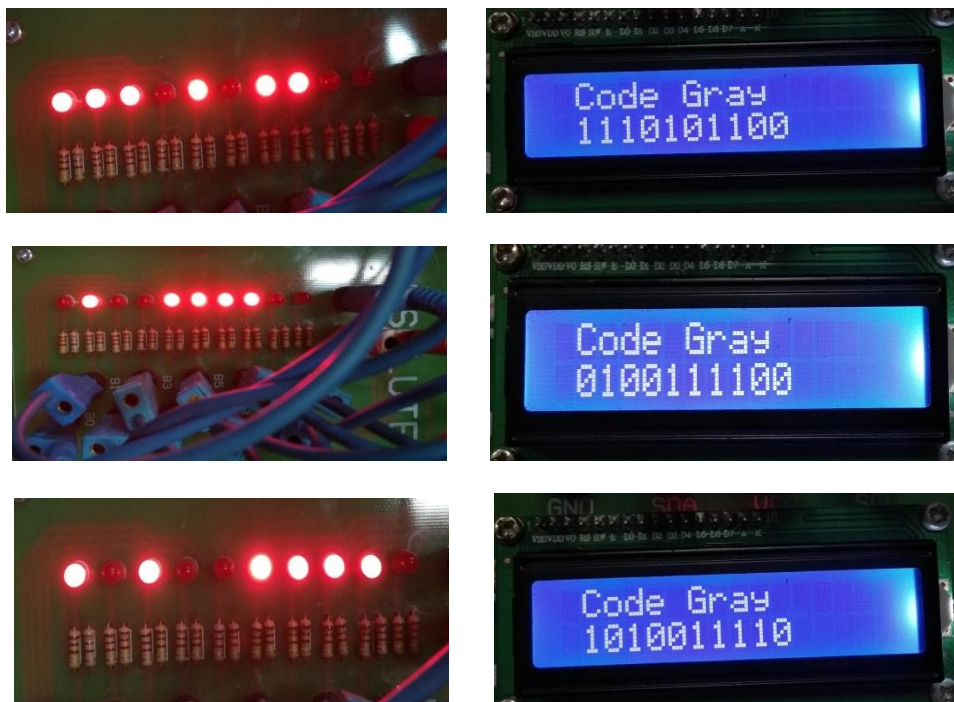


Gambar 4.20 Hasil Pengukuran Tegangan *Absolute* Pada Logika Rendah



Gambar 4.21 Hasil Pengukuran Tegangan *Absolute* Pada Logika Tinggi

Ketika LED menyala maka *output absolute encoder* akan mengeluarkan logika 1 dengan tegangan *output* yang terukur adalah 2,3 V ketika sensor *absolute encoder* mengeluarkan logika 0 maka tegangan *output* yang terukur adalah 0,04 V sehingga menyebabkan LED akan padam. Dari kondisi pada Gambar 4.20 dan Gambar 4.21 dapat disimpulkan bahwa sensor *absolute encoder* merupakan sensor yang aktif pada kondisi *high (active high)*. Berikut ini Gambar 4.22 merupakan gambar hasil uji coba *absolute encoder*.



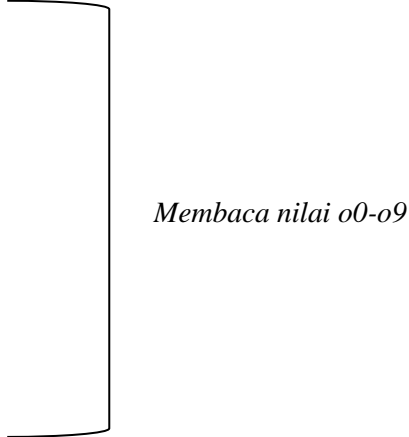
Gambar 4.22 Hasil Pengujian *Absolute Encoder*

Keterangan : 1 = LED menyala

0 = LED padam

---

```
void loop() {  
o0 = digitalRead(B0);  
o1 = digitalRead(B1);  
o2 = digitalRead(B2);  
o3 = digitalRead(B3);  
o4 = digitalRead(B4);  
o5 = digitalRead(B5);  
o6 = digitalRead(B6);  
o7 = digitalRead(B7);  
o8 = digitalRead(B8);  
o9 = digitalRead(B9);  
}
```

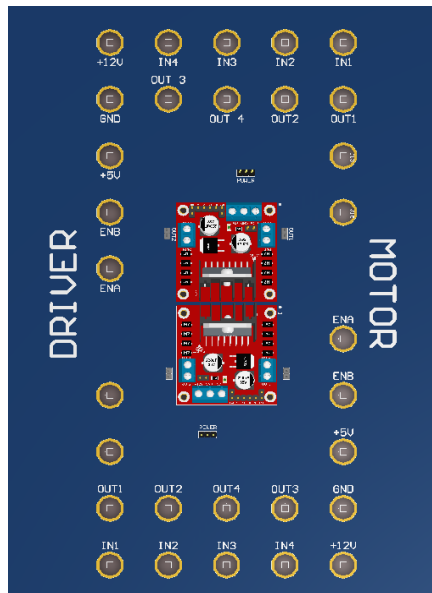


Membaca nilai o0-o9

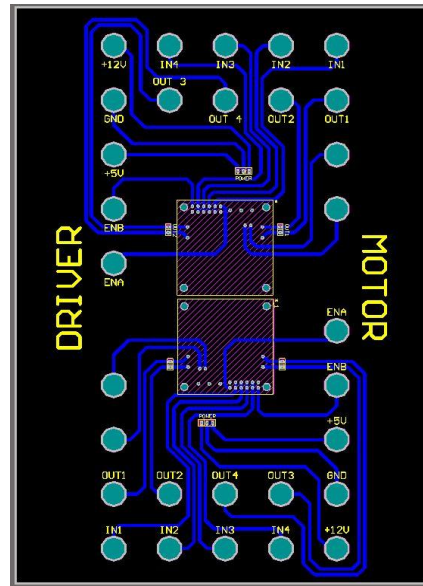
---

#### 4.4 Pengaturan Kecepatan Menggunakan PWM

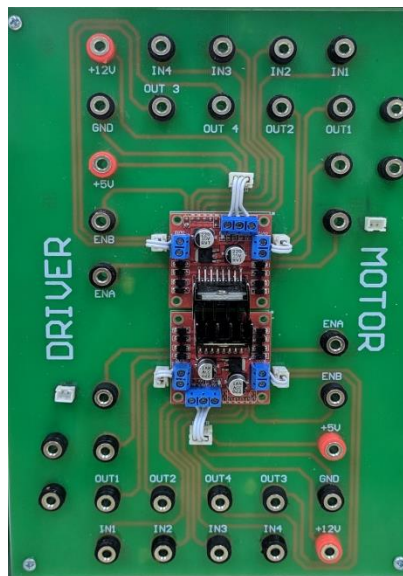
Pengaturan kecepatan menggunakan teknik PWM butuh sebuah *driver* motor. *Driver* motor dapat digunakan untuk mengendalikan arah putaran motor. *Driver* motor L298N didalamnya terdapat rangkaian H-*bridge* yang artinya dapat mengendalikan dua buah motor DC dengan dua arah putaran yaitu *clockwise* (CW) dan *counter clockwise* (CCW). Berikut pada Gambar 4.23 merupakan desain *layout* dan gambar *trainer kit* pada *driver* motor yang akan digunakan.



(a)



(b)

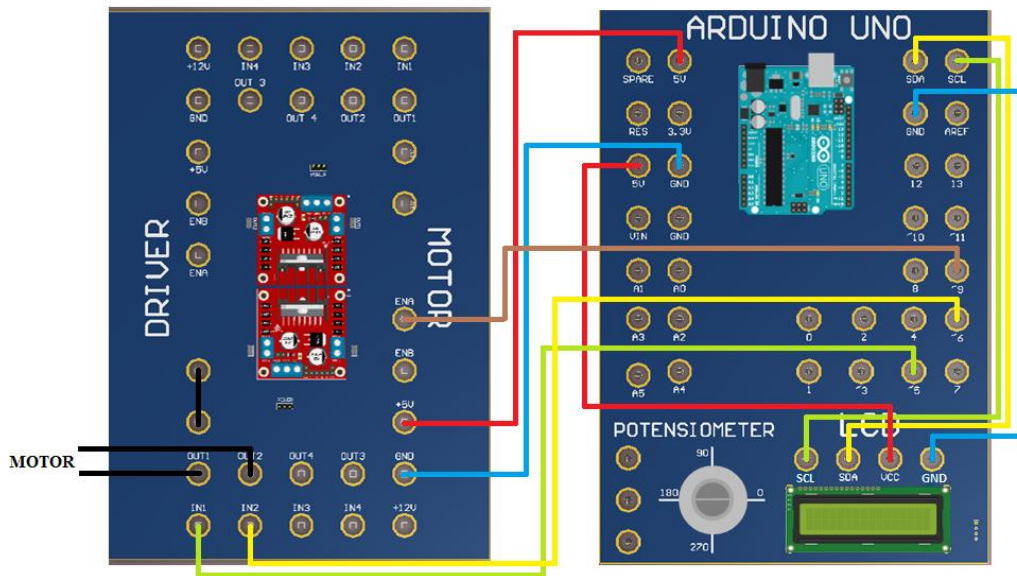


(c)

Gambar 4.23 (a) Desain *Layout* Tampak Atas *Driver* Motor (b) *Driver Layout* Tampak Bawah *Driver* Motor (c) *Driver* Motor pada *Trainer Kit*

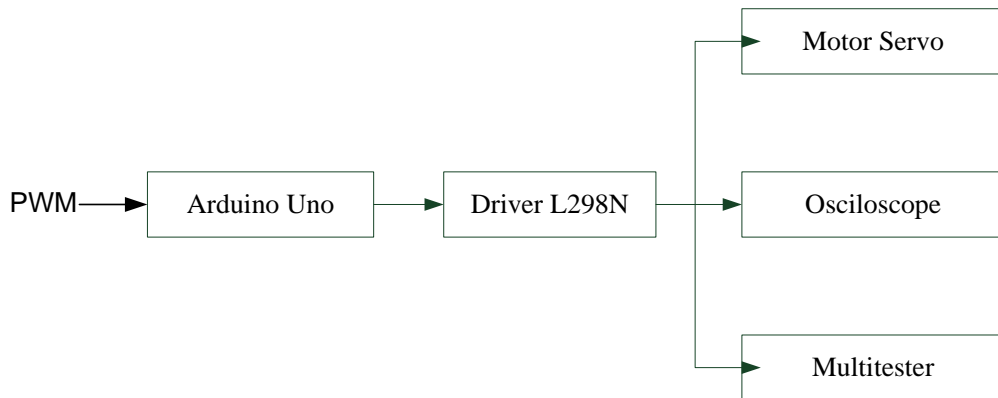
Berikut Gambar 4.24 merupakan koneksi perkabelan *driver* motor pada *trainer kit* saat proses uji coba.





Gambar 4.24 Koneksi Perkabelan *Driver Motor* Pada *Trainer Kit*

Untuk lebih memahami proses pengujian *trainer kit*, lihat Gambar 4.25 berikut ini.



Gambar 4.25 Blok Diagram Pengukuran *Driver Motor* L298N

Arah putaran motor dapat dikendalikan dengan cara memberikan logika berupa 1 (*high*) atau 0 (*low*) pada pin IN1 sampai IN4. Apabila hanya mengontrol satu buah motor, pin-pin yang diberi logika hanya pin IN1 dan IN2. Ketika IN1 berlogika 1 (*high*) sedangkan IN2 berlogika 0 (*low*) maka motor akan berputar *clockwise* (CW), tetapi ketika IN1 berlogika 0 (*low*) dan IN2 berlogika 1 (*high*) maka motor akan berputar *counter clockwise* (CCW). Sedangkan ketika IN1 dan

IN2, kedua-duanya diberikan logika 1 (*high*) atau 0 (*low*) maka motor tidak akan berputar.

Untuk mengendalikan arah putaran maka dapat melihat tabel hasil pengujian arah putaran motor pada Tabel 4.6.

