



UNIVERSITAS
Dinamika

**Perancangan Modul Pembelajaran Sistem Kontrol Menggunakan PID
Berdasarkan Input User di Laboratorium FTI Universitas Dinamika**



KERJA PRAKTIK

Program Studi

S1 Teknik Komputer

UNIVERSITAS
Dinamika

Oleh:

Fillah Arhinzah

22410200004

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2025

Perancangan Modul Pembelajaran Sistem Kontrol Menggunakan PID

Berdasarkan Input User di Laboratorium FTI

Universitas Dinamika

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Strata Satu (S1)

Disusun Oleh :



Nama : FILLAH ARHINZAH
Nim : 22410200004
Program : S1 (Strata Satu)
Jurusan : Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2025

LEMBAR PENGESAHAN

Perancangan Modul Pembelajaran Sistem Kontrol Menggunakan PID

Berdasarkan Input User di Laboratorium FTI

Universitas Dinamika

Laporan Kerja Praktik oleh

FILLAH ARHINZAH

NIM. 22410200004

Telah diperiksa, diuji, dan disetujui

Surabaya, 7 Juli 2025

Disetujui:



Pembimbing

Pauladie Susanto, S.Kom., M.T.
NIDN. 0729047501

Penyelia



Laboratorium
UNIVERSITAS

Dinamika

Teguh Sutanto, M.Kom.
NIK. 000290

Mengetahui,
Ketua Program Studi Teknik Komputer



Fakultas Teknologi dan Informatika
UNIVERSITAS

Dinamika

Pauladie Susanto, S.Kom., M.T.
NIDN. 0729047501

**PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : **Fillah Arhinzah**
NIM : **22410200004**
Program Studi : **S1 Teknik Komputer**
Fakultas : **Fakultas Teknologi dan Informatika**
Jenis Karya : **Laporan Kerja Praktik**
Judul Karya : **PERANCANGAN MODUL PEMBELAJARAN
SISTEM KONTROL MENGGUNAKAN PID
BERDASARKAN INPUT USER DI LABORATORIUM
FTI UNIVERSITAS DINAMIKA**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.


Surabaya, 7 Juli 2025



Fillah Arhinzah
NIM : **22410200004**

ABSTRAK

Kerja praktek ini bertujuan untuk merancang dan mengembangkan modul pembelajaran sistem kontrol PID yang bersifat praktis dan interaktif, sebagai media bantu pembelajaran mahasiswa di Laboratorium FTI Universitas Dinamika. Modul ini dirancang agar mahasiswa dapat memahami penerapan prinsip dasar kontrol PID (Proportional, Integral, Derivative) secara langsung melalui pengaturan parameter K_p , K_i , dan K_d untuk mengontrol kecepatan motor DC berdasarkan perhitungan nilai RPM (Rotasi per Menit). Sistem ini mengatur nilai Set Point (kecepatan motor), sehingga fokus pembelajaran tertuju pada bagaimana perubahan parameter PID memengaruhi sistem. Input parameter PID dilakukan melalui Keypad 4x4, sedangkan nilai PID dan hasil RPM ditampilkan pada LCD 16x2. Motor DC akan berputar sesuai dengan hasil pengendalian sistem. Perangkat yang digunakan adalah Arduino IDE dengan Arduino UNO sebagai mikrokontroler utama. Dengan adanya modul ini, diharapkan mahasiswa tidak hanya memahami konsep teoritis PID, tetapi juga mampu mengamati respons sistem secara nyata dan mengembangkan keterampilan praktis dalam merancang sistem kontrol berbasis mikrokontroler.



Kata Kunci : Sistem Kontrol PID, Arduino UNO, Motor DC, RPM, Modul Pembelajaran.

UNIVERSITAS
Dinamika

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat yang telah diberikan-Nya, sehingga penulis dapat menyelesaikan laporan kerja praktek ini sebagai salah satu persyaratan untuk menyelesaikan program sarjana pada program studi S1 Teknik Komputer Universitas Dinamika Surabaya.

Laporan kerja praktik di Laboratorium IOT Univesitas Dinamika Surabaya, disusun berdasarkan hasil kerja yang telah dilaksanakan di bulan february sampai bulan maret. Dan hasil dari pelaksanaan kerja praktik ini, bertujuan untuk mengembangkan dan membuat Modul Pembelajaran Sistem Kontrol Menggunakan PID Berdasarkan Input User.

Penulis menyadari bahwa keberhasilan dalam pelaksanaan kerja pratik dan penyusunan laporan ini tidak lepas dari berbagai pihak. Oleh karena itu, penulis mengungkapkan terimakasih yang sebesar-besarnya kepada:

1. Allah SWT atas rahmat, hidayah, dan kesempatan yang diberikan, sehingga penulis dapat menyelesaikan Kerja Pratik ini dengan baik.
2. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer sekaligus dosen pembimbing yang telah memberikan arahan dan bimbingan selama proses kerja pratik dan peyusunan laporan ini.
3. Bapak Teguh Sutanto, M.Kom., selaku penyelia. Terima kasih atas bimbingan yang diberikan baik itu materi secara tertulis maupun lisan, sehingga penulis dapat melaksanakan Kerja Pratik di Laboratorium Fakultas Teknologi dan Informatika Universitas Dinamika.
4. Pihak-pihak lain yang tidak bisa saya sebutkan satu-persatu yang telah memberikan bantuan dan dukungan kepada penulis.

Surabaya, 7 Juli 2025

Fillah Arhinzah

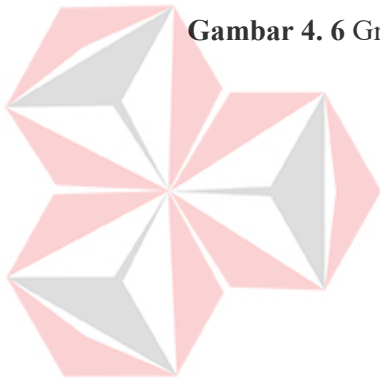
DAFTAR ISI

| | |
|---|-----|
| ABSTRAK..... | v |
| KATA PENGANTAR | vi |
| DAFTAR ISI | vii |
| DAFTAR GAMBAR | ix |
| DAFTAR TABEL | x |
| BAB I..... | 1 |
| PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 1 |
| 1.3 Batasan Masalah..... | 2 |
| 1.4 Tujuan | 2 |
| 1.5 Manfaat | 2 |
| BAB II GAMBARAN UMUM PERUSAHAAN | 4 |
| 2.1. Sejarah Singkat Perusahaan | 4 |
| 2.1.1. Visi dan Misi..... | 5 |
| 2.1.2. Tujuan | 5 |
| 2.1.3. Struktur Organisasi | 6 |
| 2.1.4. Informasi Kontak | 6 |
| 2.2. Program Studi S1 Teknik Komputer..... | 7 |
| 2.2.1. Visi dan Misi..... | 7 |
| 2.2.2. Tujuan | 8 |
| 2.2.3. Profesi Lulusan | 8 |
| BAB III..... | 10 |
| LANDASAN TEORI | 10 |
| 3.1 Sistem Kontrol | 10 |
| 3.1.1. Jenis-Jenis Sistem Kontrol..... | 10 |
| 3.1.2. Elemen Penting dalam Sistem Kontrol | 11 |
| 3.1.3. Representasi dan Analisis Sistem Kontrol | 11 |
| 3.1.4. Penerapan Sistem Kontrol dalam Kehidupan Nyata..... | 12 |
| 3.1.5. Perkembangan Modern dalam Sistem Kontrol | 13 |
| 3.2 PID (Proporsional-Integral-Derivatif) | 13 |
| 3.2.1. Aplikasikan Kontroller PID dalam Pengendalian Proses | 15 |

| | | |
|--------------------------------------|--|----|
| 3.2.2. | Keuntungan Kontoller PID | 16 |
| 3.2.3. | Tantangan dalam Penggunaan Kontroller PID | 17 |
| 3.3 | Arduino Uno | 17 |
| 3.4 | Keypad | 20 |
| 3.5 | LCD (Liquid Crystal Display) | 21 |
| 3.6 | Motor DC JGA25-370 | 23 |
| 3.7 | Ardumoto | 26 |
| BAB IV | | 29 |
| Hasil Pekerjaan dan Pembahasan | | 29 |
| 4.1 | Deskripsi Kerja Pratik | 29 |
| 4.2 | Diagram Blok | 30 |
| 4.3 | Perancangan Sistem | 35 |
| 4.3.1 | Perancangan Hardware | 35 |
| 4.3.2 | Perancangan Software | 36 |
| 4.4 | Implementasi dan Pengujian | 38 |
| 4.5 | Karakteristik Motor | 39 |
| 4.5.1 | Uji Respons Dinamis Tanpa Beban | 40 |
| 4.5.2 | Analisis Grafik dan Parameter Sistem | 41 |
| 4.5.3 | Pemodelan Sistem Orde Satu | 41 |
| 4.5.4 | Fungsi Alih (Transfer Function) | 41 |
| 4.6 | Pengujian Transef Function Pada Motor | 42 |
| 4.7 | Hasil Tuning PID Terbaik untuk Implementasi pada Arduino | 44 |
| 4.8 | Analisa dan Pembahasan Matlab dengan Arduino | 46 |
| BAB V | | 48 |
| Kesimpulan dan Saran | | 48 |
| 5.1 | Kesimpulan | 48 |
| 5.2 | Saran | 49 |
| Daftar Pustaka | | 50 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1. Struktur Organisasi | 6 |
| Gambar 3.1. PID (Sumber : https://www.kmtech.id) | 14 |
| Gambar 3.2. Arduino Uno (Sumber : https://www.pngfind.com) | 19 |
| Gambar 3. 3. Keypad (Sumber : https://www.fruugo.co.uk) | 20 |
| Gambar 3. 4. LCD (Liquid Crystal Display) | 22 |
| Gambar 3. 5. Motor DC JGA25-370 (Sumber : https://www.amazon.ca) | 24 |
| Gambar 3. 6. Ardumoto (Sumber : https://forum.arduino.c | 27 |
| Gambar 4. 1 Diagram Blok | 30 |
| Gambar 4. 2 Flowchart..... | 36 |
| Gambar 4. 3 Rangkaian lengkap | 39 |
| Gambar 4. 4 Grafik tanggapan motor tanpa beban..... | 40 |
| Gambar 4. 5 Grafik menggunakan MATLAB dengan nilai PID terbaik | 43 |
| Gambar 4. 6 Grafik secara Real Time | 44 |



UNIVERSITAS
Dinamika

DAFTAR TABEL

| | |
|---|----|
| Tabel 4. 1 konfigurasi pin | 36 |
| Tabel 4. 2 spesifikasi motor JGA25-370..... | 39 |
| Tabel 4. 3 Percobaan untuk mencari nilai PID terbaik dengan MATLAB | 43 |
| Tabel 4. 4 Hasil secara Real Time..... | 45 |



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem kontrol PID (Proportional, Integral, Derivative) merupakan salah satu metode pengendalian yang banyak digunakan dalam dunia teknik, terutama untuk mengatur suhu, tekanan, dan kecepatan motor. Namun dalam proses pembelajaran ini, mahasiswa sering kali hanya memahami konsep PID secara teori tanpa adanya praktik langsung yang mendukung. Hal ini membuat pemahaman menjadi kurang maksimal. Untuk itu, dibutuhkan sebuah modul pembelajaran yang sederhana dan mudah digunakan agar mahasiswa dapat memahami penerapan PID secara nyata.

