



**PERHITUNGAN FUNGSI ALIH DAN KONTROL PID MOTOR
PG45 PADA ROBOT DIPO-MECARO (MEDICAL ASSISTANT
ROBOT) DI PT PERMALAT BERDIKARI JAYA**

KERJA PRAKTIK

Ahmad Didik Setiyadi

21060117120024

**TEKNIK ELEKTRO FAKULTAS TEKNIK
UNIVERSITAS DIPONEGORO
SEMARANG**

DESEMBER 2020

HALAMAN PENGESAHAN JURUSAN

**LAPORAN KERJA PRAKTIK
PT. PERMALAT BERDIKARI JAYA SEMARANG
DIVISI MECHANICAL, ELECTRICAL AND PLUMBING (MEP)**

Dengan judul:
**“PERHITUNGAN FUNGSI ALIH DAN KONTROL PID MOTOR PG45
PADA ROBOT DIPO-MECARO (MEDICAL ASSISTANT ROBOT) DI PT
PERMALAT BERDIKARI JAYA”**

Disusun oleh:
**Ahmad Didik Setiyadi
21060117140052**

Universitas Diponegoro Semarang
20 Juli s/d 21 Agustus 2020

Telah disahkan pada tanggal:

.....

Mengetahui,

Ketua Departemen Teknik Elektro
Universitas Diponegoro

Dosen Pembimbing Kerja Praktik

Dr. Wahyudi, S.T., M.T.
NIP 196906121994031001

Hadha Afrisal, S.T., M.Sc.
NIP H.7. 199104172018071002

HALAMAN PENGESAHAN PERUSAHAAN

LAPORAN KERJA PRAKTIK
PT. PERMALAT BERDIKARI JAYA SEMARANG
DIVISI MECHANICAL, ELECTRICAL AND PLUMBING (MEP)

Dengan judul:
**“PERHITUNGAN FUNGSI ALIH DAN KONTROL PID MOTOR PG45
PADA ROBOT DIPO-MECARO (MEDICAL ASSISTANT ROBOT) DI PT
PERMALAT BERDIKARI JAYA”**

Disusun oleh:
Ahmad Didik Setiyadi
21060117120024

Universitas Diponegoro Semarang
20 Juli s/d 21 Agustus 2020

Telah disahkan pada tanggal:

.....

Mengetahui,

Direktur

Pembimbing Lapangan
Tenaga Ahli MEP

Siswo Herlaut, ST, IAI

Dr. Eng. Munadi, ST., MT

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan kerja praktik yang berjudul **“PERHITUNGAN FUNGSI ALIH DAN KONTROL PID MOTOR PG45 PADA ROBOT DIPO-MECARO (MEDICAL ASSISTANT ROBOT) DI PT PERMALAT BERDIKARI JAYA”**

.Kerja praktik merupakan kegiatan yang harus dilaksanakan oleh mahasiswa sebagai persyaratan untuk menyelesaikan studi di Departemen Teknik Elektro Fakultas Teknik Universitas Diponegoro, Semarang. Hal ini dianggap penting sebagai pengembangan pengetahuan mahasiswa dan mempersiapkan mahasiswa sebelum terjun ke dunia profesi.

Kerja praktik merupakan kegiatan yang harus dilaksanakan oleh mahasiswa sebagai salah satu persyaratan untuk menyelesaikan studi di Departemen Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang. Hal ini dianggap penting sebagai pengembangan pengetahuan mahasiswa guna mempersiapkan mahasiswa sebelum terjun ke dunia profesi.

Selama melakukan kerja praktik di PT. Permalat Berdikari Jaya, penulis mendapatkan tambahan ilmu pengetahuan. Sesuai dengan tujuannya bahwa selama kerja praktik, mahasiswa diharapkan dapat menerapkan dan memahami hal-hal teknis di bidang teknologi kelistrikan.

Pelaksanaan kerja praktik ini dapat berjalan dengan baik berkat bantuan yang telah diberikan oleh berbagai pihak. Pada kesempatan ini penulis ingin menyampaikan ucapan terimakasih kepada:

1. Bapak Dr. Wahyudi, S.T., M.T. selaku Ketua Departemen Teknik Elektro Fakultas Teknik Universitas Diponegoro.
2. Bapak Yuli Christyono, S.T., M.T. selaku Ketua Program Studi S-1 Teknik Elektro Fakultas Teknik Universitas Diponegoro.
3. Bapak Hadha Afrisal, S.T., M.Sc. selaku Dosen Pembimbing Kerja Praktik.
4. Bapak Dr. Eng. Munadi, S.T., M.T. selaku pembimbing lapangan selama Kerja Praktik.
5. Mas Abdul Ghoni, S.T. selaku ketua proyek selama Kerja Praktik.
6. Teman-teman S-1 Teknik Elektro Universitas Diponegoro angkatan 2017 yang selalu memberi dukungan.
7. Semua pihak yang telah membantu dalam penyusunan laporan yang tidak dapat disebutkan satu persatu.

Semoga laporan ini dapat bermanfaat bagi pembaca. Penulis menyadari

bahwa dalam penulisan laporan kerja praktik ini masih terdapat banyak kekurangan, untuk itu penulis mengharapkan masukan serta kritik yang membangun yang dapat membantu untuk penyusunan yang lebih baik lagi.

Akhir kata, semoga laporan kerja praktik ini dapat memberikan manfaat bagi semua pihak.

Semarang, 20 November 2020

A handwritten signature in black ink, appearing to be 'DDR' with a stylized flourish at the end.

Ahmad Didik Setiyadi

ABSTRAK

Fungsi alih dari suatu komponen merupakan perbandingan antara output transformasi laplace dengan input transformasi laplace dengan kondisi awal sama dengan nol. Dengan mengetahui fungsi alih suatu komponen kita dapat memprediksi keluaran dari komponen tersebut terhadap masukan yang diberikan. Begitu juga dengan fungsi alih motor DC, dengan mengetahui fungsi alih motor DC maka kita bisa mengetahui keluaran motor DC tersebut terhadap masukan yang kita berikan. Dipo-Mecaro (Medical Assistant Robot) merupakan sebuah omni-wheel mobile robot dengan empat motor PG45 yang dapat dioperasikan secara nirkabel melalui koneksi Wi-Fi sehingga dapat mengurangi kontak antara pasien Covid-19 dan tenaga medis. Hal ini dianggap sebagai solusi dari tingginya resiko yang dihadapi oleh para tenaga medis saat pandemi Covid-19 berlangsung karena harus berada didalam ruangan atau kontak langsung dengan pasien Covid-19. Perusahaan yang membuat motor PG45 yang digunakan Dipo-Mecaro tidak memberikan spesifikasi secara rinci, sehingga diperlukan pencarian fungsi alih yang berguna untuk digunakan perhitungan metode kontrol berikutnya. Pencarian fungsi alih dilakukan menggunakan masukan sinyal PWM acak dan keluaran kecepatan motor yang kemudian diolah dengan tool sistem identifikasi MATLAB. Hasil perhitungan yang dilakukan menghasilkan fungsi alih motor PG45 $\frac{45.71S + 0.6757}{S^3 + 14.15S^2 + 24.59S + 10.93}$ dengan *best fit* 97.54% dan sistem teruji stabil. Perhitungan parameter Kontrol PID dihasilkan $K_P = 32.9279740191956$, $K_I = 196.610205687394$, dan $K_D = 1.20746780266902$.

Kata Kunci: Fungsi Alih, Kontrol PID, Motor DC PG45

ABSTRACT

The transfer function of a component is the ratio between the laplace transform output and the laplace transform input with the initial condition equal to zero. By knowing the transfer function of a component, we can predict the output of that component against the given input. Likewise with the DC motor transfer function, by knowing the DC motor transfer function, we can find out the DC motor output to the input we provide. Dipo-Mecaro (Medical Assistant Robot) is an omni-wheel mobile robot with four PG45 motors that can be operated wirelessly via a Wi-Fi connection so as to reduce contact between Covid-19 patients and medical personnel. This is considered a solution to the high risk faced by medical personnel during the Covid-19 pandemic because they have to be in the room or have direct contact with Covid-19 patients. The company that made the PG45 motorbike used by Dipo-Mecaro did not provide detailed specifications, so it was necessary to find a transfer function that was useful for calculating the next control method. The search for the transfer function is carried out using random PWM signal input and motor speed output which are then processed using the MATLAB identification system tool. The results of calculations performed PG45 motor transfer function is $\frac{45.71S + 0.6757}{S^3 + 14.15S^2 + 24.59S + 10.93}$ with best fit 97.54% and system is stable. The result of PID control parameters calculation performed $K_P = 32.9279740191956$, $K_I = 196.610205687394$, dan $K_D = 1.20746780266902$.

Key Word: *Transfer Function, PID Control, Motor DC PG45*

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan kerja praktik yang berjudul **“PERHITUNGAN FUNGSI ALIH DAN KONTROL PID MOTOR PG45 PADA ROBOT DIPO-MECARO (MEDICAL ASSISTANT ROBOT) DI PT PERMALAT BERDIKARI JAYA”**

.Kerja praktik merupakan kegiatan yang harus dilaksanakan oleh mahasiswa sebagai persyaratan untuk menyelesaikan studi di Departemen Teknik Elektro Fakultas Teknik Universitas Diponegoro, Semarang. Hal ini dianggap penting sebagai pengembangan pengetahuan mahasiswa dan mempersiapkan mahasiswa sebelum terjun ke dunia profesi.

Kerja praktik merupakan kegiatan yang harus dilaksanakan oleh mahasiswa sebagai salah satu persyaratan untuk menyelesaikan studi di Departemen Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang. Hal ini dianggap penting sebagai pengembangan pengetahuan mahasiswa guna mempersiapkan mahasiswa sebelum terjun ke dunia profesi.

Selama melakukan kerja praktik di PT. Permalat Berdikari Jaya, penulis mendapatkan tambahan ilmu pengetahuan. Sesuai dengan tujuannya bahwa selama kerja praktik, mahasiswa diharapkan dapat menerapkan dan memahami hal-hal teknis di bidang teknologi kelistrikan.

Pelaksanaan kerja praktik ini dapat berjalan dengan baik berkat bantuan yang telah diberikan oleh berbagai pihak. Pada kesempatan ini penulis ingin menyampaikan ucapan terimakasih kepada:

8. Bapak Dr. Wahyudi, S.T., M.T. selaku Ketua Departemen Teknik Elektro Fakultas Teknik Universitas Diponegoro.
9. Bapak Yuli Christyono, S.T., M.T. selaku Ketua Program Studi S-1 Teknik Elektro Fakultas Teknik Universitas Diponegoro.
10. Bapak Hadha Afrisal, S.T., M.Sc selaku Dosen Pembimbing Kerja Praktik.
11. Bapak Dr. Eng. Munadi, S.T., M.T. selaku pembimbing lapangan selama Kerja Praktik.

12. Mas Abdul Ghoni, S.T. selaku ketua proyek selama Kerja Praktik.
13. Teman-teman S-1 Teknik Elektro Universitas Diponegoro angkatan 2017 yang selalu memberi dukungan.
14. Teman-teman Kerja Praktik di PT. Permalat Berdikari Jaya atas nama Aan Aria Nanda dan Vinsensius Hans Prasetya yang membantu selama Kerja Praktik.
15. Semua pihak yang telah membantu dalam penyusunan laporan yang tidak dapat disebutkan satu persatu.

Semoga laporan ini dapat bermanfaat bagi pembaca. Penulis menyadari bahwa dalam penulisan laporan kerja praktik ini masih terdapat banyak kekurangan, untuk itu penulis mengharapkan masukan serta kritik yang membangun yang dapat membantu untuk penyusunan yang lebih baik lagi.

Akhir kata, semoga laporan kerja praktik ini dapat memberikan manfaat bagi semua pihak.

Semarang, 20 November 2020



Ahmad Didik Setiyadi

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN JURUSAN.....	ii
HALAMAN PENGESAHAN PERUSAHAAN.....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	2
1.3 Rumusan Masalah.....	2
1.4 Batasan Masalah.....	3
1.5 Waktu dan Tempat Pelaksanaan.....	3
1.6 Metode Penulisan.....	3
1.7 Sistematika Penulisan.....	4
BAB II TINJAUAN UMUM PERUSAHAAN.....	5
2.1 Profil Perusahaan.....	5
2.2 Sejarah Umum Perusahaan.....	5
2.3 Visi dan Misi Perusahaan.....	6
2.4 Logo dan Makna Perusahaan.....	6
2.5 Struktur Organisasi PT.....	6
2.6 Lokasi PT.....	6
BAB III KAJIAN PUSTAKA.....	8
3.1 Motor DC.....	8
3.1.1 <i>Rotary Encoder</i>	9
3.1.2 <i>Driver Motor H-Bridge</i>	10
3.3 Mikrokontroler.....	11
3.3 Fungsi Alih.....	12

3.3.1 Fungsi Alih Motor DC	12
3.3.2 Identifikasi Parameter Motor DC	13
BAB IV PERHITUNGAN FUNGSI ALIH DAN PARAMETER PID MOTOR PG45 PADA ROBOT DIPO-MECARO (MEDICAL ASSISTANT ROBOT) ...16	
4.1 Robot Dipo-Mecaro (Medical Assistant Robot).....	16
4.2 Pencarian Fungsi Alih	17
4.2.1 Pengujian Driver Motor	17
4.2.2Pengambilan Data dengan PWM Acak	20
4.2.3Memasukan Data ke <i>Workspace Matlab</i>	21
4.2.4Identifikasi Sistem Dengan <i>System Identifier Toolbox Matlab</i>	22
4.3 Analisa Kestabilan Sistem	25
4.4 Pencarian Parameter PID	26
4.5 Aplikasi Hasil PID Untuk Kontrol Kecepatan Motor DC PG45.....	28
5.1 Kesimpulan	30
5.2 Saran.....	30
BIODATA PESERTA KERJA PRAKTIK	32
LAMPIRAN.....	35