Tabel 4.6 Hasil Pengujian Arah Putaran Motor

IN 1	IN 2	Arah Putaran
LOW	LOW	STOP
LOW	HIGH	CCW
HIGH	LOW	CW
HIGH	HIGH	STOP

*Driver* Motor L298N juga dapat mengendalikan kecepatan motor dengan teknik PWM (*Pulse Width Modulation*). PWM dapat mengendalikan kecepatan dengan cara mengatur besarnya *dutycycle* atau lebar pulsa *high* dan *low*. Untuk mengendalikan kecepatan motor ini telah menyediakan pin khusus yaitu ENA dan ENB. Cara mengendalikannya dengan menghubungkan pin ENA atau ENB dengan pin arduino yang memiliki PWM. *Output driver* terhubung pada *oscilloscope* untuk mendapatkan data gelombang PWM yang terhubung ke pin ENA. Kemudian dapat dikontrol menggunakan *software* IDE berapa persentase *dutycycle* atau dengan cara memasukkan nilai ADC yang kita inginkan melalui serial monitor. Nilai parameter berkisar antara 0 hingga 255. Bila hendak mengatur *dutycycle* ke 0%, maka kita akan mengisi nilai parameter ke 0 dan untuk *dutycycle* ke 100% maka atur nilai ke 255. Jika mengatur *dutycycle* ke 50% berarti nilai yang harus diisi adalah 127. Hal ini didapatkan dari nilai parameter dikali dengan berapa persentase nilai *dutycycle*. Sehingga dapat dirumuskan seperti berikut.

$$Dutycycle = \frac{T_{on}}{T_{on}+T_{off}} \dots\dots\dots(4.1)$$

$$Dutycycle = \frac{T_{on}}{T_{total}} \dots\dots\dots(4.2)$$

Tabel 4.7 untuk melihat hasil pengukuran tegangan *output* dan kecepatan motor menggunakan teknik PWM.

Tabel 4.7 Hasil Pengukuran Tegangan *Output* dan Kecepatan Motor

No	<i>Dutycycle</i> (%)	Nilai ADC	Tegangan <i>Output</i> (V)	Kecepatan (rpm)
1	0	0	0	0
2	10	26	4,18	1331
3	20	51	4,77	1375
4	30	77	5,55	2590
5	40	102	6,18	2306
6	50	128	8,77	2588
7	60	153	9,55	2847
8	70	179	10,37	2882
9	80	204	10,45	2907
10	90	230	10,51	2931
11	100	255	10,58	2957

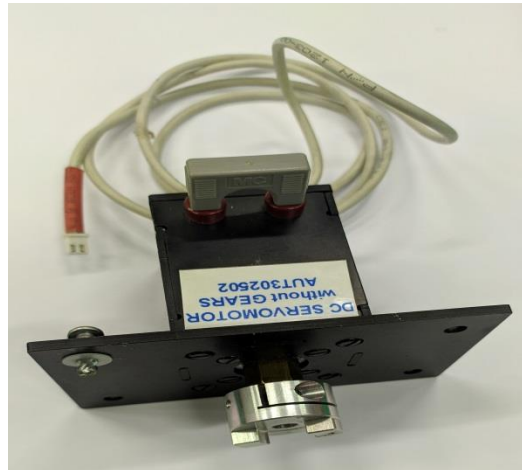
Pengukuran tegangan *output* menggunakan multimeter sedangkan kecepatan diukur menggunakan tachometer. Gambar 4.26 merupakan sinyal PWM *output driver* dengan *dutycycle* yang diatur dari Arduino.



Gambar 4.26 Sinyal PWM dengan *dutycycle* 30,2%

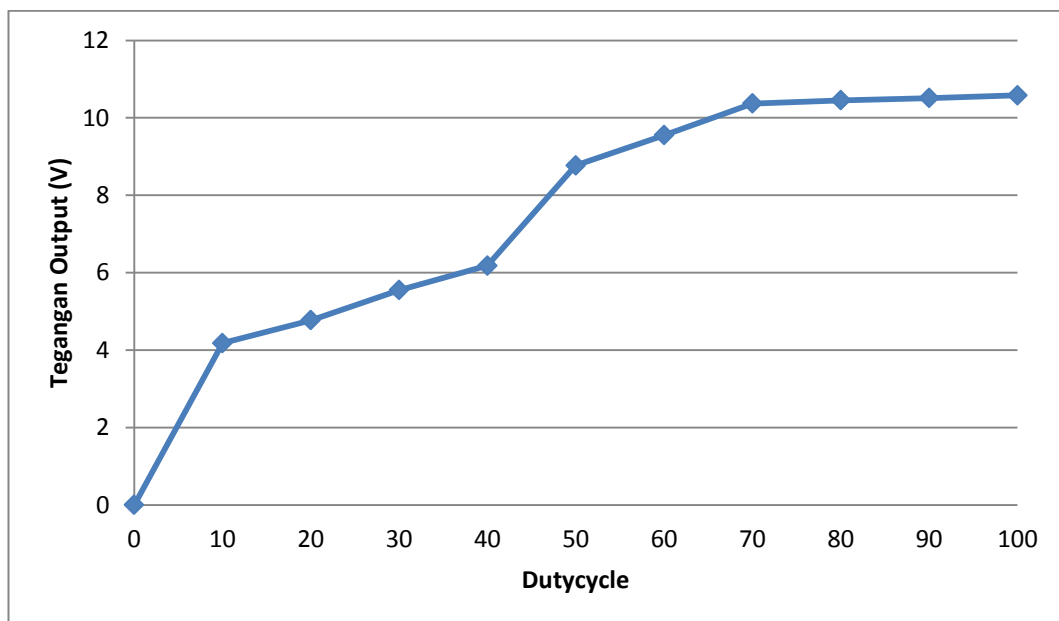


Dari Tabel 4.7 merupakan hasil pengukuran serta kecepatan motor dan tegangan *output* dengan menggunakan beban motor berupa motor servo tanpa gear lihat Gambar 4.27.



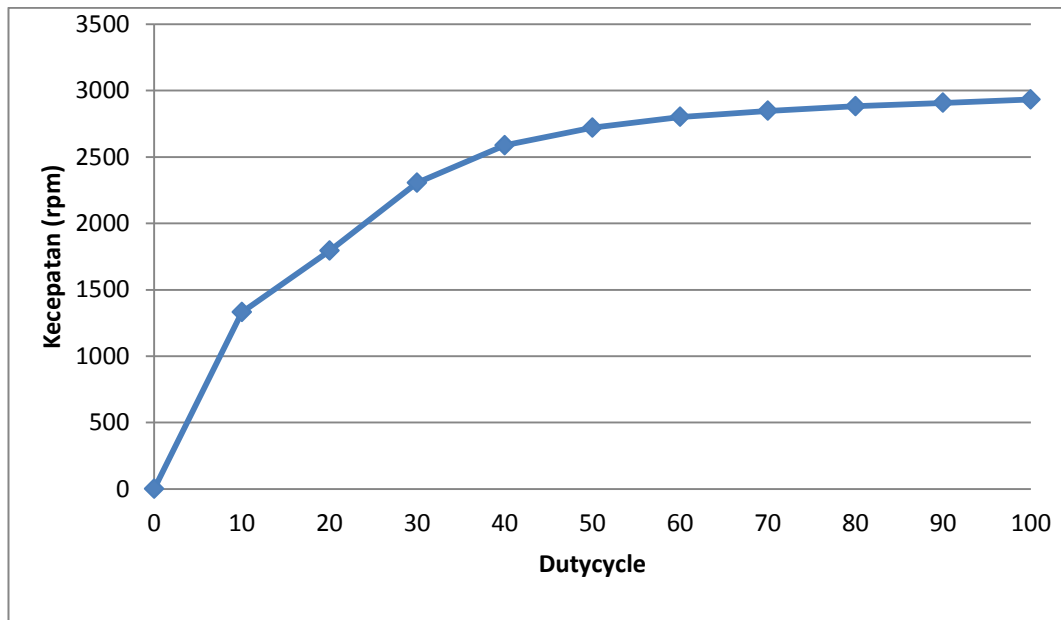
Gambar 4.27 Motor Servo Tanpa Gear

Gambar 4.28 merupakan grafik hasil pengukuran PWM dengan tegangan *output* dengan beban motor. Dapat dilihat dari grafik pada Gambar 4.28 bahwa semakin besar nilai *dutycycle* yang diberikan maka tegangan *output* yang terukur semakin bertambah.



Gambar 4.28 Grafik hubungan PWM dengan tegangan *output* beban motor

Pengukuran kecepatan putaran motor menggunakan alat ukur tachometer. Berikut ini Gambar 4.29 merupakan grafik hasil pengukuran PWM dengan tegangan *output* dengan beban motor.



Gambar 4.29 Grafik hubungan antara PWM dengan kecepatan

Berikut Gambar 4.30 merupakan gambar hasil pengujian kecepatan motor servo menggunakan alat ukur tachometer.



Gambar 4.30 Hasil Pengujian Kecepatan pada PWM 30 % 50% 100%

---

```

void loop()
{
  if (Serial.available())
  {
    nilaiADC=Serial.readStringUntil('\n');
    Serial.println(nilaiADC);
    inputADC=nilaiADC.toInt();
    Serial.println(inputADC);
  }
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  analogWrite (ENA,inputADC);
  pwm = map(inputADC,0,255,0,100);
  delay(1000);
}

```

Serial monitor untuk menginput nilai ADC

instruksi motor berputar CW

nilai ADC yang terinput (PWM)

mapping nilai ADC

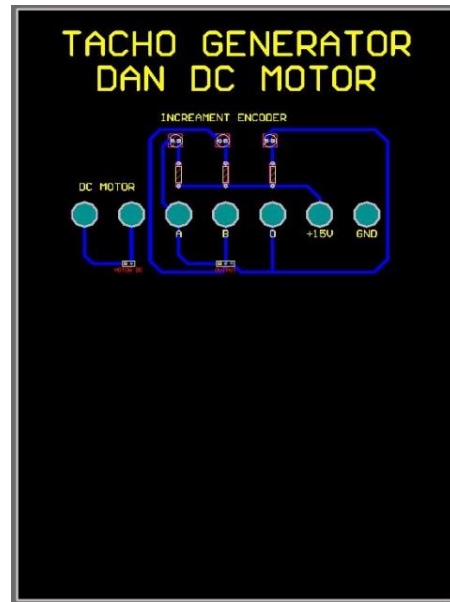
---

#### 4.5 Pengaturan Kecepatan Menggunakan Tachogenerator

Tachogenerator membangkitkan tegangan DC. Yang mana pada *trainer kit* industrial motor servo ini, tachogenerator dihubungkan dengan motor DC menggunakan elemen transmisi. DC motor digunakan untuk menggerakkan tachogenerator dibantu dengan *gear* dan *belt* yang merupakan elemen transmisi yang digunakan pada *trainer kit* ini. Untuk melihat desain *trainer kit* dapat dilihat pada Gambar 4.31.



(a)



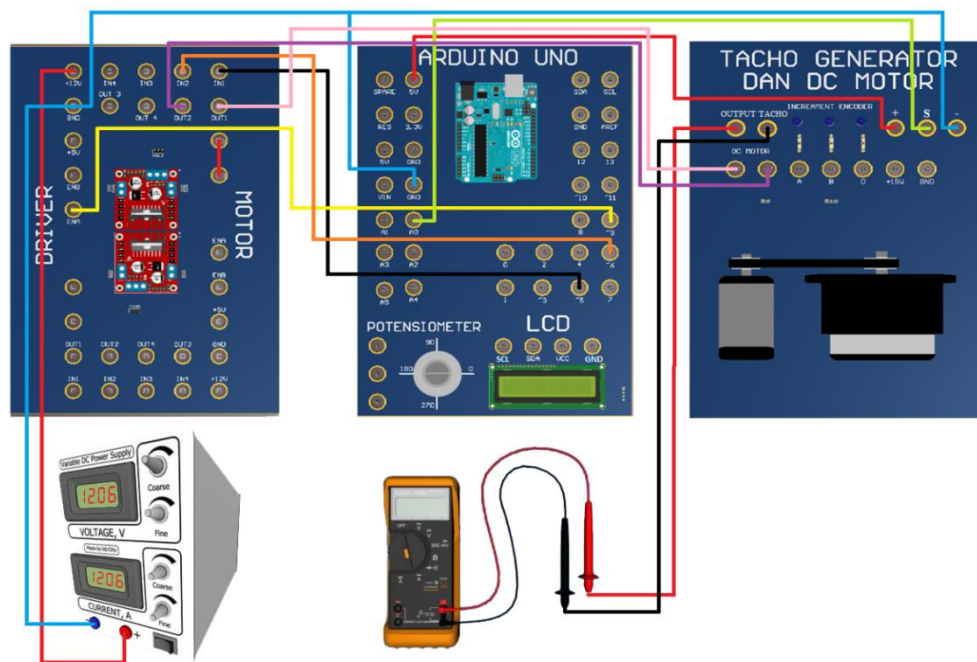
(b)



(c)

Gambar 4.31 (a) Desain *Layout* Tampak Atas Tachogenerator dan Motor DC (b) Desain *Layout* Tampak Atas Tachogenerator dan Motor DC (c) Tachogenerator dan Motor DC pada *Trainer Kit*

Berikut Gambar 4.32 merupakan koneksi perkabelan tachogenerator dan motor DC ketika proses uji coba *trainer kit*.