Modul ini dirancang sebagai media pembelajaran di Laboratorium IoT Universitas Dinamika. Modul ini berfungsi untuk mengatur kecepatan motor DC berdasarkan nilai Input User, dengan parameter PID (K_p , K_i , dan K_d), dan set point yang dapat diatur melalui Keypad 4x4. Nilai RPM dan hasil perhitungan PID ditampilkan pada LCD 16x2, sedangkan nilai set point dan batas suhu telah ditentukan agar pengguna hanya fokus pada pengaturan PID. Sistem ini menggunakan Arduino UNO sebagai mikrokontroler dan Arduino IDE sebagai software pemrogramannya.

Dengan adanya modul ini, mahasiswa diharapkan dapat lebih memahami prinsip kerja sistem kontrol PID secara langsung. Modul ini juga memberikan pengalaman praktik dalam merancang sistem kontrol sederhana yang berbasis mikrokontroler, sehingga dapat mendukung proses pembelajaran yang lebih efektif di Laboratorium IoT Universitas Dinamika.

1.2 Rumusan Masalah

Dari latar belakang, maka dapat perumusan masalah sebagai berikut:

1. Bagaimana merancang modul pembelajaran yang dapat membantu mahasiswa memahami konsep dasar sistem kontrol PID secara praktis?
2. Apa saja pengaturan parameter PID (K_p , K_i , K_d) yang dapat diubah oleh pengguna untuk mempengaruhi kecepatan motor?
3. Bagaimana menampilkan hasil perhitungan PID dan nilai RPM secara jelas

dan mudah dipahami menggunakan LCD 16x2?

1.3 Batasan Masalah

Batasan masalah pada pelaksanaan kerja praktik ini adalah sebagai berikut:

1. Modul ini hanya mengatur kecepatan motor DC berdasarkan sistem kontrol PID yang menggunakan Input User.
2. Parameter PID yang digunakan hanya terbatas pada tiga komponen utama, yaitu Kp (proporsional), Ki (integral), dan Kd (derivatif), yang dapat diatur oleh pengguna melalui keypad.
3. Input yang digunakan dalam sistem ini berasal langsung dari pengguna (user) melalui perangkat Keypad 4x4, bukan dari sensor otomatis seperti sensor suhu atau sensor jarak.
4. Sistem tidak melakukan kalibrasi otomatis atau adaptasi parameter PID secara *real-time* berdasarkan kondisi lingkungan; seluruh pengaturan berasal dari input manual.

1.4 Tujuan

Tujuan Kerja praktik di Laboratorium Laboratorium IOT Universitas Dinamika Surabaya adalah sebagai berikut:

1. Merancang dan mengembangkan modul pembelajaran yang dapat membantu mahasiswa memahami konsep dasar sistem kontrol PID secara praktis dan interaktif.
2. Mengimplementasikan kontrol kecepatan motor DC dengan menggunakan parameter PID (Kp, Ki, Kd), dan set point yang dapat disesuaikan secara manual oleh pengguna melalui input keypad.
3. Menyediakan alat bantu pembelajaran pengguna untuk mengamati perubahan parameter PID secara langsung terhadap kecepatan motor DC yang akan ditampilkan menggunakan LCD 16x2.

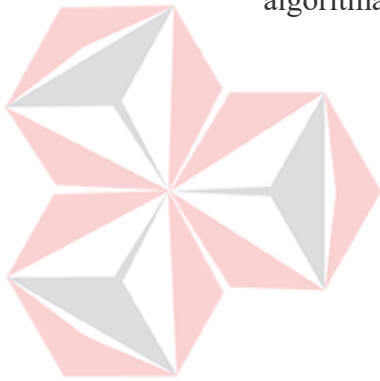
1.5 Manfaat

Modul pembelajaran sistem kontrol PID yang dirancang di Laboratorium FTI Universitas Dinamika memiliki berbagai manfaat, baik bagi pengguna maupun lingkungan laboratorium itu sendiri, antara lain:

1. Dengan modul ini, mahasiswa dapat lebih mudah memahami konsep dasar

dan aplikasi sistem kontrol PID secara praktis. Mereka akan belajar bagaimana cara mengatur parameter PID (K_p , K_i , K_d), set point sehingga dapat dilihat dampaknya terhadap pengendalian motor DC, yang dapat memperdalam pengetahuan mereka dalam bidang kontrol otomatis.

2. Modul ini berfungsi sebagai alat pendidikan yang sangat berguna di Laboratorium FTI Universitas Dinamika. Mahasiswa dapat memanfaatkan modul ini untuk eksperimen langsung dalam pengaturan kecepatan motor melalui input pengguna, yang akan mendukung pembelajaran dalam mata kuliah terkait otomasi dan sistem kontrol.
3. Melalui penggunaan perangkat keras seperti Arduino UNO, keypad 4x4, LCD, mahasiswa akan memperoleh keterampilan praktis dalam merancang dan mengimplementasikan sistem kontrol menggunakan algoritma PID.



UNIVERSITAS
Dinamika

BAB II

GAMBARAN UMUM PERUSAHAAN

2.1. Sejarah Singkat Perusahaan

Pada 30 April 1983 terjadi pengembangan teknologi dan informasi menjadi hal penting dalam pembangunan dan pengembangan nasional. Kedua hal tersebut juga harus diiringi dengan di bidang ekonomi dan bisnis untuk bisa bersaing di era yang terus berkembang. Seni dan budaya harus tetap dipertahankan agar identitas bangsa tidak musnah. Melalui empat (4) hal utama, yaitu kreatif, kolaborasi, dan komunikasi, para pendiri terdiri dari laksda. TNI (Purn) Mardiono, Ir. Andrian A.T, Ir. Handoko A.T, Dra. Rosy Merianti, Ak. Dalam bidang teknologi informasi dengan nama AKIS (Akademi Komputer dan Informatika Surabaya).

Pada 10 Maret 1984 izin operasional penyelenggara program Diploma III Manajemen Informatika diberikan kepada AKIS melalui SK Kopertis Wilayah VII Jawa Timur.

Pada 19 Juni 1984 AKIS yang berlokasi di Ketintang Surabaya memperoleh status terdaftar dari DIKTI.

Pada 20 Maret 1986 terus meningkatnya kebutuhan pendidikan, Yayasan Putra Bhakti memutuskan untuk merubah Akademi menjadi Sekolah Tinggi. AKIS (Akademi Komputer dan Informatika Surabaya) berubah menjadi Sekolah Tinggi Manajemen Informatika dan Teknik Komputer Surabaya, yang lebih dikenal dengan STIKOM Surabaya.

Pada 11 Desember 1987 STIKOM Surabaya membangun kampus pertama yang berlokasi di Jalan Kutisari No.66 Surabaya, yang diresmikan oleh Letnan Jendral TNI Wahono selaku Gubernur Jawa Timur pada saat itu.

Pada 28 Oktober 1997 Awal pemasangan tiang pancang pertama STIKOM Surabaya di Jalan Raya Kedung Baruk No.98 Surabaya bersamaan dengan Hari Sumpah Pemuda.

Pada 04 September Seiring dengan perubahan zaman serta kebutuhan masyarakat, STIKOM Surabaya resmi berubah menjadi Institut dengan nama Institut Bisnis dan Informatika STIKOM Surabaya yang memiliki 2 fakultas dengan 9 program studi.

Pada 29 Juli 2019 Melalui Surat Keputusan Riset Dikti, Institut Bisnis dan Informatika STIKOM Surabaya resmi berubah menjadi Universitas Dinamika yang memiliki 2 fakultas dengan 9 program studi, yakni Fakultas Teknologi dan Informatika, Prodi S1 Teknik Komputer, Prodi S1 Desain Komunikasi Visual, Prodi S1 Desain Produk, Prodi D4 Produksi Film dan Televisi, dan Prodi D3 Sistem Informasi. Serta Fakultas Ekonomi dan Bisnis (FEB) dengan Prodi S1 Manajemen, Prodi S1 Akuntansi, dan Prodi S1 Administrasi Perkantoran.

Pada 31 Mei 2021 Melalui Surat Keputusan Rektor, Universitas Dinamika melakukan perubahan struktur organisasi dengan membentuk fakultas baru, yakni Fakultas Desain dan Industri Kreatif (FDIK) dengan 3 program studi, yaitu Prodi S1 Desain Produk, Prodi S1 Desain Komunikasi Visual, dan D4 Produksi Film dan Televisi yang sebelumnya berada dibawah naungan Fakultas Teknologi dan Informatika (FTI).

2.1.1. Visi dan Misi

Visi

Menjadi *smart entrepreneurial university* berskala global yang produktif dalam berinovasi.