DAFTAR GAMBAR

Gambar 2.1 Logo PT. Permalat Berdikari Jaya.....	6
Gambar 2.2 Struktur Organisasi Perusahaan.....	6
Gambar 2.3 Lokasi Kantor Utama PT. Permalat Berdikari Jaya	7
Gambar 2.4 Lokasi Bengkel Utama PT. Permalat Berdikari Jaya	7
Gambar 3.1 Motor DC PG45	8
Gambar 3.2 Keluaran Sinyal <i>Rotary Encoder</i>	9
Gambar 3.3 <i>Wiring</i> EMS H-Bridge 30A	10
Gambar 3.4 Visualisasi Lebar Pulsa PWM.....	11
Gambar 3.5 Arduino Mega 2560.....	11
Gambar 3.6 Model Motor DC.....	13
Gambar 3.7 Blok Diagram Kontrol PID	15
Gambar 4.1 Robot Dipo-Mecaro (Medical Assistant Robot).....	16
Gambar 4.2 Alur Pencarian Fungsi Alih.....	17
Gambar 4.3 <i>Wiring</i> Pengujian Driver Motor dan Pengambilan Data.....	18
Gambar 4.4 Grafik Kecepatan Motor PG45 Dengan Masukan PWM.....	19
Gambar 4.5 Grafik Kecepatan Motor PG45 Dengan Masukan PWM Acak 10-70%	21
Gambar 4.6 Memasukan Hasil Data Ke <i>Workspace Matlab</i>	21
Gambar 4.7 Memanggil <i>System Identifier Toolbox</i>	22
Gambar 4.8 Memasukan Data ke <i>System Identifier Toolbox</i>	23
Gambar 4.9 Estimasi Fungsi Alih Dengan <i>System Identifier Toolbox</i>	23
Gambar 4.10 Fungsi Alih Motor PG45.....	25
Gambar 4.11 Analisa Kestabilan Fungsi Alih Loop Tertutup	26
Gambar 4.12 Pole Zero <i>Mapping</i>	26
Gambar 4.13 Rangkaian Simulasi Simulink	26
Gambar 4.14 Hasil Parameter PID dengan <i>PID Tunner Matlab</i>	27
Gambar 4.15 Hasil Simulasi Dengan Setpoint = 300, $K_P = 32.9279740191956$, $K_I = 196.610205687394$, dan $K_D = 1.20746780266902$	27
Gambar 4.16 Performa PID Dengan Setpoint 300 RPM.....	28
Gambar 4.17 Performa PID Dengan Setpoint 300 Beubah-ubah.....	29

DAFTAR TABEL

Tabel 3.1 Spesifikasi Motor DC PG45.....	9
Tabel 3.2 Spesifikasi EMS H-Bridge 30A.....	10
Tabel 3.3 Spesifikasi Arduino Mega 2560.....	12

BAB I PENDAHULUAN

1.1 Latar Belakang

Mahasiswa saat ini mendapatkan informasi dari berbagai sumber baik di bangku perkuliahan maupun di luar perkuliahan. Mahasiswa program sarjana mendapatkan program perkuliahan selama 4 tahun. Selama program perkuliahan, banyak hal yang dilakukan agar mahasiswa mampu mendapatkan ilmu secara maksimal. Beberapa prodi mewajibkan mahasiswa melakukan praktek kerja lapangan untuk mendapatkan ilmu penerapan yang selama ini dipelajari di bangku perkuliahan. Hal ini bertujuan agar kelak mahasiswa dapat menjadi tenaga ahli yang kompeten dan profesional. Salah satu media yang digunakan untuk menambah pengalaman mahasiswa dan siap menghadapi dunia kerja adalah dengan Kerja Praktik.

Kerja praktik merupakan salah satu mata kuliah wajib di Fakultas Teknik Universitas Diponegoro. Tujuan dari kerja praktik adalah memberikan fasilitas kepada mahasiswa untuk mengaplikasikan ilmu yang telah didapatkan selama berada di bangku perkuliahan, membuktikan kebenaran teori-teori dan dapat menimba ilmu pengetahuan guna peningkatan dan penguasaan ilmu pengetahuan dan teknologi. Pembuatan laporan ini didasari oleh pengalaman penulis selama melakukan kerja praktik di PT. Permalat Berdikari Jaya, Divisi MEP (Mechanical, Electrical, and Plumbing).

PT. Permalat Berdikari Jaya memiliki salah satu divisi utama yaitu Divisi MEP (*Mechanical, Electrical, and Plumbing*). yang memiliki fungsi sebagai perancangan dan pengawasan pada kegiatan yang berhubungan dengan bidang *mechanical, electrical, dan plumbing* pada proyek yang dijalankan di PT. Permalat Berdikari Jaya. Perancangan dan pengawasan dilakukan pada proyek yang dikerjakan oleh vendor saat kontrak diambil dengan PT. Permalat Berdikari Jaya sehingga sesuai dengan keinginan customer.

Dipo-Mecaro (*Medical Assistant Robot*) merupakan proyek riset yang tengah dikerjakan oleh PT. Permalat Berdikari Jaya saat kerja praktik dilaksanakan. Customer dari proyek ini merupakan sebuah rumah sakit yang berkerjasama dengan PT. Permalat Berdikari Jaya. Dipo-Mecaro (*Medical Assistant Robot*) merupakan

robot yang dapat dioperasikan secara nirkabel melalui koneksi Wi-Fi sehingga dapat mengurangi kontak antara pasien Covid-19 dan tenaga medis. Hal ini dianggap sebagai solusi dari tingginya resiko yang dihadapi oleh para tenaga medis saat pandemi Covid-19 berlangsung karena harus berada didalam ruangan atau kontak langsung dengan pasien Covid-19. Progres riset saat ini masih pada pembangunan prototype Dipo-Mecaro secara konstruksi mekanik, elektronika serta perancangan program dan kontrol. Salah satu kendala yang dihadapi pada proyek riset ini adalah perancangan HMI dan kontroler agar robot Dipo-Mecaro dapat dikendalikan secara nirkabel dan dapat mentransmisikan video dengan waktu tunda yang minimal.

Robot Dipo-Mecaro menggunakan empat motor PG45 sebagai penggerak utamanya. Namun perusahaan yang membuat motor PG45 yang digunakan oleh robot tidak memberikan spesifikasi dari motor PG45 secara teknis dan spesifik, sehingga perlu diadakannya perhitungan karakteristik dari motor tersebut. Untuk mengetahui hubungan antara masukan dari keluaran suatu komponen dapat dilakukan dengan mengetahui fungsi alih dari komponen tersebut. Dengan mengetahui fungsi alih suatu komponen kita bisa menghitung keluaran komponen berdasarkan masukan yang diberikan.

Berdasarkan latar belakang tersebut, pada kerja praktik ini penulis mengambil judul “Perhitungan Fungsi Alih dan Kontrol PID Motor PG45 pada Robot Dipo-Mecaro (Medical Assistant Robot) di PT. Permalat Berdikari Jaya”.

1.2 Tujuan

Tujuan dari pelaksanaan Kerja Praktik ini adalah untuk mengetahui fungsi alih motor PG45 dan cara mengontrol kecepatan motor PG45 dengan metode PID untuk Robot Dipo-Mecaro (Medical Assistant Robot) di PT. Permalat Berdikari Jaya.

1.3 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan masalah yang akan dibahas pada laporan kerja praktik ini yaitu:

1. Bagaimana cara melakukan perhitungan fungsi motor PG45?
2. Bagaimana cara melakukan perhitungan parameter PID untuk pengendalian kecepatan motor PG45?

3. Bagaimana cara pengaplikasian kontrol kecepatan dengan metode kontrol PID pada motor PG45?

1.4 Batasan Masalah

Batasan Masalah dari pelaksanaan Kerja Praktik ini adalah untuk mengetahui fungsi alih motor DC PG45 serta cara mengontrol motor PG45 dengan metode PID untuk Robot Dipo-Mecaro (Medical Assistant Robot) di PT. Permalat Berdikari Jaya.

1.5 Waktu dan Tempat Pelaksanaan

Waktu dan tempat pelaksanaan kerja praktik adalah:

Waktu : 20 Juli s.d. 21 Agustus 2020

Tempat : PT. Permalat Berdikari Jaya, Divisi MEP (*Mechanical, Electrical, and Plumbing*)

1.6 Metode Penulisan

Metode yang digunakan dalam penyusunan laporan kerja praktik adalah:

1. Wawancara dan diskusi

Penulis melakukan proses pengumpulan informasi melalui interaksi secara langsung kepada pembimbing kerja praktik cara kerja peralatan yang terdapat di PT. Permalat Berdikari Jaya, Divisi MEP (*Mechanical, Electrical, and Plumbing*).

2. Studi Literatur

Penulis melakukan pencarian literatur yang bersesuaian dan berhubungan dengan kerja praktik yang dilakukan mengenai perancangan kontrol PID pada motor PG45 serta mencari fungsi alih pada motor tersebut agar bisa dikontrol dan stabil serta dengan mengetahui fungsi alih motor akan bisa dikontrol dengan metode kontrol lain dengan mudah.

3. Metode *Research and Development*

Penulis melakukan pelaksanaan riset dengan melakukan pencarian fungsi alih dari motor PG45 dengan membuat *dataset* dari masukan berupa PWM dengan nilai random dan keluarannya, setelah fungsi alih diperoleh penulis merancang kontrol PID dengan *PID Tunner* dengan MATLAB kemudian diaplikasikan langsung ke motor PG45 dengan mengikuti arahan pembimbing serta melakukan percobaan implementasi *project* yang diberikan.

1.7 Sistematika Penulisan

Penulis membahas susunan laporan berdasarkan pada sistematika berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang, tujuan kerja praktik, pembatasan masalah, tempat dan waktu pelaksanaan, metode penulisan, dan sistematika penulisan.

BAB II TINJAUAN UMUM PERUSAHAAN

Bab ini menjelaskan tentang sejarah dan profil umum PT. Permalat Berdikari Jaya, Divisi MEP (*Mechanical, Electrical, and Plumbing*)

BAB III KAJIAN PUSTAKA

Bab ini menjelaskan tentang pengertian motor DC PG45, Perhitungan fungsi alih dan analisa keterkontrolan sistem, Perhitungan parameter PID untuk mengontrol kecepatan motor DC PG45, Mikrokontroler ATMEGA328P, Robot DIPO-MECARO (MEDICAL ASSISTANT ROBOT).

BAB IV ANALISIS PERHITUNGAN FUNGSI ALIH MOTOR DC PG45 DAN PARAMETER PID KONTROL KECEPATAN MOTOR DC PG45

Bab ini menjelaskan tentang perhitungan fungsi alih dari motor PG45 menggunakan *System Identification Toolbox MATLAB*, perhitungan parameter PID untuk kontrol kecepatan motor DC PG45 pada robot DIPO-MECARO (MEDICAL ASSISTANT ROBOT).

BAB V PENUTUP

Bab ini berisi kesimpulan dan saran dari laporan kerja praktik.

BAB II

TINJAUAN UMUM PERUSAHAAN

2.1 Profil Perusahaan

PT. Permalat Berdikari Jaya yang terletak di Semarang, Jawa Tengah merupakan perusahaan yang bergerak pada beberapa bidang, antara lain:

a. Jasa perawatan kaca dan gedung

Beberapa pekerjaan jasa perawatan kaca dan gedung pada awalnya menjadi prioritas perusahaan ini, termasuk kebersihan gedung (cleaning service). Beberapa instansi pemerintah dan swasta mempercayakan jasa kebersihan perusahaan kepada PT. Permalat Berdikari Jaya

b. Arsitektur dan jasa konstruksi “MTR”

Jasa desain arsitektur bangunan rumah dan gedung pada awalnya diberikan kepada konsumen. Selanjutnya dibuat beberapa cluster perumahan, termasuk jasa pembuatan konstruksi bangunan. Adapun jasa arsitektur dan konstruksi yang digunakan adalah “MTR Arsitektur”.

c. Supply komponen untuk industri

Terdapat workshop untuk mass product. Di antara peralatan yang dimiliki adalah mesin-mesin perkakas CNC yang digunakan untuk memproduksi pesanan komponen dari perusahaan lainnya, khususnya untuk komponen ekspor.

d. Peralatan otomasi untuk industri kesehatan “Media Geometri”

Salah satu hasil riset yang sedang dikembangkan saat ini adalah peralatan medis seperti Robot Dipo-Mecaro (Medical Assistant Robot) yang berfungsi membantu tenaga medis di kondisi pandemi Covid-19 sedang berlangsung. Adapun merk alat sistem yang dikembangkan bernama “Media Geometri (MG)”.

2.2 Sejarah Umum Perusahaan

PT. Permalat Berdikari Jaya merupakan pengembangan dari CV. Permalat Berdikari yang didirikan pada tanggal 18 Desember 2013 bergerak di bidang Jasa. Dengan semakin banyaknya permintaan pelayanan di bidang Jasa dan lainnya, maka pada didirikan PT. Permalat Berdikari Jaya melalui Akta Notaris tanggal 07 Desember 2018 dan dinyatakan dalam Keputusan Menteri Hukum dan Hak Asasi Manusia Republik Indonesia Nomor AHU0001106.AH.01.01.Tahun 2019.