Gambar 4.32 Koneksi perkabelan tachogenerator dan motor DC pada *trainer kit*

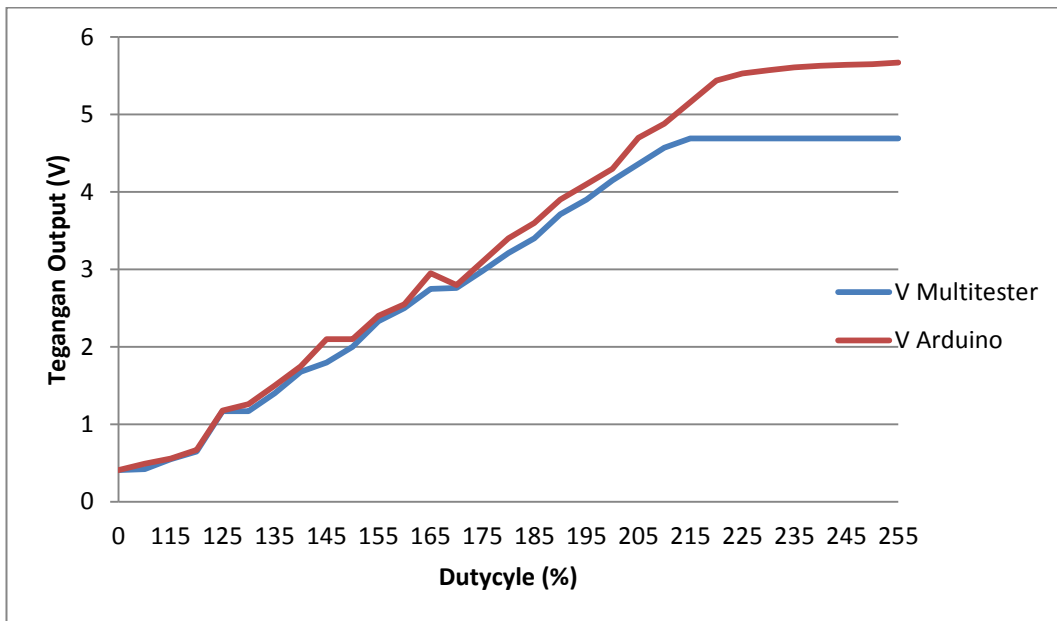
Pengujian dilakukan dengan mengukur tegangan yang dihasilkan oleh tachogenerator. Adapun data hasil pengujian adalah seperti Tabel 4.8

Tabel 4.8 Hasil Pengukuran Tegangan Tachogenerator dan Motor DC

No	<i>Input Duty cycle/ADC</i>	Tegangan Multitester	Tegangan Arduino	Kecepatan (rpm)
1	105	0,41	0,41	28
2	110	0,49	0,42	157
3	115	0,56	0,55	170
4	120	0,67	0,65	258
5	125	1,18	1,17	312
6	130	1.26	1.17	350
7	135	1.5	1.4	423
8	140	1.75	1.68	529
9	145	2.1	1.8	622
10	150	2.1	2.00	744
11	155	2.4	2.33	800

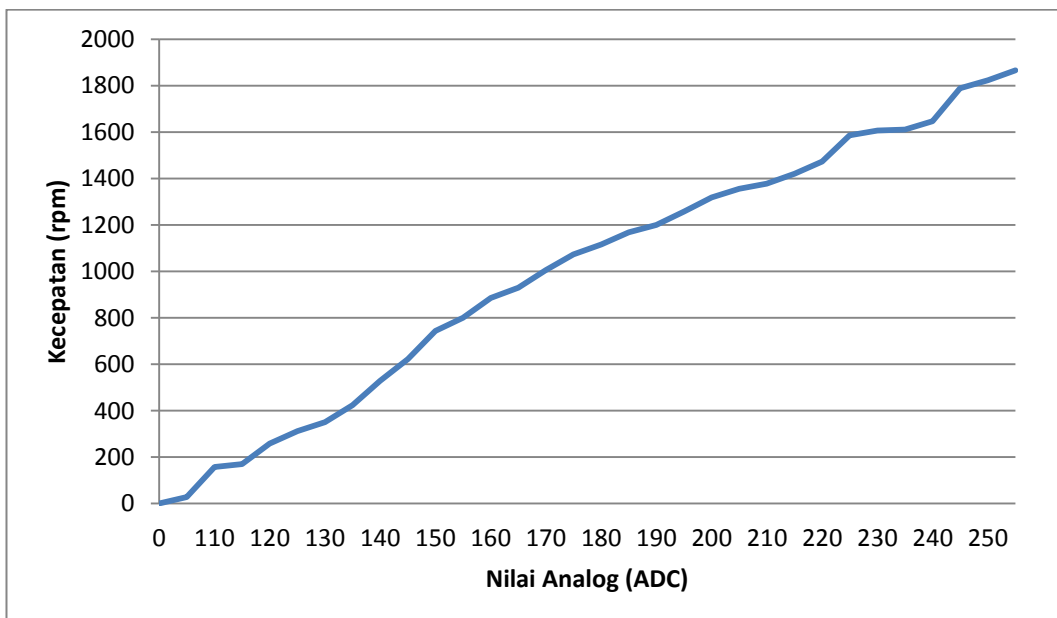
12	160	2.55	2.50	885
13	165	2.95	2.75	929
14	170	2.8	2.76	1006
15	175	3.1	2.98	1073
16	180	3.4	3.21	1115
17	185	3.6	3.40	1168
18	190	3.9	3.71	1200
19	195	4.1	3.9	1258
20	200	4.3	4.15	1319
21	205	4.7	4.36	1355
22	210	4.88	4.57	1377
23	215	5.16	4.69	1420
24	220	5.44	4.69	1473
25	225	5.53	4.69	1586
26	230	5.57	4.69	1606
27	235	5.61	4.69	1611
28	240	5.63	4.69	1647
29	245	5.64	4.69	1788
30	250	5.65	4.69	1823
31	255	5.67	4.69	1866

Dari Tabel 4.8 dapat disimpulkan bahwa semakin besar nilai *dutycycle* yang diberikan maka nilai tegangan *output* yang terukur pada multimeter dan arduino akan semakin bertambah besar. Untuk lebih memahami lihat Gambar 4.33 grafik hubungan antara *tegangan* dengan *dutycycle*.



Gambar 4.33 Grafik hubungan antara tegangan *output* dengan *dutycycle*

Sama halnya dengan kecepatan motor, semakin besar nilai *dutycycle* maka motor akan berputar semakin besar. Untuk lebih memahami lihat Gambar 4.34 grafik hubungan antara *dutycycle* dengan kecepatan motor.



Gambar 4.34 Grafik hubungan antara kecepatan dengan *dutycycle*

Berikut Gambar 4.35 merupakan gambar hasil pengujian kecepatan menggunakan tachometer pada nilai analog (ADC) 150, 200 dan 255.



Gambar 4.35 Hasil Pengujian Kecepatan Motor dengan *dutycycle* 150, 200 dan 255

#### 4.6 Pengaturan Kecepatan Menggunakan Kontroler PID

Kontroler PID terdiri dari parameter P, I, dan D dengan karakteristik yang berbeda-beda. Yang mana ketiga parameter bisa dipakai bersamaan ataupun hanya memakai salah satunya tergantung dari sistem pemakaian yang diinginkan. Berikut ini penjelasan dari kontroler PID:

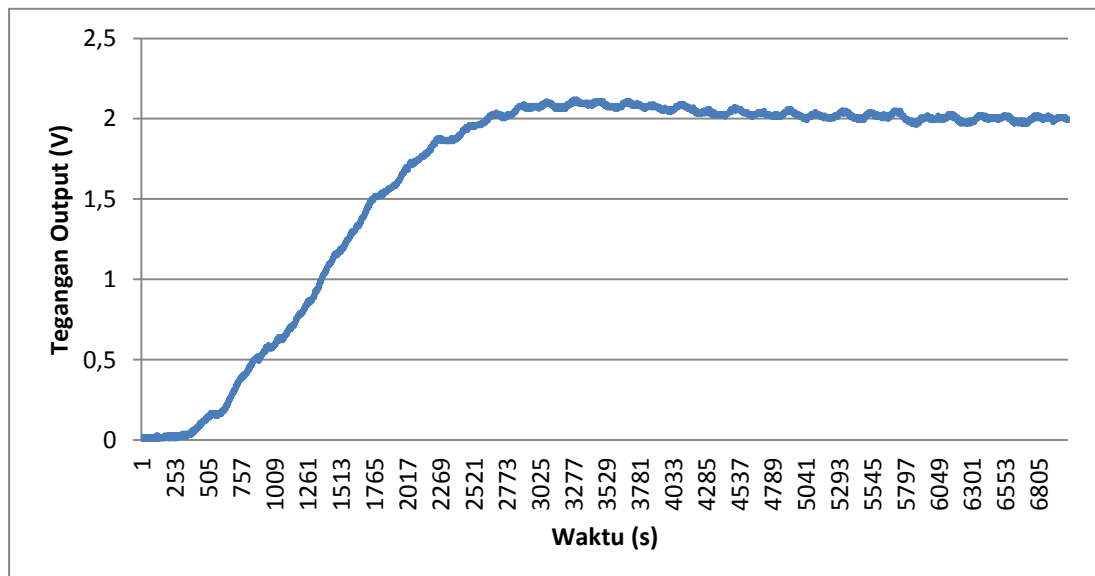
- a. P kontroler merupakan kontroler yang merespon pada nilai *error* yang terjadi pada saat ini. P kontroler berfungsi untuk mempercepat *start* kecepatan motor.
- b. Pi kontroler merupakan PI yang bertanggung jawab pada nilai *error* yang terjadi sebelumnya. PI kontroler berfungsi untuk menjaga kecepatan motor agar tidak terjadi penurunan dan kenaikan kecepatan.
- c. PID kontroler merupakan kontroler yang secara berulang menghitung nilai *error* atau kesalahan yang terjadi antara *setpoint* yang diinginkan dengan



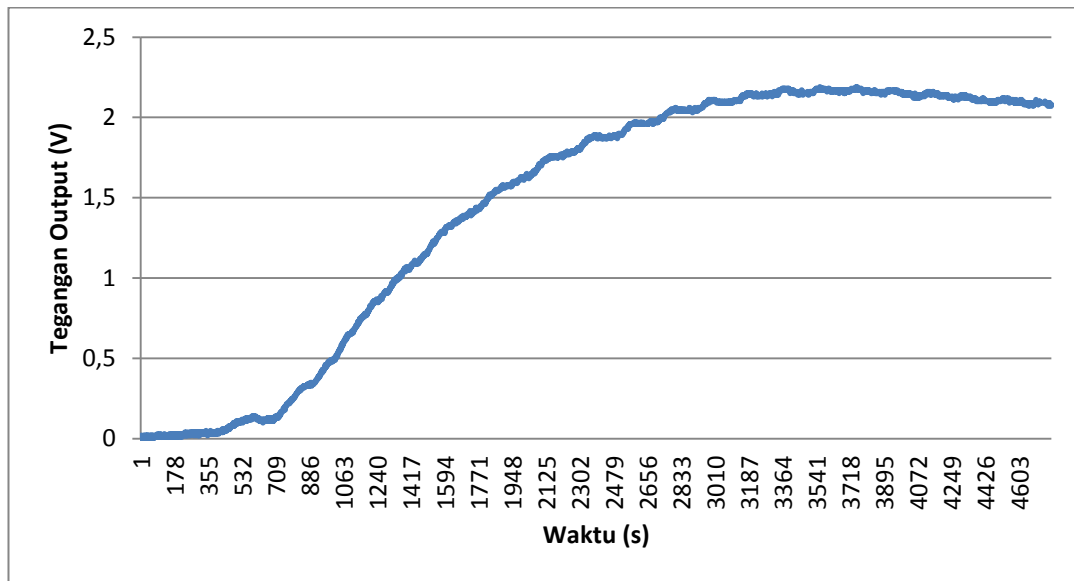
kecepatan yang terukur sekarang. Disini *derivatif* bertanggung jawab untuk nilai *error* yang akan datang.