Misi

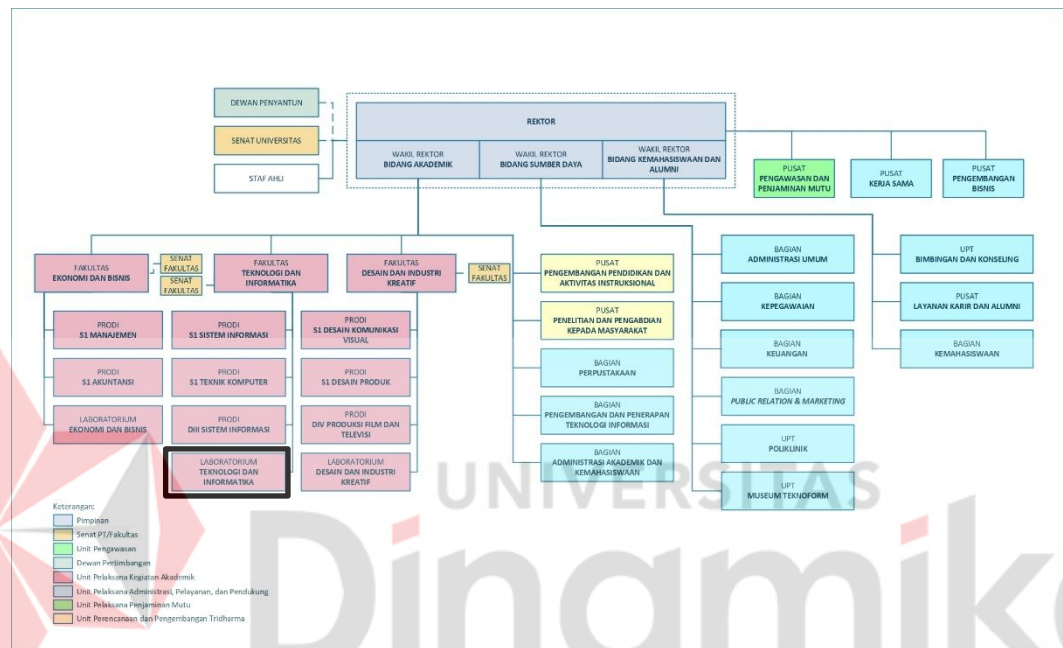
1. Menyelenggarakan dan mengembangkan pendidikan berbasis teknologi informasi yang bermutu dan berdaya saing global.
2. Melaksanakan penelitian yang berfokus pada pengembangan inovasi untuk mewujudkan entrepreneurial university.
3. Melakukan pengabdian untuk menyebar luaskan ipteks dan hasil inovasi bagi kesejahteraan masyarakat.
4. Melaksanakan kemitraan berskala global.
5. Mengembangkan bisnis dan kewirausahaan secara otonom yang akuntabel dan transparan.

2.1.2. Tujuan

1. Menyelenggarakan pendidikan yang berkualitas, inovatif, dan futuristik.
2. Menciptakan SDM berdaya saing global dan berjiwa *entrepreneur*.
3. Menghasilkan penelitian berkualitas dan berskala global.

4. Menghasilkan inovasi yang bernilai jual dan bermanfaat bagi masyarakat.
5. Melaksanakan diseminasi ipteks dan/atau hasil inovasi untuk meningkatkan kesejahteraan masyarakat.
6. Mewujudkan kemitraan berskala global.
7. Menjamin keberlanjutan Perguruan Tinggi.

2.1.3. Struktur Organisasi



Gambar 2.1. Struktur Organisasi

Pada bagian yang di tandai dengan kotak yang berwarna hitam disitulah tempat saya melakukan kerja pratik tersebut.

2.1.4. Informasi Kontak

Tempat : Jl. Raya Kedung Baruk 98 Surabaya

Email : official@dinamika.ac.id

Website : dinamika.ac.id

No telfon : +62(31)8721731

Sosial Media

Facebook : Universitas Dinamika

Youtube : Universitas Dinamika

Instagram : @universitasdinamika

2.2. Program Studi S1 Teknik Komputer

S1 Teknik Komputer mulai beroperasi mulai tahun 1991. Program Studi S1 Teknik Komputer bertujuan mencetak lulusan yang kompeten dalam bidang sistem komputer dan otomasi industri. Mahasiswa dibekali dengan berbagai mata kuliah terkait, seperti Microcontroller, Programmable Logic Controller (PLC), Sistem Digital, Jaringan Komputer, dan Robotika.

Lulusan program ini diharapkan mampu menganalisis, merancang, dan mengembangkan sistem otomasi industri berbasis komputer. Profesi yang dapat ditempuh antara lain Computer System Engineer, Embedded System Developer, dan Automation Engineer.

Program ini juga menekankan pada pemahaman teori dan praktik melalui berbagai kegiatan akademik, seperti praktikum, penelitian, dan proyek akhir. Selain itu, mahasiswa dibekali dengan keterampilan dalam pemrograman tingkat tinggi, pengolahan data, serta pengembangan sistem berbasis kecerdasan buatan. Dengan kurikulum yang terus disesuaikan dengan perkembangan industri, lulusan S1 Teknik Komputer Universitas Dinamika diharapkan dapat bersaing di dunia kerja maupun melanjutkan studi ke jenjang yang lebih tinggi.

Selain itu, mahasiswa juga diberikan kesempatan untuk mengikuti sertifikasi industri seperti Cisco Certified Network Associate (CCNA) dan sertifikasi pemrograman embedded system, guna meningkatkan daya saing di dunia kerja. Dengan adanya kerja praktik, mahasiswa dapat mengasah keterampilan yang relevan dengan kebutuhan industri dan memiliki pengalaman nyata sebelum terjun ke dunia profesional.

2.2.1. Visi dan Misi

Visi:

Mengembangkan keilmuan di bidang IoT yang didukung oleh kecerdasan artifial, dan diintegrasikan dengan konsep *technopreneurship*, sehingga mampu menciptakan inovasi yang bermanfaat bagi masyarakat dan industri berskala global.

Misi:

1. Mengembangkan pendidikan dan pengajaran di bidang Teknik Komputer yang bermutu, berwawasan global, dan mengarah pada *techonopreneurship*.
2. Melaksanakan penelitian di bidang Teknik Komputer yang inovatif dan solutif

bagi masyarakat dan industri berskala global.

3. Melaksanakan pengabdian atau penerapan hasil inovasi di bidang Teknik Komputer yang bermanfaat bagi masyarakat dan industri.

2.2.2. Tujuan

Tujuan Program Studi S1 Teknik Komputer adalah sebagai berikut :

1. Lulusan memiliki kemampuan menganalisis permasalahan sistem komputer khususnya pada aspek perangkat keras untuk menghasilkan solusi bagi organisasi.
2. Lulusan memiliki kemampuan menganalisis perangkat lunak (meliputi pemrograman anatar muka, pemrograman *real-time*) dan perangkat keras (meliputi pemantauan, pengendalian) sistem komputer sebagai solusi bagi permasalahan organisasi. Lulusan memiliki kemampuan menganalisis dan merancang sistem komputer dengan menerapkan sistem tertanam, *Internet of Things*(IoT), kecerdasan artifisial, dan atau jaringan komputer untuk menghasilkan solusi bagi organisasi.
3. Lulusan yang memiliki kemampuan dalam merumuskan keputusan yang tepat berdasarkan analisis informasi dan data, beretika, dan bertanggung jawab pada pekerjaan dalam lingkup tugasnya.

2.2.3. Profesi Lulusan

Profesi Lulusan Program Studi

1. **IoT Engineer** : Menyediakan produk dan atau solusi IoT sesuai dengan kebutuhan pengguna.
2. **Artificial Intelligent Engineer** : Membangun solusi berbasis kecerdasan artifisial (Artificial Intelligence/AI)
3. **Network Designer** : Melaksanakan penyediaan desain instalasi jaringan dan infrastruktur meliputi kegiatan pemetaan kebutuhan, monitoring dan pengawasan dampak design pembangunan dan pengembangan instalasi jaringan dan infrastruktur yang dibutuhkan oleh user sejalan dengan rencana dan pengembangan organisasi.
4. **Digital Computer Technology Advisor** : Memecahkan masalah teknis, memberikan saran tentang perangkat keras dan perangkat lunak yang tepat,

serta mengoptimalkan penggunaan teknologi dalam bisnis atau kehidupan sehari-hari.

5. **Industrial Automation** : Meningkatkan efisiensi dan produktivitas sistem produksi di industri



UNIVERSITAS
Dinamika

BAB III

LANDASAN TEORI

3.1 Sistem Kontrol

Sistem kontrol merupakan bagian dari ilmu teknik yang mempelajari bagaimana suatu sistem fisik atau elektronik dapat diarahkan perilakunya agar mencapai kinerja yang telah ditentukan. Tujuan utama dari sistem kontrol adalah memastikan bahwa keluaran (output) dari suatu proses mengikuti nilai yang diinginkan (setpoint) meskipun terdapat gangguan dari luar atau perubahan kondisi internal. Untuk mencapai hal ini, sistem kontrol harus mampu mengatur besaran masukan (input) secara teratur dan tepat waktu (Nise, 2015).

Dalam konteks rekayasa, sistem kontrol sangat penting karena hampir semua sistem teknik, mulai dari pengaturan suhu ruangan, kendaraan otonom, hingga robot industri, membutuhkan pengendalian yang presisi agar dapat bekerja secara stabil dan efisien. Proses pengendalian ini umumnya dilakukan berdasarkan representasi matematis yang menggambarkan hubungan antara input, proses, dan output dari sistem (Ogata, 2010).

3.1.1. Jenis-Jenis Sistem Kontrol

a. Sistem Kontrol Terbuka (*Open-Loop*)

Sistem kontrol terbuka merupakan sistem yang menjalankan perintah berdasarkan input yang telah ditentukan, tanpa adanya pengukuran atau evaluasi terhadap output yang dihasilkan. Sistem jenis ini umumnya lebih sederhana, cepat, dan ekonomis karena tidak membutuhkan sensor umpan balik. Namun, kelemahan utamanya adalah tidak adanya koreksi otomatis jika terjadi deviasi atau gangguan. Contoh sederhana adalah pengatur waktu pada oven listrik yang tetap akan mati setelah waktu habis, terlepas dari apakah makanan sudah matang atau belum.

b. Sistem Kontrol Tertutup (*Closed-Loop*)

Sebaliknya, sistem kontrol tertutup dilengkapi dengan mekanisme umpan balik yang memungkinkan pengukuran output secara langsung dan perbandingan terhadap referensi. Jika terdapat perbedaan antara keluaran aktual dan nilai yang diharapkan, sistem secara otomatis menyesuaikan input-nya. Sistem seperti ini lebih kompleks namun jauh lebih akurat dan stabil dalam kondisi dinamis. Salah

satu aplikasi umum adalah sistem kontrol kecepatan pada kendaraan, di mana sensor kecepatan memberikan informasi kepada kontroler untuk menyesuaikan posisi throttle (Dorf & Bishop, 2017).