2.3 Visi dan Misi Perusahaan

Visi

Menjadikan PT. Permalat Berdikari Jaya yang mampu memberikan pelayanan sebaik-baiknya kepada klien, menghasilkan karya yang tidak hanya sebatas *engineering* tetapi lebih ke buah pemikiran

Misi

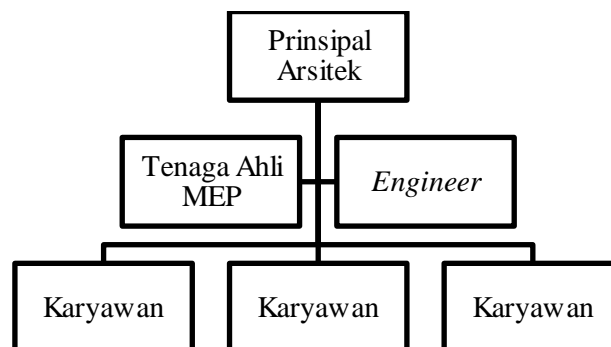
1. Menghasilkan karya *engineering* yang berkualitas dalam mendukung pembangunan Indonesia
2. Memberikan layanan secara profesional dengan mengedepankan kepuasan pelanggan.
3. Menjadikan perusahaan jasa arsitektur dan konstruksi yang memberikan nilai lebih bagi karyawan dan masyarakat.

2.4 Logo dan Makna Perusahaan



Gambar 2.1 Logo PT. Permalat Berdikari Jaya

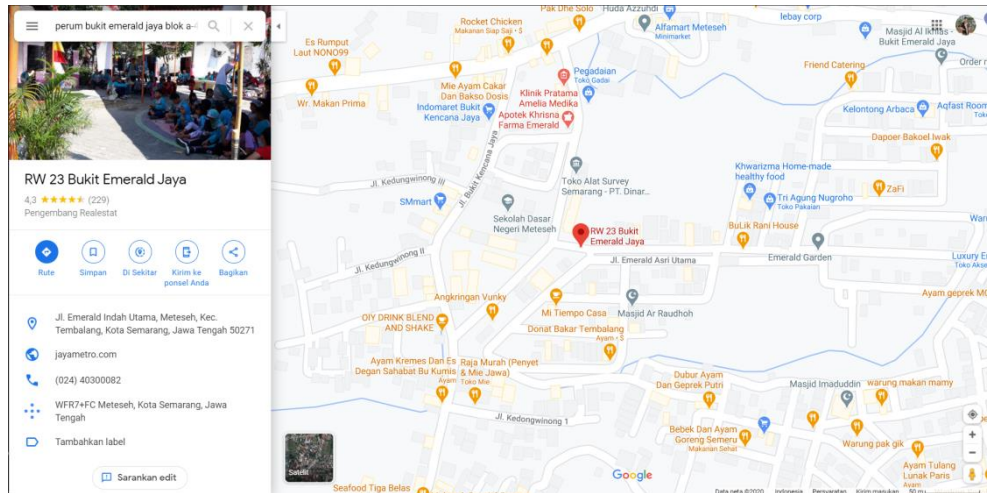
2.5 Struktur Organisasi PT



Gambar 2.2 Struktur Organisasi Perusahaan

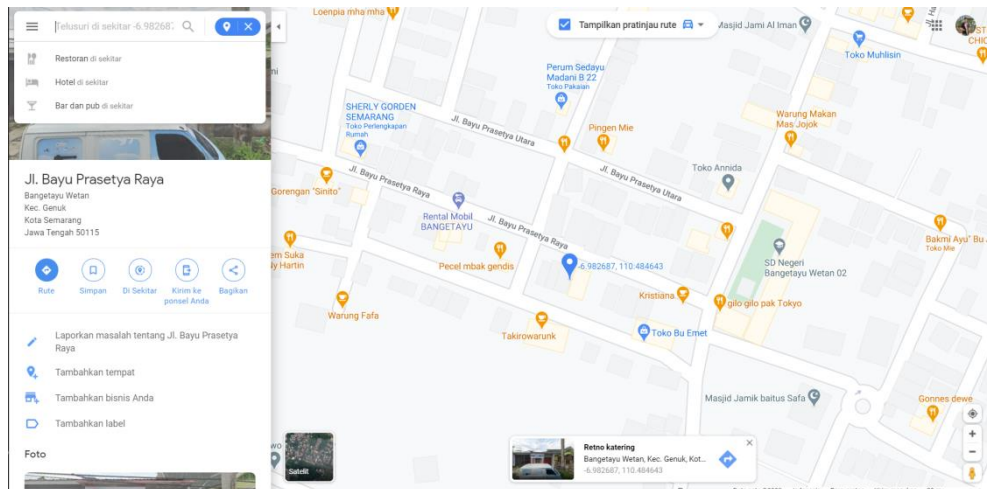
2.6 Lokasi PT

PT. Permalat Berdikari Jaya Semarang memiliki dua buah lokasi utama yaitu kantor utama yang terletak di Perum Bukit Emerald Jaya Blok A-4 No. 104, Meteseh, Tembalang, Semarang, Jawa Tengah dan bengkel utama yang terletak di Jl. Bayu Prasetya Raya, Bangetayu Wetan, Kecamatan Genuk, Semarang, Jawa Tengah, . Lokasi kantor utama dapat dilihat pada **Gambar 2.3** yang ditandai dengan pin berwarna merah.



Gambar 2.3 Lokasi Kantor Utama PT. Permalat Berdikari Jaya

Sementara lokasi bengkel utama dapat dilihat pada **Gambar 2.4** yang ditandai dengan pin berwarna merah.



Gambar 2.4 Lokasi Bengkel Utama PT. Permalat Berdikari Jaya

BAB III

KAJIAN PUSTAKA

3.1 Motor DC

Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah (DC) pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/*direct-unidirectional* (Djalal, 2017).

Robot Dipo-Mecaro menggunakan empat buah motor DC PG45 mekanum sebagai penggerak utamanya. Motor PG45 pada robot dikontrol dengan mikrokontroller dengan bantuan *driver motor EMS 30A H-Bride*. Mikrokontroler mengatur kecepatan motor PG45 dengan cara mengatur besar pulsa PWM (*Pulse Width Modulation*). Yang mana penelitian yang sudah dilakukan sebelumnya, kecepatan motor DC dapat diatur menggunakan kerapatan pulse PWM yang terdapat dalam mikrokontroler ATmega2560 pada Arduino (Setiawan, 2017).



Gambar 3.1 Motor DC PG45

Motor DC PG45 merupakan seri motor DC yang memiliki torsi besar dan kecepatan cukup. Secara teknis spesifikasi motor DC PG45 yang digunakan seperti pada **Tabel 3.1** dibawah.

Tabel 3.1 Spesifikasi Motor DC PG45

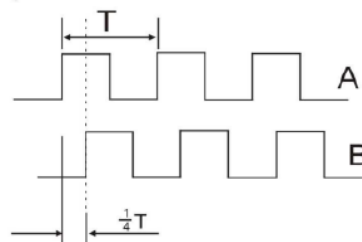
Fitur Teknis	Spesifikasi
<i>Motor Type</i>	<i>Brushed Motor</i>
<i>Voltage</i>	24V
<i>Power Rated</i>	100W
<i>No Load Current</i>	<1.5 A
<i>Torque</i>	15kgcm
<i>No Load Speed</i>	468 ($\pm 10\%$) Rpm
<i>Rated Load Speed</i>	398 ($\pm 10\%$) Rpm
<i>Internal Encoder</i>	37PPR (datasheet) 150PPR (praktis)

3.1.1 Rotary Encoder

Rotary Encoder atau encoder poros adalah perangkat elektromekanis yang mengubah sudut posisi/gerakan poros atau mengubah poros ke kode analog ataupun digital (eitel, 2014). Ada dua jenis utama rotary encoder, yaitu : absolut dan incremental. Sebuah rotary encoder incremental menyediakan output siklus (hanya) ketika encoder diputar. *Rotary encoder* dapat berupa mekanik atau optik. Jenis pada benda mekanik membutuhkan *debouncing* dan biasanya digunakan sebagai potensiometer digital pada peralatan termasuk perangkat konsumen.

Rotary encoder incremental merupakan yang paling banyak digunakan dari semua *rotary encoder* karena biaya rendah dan kemampuan untuk memberikan sinyal yang dapat dengan mudah ditafsirkan untuk memberikan informasi terkait seperti gerak kecepatan. Encoder Incremental digunakan untuk melacak gerakan dan dapat digunakan untuk menentukan posisi dan kecepatan. Hal ini dapat berupa linear atau gerakan berputar. Karena arah dapat ditentukan, pengukuran yang sangat akurat dapat dibuat. Rotary Encoder Incremental memiliki dua buah output yang disebut channel A dan channel B, yang disebut output quadrature, karena pada channel A dan channel B memiliki perbedaan sebesar 90 derajat keluar dari fase.

Dalam kerja praktik ini menggunakan sensor encoder internal dari motor DC PG45 yang mempunyai spesifikasi 37PPR, namun dalam praktiknya setelah dites memiliki ± 150 PPR.

**Gambar 3.2** Keluaran Sinyal Rotary Encoder

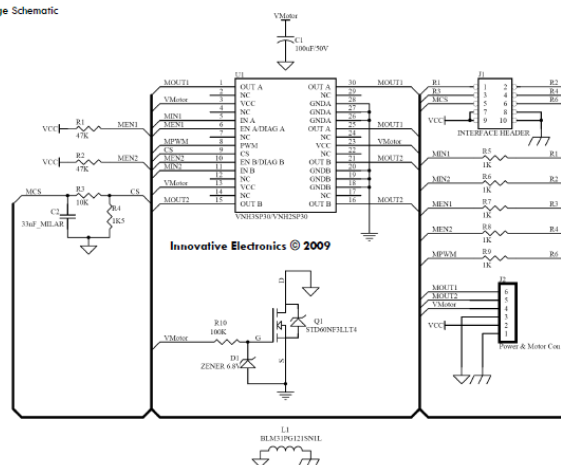
3.1.2 Driver Motor H-Bridge

Motor DC tidak bisa dikontrol secara langsung tanpa menggunakan *driver motor*. Driver motor ini dipasaran ada beberapa jenis, ada yang menggunakan IC, realy, ataupun mosfet. *Driver motor* jenis ini hanya memiliki satu keluaran, sehingga jika untuk mengontrol motor DC maka hanya bisa maju atau berhenti saja. Ada jenis *driver motor* yang lebih *advance* dalam penggunaannya, yaitu *driver motor H-Bridge*, driver motor ini memiliki dua keluaran, sehingga dalam pengaplikasiannya, motor DC dapat dikontrol untuk maju dan mundur. Dalam kerja praktik ini digunakan *driver motor* EMS H-Bridge 30A yang artinya bisa mengeluarkan arus sebesar 30A secara kontinu dalam penggunaannya. Spesifikasi *driver motor* yang digunakan bisa dilihat pada **Tabel 3.2**.

Tabel 3.2 Spesifikasi EMS H-Bridge 30A

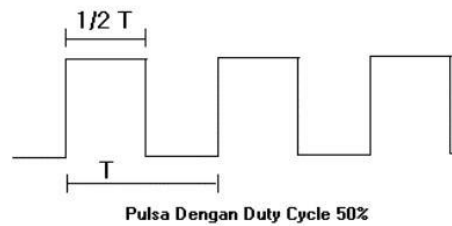
Fitur Teknis	Spesifikasi
Driver	VNH2SP30
Tegangan Load	5.5 – 36 V
Arus	30A (kontinu)
Frekuensi PWM maksimum	10KHz
Keamanan	Arus pendek, kelebihan dan kekurangan tegangan, baterai terbalik, <i>overheating</i>
Mode	3 mode (Maju, Mundur, dan Normal/Diam)

Attachment
EMS 30 A H-Bridge Schematic



Gambar 3.3 Wiring EMS H-Bridge 30A

Fungsi dari driver motor ini adalah untuk mengatur PWM ke motor PG45. Dalam pengaplikasian kontrol motor, PWM teknik modulasi dengan mengubah lebar pulsa (*duty cycle*). Dengan mengubah lebar pulsa maka kecepatan motor akan berubah dan kecepatan motor sebanding dengan lebar pulsa. Lebar pulsa dapat diatur dengan mikrokontroler yang akan dibahas pada subbab berikutnya. Visualisasi lebar pulsa pada PWM dapat dilihat pada **Gambar 3.4**.



Gambar 3.4 Visualisasi Lebar Pulsa PWM

3.3 Mikrokontroler

Mikrokontroler merupakan sebuah chip IC yg dapat di program sesuai dengan kebutuhan penggunaannya. Untuk mengisi program yang sesuai dengan kebutuhan pengguna dengan tipe data heksa (Hex file) dengan berisikan intruksi atau perintah untuk menjalankan sistem kontrol (Desnanjaya, 2018).

Dalam kerja praktik ini menggunakan mikrokontroler ATMEGA 2560 dengan sistem minimum Arduino Mega. Untuk mengisi program Arduini Mega dapat menggunakan Arduino IDE yang merupakan program *open-source* yang dibuat khusus untuk sistem minimum berbasis Arduino. Arduino Mega digunakan untuk mengambil *data sample* dari kecepatan motor PG45 terhadap masukan yang berupa PWM (*Pulse Width Modulation*) sehingga data masukan dan keluaran tersebut menjadi *dataset* yang akan diolah pada proses selanjutnya.

Arduino Mega merupakan salah satu board mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan breadboard. Arduino Mega diciptakan dengan basis microcontroller ATmega 2560 (untuk Arduino Mega versi 3.x) atau Atmega 1280(untuk Arduino versi 2.x) (Arifin, 2016).



Gambar 3.5 Arduino Mega 2560

Dalam pengumpulan *dataset*, mikrokontroler membangkitkan sinyal PWM ke *driver motor*, kemudian driver motor mengolah sinyal PWM menjadi tegangan yang dilanjutkan ke motor PG45. Ketika motor PG45 mendapat arus listrik searah maka motor berputar dan sensor encoder internal dari motor PG45 membaca pergerakan kemudian dikirim ke dan diolah mikrokontroler ATMEGA 2560. Papan mikrokontroler yang digunakan memiliki spesifikasi seperti pada **Tabel 3.2**.