Mengendalikan kecepatan motor dengan menggunakan teknik PID dengan *feedback* sensor tegangan. Sensor tegangan ini yang mana berfungsi untuk menurunkan tegangan *output* yang dihasilkan oleh tachogenerator. *Setpoint* yang diinginkan berupa tegangan.

Gambar 4.36 dan Gambar 4.37 menampilkan grafik hasil pengujian nilai  $K_p$  pada sistem dengan menggunakan  $K_p$  bernilai 1 dan 2 dengan *setpoint* 2 Volt.



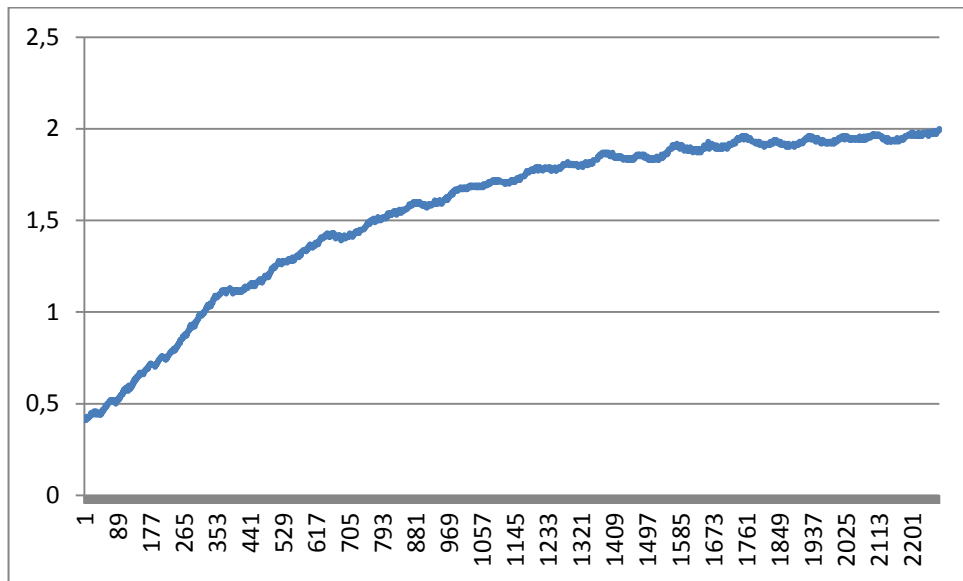
Gambar 4.36 Grafik Respon  $K_p = 1$



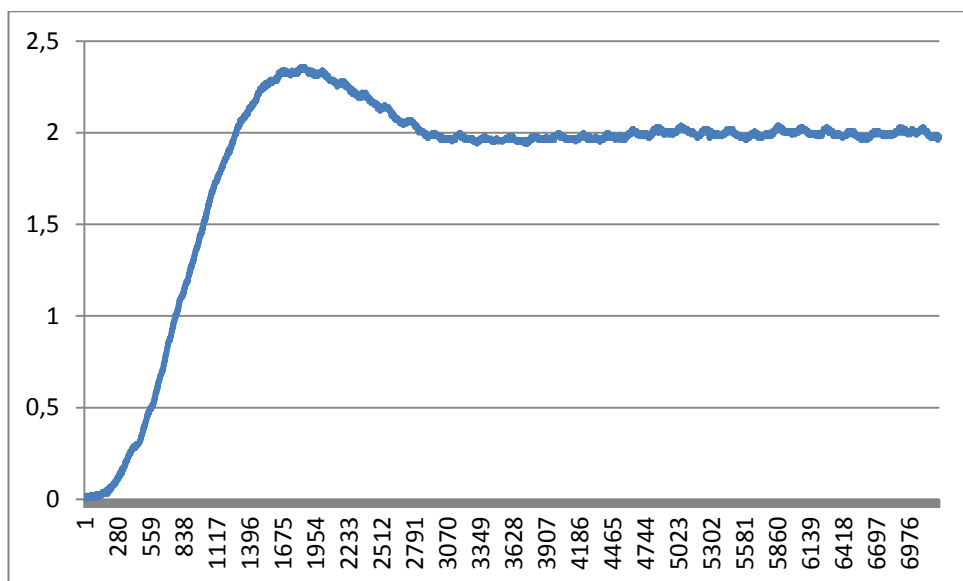
Gambar 4.37 Grafik Respon  $K_p = 2$

Pada Gambar 4.36 dan Gambar 4.37 dapat dilihat perbedaan antara  $K_p$  yang 1 dengan  $K_p$  2. Pada penggunaan  $K_p$  1 motor berputar dengan lambat namun mencapai titik *setpoint*, setelah mengalami sedikit osilasi. Hal ini dikarenakan saat  $K_p$  1 motor berputar lambat membuat gerakan putaran motor yang menghasilkan tegangan mudah terbaca oleh sensor sedangkan saat penggunaan  $K_p$  2 motor berputar dengan cepat sehingga mencapai *setpoint* dengan cepat dan stabil namun sistem bekerja tidak stabil, sensor agak kesulitan membaca tegangan yang dihasilkan oleh kecepatan motor sehingga terjadilah osilasi yang besar. Semakin besar  $K_p$  yang diberikan maka osilasi yang terjadi akan semakin besar. Namun semakin besar nilai  $K_p$  maka respon sistem akan semakin cepat.

Percobaan PI controller dengan menggunakan  $K_p = 1$  dan  $K_i = 0,5$  dengan membandingkan  $K_p = 2$  dan  $K_i = 1$  dengan *setpoint* 2 Volt.



Gambar 4.38 Grafik Respon  $K_p = 1$  dan  $K_i = 0,5$

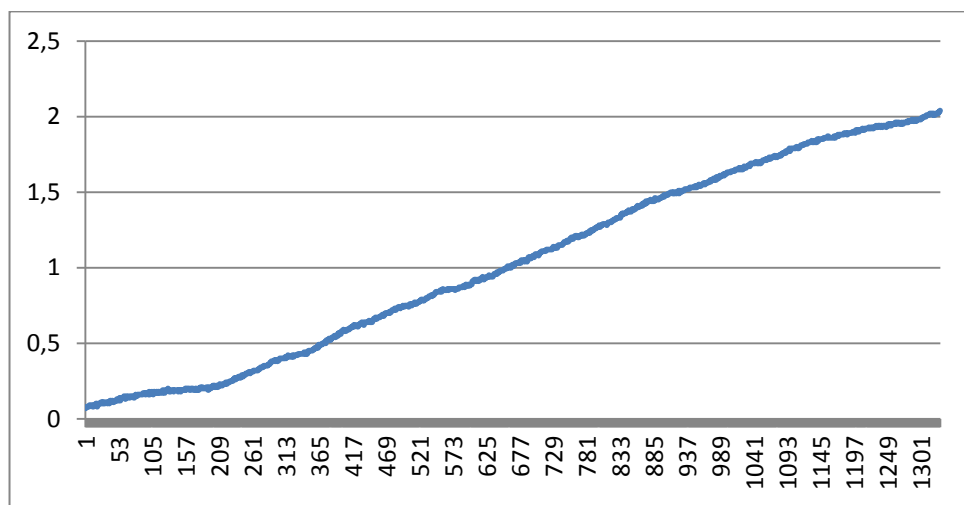


Gambar 4.39 Grafik Respon  $K_p = 2$  dan  $K_i = 1$

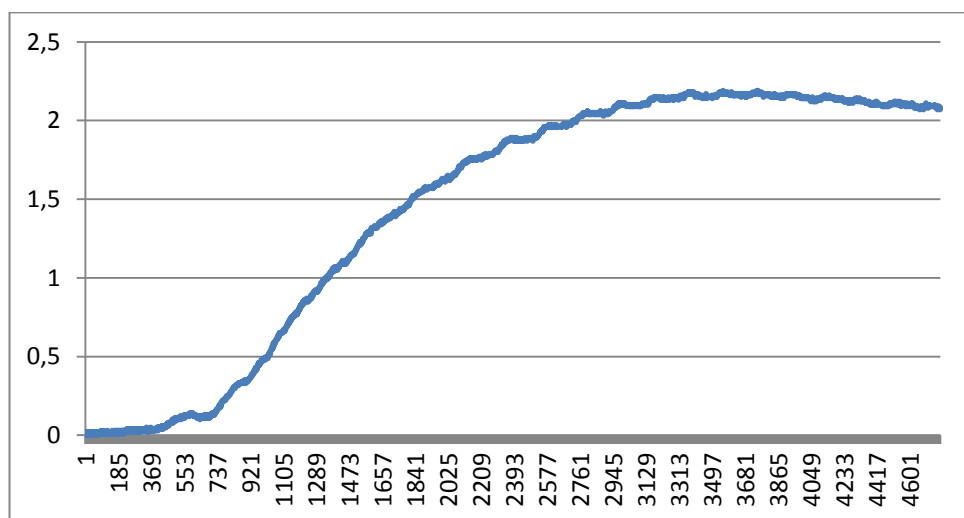
Pada Gambar 4.38 merupakan grafik respon  $K_p = 1$  dan  $K_i = 0,5$ . Pada Gambar 4.38 ini menunjukkan bahwa kecepatan motor berputar lama sehingga tegangan yang dihasilkan bergerak perlahan untuk mencapai *setpoint* dengan terjadi osilasi kecil serta *error* yang dihasilkan kecil. Sedangkan pada Gambar 4.39 menggunakan  $K_p = 2$  dan  $K_i = 1$  menunjukkan motor berputar cepat sehingga mencapai *setpoint* dengan cepat namun *overshoot* yang timbul menjadi berlebihan

dengan osilasi yang sangat besar, membutuhkan waktu yang lama untuk membuat motor berputar stabil. Kesimpulannya bahwa PI merupakan kontroler yang membuat osilasi semakin kecil tergantung sebesar apa nilai  $K_i$  dan  $K_p$  yang diberikan. Semakin besar nilai  $K_i$  yang diberikan maka akan menghilangkan *steady-state error* namun mengakibatkan peningkatan nilai osilasi.

Percobaan PID controller dengan menggunakan  $K_p = 1$ ,  $K_i = 0,5$  dan  $K_d = 0,5$  dengan membandingkan  $K_p = 2$ ,  $K_i = 1$  dan  $K_d = 1$  dengan *setpoint* 2 Volt.



Gambar 4.40 Grafik Respon  $K_p=1$ ,  $K_i = 0,5$  dan  $K_d = 0,5$



Gambar 4.41 Grafik Respon  $K_p=1$ ,  $K_i = 0,5$  dan  $K_d = 0,5$

Pada Gambar 4.40 menampilkan grafik dari penggunaan PID kontroler pada sistem yang digunakan dengan menggunakan Kp bernilai 1 Ki 0,5 dan Kd 0,5 dengan *setpoint* 2 Volt. Dapat dilihat pada Gambar 4.41 menggunakan Kp 2 Ki 1 dan Kd 1. Bentuk grafik waktu yang ditempuh pada Gambar 4.40 lebih cepat dari pada Gambar 4.41 hal ini menyebabkan Gambar 4.39 mencapai *setpoint* lebih cepat semakin mendekati *setpoint* maka kecepatannya semakin stabil. Sama seperti halnya P, PI, dan PID semakin besar nilai Kp, Ki, Kd maka *osilasi* yang dihasilkan akan semakin besar.