3.1.2. Elemen Penting dalam Sistem Kontrol

Dalam praktiknya, sistem kontrol terdiri dari beberapa elemen utama yang saling terintegrasi, yaitu:

1. Sensor

Bertugas menangkap data dari lingkungan atau sistem seperti suhu, tekanan, atau posisi. Sensor mengubah besaran fisik menjadi sinyal listrik yang dapat dibaca oleh kontroler.

2. Kontroler

Merupakan pusat pengambilan keputusan yang menentukan bagaimana input harus disesuaikan berdasarkan sinyal error. Kontroler dapat menggunakan algoritma sederhana hingga kompleks, seperti kontrol PID (Proportional, Integral, Derivative).

3. Aktuator

Perangkat yang mengimplementasikan perintah dari kontroler ke sistem fisik, seperti motor, katup, atau elemen pemanas.

4. Plant (tanaman atau sistem):

Objek yang dikendalikan, baik berupa proses mekanik, elektronik, maupun kimia.

Setiap komponen di atas memiliki peran kritis dalam menjaga kesesuaian antara keluaran aktual dan keluaran yang diharapkan.

3.1.3. Representasi dan Analisis Sistem Kontrol

Agar dapat dirancang dan dianalisis dengan baik, sistem kontrol biasanya direpresentasikan dalam bentuk matematis. Representasi umum meliputi:

1. Fungsi Transfer: Digunakan untuk sistem linier dan stasioner, menggambarkan hubungan input-output dalam domain Laplace.
2. Model Keadaan (State-Space): Digunakan untuk sistem kompleks atau multivariabel. Representasi ini menggambarkan sistem dalam bentuk himpunan persamaan diferensial tingkat pertama.

3. Diagram Blok: Menunjukkan alur sinyal antar komponen sistem dan sering digunakan untuk menyederhanakan struktur sistem kontrol.

Tujuan analisis sistem kontrol adalah untuk memastikan bahwa sistem tersebut memenuhi kriteria stabilitas, akurasi, dan kecepatan respon yang sesuai. Parameter penting yang dianalisis meliputi rise time, settling time, overshoot, dan error steady-state (Ogata, 2010).

3.1.4. Penerapan Sistem Kontrol dalam Kehidupan Nyata

Sistem kontrol memiliki peranan yang sangat penting dalam berbagai bidang kehidupan modern karena kemampuannya dalam menjaga kestabilan, efisiensi, dan keakuratan suatu proses secara otomatis. Salah satu penerapan yang paling umum dan mudah ditemukan adalah pada sistem pengatur suhu otomatis, seperti pada air conditioner (AC), pemanas air, dan sistem HVAC (Heating, Ventilation, and Air Conditioning). Dalam aplikasi ini, sensor suhu akan mendeteksi kondisi aktual ruangan, kemudian kontroler membandingkan nilai suhu tersebut dengan suhu yang diinginkan (setpoint). Jika terjadi selisih (error), maka sistem akan mengaktifkan aktuator seperti kompresor atau elemen pemanas untuk mengoreksi suhu hingga sesuai dengan target (Ogata, 2010).

Penerapan sistem kontrol juga sangat dominan dalam dunia robotika industri. Di lini produksi otomatis, lengan robot harus dapat bergerak dengan tingkat presisi tinggi untuk melakukan tugas seperti perakitan, pengelasan, atau pengepakan. Sistem kontrol memastikan bahwa posisi, kecepatan, dan orientasi dari masing-masing segmen robotik sesuai dengan perintah yang diberikan oleh kontroler utama, berdasarkan data dari sensor posisi (seperti encoder) dan sensor gaya (Bolton, 2015). Penggunaan kontrol tertutup (closed-loop control) memungkinkan sistem mendeteksi dan memperbaiki kesalahan secara real-time, sehingga hasil kerja robot menjadi lebih presisi dan andal.

Di bidang transportasi, sistem kontrol telah menjadi komponen krusial dalam pengembangan kendaraan otonom. Sensor-sensor seperti LIDAR, kamera, dan GPS bekerja sama dengan kontroler untuk mengatur kecepatan, arah, serta menjaga jarak aman terhadap kendaraan di sekitarnya secara otomatis. Sistem semacam ini disebut dengan adaptive cruise control dan lane-keeping assist, yang merupakan bagian dari Advanced Driver Assistance Systems (ADAS). Teknologi

ini tidak hanya meningkatkan kenyamanan berkendara, tetapi juga sangat penting untuk keselamatan, terutama dalam sistem pengereman otomatis dan penghindaran tabrakan (Rajamani, 2012).

Dari contoh tersebut, dapat disimpulkan bahwa sistem kontrol bukan hanya berperan dalam meningkatkan performa dan efisiensi suatu sistem, tetapi juga menjadi komponen fundamental dalam pengembangan teknologi otomatis dan cerdas. Dengan integrasi sistem kontrol yang tepat, berbagai bidang seperti manufaktur, transportasi, dan penerbangan dapat terus berkembang menuju otomatisasi yang lebih luas dan lebih aman (Dorf & Bishop, 2017).

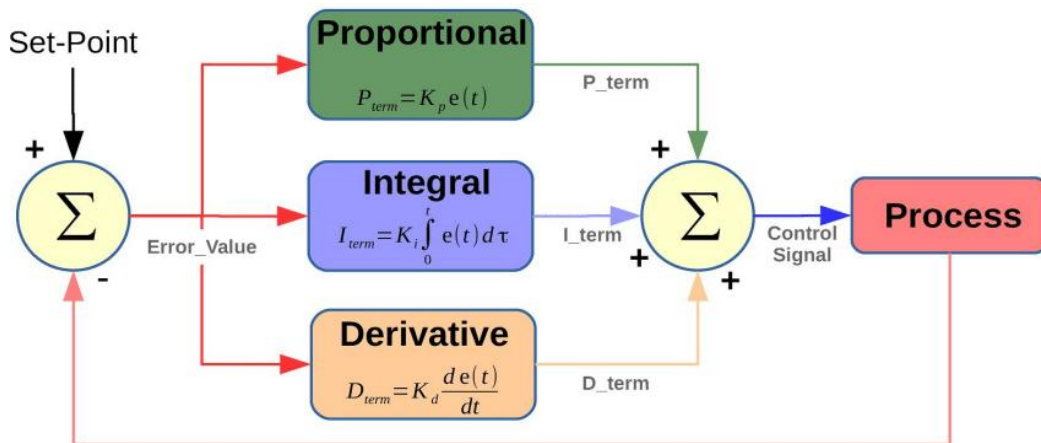
3.1.5. Perkembangan Modern dalam Sistem Kontrol

Seiring perkembangan teknologi, sistem kontrol tidak lagi hanya berbasis logika deterministik dan model matematis konvensional. Pendekatan baru berbasis kecerdasan buatan mulai banyak diadopsi, seperti kontrol fuzzy, logika adaptif, dan machine learning. Sistem kontrol cerdas ini memungkinkan pengendalian sistem nonlinier dan sistem yang sangat kompleks tanpa harus mengetahui model matematis secara lengkap. Selain itu, tren menuju sistem kontrol real-time dan kontrol terdistribusi berbasis jaringan menjadi sangat relevan dalam era industri 4.0 dan Internet of Things (IoT), di mana ribuan sensor dan aktuator dapat bekerja secara bersamaan dan saling berkomunikasi (Bolton, 2015).

3.2 PID (Proporsional-Integral-Derivatif)

Kontroller Proporsional-Integral-Derivatif, atau yang bisa dikenal sebagai Kontroller PID. Pengertian PID sendiri ialah algoritma kontrol umpan balik yang sering digunakan dalam industri atau dengan kata lain pengendalian otomatis yang beroperasi berdasarkan perbandingan antara nilai setpoint (nilai yang ingin dicapai/yang diinginkan) dan umpan balik (nilai aktual). Tujuan dari Kontroller PID ialah untuk mempertahankan variabel proses pada nilai setpoint yang diinginkan dengan mengoreksi perubahan yang terjadi. Kontroller PID dapat menghasilkan sebuah sinyal kontrol yang berfungsi untuk mengatur sistem agar mencapai nilai setpoint yang diinginkan dengan cepat dan tepat.

Prinsip Kerja Kontroller PID :



Gambar 3.1. PID
(Sumber : <https://www.kmtech.id>)

Prinsip kerja Kontroller PID ada tiga komponen utama yaitu :

1. Proporsional (P) Control

Komponen P atau Proporsional bertanggung jawab untuk memberikan respons kontrol sebanding dengan besarnya error. Semakin besar perbedaan antara variabel proses dan setpoint maka semakin besar respon kontrol yang dihasilkan.

2. Integral (I) Control

Komponen I atau integral berfungsi untuk mengatasi offset atau bias yang mungkin terjadi dalam sistem. Hal ini terjadi ketika error yang terus menerus terjadi, namun respons kontrol belum mencapai nilai yang tepat. Komponen I akan secara langsung bertahap mengakumulasi dan mengurangi offset ini hingga mencapai pengendalian yang akurat.

3. Derivatif (D) Control

Komponen D atau Derivative berperan dalam mengantisipasi perubahan cepat dalam variabel proses dengan perubahan nilai yang cepat dalam merespons tingkat perubahan saat ini dapat membantu mengurangi overshoot dan menstabilkan sistem. Ketika perubahan yang terjadi, komponen D akan memberikan respons kontrol yang tajam untuk mencegah variabel proses melalui setpoint.

Kontroller PID beroperasi dengan mengukur variabel proses aktual yang membandingkannya dengan setpoint yang ditargetkan. Selanjutnya, kontroller

menghitung dan membandingkan perbedaan antara keduanya, yang biasa dikenal dengan *error*. Setiap komponen PID memberikan kontribusi yang menghasilkan sinyal kontrol yang akurat untuk mengoreksi variabel proses.

Misalnya kita memiliki sebuah motor DC yang kecepatan putarnya 100 RPM (Revolutions Per Minute) dan ingin menjaga agar kecepatan putarannya tetap stabil, bisa menggunakan controller PID untuk melakukan hal tersebut.