Tabel 3.3 Spesifikasi Arduino Mega 2560

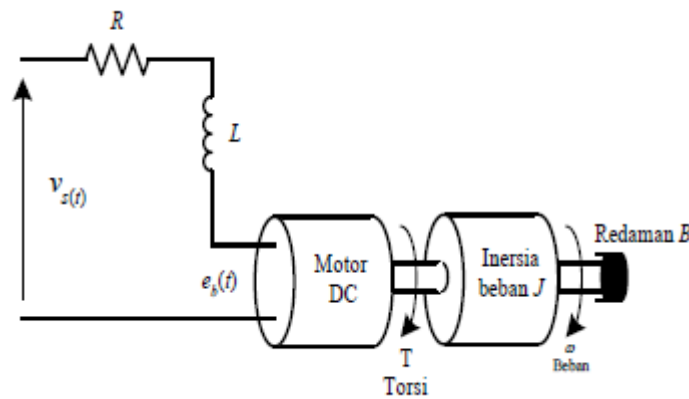
Fitur Teknis	Spesifikasi
Tegangan Kerja	5V
I/O Pin Digital	54 (15 PWM) Pin
O/O Pin Analog	16 Pin
Arus per I/O	20 mA
Flash Memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Dimensi	10,16 cm x 5,3 cm
Berat	37 Gram

3.3 Fungsi Alih

Dalam teori kontrol, fungsi yang disebut “fungsi alih” sering digunakan untuk mencirikan hubungan masukan dan keluaran dari sistem linier parameter konstan. Konsep fungsi alih hanya digunakan pada sistem linier parameter konstan, walaupun dapat diperluas untuk suatu sistem kontrol nonlinier (Sumardi, 2019). Secara mudahnya fungsi alih adalah variabel keluaran $Y_{(s)}$ dibagi variabel masukan $X_{(s)}$ dengan kondisi awal nol. Dengan fungsi alih kita bisa menganalisa dan menghitung keluaran sistem secara matematis tanpa menghidupkan komponen tersebut, dengan begitu kita bisa mengetahui keluaran suatu komponen hanya dengan perhitungan matematis. Dalam penggunaannya untuk motor PG45, kita bisa menghitung secara matematis percepatan, kecepatan, dan posisi dari motor. Secara tidak langsung fungsi alih ini sangat penting untuk mengetahui karakteristik suatu komponen sehingga mempermudah dalam pengontrolan komponen tersebut.

3.3.1 Fungsi Alih Motor DC

Secara umum motor DC memiliki komponen yang perlu diperhitungkan dalam perhitungan matematis, komponen tersebut : resistansi jangkar, Induktansi jangkar, arus medan, momen inersia, dan koefisien gesek. Dari semua komponen tersebut dapat digambarkan hubungan antar komponen seperti pada **Gambar 3.6** dibawah.



Gambar 3.6 Model Motor DC

Dari rangkaian listrik dapat dibuat persamaan tegangan menurut hukum Kirchhoff tegangan seperti dinyatakan oleh persamaan (1)

$$V_s(t) = Ri_a(t) + L \frac{di_a(t)}{dt} + e_b(t) \quad (1)$$

Dimana:

R = tahanan belitan armatur (Ω)

L = Impedans belitana armatur (H)

$i_a(t)$ = Arus armatur (A)

$e_b(t)$ = Tegangan induksi armatur (V)

$V_s(t)$ = Tegangan terminal motor (V)

$e_b(t)$ adalah tegangan induksi yang tergantung pada putaran sudut $\omega(t)$, dinyatakan oleh persamaan (2)

$$e_b(t) = k_1 \phi \frac{60}{2\pi} \omega(t) \quad (2)$$

Dimana:

k_1 = Konstanta dimensi motor

ϕ = Fluks magnet kutub motor (Wb)

$\frac{60}{2\pi} \omega(t)$ = Putaran rotor (rpm)

$\omega(t)$ = Kecepatan sudut rotor (rad/s)

3.3.2 Identifikasi Parameter Motor DC

Untuk mengidentifikasi parameter motor DC, ada beberapa metode yang bisa digunakan. Pada tahun 2012 Setiawan menghitung fungsi alih menggunakan *system identifier toolbox Matlab* dengan data masukan PWM acak dan keluaran kecepatan motor menghasilkan *best fit* 93%. Pada tahun yang sama Wu berhasil

menghitung persamaan fungsi alih motor DC menggunakan perhitungan matematis dengan *speed step responses*. Pada tahun 2018 Septiarini berhasil menghitung pemodelan matematika kecepatan motor DC menggunakan identifikasi dengan metode 2s dan hasil perhitungan tersebut digunakan untuk perhitungan kontrol PID. Dalam kerja praktik ini penulis menggunakan metode pertama untuk menghitung fungsi alih motor DC PG45 robot Dipo-Mecaro (*Medical Assistant Robot*). Pada tahun 2017 Tang berhasil melakukan identifikasi sistem dan PID *Tunning* dengan menggunakan *system identifier toolbox Matlab* dan *PID Tunner Matlab* menghasilkan *best fit* sebesar 68.71%. Dalam kerja praktik ini penulis menggunakan metode yang sama dilakukan oleh Tang pada tahun 2017 yaitu menggunakan *system identifier toolbox Matlab* dan *PID Tunner Matlab* karena *system identifier toolbox Matlab* mampu mengestimasi *best fit* hasil yang diperoleh dan *PID Tunner Matlab* mampu mensimulasi hasil fungsi alih yang diperoleh untuk membuat sistem dengan karakteristik sesuai kebutuhan hanya dengan menggeser *time respon*-nya.

3.3.3 Kestabilan Sistem

Fungsi alih dikatakan apabila semua akar-akarnya berada disebelah kiri sumbuk y (bernialai negatif) (Triwiyatno, 2019). Dalam kerja praktik ini penulis mengubah fungsi alih yang dihasilkan menjadi loop tertutup terlebih dahulu, karena fungsi alih yang akan dihasilkan akan digunakan dalam metode kontrol tertentu. Sehingga sistem yang digunakan adalah loop tertutup dan akar-akar dari loop tertutup tersebut yang menjadi acuan uji kestabilan sistem.

3.3.4 Kontrol PID

Kontrol PID adalah merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Persamaan PID merupakan kontrol yang paling banyak digunakan dalam dunia industri (Setiawan, 2015). PID sesuai memiliki parameter kontrol proporsional (Kp), kontrol integratif (Ki), dan kontrol derivatif (Kd). Pada tahun 2018 Septiarini berhasil mengontrol motor DC dengan perhitungan fungsi alih yang telah dihasilkan menggunakan metode 2s. Dalam kerja praktik ini menggunakan metode *trial error* dengan *PID Tunner Matlab* menggunakan fungsi alih dari motor DC PG45 yang dihasilkan. Adapun persamaan kontrol PID sebagai berikut:

$$mv(t) = K_p(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt})$$

Dimana:

mv(t) = output dari pengontrol PID atau Manipulated Variable

Kp = konstanta Proporsional

Ti = konstanta Integral

T_d = konstanta Derivatif

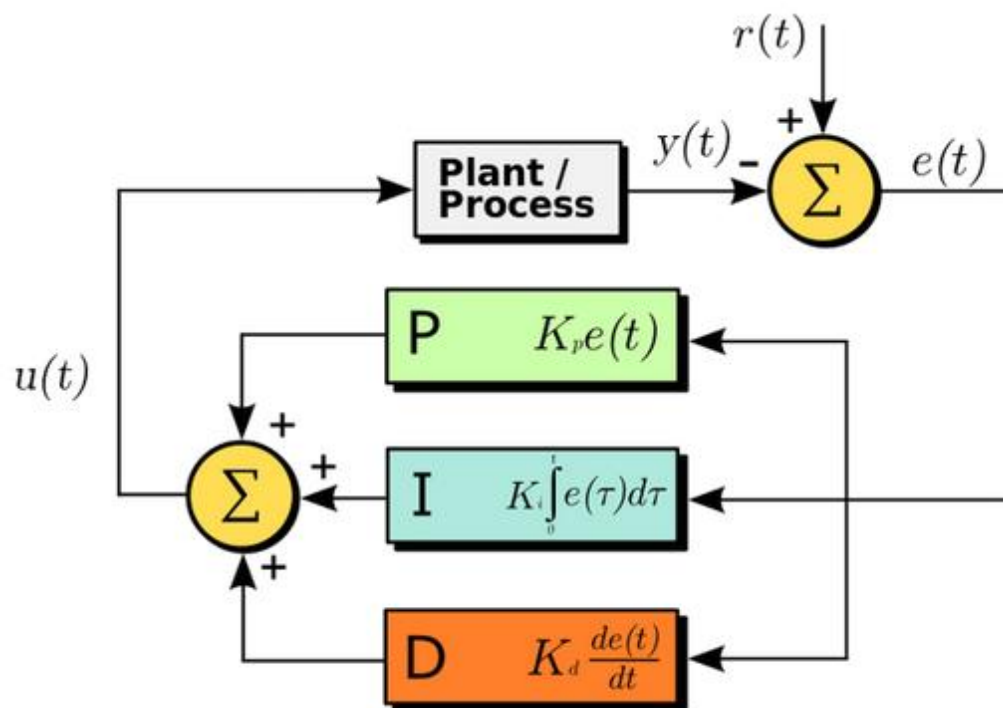
$e(t)$ = error (selisih antara set point dengan level aktual)

Persamaan kontrol PID diatas juga bisa dituliskan sebagai berikut:

$$mv(t) = K_p(e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt})$$

Dengan:

$$K_i = K_p \times \frac{1}{T_i} \text{ dan } K_d = K_p \times T_d$$



Gambar 3.7 Blok Diagram Kontrol PID

BAB IV

PERHITUNGAN FUNGSI ALIH DAN PARAMETER PID MOTOR PG45 PADA ROBOT DIPO-MECARO (MEDICAL ASSISTANT ROBOT)

4.1 Robot Dipo-Mecaro (Medical Assistant Robot)

Robot Dipo-Mecaro (Medical Assistant Robot) merupakan salah satu riset yang berfungsi untuk membantu tenaga medis di kondisi pandemi Covid-19 sedang berlangsung. Robot ini mampu melakukan beberapa kegiatan yang dilakukan oleh perawat pada umumnya, seperti mengantarkan makanan dan obat-obatan, melakukan pemeriksaan sederhana seperti mengukur tekanan darah, serta mampu berkomunikasi dengan pasien. Robot Dipo-Mecaro merupakan mobile robot yang menggunakan konfigurasi omni-wheel yang dapat memungkinkan robot dapat bergerak kesegala arah.

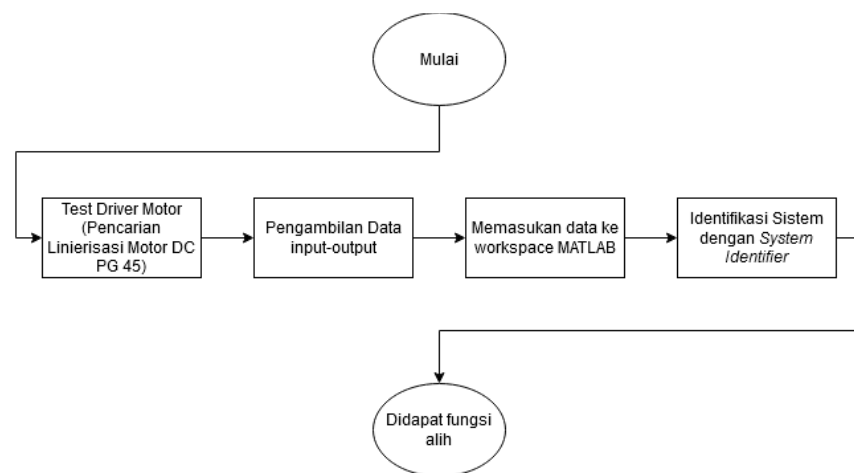


Gambar 4.1 Robot Dipo-Mecaro (Medical Assistant Robot)

Riset ini dilakukan oleh salah bidang dari PT. Permalat Berdikari Jaya yaitu dengan “Media Geometri (MG)” yang bergerak dibidang produksi alat medis.

4.2 Pencarian Fungsi Alih

Dengan fungsi alih kita bisa menghitung keluaran sistem terhadap masukan yang diberikan, dengan begitu kita perlu mengumpulkan data masukan dan keluaran yang diberikan. Perhitungan fungsi alih dilakukan dengan beberapa tahap, yang pertama adalah melihat linierisasi dari Motor DC PG45 yang akan dicari fungsi alihnya dengan cara memasukkan nilai PWM dari nilai minimum sampai nilai maksimum. Setelah mengetahui linierisasi motor DC PG45 di PWM berapa persen selanjutnya adalah mengambil data dengan nilai masukan PWM acak pada *range* liniernya yang sudah diketahui sebelumnya. Pengambilan data dilakukan menggunakan Arduino Mega 2560 dengan Arduino IDE. Setelah didapat data keluaran terhadap masukan, data tersebut dilolah dan dilakukan identifikasi sistem menggunakan *System Identifier Toolbox Matlab*. Alur pencarian fungsi alih Motor DC PG45 ini ditunjukkan pada **Gambar 4.2** dibawah.