## BAB V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil pengujian alat serta analisa pengujian alat maka dapat disimpulkan bahwa:

1. Potensiometer digunakan sebagai pengatur PWM dengan cara memutar potensiometer yang akan mengeluarkan nilai ADC. Hubungan antara tegangan dan nilai hambatan berbanding lurus. Semakin besar nilai ADC maka tegangan dan hambatan yang terukur akan semakin besar pula begitu pula sebaliknya. Nilai ADC maksimal 255 menghasilkan tegangan output sebesar 5V pada arduino dan 5,04V pada multimeter serta nilai tahanan 12 K $\Omega$  sedangkan nilai ADC 26 menghasilkan tegangan 0,51V pada arduino dan 0,53V pada potensiometer dengan nilai tahanan 2,76 K $\Omega$  memiliki kepresisian 92,6% dengan persentase *error* 7,4%.
2. *Absolute encoder* merupakan sensor yang aktif pada kondisi high. Ketika LED menyala mengeluarkan logika 1 dengan tegangan *output* yang terukur 2,03V dan ketika LED padam mengeluarkan logika 0 dengan tegangan *output* 0,04V. *Absolute encoder* memiliki kepresisian 98% dengan persentase *error* 2%
3. *Increment encoder* merupakan sensor yang aktif pada kondisi low. Ketika LED menyala mengeluarkan logika 0 dengan tegangan *output* yang terukur 33mV dan ketika LED padam mengeluarkan logika 1 dengan tegangan *output* yang terukur 5,03 V. *Increment encoder* memiliki kepresisian 98% dengan persentase *error* 2%.
4. Kecepatan motor dapat diatur menggunakan teknik PWM dengan cara mengatur lebar pulsa *high* dan *low*. Hubungan antara nilai PWM dengan kecepatan motor berbanding lurus yang artinya semakin besar nilai PWM yang diberikan maka semakin bertambah pula kecepatan motor. Nilai *duty cycle* 10% menghasilkan tegangan *output* 4,18 V dengan kecepatan 1331

- rpm. Sedangkan *duty cycle* 100% menghasilkan tegangan *output* 10,58 dengan kecepatan 2957 rpm dengan kepresisian 90% dan persentase *error* 10%.
5. Tachogenerator digerakkan oleh motor DC yang akan menghasilkan tegangan dan juga kecepatan. Semakin besar nilai PWM yang diberikan maka tegangan yang terukur dan juga kecepatan yang dihasilkan akan semakin besar pula. Nilai ADC 105 menghasilkan tegangan *output* 0,41 V dengan kecepatan 28 rpm sedangkan nilai ADC 255 menghasilkan tegangan *output* pada arduino dan multimeter sebesar 4,69 dan 5,67 V dengan kecepatan 1866 rpm dengan kepresisian 87% dan persentase *error* 13%.
  6. PID merupakan kontroller yang memiliki karakteristik tidak punya *overshoot*, *rise time* cepat dan tidak punya *steady state error*. Semakin besar nilai  $K_p$ ,  $K_i$ ,  $K_d$  maka osilasi yang dihasilkan akan semakin besar. Nilai  $K_p = 1$   $K_i = 0,5$  dan  $K_d = 0,5$  merupakan nilai PID yang paling baik dengan nilai *error* mencapai 0 dengan persentase *error* 0,4%.

## 5.2 Saran

*Trainer kit* industrial motor servo ini diharapkan kedepannya lebih dikembangkan dengan banyak menambah lebih banyak sensor-sensor untuk mengendalikan kecepatan dan posisi motor servo. Untuk meningkatkan kepresisian kecepatan motor dianjurkan untuk menghubungkan motor dengan tachogenerator menggunakan kopling hal ini menyebabkan motor berputar stabil karena terjadi kontak langsung antara motor dengan tachogenerator sehingga tidak stabil yang akan menghasilkan nilai *error* yang besar. Selain itu, diharapkan kedepannya menggunakan metode PID untuk mengatur posisi motor servo karena *trainer kit* ini hanya bisa mengendalikan kecepatan motor servo menggunakan kontrol PID (*Proportional Integral Derivative*).

## DAFTAR PUSTAKA

- [1] H. P. Putra, "Perancangan Simulasi Alat Bantu PEmindah Barang pada Industri Manufaktur dengan Robot Lengan," 21 Desember 2013. [Online]. Available: <http://eprints.ums.ac.id/27145/>. [Diakses 2 Mei 2019].
- [2] Sujarwata, "Pengendali Motor Servo Berbasis Mikrokontroler Basic Stamp 2SX Untuk Mengembangkan Sistem Robotika," *Angkasa*, vol. 5, no. 1, p. 47, 213.
- [3] W. J. dkk., *Robotika: Teori dan Aplikasi*, Pustakawan Nasional: Katalog Dalam Terbita, 2012.
- [4] T. U. Kalsum, D. A. Trianggana dan Hermawansyah, "Robot Pendeteksi Api Menggunakan Bahasa Pemrograman Basic STAMP," *Jurnal Media Infotama*, vol. 9, no. 1, p. 120, 2013.
- [5] A. Asmara, "Robotika," [Online]. Available: <http://staffnew.uny.ac.id/upload/11510860908616/pendidikan/pengantar-robotika.pdf>. [Diakses 29 April 2019].
- [6] P. A. Nalwan, "AN-0012 Jenis-Jenis Motor," 1 Januari 2007. [Online]. Available: <http://www.robotindonesia.com/article/an0012.pdf>. [Diakses 30 April 2019].
- [7] S. Syamsuddin, R. NAzir dan S. Saputra, "Pengontrolan (Posisi) Motor Servo AC dengan metode Pengaturan "VOLT/HERTZ"," vol. 2, no. 27, p. 52, 2007.
- [8] E. Satria, *Modul Elektronika dan Mekatronika Motor Servo*, Direktorat Pembinaan Sekolah Menengah Kejuruan, 2017.
- [9] Syahrul, "Karakteristik dan Pengontrolan Servomotor," *Majalah Ilmiah UNIKOM Vol. 8 No.2*, p. 143, 2011.
- [10] Jordandee, "Pulse Width Modulation," Learn Sparkfun, [Online]. Available: <https://learn.sparkfun.com/tutorials/pulse-width-modulation/all> 8-4-2019. [Diakses 8 April 2019].
- [11] S. D. Ermansyah, "Implementasi System Voice Recognition Dan Rotary



Encoder pada Mobile Robot Sebagai Sistem Navigasi dan Perhitungan Posisi Robot,” UNIVERSITAS JEMBER, 2016. [Online]. Available: [http://repository.unej.ac.id/bitstream/handle/123456789/78594/Septian%20Dwi%20Ermansyah%20-%20111910201099\\_.pdf?sequence=1](http://repository.unej.ac.id/bitstream/handle/123456789/78594/Septian%20Dwi%20Ermansyah%20-%20111910201099_.pdf?sequence=1). [Diakses 21 April 2019].

- [12] Utami dan M. Kharisma, “Penggunaan Potensiometer Sebagai Sensor Posisi Pada Lengan Robot Berjari Pengikut Gerak Lengan Manusia Berbasis Mikrokontroler,” 21 Januari 2016. [Online]. Available: <http://eprints.polsri.ac.id/171/3/BAB%20II.pdf>. [Diakses 24 April 2019].
- [13] PLCDROID, “Pengertian dan Cara Kerja Potensionmeter Beserta Fungsinya,” 17 Maret 2019. [Online]. Available: <https://www.plcdroid.com/2019/03/potensiometer.html?m=1>. [Diakses 27 Agustus 2019].
- [14] I. m. A. Niratha, Sultan, A. B. Muljono dan I. M. B. Suksmadana, “Rancang Bangun Interface untuk Visuaisasi Kecepatan dan Suhu Motor DC Shunt Berbeban Sebagai Modul Praktikum di Laboratorium Sistem Tenaga,” *Dielekrika*, vol. 1, no. 2, p. 122, 2014.
- [15] J. Handy Wicaksono, “Kontrol PID Untuk Pengaturan Kecepatan Motor DC,” *Tesis Elektro*, vol. 4, no. 1, pp. 10-17, 2004.
- [16] F. Arifin, “PID Controller,” [Online]. Available: <http://staffnew.uny.ac.id/upload/132206815/pendidikan/pid-controller.pdf>. [Diakses 08 Agustus 2019].

**LAMPIRAN 1**  
**( Daftar Riwayat Hidup )**

## DAFTAR RIWAYAT HIDUP



### 1. Data Pribadi

Nama Lengkap : Krisna Monica  
M Dedy Deva Sanjaya  
Tempat & Tanggal Lahir : Sungailiat, 09 Maret 1998  
Alamat Rumah : Jln. Hos Cokro Aminoto No.101  
Hp : +62 852 6020 6807  
Email : [krisnamonica93@gmail.com](mailto:krisnamonica93@gmail.com)  
Jenis Kelamin : Perempuan  
Agama : Islam

### 2. Riwayat Pendidikan

SD Negeri 5 Sungailiat	Lulus 2010
SMP Negeri 1 Sungailiat	Lulus 2014
SMA Negeri 1 Sungailiat	Lulus 2016
DIII Politeknik Manufaktur Negeri Bangka Belitung	Lulus 2019

### 3. Pendidikan Non Formal

Sungailiat, 12 September 2019

Krisna Monica

## DAFTAR RIWAYAT HIDUP



### 1. Data Pribadi

Nama Lengkap : Muhammad Dedy Deva Sanjaya  
Hata Babelin

Tempat & Tanggal Lahir : Manggar, 25 April 1998

Alamat Rumah : Jln. K.H. Abdul Hamid Masjid Jamik  
Pangkalpinang  
Hp : +6281250728168  
Email : [mdedydeva@gmail.com](mailto:mdedydeva@gmail.com)

Jenis Kelamin : Laki-laki

Agama : Islam

### 2. Riwayat Pendidikan

TK Kasih Ayu Kelapa	Lulus 2003
SD Negeri 1 Kelapa	Lulus 2010
SMP Negeri 1 Kelapa	Lulus 2014
SMK Negeri 2 Pangkalpinang	Lulus 2016
DIII Politeknik Manufaktur Negeri Bangka Belitung	Lulus 2019

### 3. Pendidikan Non Formal

Sungailiat, 12 September 2019

M.Dedy Deva Sanjaya Hata Babelin

**LAMPIRAN 2**  
**(Modul Praktikum)**



**MODUL PRAKTIKUM**  
**ROBOTIKA DAN MESIN OTOMATIS 2**  
**(RMO 2)**

**PROGRAM STUDI TEKNIK ELEKTRONIKA**  
**POLITEKNIK MANUFAKTUR NEGERI BANGKA**  
**BELITUNG**

**Jalan Timah Raya, Kawasan Industri Air Kantung, Sungailiat-Bangka 33211**

**Telepon (0717) 93586, 95252, Faksimili (0717) 93585**

**Email: [polman@polman-babel.ac.id](mailto:polman@polman-babel.ac.id)**

**Website: <http://www.polman-babel.ac.id>**



## KATA PENGHANTAR

Segala puji syukur kami panjatkan atas kehadiran Allah SWT yang telah memberikan rahmat serta karunianya kepada kami sehingga “Modul Praktikum Robotika dan Mesin Otomatis 2 (RMO 2)” ini dapat terselesaikan. Pembuatan modul ini dikhususkan sebagai sarana ajar bagi mahasiswa.

Modul ini disusun untuk memberikan pengetahuan dasar serta panduan dalam melaksanakan praktikum Robotika dan Mesin Otomatis 2. Dalam modul ini akan membahas tentang motor servo pengaturan posisi dan kecepatannya. Kami berharap modul ini dapat membimbing mahasiswa dalam menyelesaikan praktikum.