Kontroller PID akan mengatur kecepatan motor secara Real-time dan membandingkan dengan nilai yang diinginkan, yaitu 100 RPM yang biasa disebut dengan (*setpoint*). Jika kecepatan motornya masih rendah dari 100 RPM, maka :

1. Komponen Proporsional (P) akan memberikan tegangan atau sinyal kontrol lebih besar ke motor, agar putaran motor lebih cepat dan mendekati kecepatan target.
2. Komponen Integral (I) akan menghitung selisih waktu ke waktu, lalu menyesuaikan dengan sinyal kontrol supaya kecepatan motor tetap stabil meskipun ada gangguan kecil, seperti perubahan beban.
3. Komponen Derivatif (D) akan merespons perubahan kecepatan yang terlalu cepat, yang berfungsi sebagai peredam, agar motor tidak mengalami perubahan kecepatan yang melonjak.

Dengan ketiga komponen ini, kecepatan motor bisa terjaga tetap stabil di sekitar 100 RPM, meskipun kondisi lingkungan atau beban motor berubah-ubah.

3.2.1. Aplikasikan Kontroller PID dalam Pengendalian Proses

Kontroller PID telah digunakan secara luas diberbagai bidang, salah satunya ialah dibidang industri, karena fleksibilitas dan kemampuan dalam mengatasi variasi kompleks pada sistem proses. Berikut adalah aplikasi yang relevan dari Kontroller PID meliputi :

1. Industri Manufaktur

Dalam industri manufaktur, Kontroller PID digunakan dalam pengendalian suhu, tekanan, kelembaban, dan kualitas produk. Contoh; dalam pemanasan dan pendinginan di pabrik kimia, pengendalian suhu dalam proses pemrosesan makanan, dan pengaturan tekanan dalam mesin cetak.

2. Otomasi Industri

Kontroller PID merupakan elemen kunci dalam sistem otomasi industri, seperti

kontrol robotik, pengendalian motor, dan pengaturan kecepatan produksi. Penerapan kontroller PID dalam otomasi industri membantu mencapai presisi dan efisiensi tinggi dalam operasi.

3. Pengendalian Proses Kimia

Dalam proses kimia, stabilitas dan akurasi sangatlah krusial. Kontroller PID sering digunakan untuk mengatur laju reaksi, konsentrasi, dan aliran bahan dalam sistem kimia kompleks, seperti dalam produksi petrokimia dan farmasi.

4. Pengaturan Kualitas Air

Kontroller PID diaplikasikan dalam sistem pengendalian kualitas udara pada gedung perkantoran atau fasilitas lainnya. Kontrol suhu, kelembaban, dan sirkulasi udara menjadi lebih efisien dan tepat dengan bantuan kontroller PID.

5. Industri Energi

Dalam bidang energi, Kontroller PID digunakan dalam pengendalian pembangkit listrik, pengaturan laju aliran bahan bakar, dan mengoptimalkan proses produksi energi dari berbagai sumber, termasuk tenaga surya dan angin.

3.2.2. Keuntungan Kontroller PID

Penggunaan Kontroller PID menawarkan sejumlah keuntungan yang signifikan dalam proses. Beberapa di antaranya adalah:

1. Stabilitas Proses

Kontroller PID dapat mempertahankan variabel proses pada *setpoint* yang diinginkan mengurangi fluktuasi dan menjaga stabilitas prosesnya.

2. Akurasi Kontrol

Kontroller PID mampu memberikan respon kontrol yang akurat dan responsif, sehingga memastikan variabel proses tetap berada dalam batas yang diinginkan.

3. Penyesuaian Otomatis

Kontroller PID dapat secara otomatis menyesuaikan diri dengan kondisi proses yang berubah, mengoptimalkan kinerja pengendalian.

4. Pengendalian yang Efisien

Dengan menggunakan Kontroller PID, proses pengendalian dapat dilakukan dengan lebih efisien, mengurangi konsumsi energi dan bahan baku.

5. Pengurangan Biaya Produksi

Ketepatan pengendalian yang ditawarkan oleh controller PID dapat mengurangi pemborosan dan kerugian dalam proses produksi, sehingga mengurangi biaya produksi secara keseluruhan.

3.2.3. Tantangan dalam Penggunaan Controller PID

Controller PID memiliki banyak kelebihan, namun dengan kelebihan tersebut ternyata ada beberapa tantangan yang dihadapi dalam penggunaannya:

1. Variabilitas Proses

Variabilitas dalam proses produksi dapat membuat penyetelan controller PID menjadi lebih rumit, karena setiap proses mungkin membutuhkan penyesuaian yang berbeda.

2. Gangguan Eksternal

Gangguan eksternal, seperti perubahan bahan baku atau kondisi lingkungan, dapat mempengaruhi kinerja controller PID dan memerlukan strategi pengendalian yang lebih canggih.

3.3 Arduino Uno

Arduino merupakan platform open-source berbasis mikrokontroler yang dirancang untuk memudahkan perancangan dan pengembangan sistem elektronik interaktif. Arduino pertama kali dikembangkan di Italia pada tahun 2005 oleh Massimo Banzi dan David Cuartielles sebagai sarana pembelajaran elektronika dan pemrograman mikrokontroler yang terjangkau dan mudah diakses (Banzi & Shiloh, 2014). Papan Arduino terdiri atas sebuah unit mikrokontroler, seperti ATmega328 pada varian Arduino Uno, yang berfungsi untuk menerima input dari sensor, memprosesnya sesuai instruksi program, dan kemudian menghasilkan sinyal output yang mengendalikan perangkat lain seperti motor, lampu LED, atau modul komunikasi (McRoberts, 2013).

Salah satu keunggulan utama Arduino adalah kemudahan penggunaannya baik dari sisi perangkat keras maupun perangkat lunak. Perangkat lunak Arduino IDE (Integrated Development Environment) mendukung pemrograman menggunakan bahasa pemrograman berbasis C/C++ yang telah disederhanakan dengan berbagai library bawaan, memungkinkan pemula untuk dengan cepat memahami dan mengembangkan program (Monk, 2016). Dari sisi perangkat keras,

Arduino dilengkapi dengan pin input/output digital dan analog, serta mendukung protokol komunikasi serial seperti UART, SPI, dan I2C, yang menjadikannya platform fleksibel untuk berbagai keperluan sistem tertanam dan kendali otomatis (Margolis, 2011).

Arduino telah menjadi fondasi dalam berbagai aplikasi sistem kontrol dan embedded system, seperti kendali pencahayaan otomatis, pengaturan suhu berbasis sensor termistor atau DHT11, sistem keamanan berbasis RFID, hingga robotika dan proyek berbasis Internet of Things (IoT). Kemudahan integrasi dengan berbagai sensor dan aktuator menjadikan Arduino sangat populer di kalangan akademisi, peneliti, maupun pengembang industri untuk prototyping sistem otomatis (Geddes, 2011). Fleksibilitas Arduino juga tampak dari kemampuannya dalam membaca berbagai jenis data sensor, memproses informasi dalam waktu nyata (real-time), serta menghasilkan aksi yang sesuai terhadap sistem fisik, menjadikannya sangat ideal sebagai pusat kendali dalam sistem kendali tertanam (Kumar & Singh, 2019).

Ekosistem open-source Arduino memungkinkan kontribusi global dalam bentuk modul tambahan (shields), dokumentasi, serta forum diskusi yang sangat aktif, sehingga mendukung penyelesaian masalah secara kolaboratif dan mempercepat pengembangan teknologi. Dengan demikian, Arduino bukan hanya alat bantu pendidikan, melainkan juga platform inovatif yang memfasilitasi lahirnya berbagai solusi teknologi di berbagai sektor (Banzi & Shiloh, 2014; Monk, 2016).

Arduino Uno merupakan salah satu varian paling populer dari keluarga papan pengembangan Arduino yang berbasis mikrokontroler ATmega328P. Papan ini dirancang untuk memberikan kemudahan dalam proses prototyping dan edukasi elektronika serta sistem kendali digital. Arduino Uno memiliki 14 pin input/output digital (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, satu port USB tipe B untuk pemrograman dan komunikasi serial, serta port daya eksternal yang mendukung tegangan masukan antara 7–12 volt (McRoberts, 2013). Mikrokontroler ATmega328P yang tertanam dalam Arduino Uno bekerja pada kecepatan 16 MHz dan memiliki 32 KB memori flash untuk menyimpan program, 2 KB SRAM, dan 1 KB EEPROM, yang cukup memadai untuk menangani berbagai aplikasi mikrokontroler berskala kecil hingga menengah (Margolis, 2011).



Gambar 3.2. Arduino Uno
(Sumber : <https://www.pngfind.com>)

Keunggulan utama dari Arduino Uno terletak pada kestabilannya, kemudahan pemrogramannya, dan kompatibilitasnya dengan berbagai modul eksternal seperti sensor suhu, modul Bluetooth, LCD, serta motor driver. Berkat penggunaan bootloader yang telah diinstal sebelumnya, pengguna tidak memerlukan perangkat tambahan seperti programmer eksternal untuk mengunggah program ke dalam chip mikrokontroler. Arduino Uno juga mendukung komunikasi serial (UART), SPI, dan I2C, yang memungkinkan koneksi dengan berbagai sensor dan perangkat komunikasi secara simultan, sehingga menjadikannya sangat ideal sebagai pusat pengendali dalam sistem tertanam (Monk, 2016). Dari segi perangkat lunak, Arduino Uno sepenuhnya didukung oleh Arduino IDE, yang kompatibel dengan berbagai sistem operasi termasuk Windows, macOS, dan Linux.

Selain digunakan secara luas dalam lingkungan akademik dan pendidikan sebagai alat bantu pembelajaran mikrokontroler, Arduino Uno juga banyak digunakan dalam penelitian dan proyek inovasi, seperti sistem monitoring suhu dan kelembaban otomatis, robot pengikut garis (line follower), sistem kontrol lampu otomatis berbasis sensor PIR, dan sistem keamanan berbasis keypad dan RFID. Fleksibilitas dan dokumentasi luas dari komunitas global menjadikan Arduino Uno sebagai standar de facto untuk pengembangan awal sistem kontrol berbasis mikrokontroler (Geddes, 2011; Kumar & Singh, 2019). Karena kemudahan dan kestabilannya, Arduino Uno terus menjadi pilihan utama dalam berbagai proyek teknik elektro, sistem otomasi, serta pengembangan produk berbasis IoT.