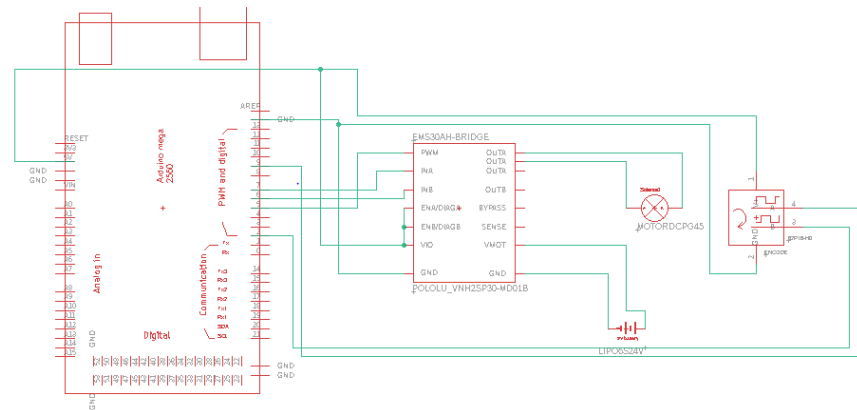


Gambar 4.2 Alur Pencarian Fungsi Alih

Perhitungan fungsi alih dilakukan dengan *System Identifier Toolbox Matlab*, Dalam pengumpulan data ini digunakan Arduino Mega 2560 dengan perangkat lunak Arduino IDE. Sebelum melakukan pengumpulan data, dilakukan pengujian driver motor untuk melihat linierisasi kecepatan motor terhadap masukan PWM yang diberikan.

4.2.1 Pengujian Driver Motor

Pengujian *driver motor* bertujuan untuk melihat *range* linierisasi dari motor PG45. Driver motor yang digunakan adalah EMS H-Bridge 30A yang dikontrol menggunakan Arduino Mega. Pada Wiring pengujian Driver motor dapat dilihat pada **Gambar 4.3** dibawah.



Gambar 4.3 Wiring Pengujian Driver Motor dan Pengambilan Data

Pada pengujian *driver motor* ini menggunakan perangkat lunak Arduino IDE untuk mengisi program yang akan dieksekusi mikrokontroler. Pada pengujian ini mikrokontroler mengirim nilai PWM dari nilai minimal sampai maksimal, yang mana pada kerja praktik ini menggunakan frekuensi 1Khz sehingga nilainya 0 sampai 1000 yang akan di konversi dengan fungsi *map* menjadi 0-255, karena PWM dari Arduino hanya support 0-255. Dengan memasukan sinyal PWM ke *motor driver* maka akan menggerakkan motor PG45, *feedback* pergerakan motor didapat dari sensor *encoder internal* motor PG45 yang memiliki 150 PPR (*Pulse per Rotation*). Data sensor encoder sendiri didapat dengan *interrupt* sehingga tidak mengganggu program utama. *Listing program interrupt* untuk menerima data sinyal encoder dan perhitungan RPM sebagai berikut:

```
void onPin2CHANGECallBackFunction(uint32_t Time, uint32_t PinsChange
d, uint32_t Pins){
    static uint32_t lTime; // Saved Last Time of Last Pulse
    uint32_t cTime; // Current Time
    cTime = micros(); // Store the time for RPM Calculations
    int32_t dTime; // Delt in time

// Encoder Code
    bool DataPinVal = digitalRead(DataPin);
// We know pin 2 just went high to trigger the interrupt
// depending on direction the data pin will either be high or low
    EncoderCounter += (DataPinVal) ? 1 : -
1; // Should we step up or down?
// End Encoder Code
// calculate the DeltaT between pulses
    dTime = cTime - lTime;
    lTime = cTime;
    speedInRPM = Multiplier / ((DataPinVal) ? dTime: (-
1 * dTime)); // Calculate the RPM Switch DeltaT to either positive o
r negative to represent Forward or reverse RPM
}
```

Sensor encoder internal dari motor PG45 sendiri memiliki empat (4) pin, yaitu VCC, GND, PINA, dan PINB. Pin A terhubung dengan digital pin 2 Arduino Mega, pin B terhubung dengan pin digital 9. Apabila pin digital 2 yang terhubung dengan encoder PIN A menerima sinyal HIGH maka akan men-trigger untuk menambah nilai encoder (*increment*). Perhitungan RPM dengan mengitung perbedaan encoder waktu akhir dan waktu awal dikali dengan pengali yang mana ini didapat dengan menghitung (60 detik dikali 1000000 mikrodetik) dibagi spesifikasi PPR sensor encoder yaitu 150, sehingga pengalinya adalah 399999.8.

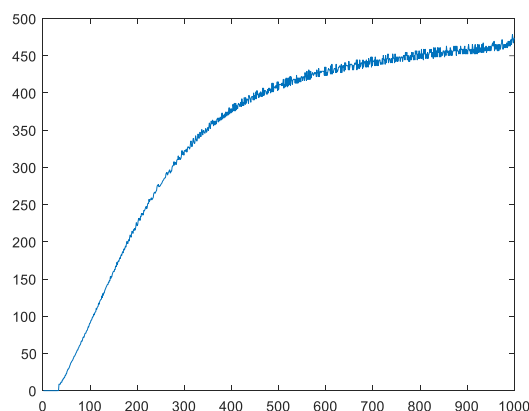
Dengan memasukan nilai PWM 0 sampai maksimal. Program untuk menjalankan perintah tersebut sebagai berikut:

```
void motorDriverTest(){
    nilaiRandom += 1;
    analogWrite(motor,map(nilaiRandom, 0, 1000, 0, 255)); //set pwm
    motor as random number
    //show nilaiRandom value as a PWM
    Serial.print(nilaiRandom);
    Serial.print(" ");

    //show Encoder value
    //Serial.println(EncoderCounter);
    //Serial.print(" ");

    //show speed in RPM
    Serial.print(speedInRPM, 5);
    Serial.println(",");
}
```

Program tersebut menjalankan penambahan variabel nilaiRandom dari 1 sampai 1000 yang akan dikonversi menjadi 0-255. Data hasil dari percobaan ini didapatkan data sebagaimana pada **Gambar 4.4**.



Gambar 4.4 Grafik Kecepatan Motor PG45 Dengan Masukan PWM

Bisa kita lihat bahwa motor bergerak meningkat kecepatannya dari PWM 100 sampai 700 dengan kecepatan maksimal ± 465 RPM. Nilai percepatan naik secara signifikan dari PWM ± 100 sampai ± 700 , setelah itu kecepatan motor naik namun tidak signifikan. Dengan data ini bisa kita katakan linieritas motor berapa pada

range PWM 100-700. Data ini akan digunakan untuk masukan nilai acak pada *dataset* kecepatan motor terhadap masukan PWM yang akan kita gunakan untuk *dataset system identifier* pada Matlab.

4.2.2 Pengambilan Data dengan PWM Acak

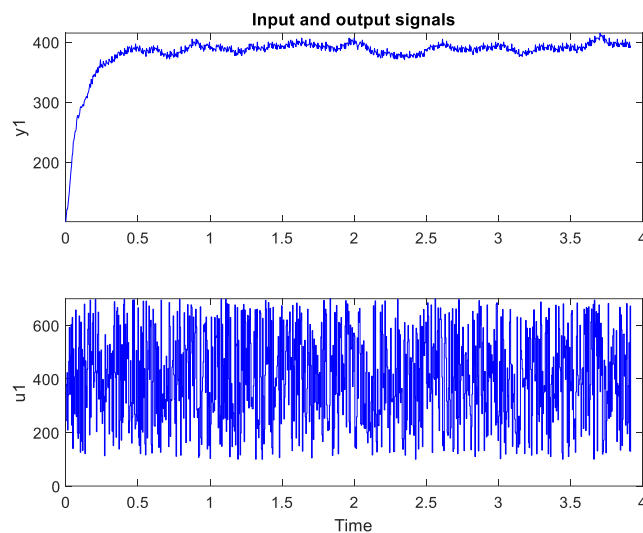
Setelah diketahui batasan linieritas motor, maka tahap selanjutnya adalah mengambil data kecepatan motor PG45 terhadap masukan PWM dengan nilai acak rentang nilai 100-700. Pengambilan data kerja praktik ini dilakukan menggunakan perangkat yang sama seperti *test driver motor*. Namun berbeda program, sehingga mikrokontroler menjalankan program yang berbeda. Program untuk pengambilan data sebagai berikut:

```
void nilaiAcak(){
    nilaiRandom = random(100,700); //10-70%
    analogWrite(motor,map(nilaiRandom, 0, 1000, 0, 255)); //set pwm
    motor as random number
    //show nilaiRandom value as a PWM
    Serial.print(nilaiRandom);
    Serial.print(" ");

    //show Encoder value
    //Serial.println(EncoderCounter);
    //Serial.print(" ");

    //show speed in RPM
    Serial.print(speedInRPM, 5);
    Serial.println(",");
}
```

Program diatas mengubah variabel nilaiRandom menggunakan fungsi random dengan nilai minimum 100 dan nilai maksimum 700, nilai random tersebut dikonversi menjadi 0-255 dengan fungsi map. Data hasil dari percobaan ini didapatkan data sebagaimana pada **Gambar 4.5**



Gambar 4.5 Grafik Kecepatan Motor PG45 Dengan Masukan PWM Acak 10-70%

Data yang didapat diatas kemudian diproses menggunakan *System Identifier* di Matlab.

4.2.3 Memasukan Data ke *Workspace Matlab*

Pada tahap ini dilakukan identifikasi sistem atau kata lain mencari fungsi alih dari data yang sudah didapat sebelumnya. Pencarian fungsi alih khususnya untuk motor DC ada dua metode, yang pertama dengan mencari secara perhitungan matematis menggunakan *speed step responses* (Wu, 2012). Yang kedua menggunakan *System Identification Toolbox* Matlab. Dalam kerja praktik ini penulis menggunakan metode yang kedua. Data yang diperoleh kecepatan motor dengan masukan PWM acak 10-70% sebelumnya disalin dan dimasukkan di *workspace* Matlab. Sebagian *listing program* dan data yang dimasukkan ke Matlab seperti pada **Gambar 4.6**.

```

Editor - F:\OneDrive - Universitas Diponegoro\KERJA PRAKTEK\GITHUB\matlab\nilaiAcak.m
nilaiAcak.m
972 227 404.85812
973 473 392.15667
974 681 396.82522
975 195 399.99981
976 236 398.40618
977 374 395.25674
978 535 390.62481
979 131 399.99981
980 672 390.62481
981 128 393.70059;
982 x=nilai(:,1);
983 y=nilai(:,2);
984 plot(x,y)
985

```

Gambar 4.6 Memasukan Hasil Data Ke *Workspace Matlab*

Pada **Gambar 4.6** data sebelah kiri adalah masukan PWM acak dan yang sebelah kanan adalah *feedback* kecepatan motor DC PG45 yang dideteksi dengan sensor encoder. Memisahkan matriks data masukan dan keluaran yang masing-masing berjumlah 980 data menggunakan program:

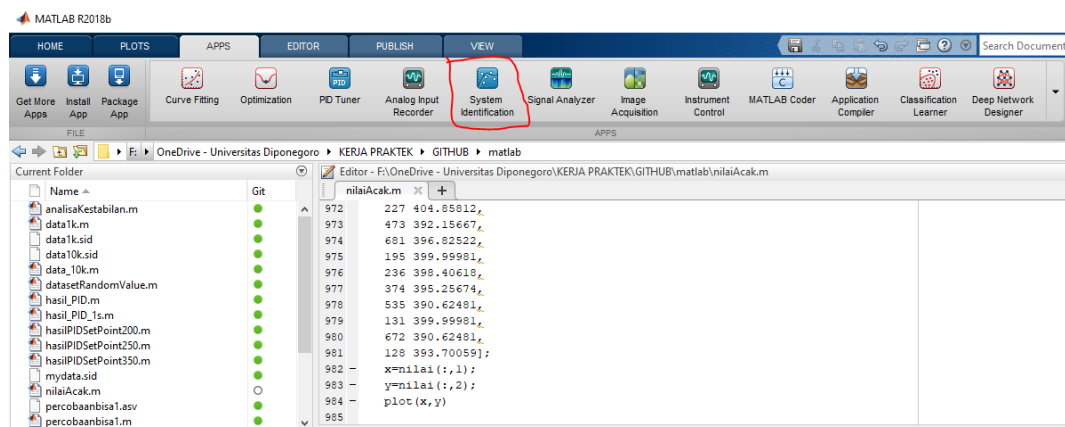
```
X=nilai(:,1);
```

```
Y=nilai(:,2);
```

Dengan sintaks diatas dipisahkan data masukan PWM dan keluaran kecepatan motor DC PG45 yang mana variabel x berisi semua masukan PWM dengan mengambil semua kolom 1 dan variabel y berisi semua keluaran kecepatan motor DC PG45 dengan mengambil semua kolom 2 dari matrik nilai. Dari data variabel x dan y, selanjutnya diproses menggunakan *System Identifier Toolbox Matlab*.

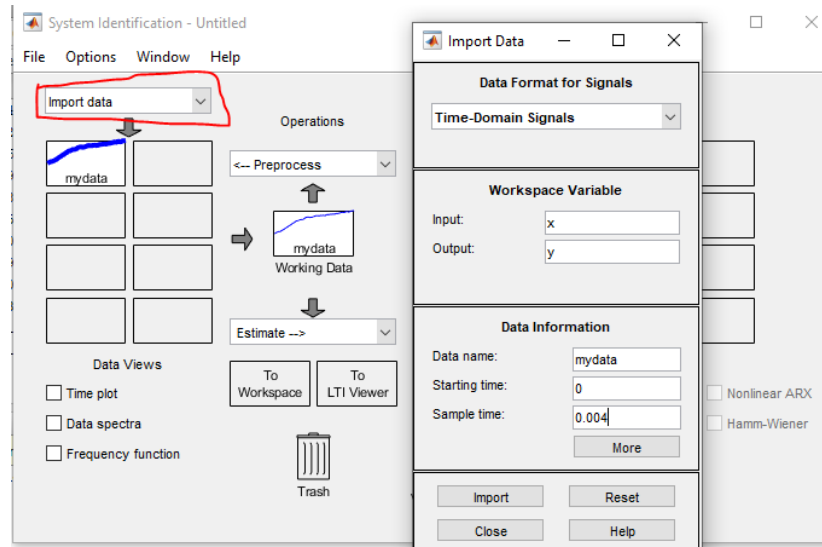
4.2.4 Identifikasi Sistem Dengan *System Identifier Toolbox Matlab*

Dengan *System Identification Toolbox* pada Matlab bisa membangun model matematis dinamis dari data masukan dan keluaran yang diukur. Data yang diukur bisa berupa dalam domain waktu atau frekuensi. Dengan *toolbox* ini bisa juga untuk mengukur ketepatan model matematis yang dihasilkan dalam bentuk persen dan error dihasilkan. Untuk memanggil *System Identification Toolbox* bisa menggunakan sintaks 'ident' atau 'systemIdentification' pada *Command Window*, selain menggunakan sintaks bisa juga dengan menekan ikon '*System Identification*' seperti pada **Gambar 4.7** yang dilingkari dengan pen warna merah.