Untuk menambah pengetahuan mahasiswa disarankan untuk dapat mencari pengetahuan berhubungan dengan motor servo pada internet dan buku literatur lainnya. Tanyakan apabila ada yang kurang dimengerti kepada instruktur. Dan berdiskusilah pada teman-teman dalam kelas anda untuk menambah pemahaman anda,

^Akhir kata kami ucapkan terima kasih kepada semua pihak yang turut membantu dalam penyusunan modul ini. Saran dan kritikan senantiasa kami harapkan demi kesempurnaan modul ini di masa mendatang.

Sungailiat, 19 Juni 2019

Penulis



## PRAKTIKUM 1

### “Sensor Posisi Menggunakan Potensiometer ”

**JUDUL :** Karakteristik Potensiometer sebagai Sensor Posisi

**TUJUAN :**

- a. Mahasiswa dapat mengetahui karakteristik potensiometer sebagai sensor posisi
- b. Mahasiswa dapat mengkalibrasi potensiometer melalui program arduino

**PENDAHULUAN :**

Potensiometer menghasilkan nilai ADC (*Analog to Digital Converter*) yang berfungsi sebagai pengatur PWM analog. Tegangan yang masuk ke potensiometer sebesar 5V dan 0V. Ketika potensiometer diputar searah jarum jam maka nilai ADC yang dihasilkan akan semakin bertambah begitu pula sebaliknya. Hal ini berlaku juga terhadap nilai tahanan dan tegangan dari potensiometer. Output dari potensiometer akan dihubungkan pada pin Analog pada Arduino. Sistem perkabelan potensiometer dapat dilihat pada Gambar 1.1

**PERALATAN :**

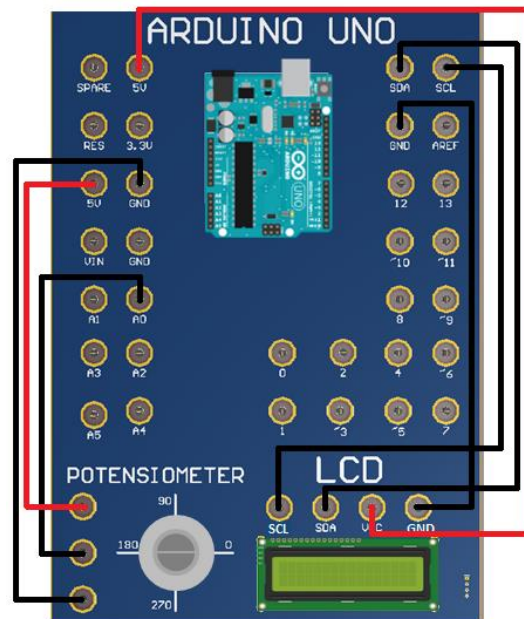
- a. Arduino Uno Board
- b. Potensiometer
- c. Multitester
- d. Kabel
- e. PC

**GAMBAR PERCOBAAN:**





Gambar 1.1 Koneksi Perkabelan Potensiometer

**PROSEDUR PRAKTIKUM:**

1. Rangkailah rangkaian percobaan seperti gambar 1.1.
2. Hubungkan input potensiometer (Pin atas dan bawah) pada pin arduino 5V dan GND.
3. Hubungkan output potensimeter (Pin tengah) pada pin arduino Analog IN (A0).
4. Ukur nilai tahanan dan tegangan output dengan menggunakan multimeter.
5. Buatlah program pada arduino untuk menampilkan nilai analog yang terukur beserta nilai tegangan.
6. Bandingkan apakah nilai yang ditampilkan pada arduino sama dengan nilai yang terukur pada multimeter.
7. Gambar grafik perubahan sudut potensiometer terhadap tegangan outputnya.
8. Gambarkan juga grafik perubahan sudut terhadap besar nilai tahanan yang terukur.

**PROGRAM**

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

int Potensio = A0; // deklarasi pin yang digunakan
int Nilai_ADC;
void setup()
{
  Serial.begin(9600); //serial monitor
  lcd.init();// inisialisasi LCD
}
```

**PROGRAM**

```
lcd.init();
}
void loop()
{
  Nilai_ADC = analogRead(potensio);

  //menampilkan nilai pada LCD
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Nilai ADC = ");
  lcd.setCursor (13,0);
  lcd.print(nilai_ADC);
}
```

**TUGAS**

1. Buatlah program pada arduino untuk menampilkan nilai analog yang terukur beserta nilai tegangannya.

**HASIL PRAKTIKUM:**

No	Nilai ADC	Tahanan ( K $\Omega$ )	Tegangan (V)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			



## PRAKTIKUM 2

### “Sensor Posisi Menggunakan Increment Encoder”

**JUDUL :** Karakteristik Increment Encoder sebagai Sensor Posisi

**TUJUAN :**

- a. Mahasiswa dapat mengetahui karakteristik increment sebagai sensor posisi

**PENDAHULUAN :**

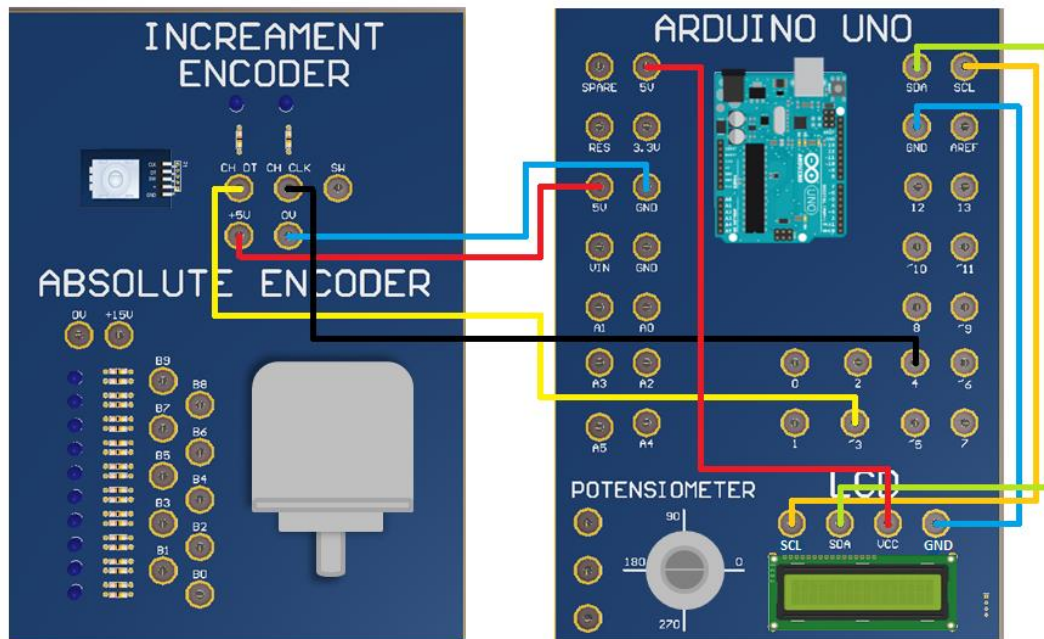
Increment encoder merupakan sensor yang dapat mendeteksi arah putaran. Increment encoder memiliki dua buah sensor yaitu Channel A (CH-DT) dan Channel B (CH-CLK). Yang mana kedua sensor itu menghasilkan output berupa kode biner 2 bit. Kedua channel ini berfungsi untuk mendeteksi arah putaran (CW atau CCW) dengan cara menghubungkan pada pin-pin Arduino Increment encoder memiliki 4 *sharf*, yang mana akan terjadi pengulangan ketika diputar searah atau berlawanan arah jarum jam. Tegangan yang mengaktifkan sensor increment encoder ini ialah 5V. Penggunaan LED pada trainer kit ini sebagai indikator output yang dikeluarkan oleh channel A dan B. Increment encoder merupakan sensor yang aktif pada kondisi LOW. Artinya ketika LED menyala maka berada pada kondisi LOW sedangkan ketika LED padam maka berada pada kondisi HIGH. Selain itu, output dari channel A dan B juga dapat dihubungkan pada pin-pin Digital Arduino. Untuk lebih jelasnya lakukanlah percobaan dibawah ini. Sistem perkabelan dari increment encoder dapat dilihat pada Gambar 2.1.

**PERALATAN :**

- a. Arduino Uno Board
- b. Rotary Encoder KY-040
- c. Kabel
- d. PC
- e. LCD



### GAMBAR PERCOBAAN:



Gambar 2.1 Rangkaian Increment Encoder

### PROSEDUR PRAKTIKUM:

1. Rangkailah rangkaian percobaan diatas.
2. Hubungkan pin +5V dan GND pada Rotary Encoder ke pin arduino 5V dan GND.
3. Hubungkan pin DT dan CLK ke pin Digital arduino.
4. Ukurlah nilai tegangan pada Rotary Encoder.
5. Buatlah program pada arduino untuk menampilkan sudut putaran terhadap arah putaran.
6. Amati Rotary Encoder ketika poros diputar searah jarum jam (CW) atau berlawanan arah jarum jam (CCW) Bandingkan apakah nilai yang ditampilkan pada arduino sama dengan nilai yang terukur pada multimeter.

### PROGRAM

```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd (0x27,16,2);

// deklarasi pin/ variable yang digunakan
int pinA = 2; // terhubung ke DT
int pinB = 3; // terhubung keCLK
int nilai_A; //
int nilai_B; //
```



```
void setup() {
pinMode (pinA,INPUT);
pinMode (pinB,INPUT);
lcd.init();//inisialisasi LCD
lcd.init();
}