3.4 Keypad

Keypad merupakan perangkat input yang umum digunakan dalam berbagai sistem elektronik digital, terutama dalam sistem kendali tertanam (embedded control systems). Keypad umumnya berbentuk matriks tombol yang disusun dalam baris dan kolom, seperti pada keypad 4x3 atau 4x4, yang masing-masing memiliki 12 dan 16 tombol. Desain ini memungkinkan efisiensi dalam penggunaan pin I/O karena tombol-tombol yang tersedia dapat dialamatkan menggunakan teknik multiplexing. Sebagai contoh, pada keypad 4x4, meskipun terdiri dari 16 tombol, pengguna hanya memerlukan 8 pin input/output mikrokontroler—4 pin untuk baris dan 4 pin untuk kolom—karena masing-masing tombol berada di persimpangan antara satu baris dan satu kolom (Pillai, 2011).



Gambar 3. 3. Keypad
(Sumber : <https://www.fruugo.co.uk>)

Prinsip kerja keypad berbasis matriks didasarkan pada metode pemindaian (scanning). Arduino atau mikrokontroler lainnya akan secara bergantian mengaktifkan satu baris dalam satu waktu (mengeluarkan logika HIGH) dan memeriksa sinyal dari setiap kolom. Ketika sebuah tombol ditekan, koneksi listrik antara satu baris dan satu kolom akan terbentuk, memungkinkan sistem mendeteksi lokasi tombol tersebut. Proses ini berlangsung sangat cepat, sehingga pengguna merasakan interaksi real-time. Selain itu, pustaka seperti Keypad.h pada platform Arduino menyederhanakan implementasi dengan menyediakan fungsi untuk mendeteksi tombol yang ditekan, menangani efek bouncing pada tombol

(debounce), serta mengasosiasikan setiap posisi baris-kolom dengan karakter atau angka yang spesifik (Kumar, 2014).

Dalam konteks sistem kontrol, keypad berfungsi sebagai antarmuka utama bagi pengguna (user interface) untuk memasukkan perintah atau parameter. Aplikasi umumnya mencakup input numerik pada sistem pengunci pintu otomatis berbasis password, pengaturan suhu target pada sistem pengendali suhu, atau pengaturan kecepatan dan waktu pada pengendali motor. Input dari keypad biasanya diproses oleh mikrokontroler, yang kemudian menentukan aksi sistem berdasarkan logika program, seperti membandingkan input dengan password yang telah disimpan, atau mengirim nilai tersebut ke perangkat tampilan seperti LCD. Dalam sistem yang lebih kompleks, input keypad juga dapat digunakan untuk navigasi menu dan kontrol berbagai mode operasi perangkat (Mazidi, Naimi, & Naimi, 2008).

Keandalan keypad sebagai antarmuka input sangat bergantung pada pengolahan sinyal digital yang tepat, termasuk teknik debounce untuk mencegah pembacaan ganda akibat getaran mekanis saat tombol ditekan. Oleh karena itu, pemrograman pada sistem mikrokontroler harus memperhitungkan validasi input, pembatasan waktu respons, dan sistem proteksi terhadap kombinasi input yang salah. Hal ini sangat penting terutama dalam aplikasi yang menyangkut keamanan seperti sistem alarm atau brankas digital (Norris, 2011).

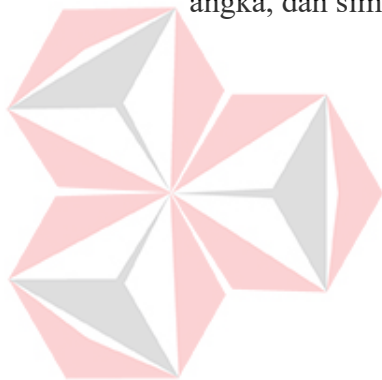
Dengan desainnya yang sederhana namun efisien, serta fleksibilitas integrasinya ke berbagai sistem mikrokontroler seperti Arduino, keypad tetap menjadi salah satu metode input yang paling banyak digunakan dalam pengembangan sistem kontrol digital, baik pada level edukasi, prototipe, maupun produksi industri.

3.5 LCD (Liquid Crystal Display)

LCD (Liquid Crystal Display) merupakan komponen tampilan yang banyak digunakan dalam sistem elektronik berbasis mikrokontroler untuk menyampaikan informasi secara visual dalam bentuk teks, angka, dan simbol. Salah satu modul LCD yang paling umum digunakan dalam proyek berbasis Arduino adalah LCD 16x2, yang mampu menampilkan dua baris teks dengan masing-masing 16 karakter. Modul ini biasanya dilengkapi dengan kontroler HD44780, sebuah chip pengendali

display yang telah menjadi standar industri untuk tampilan berbasis karakter dan kompatibel dengan berbagai platform mikrokontroler, termasuk Arduino (Sloan, 2010).

Secara prinsip, LCD bekerja dengan memanfaatkan sifat optik dari kristal cair (liquid crystal) yang mampu mengubah orientasi molekulnya ketika dikenai medan listrik. Perubahan orientasi ini mempengaruhi polarisasi cahaya yang melewatinya, sehingga dapat mengontrol apakah cahaya dapat diteruskan atau diblokir pada bagian tertentu dari tampilan. Karena pixel pada layar LCD tidak menghasilkan cahaya sendiri, maka diperlukan sumber cahaya latar (backlight), biasanya berupa LED, agar konten yang ditampilkan dapat terlihat oleh pengguna (McRoberts, 2013). Dalam modul karakter seperti LCD 16x2, setiap karakter ditampilkan dalam matriks 5x8 titik, memungkinkan pembentukan berbagai huruf, angka, dan simbol.



Gambar 3. 4. LCD (Liquid Crystal Display)
(Sumber : <https://proyectos.hellbot.xyz>)

LCD 16x2 dapat beroperasi dalam dua mode komunikasi, yaitu 8-bit dan 4-bit. Dalam mode 8-bit, semua delapan jalur data digunakan untuk mentransfer informasi dari mikrokontroler ke LCD, sedangkan dalam mode 4-bit, hanya empat jalur data yang digunakan, sehingga menghemat pin I/O tetapi memerlukan pengiriman data dalam dua siklus. Mode 4-bit lebih umum digunakan dalam aplikasi Arduino karena keterbatasan jumlah pin digital pada board seperti Arduino Uno. Untuk mempermudah proses komunikasi ini, Arduino menyediakan pustaka `LiquidCrystal.h` yang mendukung pengendalian fungsi dasar LCD, seperti

menampilkan teks, mengatur posisi kursor, dan menghapus tampilan secara efisien (McRoberts, 2013; Sloan, 2010).

Dalam sistem kontrol, LCD berfungsi sebagai alat penting dalam human-machine interface (HMI), yaitu perantara antara sistem otomatis dengan pengguna manusia. Informasi seperti nilai sensor (misalnya suhu atau kelembaban), status sistem (ON/OFF, READY, ERROR), data input dari keypad, dan berbagai notifikasi lainnya dapat ditampilkan secara real-time melalui LCD. Ini memungkinkan pengguna untuk memantau dan mengendalikan sistem secara lebih intuitif dan interaktif. Sebagai contoh, dalam sistem kontrol suhu berbasis Arduino, LCD dapat digunakan untuk menampilkan suhu saat ini serta suhu target yang telah diatur pengguna melalui keypad. Dalam aplikasi keamanan, LCD juga dapat digunakan untuk menampilkan perintah kepada pengguna seperti “Masukkan Password” atau “Akses Ditolak”, memberikan umpan balik langsung terhadap interaksi yang dilakukan.

Selain fleksibel dan hemat daya, keunggulan lain dari modul LCD berbasis HD44780 adalah dukungannya terhadap karakter khusus dan simbol buatan sendiri (custom character), yang memungkinkan desainer sistem untuk membuat ikon atau indikator grafis sederhana yang sesuai dengan kebutuhan aplikasi. Modul LCD juga dapat digunakan bersama dengan antarmuka I2C untuk mengurangi jumlah pin koneksi, memungkinkan pengguna menghubungkan lebih banyak perangkat lain ke sistem tanpa mengorbankan fungsi tampilan (Purdum, 2012).

Dengan sifatnya yang mudah diintegrasikan, didukung luas oleh komunitas dan pustaka perangkat lunak, serta fungsinya yang vital dalam memberikan umpan balik visual, LCD tetap menjadi komponen utama dalam sistem kendali tertanam, baik pada tahap prototipe maupun aplikasi produk akhir di dunia industri dan edukasi.

3.6 Motor DC JGA25-370

Motor DC JGA25-370 merupakan salah satu jenis motor DC bergearbox (geared DC motor) yang banyak digunakan dalam aplikasi robotika, kendaraan otomatis, dan sistem mekanik presisi lainnya. Motor ini terkenal karena kombinasi antara torsi tinggi, kecepatan rendah, serta ukurannya yang relatif kompak, sehingga ideal untuk aplikasi yang membutuhkan daya dorong cukup besar dengan

kontrol kecepatan yang stabil. Motor ini umumnya beroperasi pada tegangan kerja antara 6 hingga 12 volt DC dan memiliki torsi output yang tinggi karena dilengkapi dengan gearbox tipe planetary atau spur gear, tergantung pada variannya (Electronics-Salon, 2020).



Gambar 3. 5. Motor DC JGA25-370
(Sumber : <https://www.amazon.ca>)

Salah satu keunggulan utama dari JGA25-370 adalah adanya rotary encoder bawaan yang terpasang pada bagian belakang poros motor. Encoder ini biasanya berupa encoder inkremental dengan dua kanal (A dan B) serta terkadang dilengkapi sinyal kanal C sebagai sinyal referensi atau indeks. Encoder ini mampu menghasilkan hingga ratusan pulsa per rotasi (PPR), tergantung desainnya. Dengan adanya rotary encoder, motor ini dapat diintegrasikan dengan sistem kontrol tertutup (closed-loop), di mana posisi, kecepatan, atau jumlah rotasi poros motor dapat dikendalikan dan dimonitor secara real-time menggunakan mikrokontroler seperti Arduino atau STM32 (Singh & Sharma, 2019).

Dalam praktiknya, motor JGA25-370 dapat dikendalikan menggunakan sinyal PWM dari mikrokontroler untuk mengatur kecepatan rotasi, dan arah rotasi dapat diubah menggunakan rangkaian driver motor seperti L298N atau L298P (Ardumoto). Data dari encoder kemudian digunakan untuk menghitung kecepatan rotasi (dengan menghitung jumlah pulsa dalam interval waktu tertentu) atau posisi total (dengan menghitung kumulatif pulsa searah rotasi). Kombinasi ini sangat berguna dalam implementasi sistem kendali PID (*Proportional-Integral-Derivative*), di mana kontroler akan menyesuaikan sinyal PWM untuk menjaga kecepatan atau posisi motor agar tetap sesuai dengan nilai setpoint (Astrom &

Murray, 2010).