Gambar 4.7 Memanggil *System Identifier Toolbox*

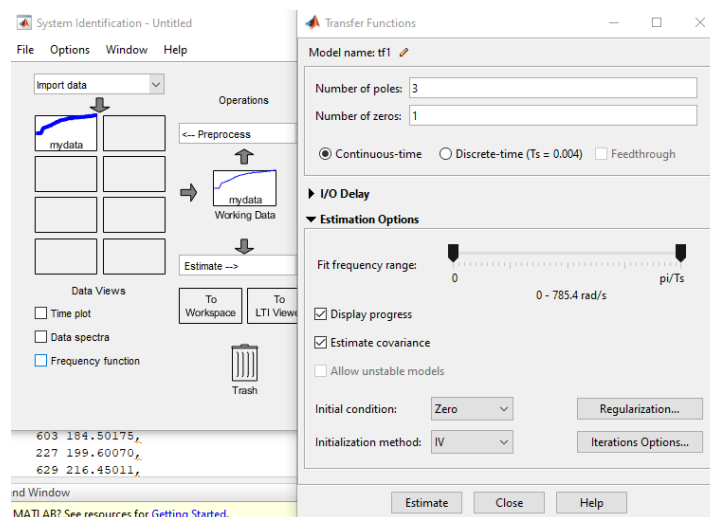
Setelah memanggil *System Identification Toolbox* maka muncul jendela dari *System Identification Toolbox* itu. Di *System Identification Toolbox* ini dari data yang telah didapat diolah menjadi fungsi alih. Sebelum mengolah data di *System Identification Toolbox* data dari *workspace* yang sudah di simpan sebelumnya (variabel x dan y) dimasukkan kedalam *System Identification Toolbox* seperti pada **Gambar 4.8** dibawah.



Gambar 4.8 Memasukan Data ke *System Identifier Toolbox*

Untuk memasukkan data dari *workspace* kedalam *System Identification Toolbox* dengan klik pada bar 'import data' yang sudah dilingkari pada gambar 4.8, selanjutnya dipilih *Time Domain Signal*, karena dalam kerja praktik ini menggunakan domain waktu dalam pengumpulan data. Setelah itu akan muncul jendela 'Import Data' yang mana diisi dengan *input* variabel *x* yang berisi masukan PWM acak dan *output* variabel *y* yang berisi kecepatan motor DC PG45, *Starting time* 0 karena pengambilan waktu dari waktu ke 0 detik dan *sampling time* 0.004 karena sampling perdata diambil setiap 4 ms.

Setelah data masuk ke *System Identification Toolbox* dengan ditandai pada kotak sebelah kiri berisi grafik warna biru pada gambar 4.8, tahap selanjutnya adalah memproses *working data* pada *System Identification Toolbox* dengan mengestimasi fungsi alih sistem. Proses ini dapat dilihat pada **Gambar 4.9**.



Gambar 4.9 Estimasi Fungsi Alih Dengan *System Identifier Toolbox*

Dalam proses ini dilakukan estimasi fungsi alih dari data di *working data* dengan variabel *mydata* yang sudah dimasukan sebelumnya. Seperti pada **Gambar 4.9**, klik '*Estimate*' yang sudah dilingkari merah, kemudian pilih '*Transfer Function Model*' karena yang ingin dicari adalah model fungsi alihnya, maka akan muncul jendela '*Transfer Function*' seperti pada **Gambar 4.9** diatas. Dalam kerja praktik ini menggunakan *dataset* data masukan dan keluaran. Disini penulis memilih 3 pole dan 1 zero dengan *initial condition zero*. Pemilihan pole dan zero ditentukan berdasarkan penelitian sebelumnya yang dilakukan oleh Setiawan pada tahun 2012 tentang kontrol kecepatan motor DC dengan metode PID menggunakan Visual Basic 6.0 dan mikrokontroler ATmega 16 berhasil menghasilkan *best fit* 93.35% dengan 3 pole dan 2 zero.

Proses estimasi parameter pada kerja praktik ini dilakukan sebanyak 5 kali dengan target minimal menghasilkan *best fit* yang sama atau lebih tinggi dari penelitian sebelumnya yang dilakukan oleh Setiawan pada tahun 2012 dan Tang pada tahun 2017. Hasil percobaan pertama mengikuti parameter yang dilakukan oleh Setiawan yaitu 3 pole dan 2 zero menghasilkan dengan *dataset* sebanyak 10000 data menghasilkan *best fit* 62.42% dengan sistem yang tidak stabil, percobaan kedua menggunakan menggunakan data yang sama dan mengurangi zero menjadi 1 menghasilkan *best fit* 62.41% dengan sistem yang stabil. Percobaan ketiga menggunakan data yang lebih sedikit, *dataset* sebanyak 1000 data dengan 3 pole dan 1 zero menghasilkan *best fit* 89.3% dengan sistem yang stabil. Disini bisa dilihat bahwa penggunaan *dataset* 1000 data menghasilkan *best fit* yang lebih baik. Percobaan 4 dan 5 menggunakan *dataset* sebanyak 980 data dengan mengabaikan data awal motor saat pertama berakselerasi. Pada percobaan 4 menggunakan 3 pole dan 2 zero menghasilkan *best fit* 97.54%. percobaan 5 menggunakan data yang sama namun dengan 3 pole dan 1 zero menghasilkan *best fit* sebesar 97.64% dengan persamaan fungsi alih $\frac{45.71S + 0.6757}{S^3 + 14.15S^2 + 24.59S + 10.93}$. Sehingga bisa dikatakan fungsi alih dari motor DC PG45 dengan *driver motor* EMS 30A H-Bridge pada robot Dipi-Mecaro adalah $\frac{45.71S + 0.6757}{S^3 + 14.15S^2 + 24.59S + 10.93}$. Hasil ini lebih baik dari penelitian terbaru identifikasi sistem dengan metode dan kontroler yang sama, yaitu dengan Arduino Mega2560. Pada tahun 2017 Tang melakukan penelitian identifikasi motor DC menggunakan *System Identifier Tool Matlab* dengan Arduino Mega2560 menghasilkan *best fit* hanya sebesar 68.71% dengan 1 zero dan 2 pole. Hasil final estimasi dapat dilihat pada **Gambar 4.10**. Secara *default* masukan dan keluaran variabel akan beganti nama menjadi u1 sebagai masukan dan y1 sebagai keluaran yang mana sebelumnya bernama x dan y. Semua *dataset* kerja praktik ini bisa diakses di <https://github.com/ahmaddidiks/SystemIdentifier/tree/master/matlab>.

```

From input "u1" to output "y1":
      45.71 s + 0.6757
-----
      s^3 + 14.15 s^2 + 24.59 s + 10.93
Name: lzero3pole
Continuous-time identified transfer function.

Use "tfdata", "getpvec", "getcov" for parameters and

Status:
Estimated using TFEST on time domain data "mydata".
Fit to estimation data: 97.54% (stability enforced)
FPE: 11.79, MSE: 11.68

```

Gambar 4.10 Fungsi Alih Motor PG45

4.3 Analisa Kestabilan Sistem

Analisa kestabilan fungsi alih dapat dilakukan dengan berbagai macam metode, salah satu metode untuk menganalisa kestabilan adalah dengan melihat karakteristik parameter dari fungsi alih yaitu akar-akar dari polenya. Sistem dikatakan stabil jika semua akar-akar persamaan karakteristiknya berada disebelah kiri sumbu y (Triwiyatno, 2019). Dalam kerja praktik ini penulis menggunakan metode ini untuk melihat kestabilan fungsi alih yang sudah didapatkan sebelumnya. Percobaan ini dilakukan menggunakan Matlab dengan *listing program* sebagai berikut:

```

%analisa kestabilan
%plot(x,y)
num = [45.71 0.6757];
den = [1 14.15 24.59 10.93];

%fungsi alih open loop
disp("Fungsi alih open loop")
sys = tf(num,den)
%fungsi alih close loop
disp("Fungsi alih close loop")
sys1 = feedback(sys,1)
%akar-akar persamaan karakteristiknya
disp("Akar-akar pers. karakteristiknya")
damp(sys1)
%Posisi akar2 pes. karakteristik
sgrid
pzmap(sys1)
grid on

```

Karena fungsi alih akan digunakan dalam metode kontrol maka fungsi alih yang sudah didapatkan dibuat menjadi loop tertutup terlebih dahulu dengan nama variable `sys1`. Dengan begitu akar-akar persamaan karakteristik yang dicari adalah dari fungsi alih yang sudah dibuat menjadi loop tertutup. Fungsi alih loop tertutup yang dihasilkan adalah $\frac{45.71s + 0.6757}{s^3 + 14.15s^2 + 24.59s + 10.93}$ dengan akar-akarnya berada pada -0.171, -6.99 + 4.37i dan -6.99 - 4.37i. Semua akar-akarnya minus yang berarti bahwa fungsi alih loop tertutup tersebut stabil sehingga tidak perlu memodifikasi fungsi alih yang sudah di dapatkan agar stabil. Hasil lengkap dari uji kestabilan fungsi alih loop tertutup ditampilkan pada **Gambar 4.11** dan letak *pole zero mapping* pada **Gambar 4.12**.

```

sys1 =

      45.71 s + 0.6757
    -----
      s^3 + 14.15 s^2 + 70.3 s + 11.61

Continuous-time transfer function.

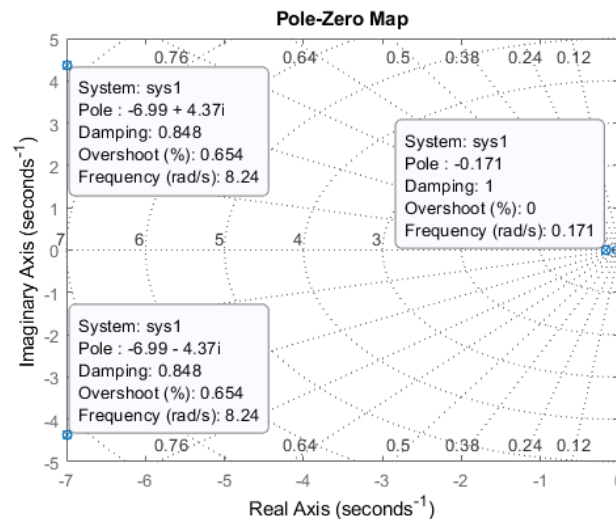
Akar-akar pers. karakteristiknya

      Pole           Damping      Frequency      Time Constant
                                (rad/seconds)    (seconds)

-1.71e-01           1.00e+00       1.71e-01       5.85e+00
-6.99e+00 + 4.37e+00i  8.48e-01       8.24e+00       1.43e-01
-6.99e+00 - 4.37e+00i  8.48e-01       8.24e+00       1.43e-01
>>

```

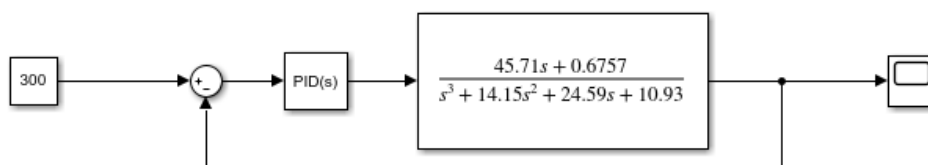
Gambar 4.11 Analisa Kestabilan Fungsi Alih Loop Tertutup



Gambar 4.12 Pole Zero Mapping

4.4 Pencarian Parameter PID

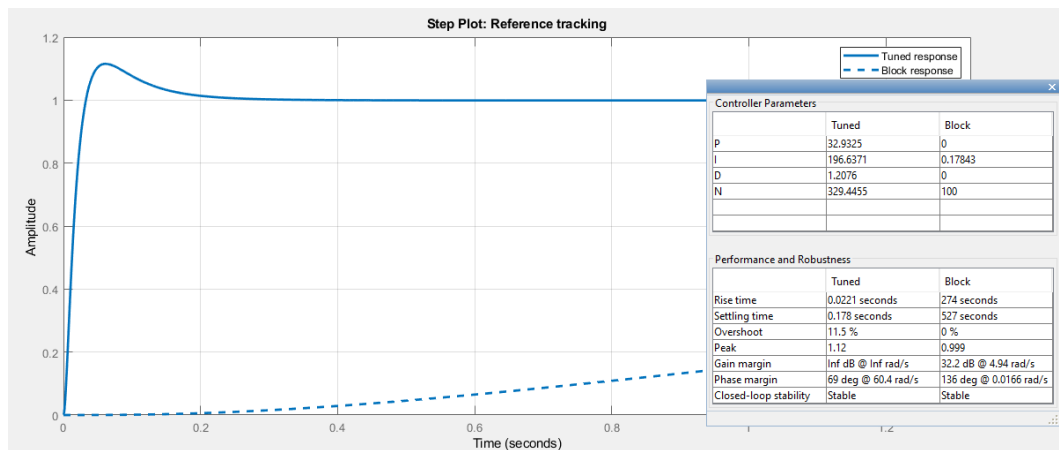
Setelah dipastikan fungsi alih yang didapatkan stabil, kontrol kecepatan motor DC PG45 dengan metode kontrol. Dalam kerja praktik ini dilakukan dengan metode kontrol PID. Pencarian parameter KP, KI, dan KD dilakukan secara *trial error* menggunakan *PID Tunner Matlab*. Teknik ini digunakan menggunakan Simulink dengan fungsi alih yang sudah didapatkan sebelumnya. Masukan dengan nilai konstan dan keluaran dipantau dengan *scope*. Rangkain simulasi simulink yang dilakukan dapat dilihat pada **Gambar 4.13**.