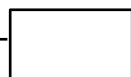
void loop() {
nilai_A = digitalRead(pinA); // baca nilai A
nilai_B = digitalRead(pinB); // baca nilai B
}
// menampilkan nilai pada LCD
lcd.clear();
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("AB: ");
lcd.setCursor(4,0);
lcd.print(nilai_A);
lcd.setCursor(5,0);
lcd.print(nilai_B);
lcd.setCursor(7,0);
}
```

### TUGAS

1. Buatlah program untuk menampilkan output dari Channel DT dan Channel CLK beserta arah putaran dan berapa nilai kali increment encoder diputar.

### HASIL PRAKTIKUM :

Buatlah tabel kebenaran ketika poros diputar CW dan CCW.





## PRAKTIKUM 3

### “Absolute Encoder”

**JUDUL :** Karakteristik Absolute Encoder sebagai sensor posisi

**TUJUAN :**

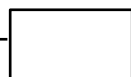
- a. Mahasiswa dapat mengetahui karakteristik Absolute Encoder sebagai sensor posisi
- b. Mahasiswa dapat program untuk menampilkan nilai output absolute encoder berupa kode *gray*

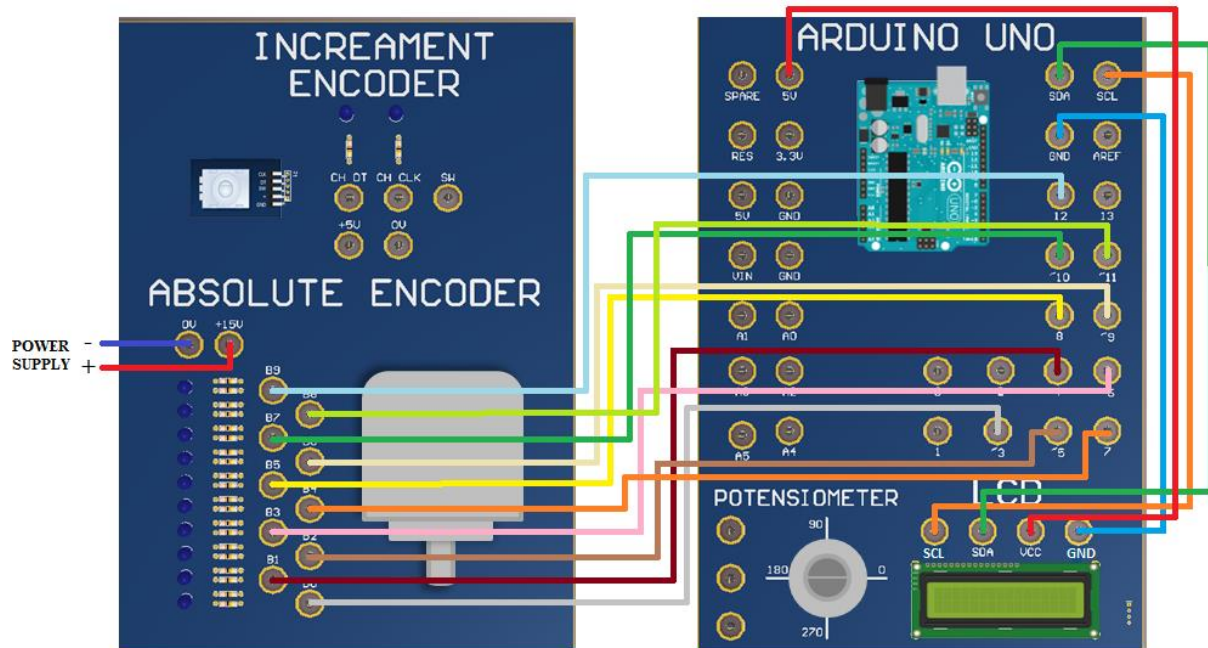
**PENDAHULUAN :**

Absolute encoder memiliki sepuluh sensor yang mana sensor-sensor tersebut akan menghasilkan output berupa kode biner atau gray code. Absolute encoder berbeda dengan increment encoder, kombinasi dari data output yang dikeluarkan absolute encoder menghasilkan output yang sama dan tetap setiap posisi. Absolute encoder memiliki sepuluh output, yang artinya memiliki 1023 kondisi. Penggunaan LED sebagai indikator output dengan detektor cahaya. Absolute encoder merupakan sensor yang aktif pada kondisi HIGH (active HIGH). Artinya ketika LED menyala maka berarti kondisi HIGH sedangkan ketika LED padam artinya berada pada kondisi LOW. Selain itu, output B0-B9 dapat dihubungkan pada pin-pin digital Arduino untuk menampilkan output dari LED berupa kode biner 10-bit pada LCD atau serial monitor. Sistem perkabelan dapat dilihat pada Gambar 3.1

**PERALATAN :**

- a. Arduino Uno Board
- b. Absolute Encoder
- c. Multitester
- d. Kabel
- e. LCD
- f. PC



**GAMBAR PERCOBAAN:**

Gambar 3.1 Koneksi Perkabelan Absolute Encoder

**PROSEDUR PRAKTIKUM:**

1. Rangkailah rangkaian percobaan seperti gambar 3.1.
2. Hubungkan output absolute encoder ke pin Digital arduino.
3. Buatlah program pada arduino untuk menampilkan nilai absolute berupa kode gray atau biner.
4. Tampilkan hasil outputan pada LCD atau serial monitor
5. Amati dan catatlah hasil praktikum ini.
- 6.

**PROGRAM**

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

// deklarasi pin yang digunakan
#define B0 3
#define B1 4
#define B2 5
#define B3 7
#define B4 6
#define B5 8
#define B6 9
#define B7 10
#define B8 13
#define B9 12
```



```
int o0,o1,o2,o3,o4,o5,o6,o7,o8,o9;
void setup()
{
  // jadikan sensor sebagai input
  pinMode (B0,INPUT);pinMode (B1,INPUT);
  pinMode (B2,INPUT);pinMode (B3,INPUT);
  pinMode (B4,INPUT);pinMode (B5,INPUT);
  pinMode (B6,INPUT);pinMode (B7,INPUT);
  pinMode (B8,INPUT);pinMode (B9,INPUT);
  Serial.begin(9600);
  lcd.init();
  lcd.init();
}
void loop() {
  // put your main code here, to run repeatedly:
  o0 = digitalRead(B0); o1 = digitalRead(B1);
  o2 = digitalRead(B2); o3 = digitalRead(B3);
  o4 = digitalRead(B4); o5 = digitalRead(B5);
  o6 = digitalRead(B6);o7 = digitalRead(B7);
  o8 = digitalRead(B8); o9 = digitalRead(B9);

  //menampilkan output dari absolute encoder
  lcd.clear();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print(" Code Gray ");
  lcd.setCursor(0,1);lcd.print(o0);
  lcd.setCursor(1,1);lcd.print(o1);
  lcd.setCursor(2,1);lcd.print(o2);
  lcd.setCursor(3,1);lcd.print(o3);
  lcd.setCursor(4,1);lcd.print(o4);
  lcd.setCursor(5,1);lcd.print(o5);
  lcd.setCursor(6,1);lcd.print(o6);
  lcd.setCursor(7,1);lcd.print(o7);
  lcd.setCursor(8,1);lcd.print(o8);
  lcd.setCursor(9,1);lcd.print(o9);
}
```

**HASIL PRAKTIKUM:**





## PRAKTIKUM 4

### “ Mengatur Kecepatan dengan Menggunakan PWM”

**JUDUL :** Pengaturan Kecepatan dan Arah Putaran Motor

**TUJUAN :**

- a. Mahasiswa dapat mengetahui bagaimana cara mengatur kecepatan motor dengan teknik PWM.
- b. Mahasiswa dapat mengetahui bagaimana cara mengatur arah putaran motor menggunakan driver motor L298N

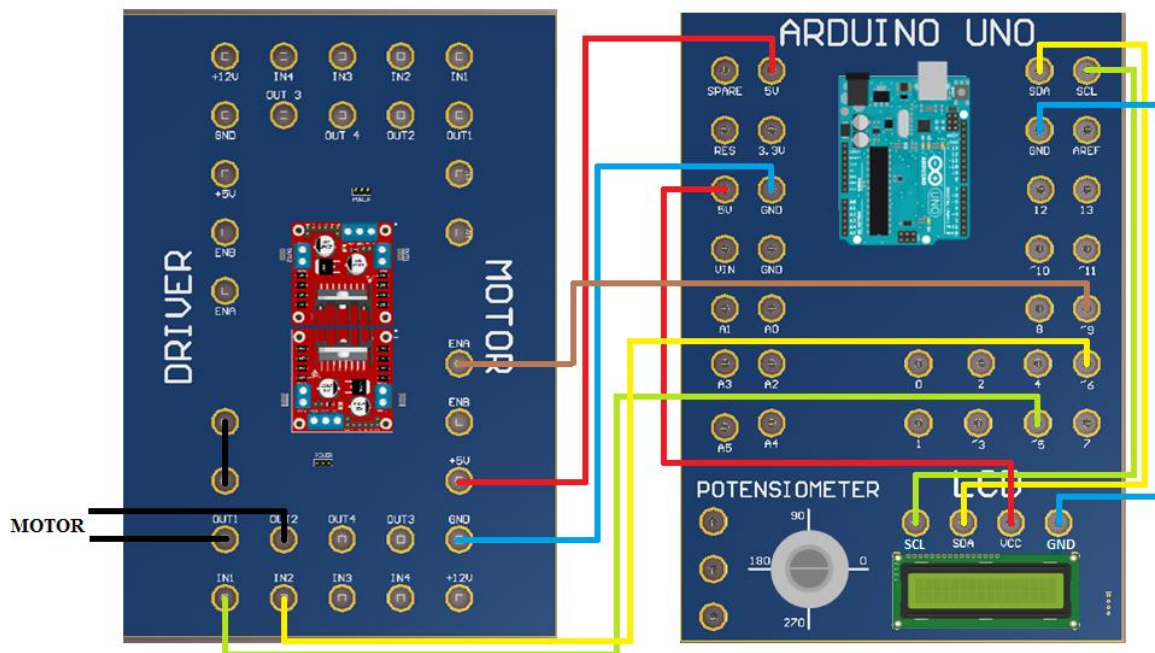
**PENDAHULUAN :**

Kecepatan motor servo dapat diatur menggunakan teknik PWM (Pulse Width Modulation). Yang mana dengan cara mengatur lebar kecilnya persentase dutycycle. Driver motor digunakan selain untuk mengatur kecepatan motor juga dapat mengatur arah putaran clockwise (CW) dan counter clockwise (CCW). Arah putaran motor dapat dikendalikan dengan cara memberi logika HIGH dan LOW pada pin IN1 sampai IN4. Sedangkan untuk mengatur kecepatan motor dengan cara memberikan PWM pada pin ENA atau ENB pada driver motor. Untuk lebih jelasnya lakukanlah percobaan dibawah ini. Sistem perkabelan dapat dilihat pada Gambar 4.1

**PERALATAN :**

- a. Arduino Uno Board
- b. Driver Motor L298N
- c. Motor Servo
- d. Multitester
- e. Tachometer
- f. LCD
- g. Kabel
- h. PC




**GAMBAR PERCOBAAN:**


Gambar 4.1 Koneksi Perkabelan Driver Motor

**PROSEDUR PRAKTIKUM:**

1. Hubungkan +5V dan GND pada Driver Motor L298N ke pin arduino + 5V dan GND. Apabila menggunakan +12V maka lepas jumperan A
2. Hubungkan OUT1 dan OUT2 pada motor servo. Apabila menggunakan dua motor servo pakailah OUT3 dan OUT4.
3. Hubungkan IN1 dan IN2 ke pin arduino pada pin Digital. Apabila menggunakan dua buah motor servo gunakan IN3 dan IN4.
4. Hubungkan ENA ke pin arduino yang memiliki PWM (~) yang berguna sebagai pengatur PWM. Ketika menggunakan dua buah motor servo gunakan ENB
5. Buatlah program pada arduino untuk mengatur kecepatan dan arah putaran motor. Seperti contoh dibawah ini.
6. Ukur nilai tegangan output dengan menggunakan Multitester dan ukur juga nilai kecepatan motor menggunakan Tachometer.
7. Untuk menampilkan hasil rangkailah
8. Amatilah arah putaran motor servo.
9. Gambar grafik hubungan antara tegangan output terhadap kecepatan motor.

**PROGRAM**

```
int IN1=4; //Deklarasi Pin Arduino yang digunakan
int IN2=6;
int ENA=9;

void setup() {
  //atur pin yang digunakan sebagai output
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(ENA,OUTPUT);
}

void loop() {
  digitalWrite(IN1,LOW); // mengubah arah putaran
  digitalWrite(IN2,HIGH);
  analogWrite(ENA,255); // pengatur kecepatan dengan mengubah nilai
  ADC
  delay(1000);
}
```

**TUGAS**

1. Buatlah program motor berputar dengan kecepatan yang dapat diatur melalui PWM sesuai dengan tabel 4.2.
2. Buatlah program motor berputar searah jarum jam (CW) dan berlawanan arah jarum jam (CCW).

**HASIL PRAKTIKUM:**

Tabel 4.1 Tabel Data Hasil Arah Putaran Motor

IN 1	IN 2	Arah Putaran
LOW	LOW	
LOW	HIGH	
HIGH	LOW	
HIGH	HIGH	



Tabel 4.2 Tabel Data Hasil Pengukuran Kecepatan

No	Nilai PWM (%)	Tegangan (V)	Kecepatan ( <i>rpm</i> )
1	0		
2	10		
3	20		
4	30		
5	40		
6	50		
7	60		
8	70		
9	80		
10	90		
11	100		

**ANALISA**

1. Bagaimana hubungan antara PWM dengan kecepatan motor?
2. Bagaimana hubungan antara PWM dengan tegangan output pada motor?



## PRAKTIKUM 5

### “Mengatur Kecepatan Motor Menggunakan Tachogenerator”

**JUDUL :** Karakteristik Tachogenerator sebagai sensor kecepatan

**TUJUAN :**

- a. Mahasiswa dapat mengetahui karakteristik dari tachogenerator dan motor DC sebagai sensor kecepatan.