Rotary encoder adalah perangkat sensorik elektromekanis yang digunakan untuk mengubah gerakan rotasi mekanik menjadi sinyal digital yang dapat diinterpretasikan oleh sistem mikrokontroler atau komputer. Sensor ini memainkan peranan penting dalam sistem kontrol gerakan karena kemampuannya dalam menyediakan informasi tentang posisi sudut, kecepatan rotasi, dan arah putaran dari suatu poros. Rotary encoder banyak digunakan dalam sistem kontrol presisi tinggi seperti robotika, otomasi industri, printer 3D, dan sistem Computer Numerical Control (CNC), di mana akurasi dan umpan balik posisi sangat vital untuk keberhasilan operasi sistem (Mitchell, 2012).

Secara umum, terdapat dua jenis utama rotary encoder yang digunakan dalam praktik: encoder inkremental (incremental encoder) dan encoder absolut (absolute encoder). Encoder inkremental menghasilkan pulsa-pulsa digital berdasarkan perubahan posisi relatif poros. Ketika poros berputar, sensor menghasilkan pulsa-pulsa dari dua kanal yang disebut kanal A dan kanal B. Kedua kanal ini disusun dalam konfigurasi quadrature, yaitu memiliki perbedaan fasa sebesar 90 derajat, yang memungkinkan deteksi arah rotasi berdasarkan urutan sinyal yang diterima. Dalam implementasinya, pulsa dari kanal A dan B dapat dihitung untuk mengetahui sejauh mana poros telah berputar, sementara arah putar dapat diidentifikasi berdasarkan pola pergeseran fasa antara keduanya (Jones, 2015). Encoder jenis ini sangat cocok untuk penghitungan kecepatan dan jumlah langkah rotasi, namun tidak dapat memberikan informasi posisi absolut jika sistem dimatikan dan dinyalakan kembali.

Dalam sistem berbasis Arduino atau mikrokontroler lainnya, rotary encoder inkremental sering diproses menggunakan fitur interrupt untuk meningkatkan responsivitas dan akurasi pembacaan. Ketika salah satu kanal encoder berubah status logika (rising atau falling edge), interrupt akan dipicu, dan program akan menghitung pulsa serta menentukan arah putaran berdasarkan status kanal lainnya. Teknik ini memungkinkan mikrokontroler menangkap sinyal dari encoder secara real-time tanpa kehilangan pulsa, yang sangat penting terutama saat poros berputar dengan kecepatan tinggi.

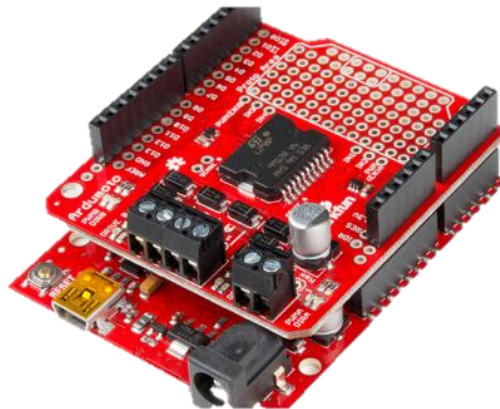
Rotary encoder sering dikombinasikan dengan motor DC, motor servo, atau

motor stepper dalam implementasi kontrol tertutup (closed-loop control system). Dalam skenario ini, encoder memberikan umpan balik posisi atau kecepatan kepada kontroler, yang kemudian membandingkan nilai aktual dengan nilai referensi (setpoint) untuk menghitung error. Selanjutnya, error ini digunakan oleh kontroler, seperti algoritma PID (Proportional-Integral-Derivative), untuk menyesuaikan sinyal kendali ke motor agar mencapai nilai target secara presisi (Astrom & Murray, 2010). Sistem seperti ini sangat efektif untuk menjaga kestabilan kecepatan putaran motor, mengontrol posisi aktuator dalam robotika, atau memastikan akurasi gerak dalam mesin CNC.

3.7 Ardumoto

Ardumoto adalah motor driver shield berbasis IC L298P yang dirancang secara khusus untuk digunakan bersama dengan papan mikrokontroler Arduino. Shield ini menawarkan solusi sederhana namun efektif dalam mengendalikan dua buah motor DC secara independen, baik dalam arah maju maupun mundur, serta menyediakan kemampuan untuk mengontrol kecepatan masing-masing motor melalui sinyal PWM (*Pulse Width Modulation*) (Margolis, 2011). Dengan desainnya yang kompatibel langsung dengan papan Arduino Uno atau Mega, Ardumoto sangat memudahkan perancangan sistem pengendali motor pada berbagai aplikasi robotika dan otomasi, karena tidak memerlukan rangkaian eksternal tambahan yang kompleks.

IC L298P yang menjadi inti dari Ardumoto merupakan sebuah dual full-bridge motor driver yang memungkinkan pengendalian arus dua arah melalui prinsip rangkaian H-Bridge. Dalam konfigurasi ini, setiap motor dikendalikan melalui dua pin logika (biasanya IN1 dan IN2) untuk menentukan arah rotasi, serta satu pin PWM untuk mengatur tingkat kecepatan motor. Dengan tegangan kerja maksimum motor hingga 12 volt dan arus hingga 2 ampere per kanal, Ardumoto dapat digunakan untuk berbagai motor DC kecil hingga menengah yang umum digunakan dalam sistem robotik edukasi maupun prototipe industri (Cook, 2010). Selain itu, shield ini menyediakan terminal sekrup (*screw terminal*) untuk memudahkan koneksi fisik ke motor, serta jalur suplai daya terpisah agar suplai motor tidak membebani catu daya utama Arduino.



Gambar 3. 6. Ardumoto
(Sumber : <https://forum.arduino.cc>)

Dari perspektif sistem kontrol, Ardumoto berperan sebagai penggerak aktuator utama, yaitu motor DC, yang dikendalikan berdasarkan logika dan algoritma yang diprogramkan dalam mikrokontroler. Dengan kemampuan untuk mengatur arah dan kecepatan motor, Ardumoto memungkinkan implementasi sistem kendali tertutup (*closed-loop control*) seperti kontrol PID (*Proportional-Integral-Derivative*). Dalam sistem seperti ini, sinyal dari sensor kecepatan atau posisi (misalnya rotary encoder) dikirim ke mikrokontroler untuk dihitung error terhadap nilai target, dan sinyal PWM disesuaikan untuk meminimalkan error tersebut. Implementasi ini sangat umum ditemukan dalam robot line follower, kendaraan otonom skala kecil, serta sistem lengan mekanik yang membutuhkan kontrol presisi gerakan motor.

Selain fleksibel dalam kontrol, Ardumoto juga mendukung integrasi dengan berbagai sensor dan antarmuka input lain seperti keypad atau sensor ultrasonik. Hal ini membuatnya sangat sesuai digunakan dalam sistem kendali adaptif, di mana masukan dari lingkungan memengaruhi perilaku motor. Sebagai contoh, dalam sistem kendali kendaraan robot berbasis Arduino, kombinasi antara sensor jarak, rotary encoder, dan driver Ardumoto memungkinkan kendaraan untuk menyesuaikan kecepatannya secara otomatis agar menghindari tabrakan atau mengikuti lintasan tertentu secara presisi.

Kepraktisan Ardumoto juga tercermin dari dokumentasi dan pustaka pendukung yang melimpah di komunitas Arduino. Dengan pustaka Arduino seperti

AFMotor atau kendali langsung menggunakan pin digital dan PWM, pengguna dapat mengakses kontrol motor secara cepat dan efisien tanpa harus memahami kompleksitas sirkuit H-Bridge secara mendalam. Hal ini menjadikan Ardumoto sebagai pilihan ideal bagi pelajar, peneliti, maupun insinyur dalam tahap prototipe maupun pengembangan sistem kendali tertanam.



UNIVERSITAS
Dinamika

BAB IV

Hasil Pekerjaan dan Pembahasan

4.1 Deskripsi Kerja Pratik

Dalam bab ini, dijelaskan proses Kerja Pratik yang dilakukan untuk Perancangan Modul Pembelajaran Sistem Kontrol Menggunakan PID Berdasarkan Input User. Perancangan modul pembelajaran ini bertujuan untuk menyediakan media praktikum yang interaktif dan aplikatif dalam mendalami prinsip pengendalian kecepatan motor DC. Modul ini dikembangkan dengan pendekatan yang integratif antara perangkat keras (hardware) dan perangkat lunak (software), dimana pusat kendali sistem menggunakan mikrokontroler Arduino Uno. Rangkaian ini mencakup sejumlah komponen utama yang saling terhubung dan memiliki fungsi yang spesifik, yakni LCD 16x2 sebagai alat untuk menampilkan informasi sistem secara real-time, Keypad 4x4 sebagai antarmuka input untuk memasukkan parameter PID dan set point, motor DC sebagai objek aktuasi, rotary encoder sebagai sensor umpan balik kecepatan, dan Ardumoto yang menggunakan LN298N sebagai media perantara distribusi daya dari mikrokontroler ke aktuator (driver motor).

Sistem ini bekerja dengan menerima inputan berupa nilai parameter PID yaitu (K_p , K_i , dan K_d) serta nilai set point kecepatan motor yang dimasukkan langsung oleh pengguna melalui keypad. Setelah input diterima, Arduino Uno memproses nilai-nilai tersebut dengan menggunakan algoritma PID yang ditanam dalam program. Output dari perhitungan PID berupa sinyal PWM (Pulse Width Modulation) dikirimkan ke motor driver L298N untuk mengendalikan tegangan yang diteruskan ke motor DC, sehingga kecepatan motor dapat disesuaikan dengan set point yang ditentukan.