Gambar 4.13 Rangkaian Simulasi Simulink

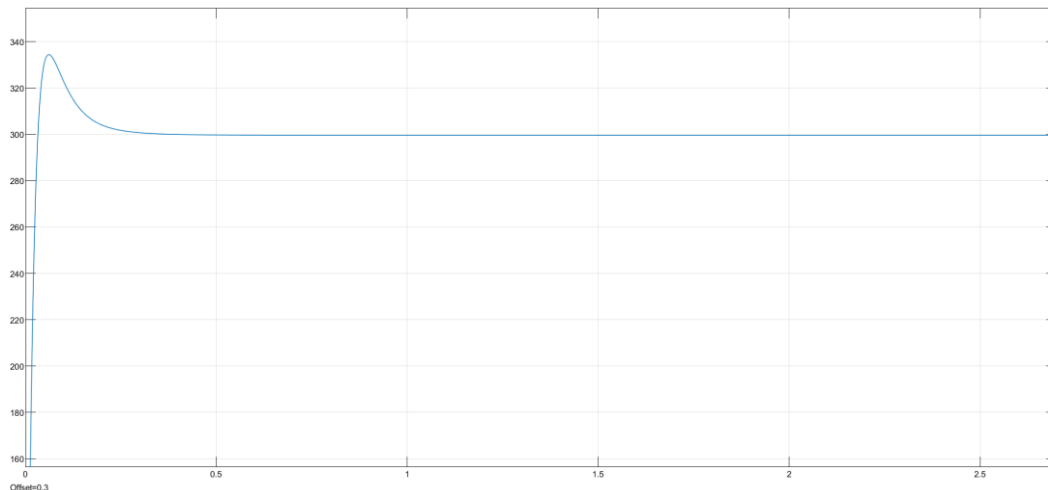
Penentuan parameter menggunakan *PID Tunner*, *trial error* dengan menggeser *time respon* untuk mendapatkan karakteristik kontrol yang dikehendaki. Pemilihan parameter KP, KI, KD pada kerja praktik ini didasari pada penggunaan

motor DC PG45 secara langsung di robot Dipo-Mecaro. Pengendalian motor menggunakan setpoint yang selalu berubah-ubah setiap waktu sesuai keluaran dari algoritma *plath planning*, Sehingga dibutuhkan kontrol yang mampu mencapai *steady state* secepat mungkin. Pencarian parameter dilakukan hanya sekali menggunakan fungsi alih yang sudah didapatkan sebelumnya dengan menggeser *time response* secepat mungkin namun mempertimbangkan *overshoot* yang dihasilkan. Hasil terbaik yang didapat adalah dengan $K_P = 32.9279740191956$, $K_I = 196.610205687394$, dan $K_D = 1.20746780266902$. karakteristik kontrol dari PID tersebut adalah dengan *Rise time* = 0.0221 detik atau 22.1ms, *Settling time* = 0.178 detik atau 178ms, *overshoot* = 11.5%.



Gambar 4.14 Hasil Parameter PID dengan *PID Tunner Matlab*

Selanjutnya dilakuakn simulasi dengan memasukan nilai setpoint 300RPM pada Simulink menghasilkan hasil simulasi seperti pada **Gambar 4.15**.



Gambar 4.15 Hasil Simulasi Dengan Setpoint = 300, $K_P = 32.9279740191956$, $K_I = 196.610205687394$, dan $K_D = 1.20746780266902$

Dari hasil simulasi memperlihatkan terjadi *overshoot* yang tinggi sampai hampir menyentuh 340 dari setpoint 300, namun setelah itu langsung mencapai *steady state* sebelum 0.5 detik.

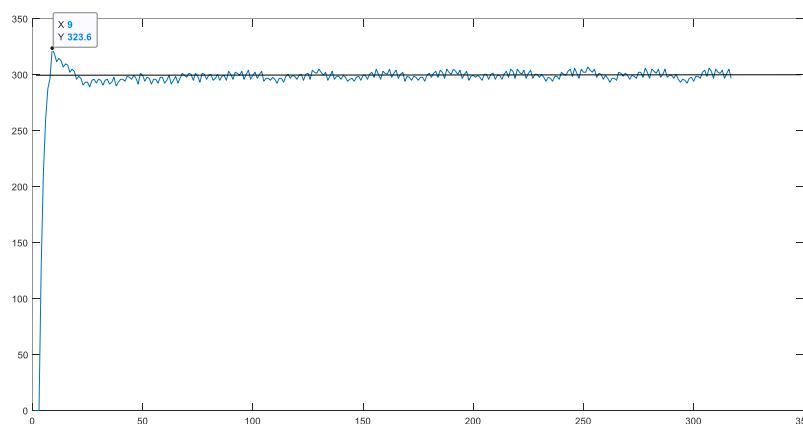
4.5 Aplikasi Hasil PID Untuk Kontrol Kecepatan Motor DC PG45

Hasil perhitungan parameter PID sebelumnya diaplikasikan langsung ke motor DC PG45. Rangkaian yang digunakan sama seperti tahap sebelumnya dari pengujian *driver motor*. Pada pengujian ini dilakukan dengan memberi setpoint pada program dan melihat performa dari kontrol PID yang sudah ditentukan sebelumnya. *Listing program* PID yang digunakan sebagai berikut:

```
void PIDTest(){
  myPID.run(); //call every loop, updates automatically at certain
  time interval
  analogWrite(motor,map(pwm, 0, 1000, 0, 255)); //use PID Lib
  //show PWM value
  Serial.print(pwm);
  Serial.print(" ");

  //show speed in RPM
  Serial.print(speedInRPM, 5);
  Serial.println(",");
  digitalWrite(LED_PIN, myPID.atSetPoint(10)); //light up LED when
  we're at setpoint +/-10 RPM
}
```

Keluaran sinyal PID diatur dengan rentang nilai 0-1000 yang kemudian di-*mapping* menjadi 0-255 karena limitasi PWM Arduino 8-bit. Hasil dari percobaan dengan setpoint 300 RPM terlihat pada gambar 4.14 yang mana seperti hasil simulasi, performa dari PID mempunyai *overshoot* yang cukup tinggi, pada percobaan mencapai 323.8 RPM (7.93%) dari setpoint 300 RPM, namun keadaan langsung stabil sebelum mencapai 50ms.

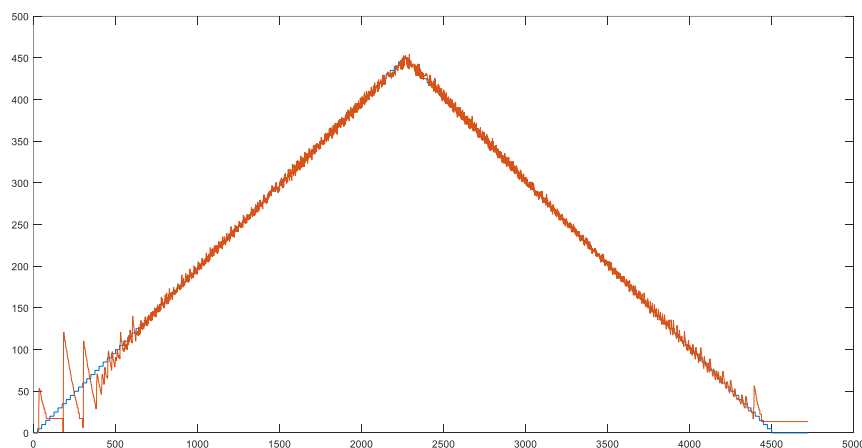


Gambar 4.16 Performa PID Dengan Setpoint 300 RPM

Hasil ini lebih cepat dari penelitian sebelumnya yang dilakukan oleh Djalal pada tahun 2017 menggunakan PID berbasis *Bee Colony* menghasilkan sistem yang mampu mencapai titik *steady state* pada 100ms. Hasil ini juga lebih cepat dari penelitian sebelumnya yang dilakukan oleh Masrukhan pada tahun 2016

mengontrol kecepatan motor DC dengan kontrol PID menggunakan *tunning ant colony* (ACO) menghasilkan kontrol motor DC yang mampu mencapai titi *steady state* pada waktu 550ms.

Karena penggunaan secara nyata dalam robot Dipo-Mecaro (Medical Assistant Robot) setpoin kecepatan berubah-ubah sesuai dengan keluaran algoritma *path planning* yang digunakan, sehingga performa PID diuji dengan melakukan perubahan setpoin RPM dengan perubahan setiap detik dari 0-450RPM dan kembali ke 0 RPM lagi. Perubahan setiap 100ms setpoin berubah ± 5 RPM. Hasil dari performa kontrol PID tersebut dapat dilihat pada **Gambar 4.17**.



Gambar 4.17 Performa PID Dengan Setpoin Berubah-ubah

Dari **Gambar 4.17** garis warna biru adalah setpoin yang diberikan dan garis warna coklat adalah kecepatan dari motor PG45. Bisa dilihat pada awal gerak setpoin kecepatan dibawah 100RPM kontrol PID tidak terlalu bagus mengikuti setpoin, disaat di jangkauan yang sama saat berhenti performa kontrol kecepatan bisa mengikuti dengan baik. Setelah setpoin diatas 100RPM performa kontrol kecepatan sangat baik dengan perubahan kecepatan tiap 100ms mampu secara cepat menuju *steady state* atau stabil.

BAB V PENUTUP

5.1 Kesimpulan

Dari Kerja Praktik yang telah dilakukan di PT. Permalat Berdikari Jaya, dapat diambil kesimpulan sebagai berikut:

1. Diketahui bahwa dapat dilakukan perhitungan identifikasi sistem model matematis fungsi alih dari motor DC PG45 dengan data masukan PWM acak dan keluaran kecepatan motor DC PG45 menggunakan *System Identifier Toolbox Matlab* menghasilkan *best fit* 97.64% dengan persamaan fungsi alih
$$\frac{45.71S + 0.6757}{S^3 + 14.15S^2 + 24.59S + 10.93}$$
2. Diketahui bahwa fungsi alih yang dihasilkan telah teruji stabil dengan menguji akar-akar persamaan karakteristiknya menghasilkan fungsi alih loop tertutupnya adalah
$$\frac{45.71S + 0.6757}{S^3 + 14.15S^2 + 70.3S + 11.61}$$
 dengan akar-akarnya berada pada -0.171 , $-6.99 + 4.37i$ dan $-6.99 - 4.37i$
3. Kontrol kecepatan motor DC PG45 dengan PID menggunakan fungsi alih pada poin 1 menghasilkan parameter $KP = 32.9279740191956$, $KI = 196.610205687394$, dan $KD = 1.20746780266902$.
4. Keandalan sistem kontrol PID pada poin 3 memiliki overshoot yang cukup tinggi namun mampu menuju *steady state* yang sangat cepat. Pada percobaan setpoint 300RPM *overshoot* yang dihasilkan mencapai 23.8 (7.93%) namun mencapai *steady state* kurang dari 100ms.

5.2 Saran

1. Dalam pengukuran kecepatan motor menggunakan sensor encoder internal memiliki PPR yang kecil sehingga keakurasiannya tidak tinggi, perlu adanya sensor encoder eksternal dengan spesifikasi PPR yang tinggi untuk memperbaiki akuisisi data
2. Perlu adanya perbaikan kontrol pada saat motor diatur pada setpoint kecepatan rendah (100RPM)
3. Menggunakan metode kontrol yang lebih baik dan modern, dengan adanya fungsi alih maka sistem bisa digunakan untuk membangun kontrol dengan metode kontrol yang lebih baik, misalnya dengan kontrol fuzzy atau dengan jaringan syaraf tiruan.

DAFTAR PUSTAKA

- Setiawan, I 2015, *Kontrol PID Untuk Proses Industri*, PT Elex Media Komputindo, Surabaya.
- Sumardi 2019, *Buku Ajar: Sistem Kontrol Multivariabel*, UNDIP Press, Semarang.
- Triwiyatno, A 2019, *Buku Ajar: Sistem Kontrol Analog*, UNDIP Press, Semarang.
- Desnanjaya, I.G.M.N. and Iswara, I.B.A.I., 2018. Trainer Atmega32 Sebagai Media Pelatihan Mikrokontroler Dan Arduino. *Jurnal RESISTOR (Rekayasa Sistem Komputer)*, 1(1), pp.55-64.
- Djalal, M.R., Nurohmah, H., Imran, A. and Yunus, M.Y., 2017. Aplikasi Metode Cerdas untuk Optimasi Controller PID Motor DC Berbasis Firefly Algorithm. *Jurnal Nasional Teknik Elektro*, 6(2), pp.76-83.
- Masrukhan, M.N., Mulyo, M.P., Ajiatmo, D. and Ali, M., 2016. Optimasi Kecepatan Motor DC Menggunakan Pid Dengan Tuning Ant Colony Optimization (ACO) Controller. *SENTIA* 2016, 8(2).
- Naung, Y., Schagin, A., Oo, H.L., Ye, K.Z. and Khaing, Z.M., 2018, January. Implementation of data driven control system of DC motor by using system identification process. In 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) (pp. 1801-1804). IEEE.
- Septiarini, A.D., 2018. *Pemodelan Matematika Kecepatan Motor DC Menggunakan Identifikasi Dengan Metode 2S* (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).
- Setiawan, D., 2017. Sistem Kontrol Motor Dc Menggunakan Pwm Arduino Berbasis Android System. *Jurnal Sains dan Teknologi Industri*, 15(1), pp.7-14.
- Tang, W.J., Liu, Z.T. and Wang, Q., 2017, July. Dc motor speed control based on system identification and pid auto tuning. In 2017 36th Chinese Control Conference (CCC) (pp. 6420-6423). IEEE.
- Wu, W., 2012. DC motor parameter identification using speed step responses. *Modelling and Simulation in Engineering*, 2012.
- Eitel, Elisabeth. 2014. *Basics of Rotary Encoders: Overview and New Technologies*. London. Machine Design Magazine.