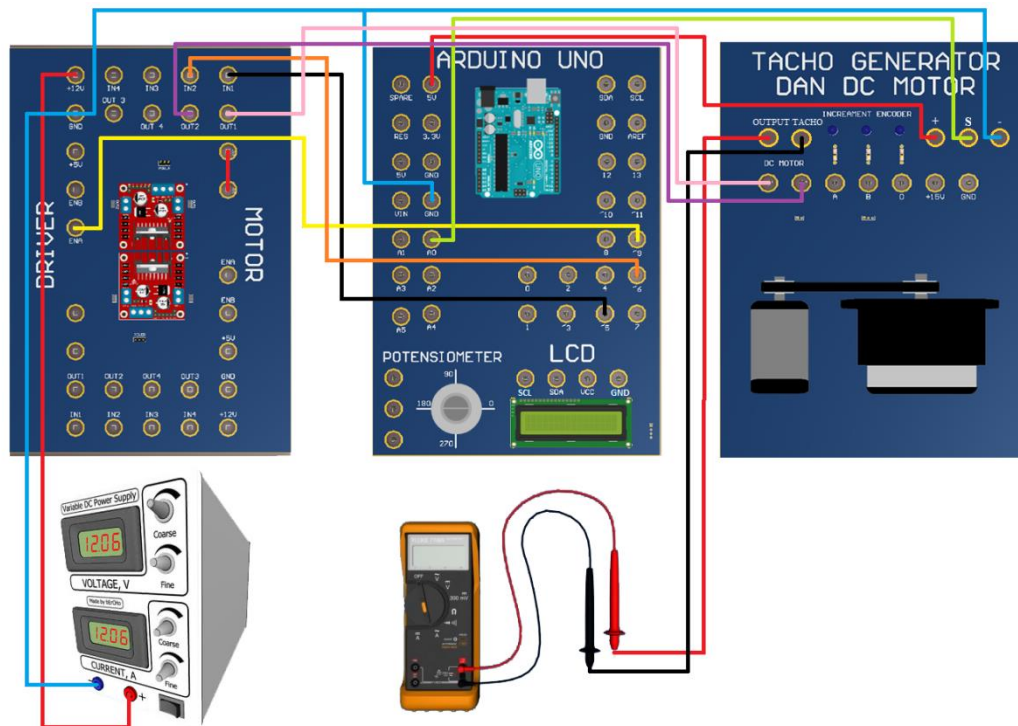
**PENDAHULUAN :**

Tachogenerator dapat menghasilkan energi listrik saat berputar. Perubahan kecepatan akan menghasilkan tegangan output yang berbeda pada tachogenerator. Motor DC menggerakkan Tachogenerator. Tachogenerator dan motor DC dihubungkan oleh elemen transmisi berupa gear dan belt dengan perbandingan 1:1 sehingga menghasilkan putaran yang sama. Untuk melihat sistem perkabelan dari Tachogenerator dan motor DC dapat dilihat pada Gambar 5.1.

**PERALATAN :**

- a. Tachogenerator
- b. Motor DC
- c. Power Supply
- d. Multitester
- e. Kabel
- f. PC




**GAMBAR PERCOBAAN:**


Gambar 5.1 Koneksi Perkabelan Tachogenerator dan Motor DC

**PROSEDUR PRAKTIKUM:**

1. Rangkailah rangkaian percobaan sesuai dengan gambar 5.1.
2. Ukurlah tegangan output yang keluar menggunakan multitester
3. Ukurlah kecepatan motor menggunakan tachometer
4. Catatlah hasil pengukuran dari percobaan tachogenerator dan motor DC ini.
5. Buatlah grafik hubungan antara tegangan input, tegangan output dan kecepatan.



**PROGRAM**

```
int IN1=4; //Deklarasi Pin Arduino yang digunakan
int IN2=6;
int ENA=9;

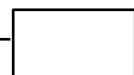
void setup() {
  //atur pin yang digunakan sebagai output
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(ENA,OUTPUT);
}

void loop() {
  digitalWrite(IN1,LOW); // mengubah arah putaran
  digitalWrite(IN2,HIGH);
  analogWrite(ENA,255); // pengatur kecepatan dengan mengubah nilai
  ADC
  delay(1000);
}
```

**HASIL PRAKTIKUM:**

No	Input Duty Cycle/ADC	Tegangan Multitester	Tegangan Arduino	Kecepatan (rpm)
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				

**ANALISA**





## PRAKTIKUM 6

### “Mengatur Kecepatan Motor Dengan PID”

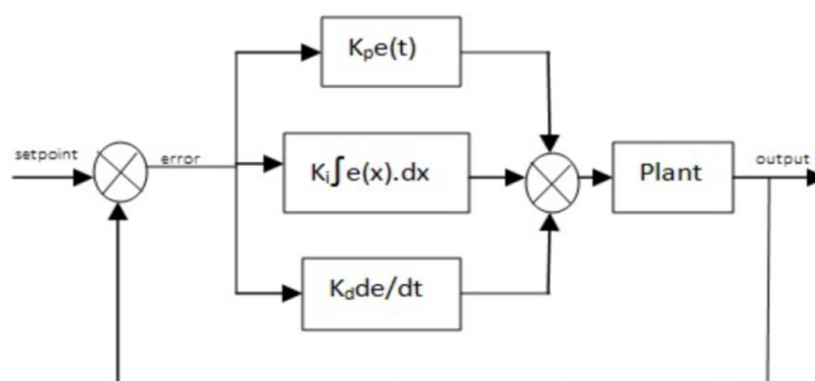
**JUDUL :** Respon PID sebagai pengatur Kecepatan Motor

**TUJUAN :**

- Mahasiswa dapat mengetahui respon PID sebagai pengatur kecepatan motor.

**PENDAHULUAN :**

Kontroler PID (*Proportional-Integral-Derivative controller*) merupakan kontroler dengan umpan balik dengan parameter berupa P, I, D yang berguna untuk memberi respon ke sistem yang diinginkan. Kontroler PID berfungsi untuk meminimalisir *error* dengan cara menghitung nilai *error* yang terjadi secara berulang-ulang.



Gambar 6.1 Blok Diagram PID Controller

**Keterangan :**

$k_p$  = Konstanta Proporsional

$k_i$  = Konstanta Integral

$k_d$  = Konstanta Derivatif

$e(t)$  = Error

Setiap kontroler baik P (Proporsional), I (Integral) maupun D (Derivatif) kontroler memiliki kelebihan dan kekurangan. *Proportional Controller* merupakan perkalian antara konstanta *proportional* dengan *setpoint* yang diinginkan dikurangi dengan *error*. *P Controller* dapat mengurangi *rise time* sehingga *rise time* menjadi cepat, menambah *overshoot* dan mengurangi *steady state*. Sedangkan *Integral Controller* memiliki karakteristik mengurangi



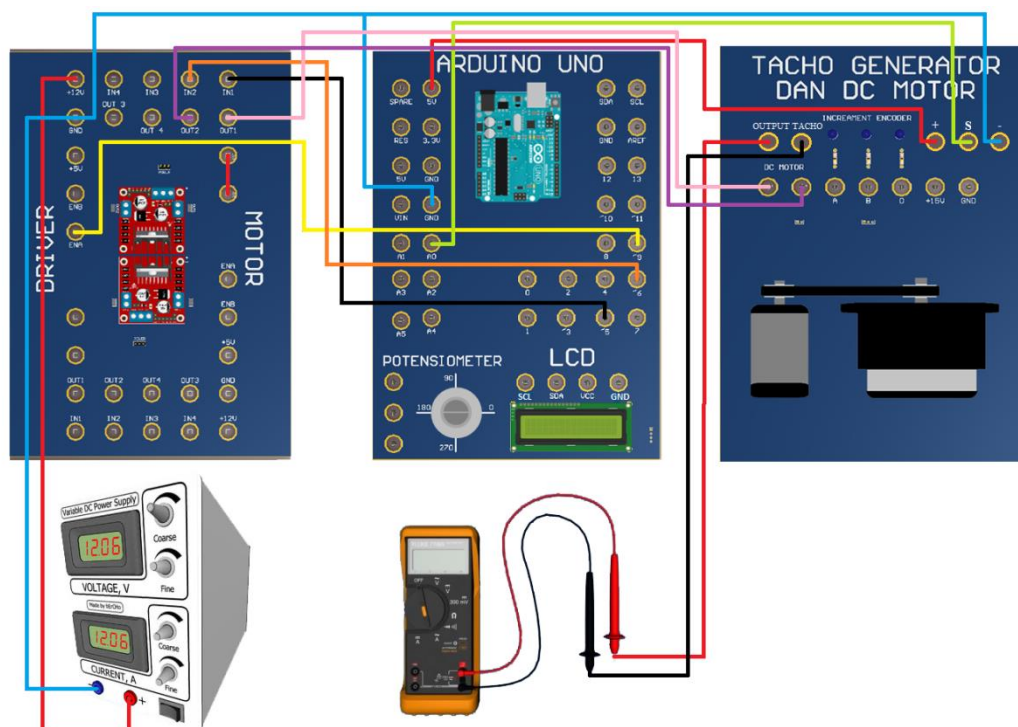


*rise time*, menambah *overshoot* dan *setling time*, serta berbeda dengan *controller* lain, *Integral Controller* dapat menghilangkan *steady state error*. *Derivatif Controller* dapat mengurangi *overshoot* dan *setling time*, tetapi tidak memberikan dampak apapun terhadap *steady state error*.

#### PERALATAN :

- a. Tachometer
- b. Motor DC
- c. Power Supply
- d. Multitester
- e. Kabel
- f. PC

#### GAMBAR PERCOBAAN:



Gambar 5.1 Koneksi Perkabelan Tachogenerator dan Motor DC

**PROSEDUR PRAKTIKUM:**

- 1) Rangkailah rangkaian percobaan sesuai dengan gambar 6.1.
- 2) Ukurlah tegangan output yang keluar menggunakan multimeter
- 3) Ukurlah kecepatan motor menggunakan tachometer
- 4) Catatlah hasil pengukuran dari percobaan ini.
- 5) Rubahlah nilai Kp Ki dan Kd sesuai dengan hasil percobaan yang diminta
- 6) Amati apa yang terjadi pada sistem dan buatlah gambar grafik pada setiap *controller*

**PROGRAM**

```
// _____PID_____

//deklarasi driver motor
#define IN1 5
#define IN2 6
#define ENA 9
float
kesalahan,tegangan,teg,proporsional,integral,derivatif,error,setpoint1;
float dutycycle, respon,pid,pid1;
int pwm, PWM;
float input1;
float kp,ki,kd, p, setpoint,hasil;
//_____sensor tegangan_____
float volt1;//nilai analog
float volt;// tegangan terbaca
void setup()
{
  // motor
  Serial.begin(9600);
  pinMode(IN1,OUTPUT);pinMode(IN2,OUTPUT);pinMode(ENA,OUTPUT);

  kp=1;
  ki=0.5;
  kd=0.5;
}

void loop()
{
  digitalWrite(IN1,HIGH); digitalWrite(IN2,LOW);// menjalankan motor
  float setpoint=0.0; //input set point(volt)
  //=====
  volt1=analogRead(A0); // baca sensor tegangan
  volt=(volt1/1023)*5; // tegangan terbaca

  respon=hitungPID( setpoint,volt);//memanggil fungsi PID
  respon=constrain(respon,0,255);
  analogWrite(ENA, respon);

  Serial.print((String)"DATA,TIME,"); // serial menampilkan waktu
  Serial.println(volt); // serialmenampilkan tegagan yang terbaca
```



```
// _____ //  
}  
float hitungPID(float tegangan, float teg)  
{  
    kesalahan=tegangan-teg;  
    proporsional=kesalahan*kp;  
    integral=integral+(error*ki);  
    derivatif=(error-kesalahan)*kd;  
    integral+=kesalahan;  
    pid=(proporsional+integral+derivatif)/50; // 50 = sistem  
    error=kesalahan;  
    pid1=pid;  
    return (pid);  
}
```

### HASIL PERCOBAAN

- *P controller* dengan nilai  $K_p = 1$   $0.5$  dan  $2$
- *PI controller* dengan  $K_p = 1$   $K_i = 0.5$  dan  $1$
- *PID controller* dengan  $K_p = 1$   $K_i = 0.5$   $K_d = 0.5$
- *PID controller* dengan  $K_p = 1$   $K_i = 1$   $K_d = 0.5$
- *PID controller* dengan  $K_p = 1$   $K_i = 0.5$   $K_d = 1$
- *PID controller* dengan  $K_p = 1$   $K_i = 1$   $K_d = 1$

### ANALISA

Bagaimana respon *P controller* ketika nilai  $K_p$  dinaikkan dan diturunkan?

Bagaimana respon *PI controller* ketika nilai  $K_i$  dinaikkan dan diturunkan?

Bagaimana respon *PID* terhadap sistem?