BIODATA PESERTA KERJA PRAKTIK

Nama	Ahmad Didik Setiyadi	
NIM	21060117120024	
Tempat, Tanggal Lahir	Boyolali, 2 Mei 1998	
Jurusan	S1-Teknik Elektro Fakultas Teknik Universitas Diponegoro Konsentrasi Teknik Kontrol dan Instrumentasi	
Semester	VII	
IPK/SKS	3.5/127 SKS	
Alamat Rumah	Ngrawoh RT 09 RW 09, Pakang, Andong, Boyolali	
Alamat Kampus	Universitas Diponegoro Fakultas Teknik Departemen Teknik Elektro Jl. Prof. H. Sudharto, SH Tembalang-Semarang Telp. (024) 7460053-7460055 Fax (024)7460055	
E-mail	didik@student.undip.ac.id	
Motto Hidup	Cita-cita itu diraih bukan diangan	

Riwayat Pendidikan:

No.	Riwayat Pendidikan	Tahun
1	SDN Pakang	2005
2	SMP Negeri 1 Andong	2011
3	SMA Negeri 1 Gemolong	2014
4	Teknik Elektro Universitas Diponegoro	2017

Pengalaman Organisasi:

No.	Nama Organisasi	Jabatan	Tahun
1.	Forum Studi Teknik	Staff Riset, Staff Ahli	2018, 2019
2.	URDC(Undip Robotic Development Center)	Staff Programming, Ketua Divisi Kerja Programming, Ketua Divisi Robot Sepak Bola <i>Humanoid</i>	2018, 2019, 2020

Pelatihan dan *Training*:

No.	Nama Kegiatan	Tahun
2.	Latihan Keterampilan Manajemen Mahasiswa Pra Dasar Departemen Teknik Elektro (LKMPD)	2017
3.	FST Academy Super Training	2017

Prestasi:

Prestasi	Tingkat
Juara 2 Technocorner 2018 Divisi Soccerbot	Tingkat Universitas
Peserta PIMNAS ke-31	Tingkat Nasional

Juara Harapan Kontes Robot Sepak Bola Indonesia <i>Humanoid</i>	Tingkat Regional
Bronze Medal di Thailand Inventor Day 2019	Tingkat Internasional
Juara 2 Technocorner 2018 Divisi Soccerbot	Tingkat Universitas

LAMPIRAN



Foto Workspace 3D Printing dan Pengerjaan Hardware



Foto Robot Dipo-Mecaro

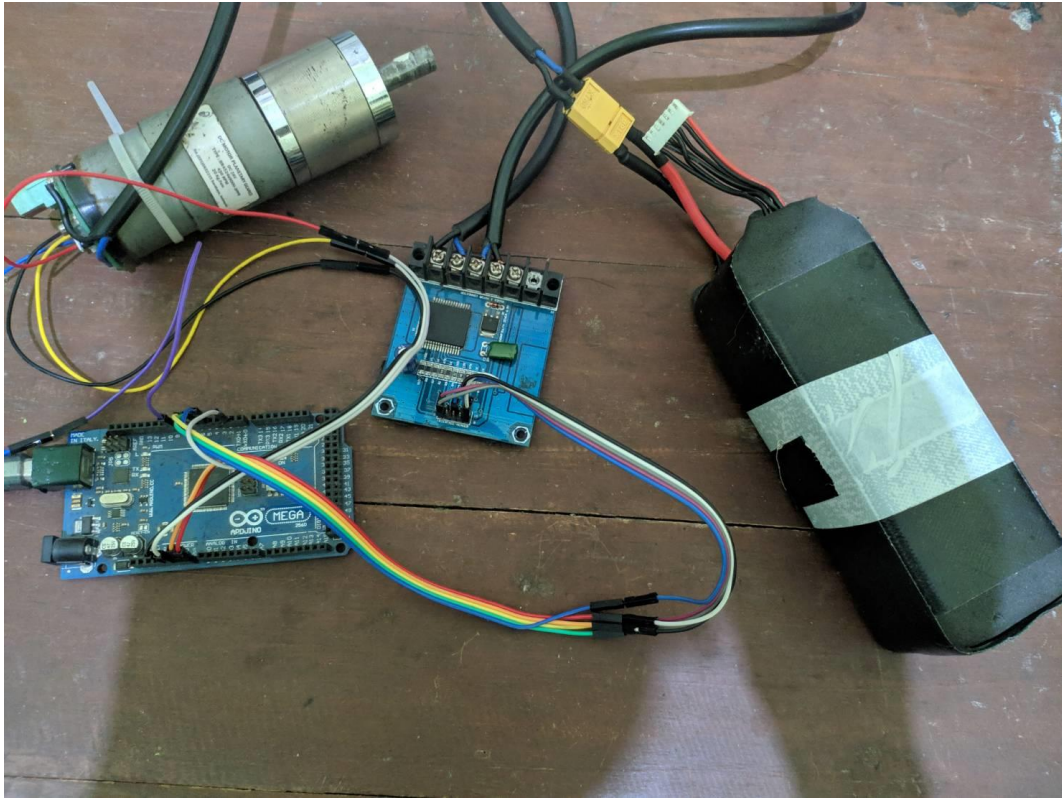
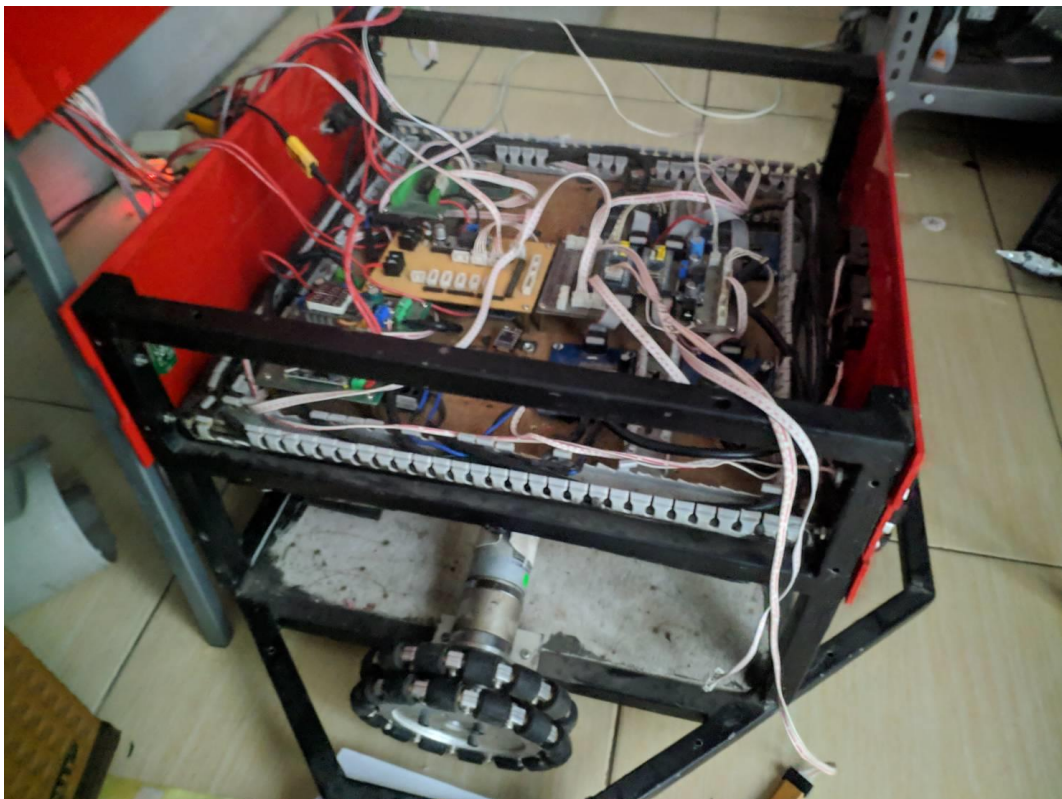


Foto Rangkaian Kontrol Motor DC PG45



Konfigurasi Motor DC PG45 dan Elektronika pada Robot Dipo-Mecaro



Foto Penulis dengan Robot Dipo-Mecaro

Listing Program Arduino IDE :

```

/* program for collecting dataset PG45
   Author      : Ahmad Didik Setiyadi
   Last edited  : 4 Des 2020

*/

#include <AutoPID.h>

double setPoint = 300;
double pwm, speedInRPM;

//waktu inc 0-1000 = 10000*10 (sampling time = 10ms) = 10000
#define waktuAmbilData 3000 //berapa lama ambil datanya (ms)

#define KP 32.9279740191956//2.0086510034733
#define KI 196.610205687394//29.8109534766749
#define KD 1.20746780266902
#define OUTPUT_MIN 0
#define OUTPUT_MAX 1000

#define LED_PIN 13
#define motor 5
#define pinA 6
#define pinB 7
#define ClockPin 2 // Must be pin 2 or 3
#define DataPin 9 // can be any other pin
    // My Encoder has 150 Clock pulses per revolution
    // note that 399999.8 = (60 seonds * 1000000 microseconds)micr
oseconds in a minute / 150 pulses in 1 revolution)
    // change the math to get the proper multiplier for RPM for yo
ur encoder
#define Multiplier 399999.8//150000.0 // don't forget a decimal plac
e to make this number a floating point number

long nilaiRandom;
unsigned long waktuAwal,waktuAkhir;

volatile long EncoderCounter = 0;

//input/output variables passed by reference, so they are updated au
tomatically
AutoPID myPID(&speedInRPM, &setPoint, &pwm, OUTPUT_MIN, OUTPUT_MAX,
KP, KI, KD);

```

```

void setup() {
    // put your setup code here, to run once:
    pinMode(motor, OUTPUT);
    pinMode(pinA, OUTPUT); digitalWrite(pinA, LOW);
    pinMode(pinB, OUTPUT); digitalWrite(pinB, HIGH); //FORWARD
    pinMode(ClockPin, INPUT);
    pinMode(DataPin, INPUT);
    pinMode(LED_PIN, OUTPUT);
    attachInterrupt(0,onPin2CHANGECallBackFunction,RISING);

    myPID.setBangBang(100);
    myPID.setTimeStep(1);

    Serial.begin(57600);
    Serial.println("Serial Test");
    delay(500);
    for (int i=0;i<=5;i++){
        Serial.print("Collecting dataset in "); Serial.println(5-i);
        delay(1000);
    }
    waktuAwal = millis();

    waktuAkhir = waktuAwal+waktuAmbilData;
}

void loop() {
    // put your main code here, to run repeatedly:
    static unsigned long SpamTimer;
    while(SpamTimer <= waktuAkhir ){
        if ( (unsigned long)(millis() - SpamTimer) >= (3)) {

            SpamTimer = millis();

            //run this code for motor driver test
            //motorDriverTest();
            //Run this code for collecting data
            //nilaiAcak();
            //Run this code for PID Test
            PIDTest();
        }
    }
    if(SpamTimer >= waktuAkhir) analogWrite(motor,0); //motor off
}

```

```

void onPin2CHANGECallbackFunction(uint32_t Time, uint32_t PinsChange
d, uint32_t Pins){
    static uint32_t lTime; // Saved Last Time of Last Pulse
    uint32_t cTime; // Current Time
    cTime = micros(); // Store the time for RPM Calculations
    int32_t dTime; // Delt in time

// Encoder Code
    bool DataPinVal = digitalRead(DataPin);
// We know pin 2 just went high to trigger the interrupt
// depending on direction the data pin will either be high or low
    EncoderCounter += (DataPinVal) ? 1 : -
1; // Should we step up or down?
// End Encoder Code

// calculate the DeltaT between pulses
    dTime = cTime - lTime;
    lTime = cTime;
    speedInRPM = Multiplier / ((DataPinVal) ? dTime: (-
1 * dTime)); // Calculate the RPM Switch DeltaT to either positive o
r negative to represent Forward or reverse RPM
}

void motorDriverTest(){
    nilaiRandom += 1;
    analogWrite(motor,map(nilaiRandom, 0, 1000, 0, 255)); //set pwm
motor as random number
    //show nilaiRandom value as a PWM
    Serial.print(nilaiRandom);
    Serial.print(" ");

    //show Encoder value
    //Serial.println(EncoderCounter);
    //Serial.print(" ");

    //show speed in RPM
    Serial.print(speedInRPM, 5);
    Serial.println(",");
}

void nilaiAcak(){
    nilaiRandom = random(100,700); //10-70%
    analogWrite(motor,map(nilaiRandom, 0, 1000, 0, 255)); //set pwm
motor as random number
    //show nilaiRandom value as a PWM
    Serial.print(nilaiRandom);
    Serial.print(" ");
}

```

```

        //show Encoder value
        //Serial.println(EncoderCounter);
        //Serial.print(" ");

        //show speed in RPM
        Serial.print(speedInRPM, 5);
        Serial.println(",");
    }

void PIDTest(){
    myPID.run(); //call every loop, updates automatically at certain
    time interval
    analogWrite(motor,pwm); //use PID Lib

    //show PWM value
    Serial.print(pwm);
    Serial.print(" ");

    //show Encoder value
    //Serial.println(EncoderCounter);
    //Serial.print(" ");

    //show speed in RPM
    Serial.print(speedInRPM, 5);
    Serial.println(",");
    digitalWrite(LED_PIN, myPID.atSetPoint(10)); //light up LED when
    we're at setpoint +-10 RPM
}

```

Semua data hasil percobaan dan program bisa diakses di:

<https://github.com/ahmaddidiks/SystemIdentifier>