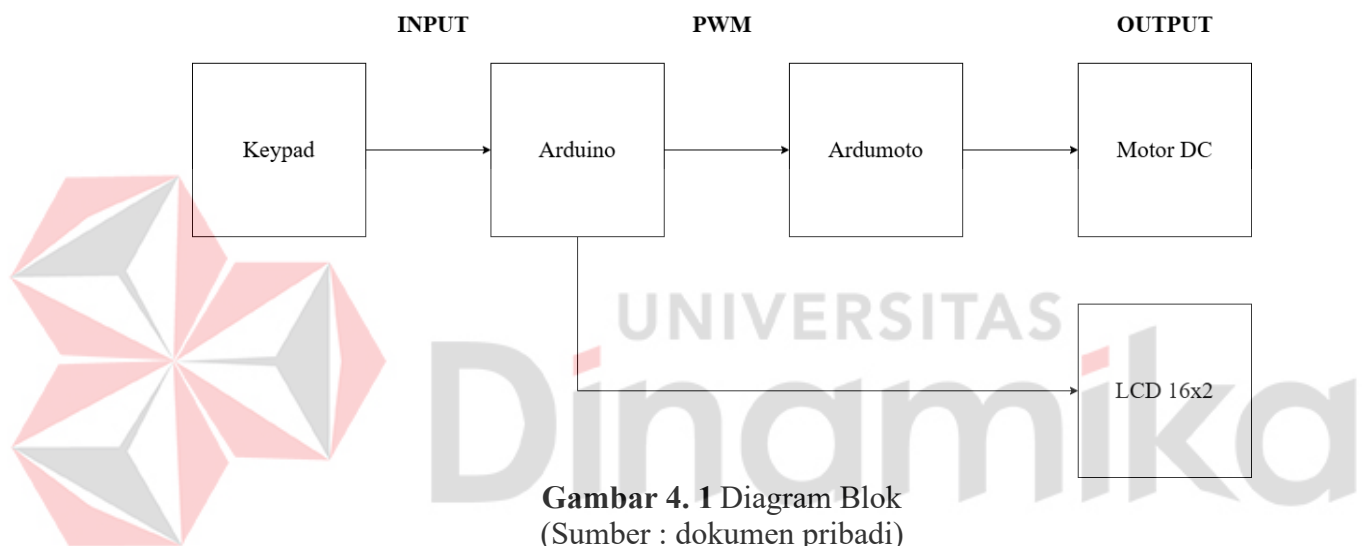
Sinyal yang dihasilkan oleh rotasi motor dalam satuan waktu ialah RPM (Rotasi per Menit), yang nantinya akan ditampilkan pada layar LCD 16x2 bersamaan dengan set point dan parameter PID. Penyajian informasi secara langsung melalui LCD memungkinkan pengguna untuk memantau dan mengevaluasi performa sistem kendali secara real-time.

Dengan konfigurasi tersebut, dapat melatih mahasiswa dalam integrasi

perangkat keras dan pemrograman mikrokontroler yang merupakan kompetensi penting dalam bidang teknik komputer dan sistem otomasi.

4.2 Diagram Blok

Diagram blok sistem yang dirancang menggambarkan arsitektur kerja dari modul pembelajaran sistem kontrol PID berbasis input pengguna. Diagram tersebut merupakan alur proses pengendalian motor DC dari awal hingga akhir, dimulai dari input, pengendali, hingga output. Setiap blok komponen dalam sistem memiliki peran spesifik yang saling terintegrasi dan berkontribusi dalam pengendalian yang presisi.



Gambar 4. 1 Diagram Blok
(Sumber : dokumen pribadi)

Pada bagian awal dari sistem kendali ini, terdapat komponen **keypad 4x4** yang berperan sebagai antarmuka utama antara pengguna (user) dengan perangkat mikrokontroler. Keypad tersebut memungkinkan pengguna untuk memberikan perintah secara langsung kepada sistem dengan cara memasukkan sejumlah parameter yang sangat krusial bagi proses pengendalian kecepatan motor. Adapun parameter yang dimasukkan meliputi tiga nilai utama dalam algoritma kontrol PID, yaitu:

1. Kp (konstanta proporsional): berfungsi untuk mengatur besar respon sistem terhadap error saat ini.
2. Ki (konstanta integral): berperan dalam menghilangkan error jangka panjang dengan cara mengakumulasi error dari waktu ke waktu.
3. Kd (konstanta derivatif): berfungsi untuk merespons perubahan error secara

cepat guna meredam osilasi atau overshoot.

Selain ketiga parameter tersebut, pengguna juga diminta untuk memasukkan nilai set point, yaitu target kecepatan (biasanya dalam satuan RPM) yang diharapkan dicapai dan dipertahankan oleh motor DC selama sistem berjalan.

Keypad 4x4 yang digunakan terdiri dari 16 tombol dalam konfigurasi baris dan kolom. Setiap tombol memiliki kode numerik atau fungsional tertentu yang telah diprogram dalam mikrokontroler. Melalui pemrograman ini, tombol-tombol pada keypad dikonversi menjadi perintah numerik yang dapat diproses oleh Arduino. Ketika pengguna menekan tombol pada keypad, sinyal digital akan dikirimkan ke pin input Arduino untuk dibaca sebagai nilai masukan.

Tahapan pembacaan input dilakukan secara sekuensial. Sistem menunggu input dari pengguna dalam urutan tertentu, misalnya diawali dari $K_p \rightarrow K_i \rightarrow K_d \rightarrow$ set point. Setelah setiap nilai dimasukkan, data tersebut akan disimpan dalam memori sementara (variabel global) yang akan digunakan dalam komputasi algoritma PID. Proses ini disertai dengan konfirmasi visual pada LCD agar pengguna mengetahui nilai apa yang sedang dimasukkan.

Penting untuk dicatat bahwa tahap input ini sangat menentukan kinerja sistem secara keseluruhan. Kesalahan dalam memasukkan parameter dapat mengakibatkan hasil kendali yang tidak sesuai, seperti osilasi yang tidak stabil, waktu respon yang terlalu lambat, atau ketidakmampuan sistem mencapai kecepatan target. Oleh karena itu, pemilihan nilai input yang tepat melalui keypad menjadi salah satu kunci utama dalam keberhasilan implementasi sistem kontrol PID.

Pada blok proses yaitu mikrokontroler Arduino yang berperan sebagai otak utama sistem yang bertanggung jawab mengelola seluruh proses komputasi dan komunikasi antar komponen. Setelah sistem diaktifkan dan pengguna memasukkan parameter melalui keypad, Arduino menyimpan nilai-nilai parameter kontrol PID yang meliputi K_p (*proportional gain*), K_i (*integral gain*), dan K_d (*derivative gain*), serta nilai set point kecepatan yang diinginkan.

Proses awal dilakukan dengan membaca nilai input dari keypad, kemudian Arduino menyusun logika untuk menunggu input secara bertahap dan menyimpannya ke dalam variabel global. Nilai-nilai tersebut selanjutnya digunakan

dalam algoritma PID yang ditanamkan dalam kode program Arduino. Algoritma PID ini bekerja dengan cara menghitung error, yaitu selisih antara set point dan kecepatan aktual motor (yang diperoleh melalui feedback dari sensor seperti rotary encoder). Berdasarkan error tersebut, Arduino menghitung nilai kontrol menggunakan formula PID dan menghasilkan sinyal PWM (*Pulse Width Modulation*) yang proporsional terhadap koreksi yang harus dilakukan pada kecepatan motor.

PWM merupakan sinyal digital yang dimodulasi berdasarkan lebar pulsa. Arduino menghasilkan PWM melalui pin digital tertentu (misalnya pin D3 atau D5), dan sinyal ini mewakili besar kecilnya tegangan rata-rata yang akan diberikan ke motor. Dengan kata lain, PWM adalah representasi analog dari hasil perhitungan PID yang memungkinkan Arduino mengatur kecepatan motor secara presisi.

Arduino juga mengatur komunikasi dengan LCD 16x2 sebagai media tampilan. LCD digunakan untuk menampilkan parameter yang sedang digunakan dan hasil kecepatan aktual motor secara real-time. Hal ini memberikan transparansi kepada pengguna terhadap apa yang sedang terjadi dalam sistem dan memungkinkan pemantauan yang efisien.

Dengan segala fungsinya, Arduino Uno dalam sistem ini tidak hanya bertindak sebagai pemroses pusat, tetapi juga sebagai penghubung antarkomponen, pemantau, serta pelaksana logika kendali waktu nyata yang esensial dalam implementasi kontrol PID.

Pada blok proses yang lainnya yaitu Ardumoto adalah driver motor berbasis IC L298N yang berfungsi sebagai perangkat penguat daya untuk sinyal PWM yang dihasilkan oleh Arduino. Perlu dipahami bahwa sinyal PWM dari Arduino hanya berupa sinyal logika dengan tegangan rendah (biasanya 5V) dan arus yang sangat kecil, sehingga tidak cukup kuat untuk menggerakkan motor DC secara langsung. Oleh karena itu, Ardumoto digunakan untuk menguatkan sinyal tersebut dan menyediakan tegangan serta arus yang memadai ke motor.

Dalam diagram blok, Ardumoto menerima input sinyal PWM dari Arduino dan mengubahnya menjadi sinyal tegangan yang sesuai untuk mengatur kecepatan motor. Modul ini memiliki dua jalur utama: satu jalur untuk input logika dari Arduino dan satu jalur catu daya terpisah yang digunakan untuk menyuplai

tegangan ke motor. Dalam sistem ini, fungsi utama Ardumoto mencakup:

1. Menerima sinyal PWM dari Arduino
2. Menyediakan arus tinggi yang dibutuhkan oleh motor
3. Mengatur arah dan kecepatan putaran motor berdasarkan sinyal kendali

Penggunaan driver motor seperti Ardumoto sangat penting agar motor dapat bekerja dalam kondisi beban tinggi tanpa membebani sirkuit kontrol utama (Arduino). Selain itu, beberapa driver motor juga memungkinkan pembacaan sinyal feedback seperti current sense atau status logic, yang dapat digunakan untuk deteksi proteksi arus lebih (overcurrent) atau gangguan motor lainnya.

Dalam konteks ini, Ardumoto bekerja secara sinkron dengan Arduino. Meskipun tidak mengambil keputusan logika, ia menjadi komponen vital dalam mewujudkan sinyal kendali yang telah dihitung. Secara fungsional, dapat dikatakan bahwa Ardumoto adalah eksekutor fisik dari perintah digital yang dikeluarkan oleh Arduino.

Pada blok output yaitu motor DC merupakan komponen utama dalam blok output yang berfungsi sebagai aktuator akhir dalam sistem kendali. Komponen ini menerima sinyal kendali berbentuk PWM (*Pulse Width Modulation*) yang telah diperkuat oleh driver motor (Ardumoto), dan mengubahnya menjadi gerakan mekanis berupa rotasi poros motor. Kecepatan rotasi motor bergantung pada lebar pulsa dari sinyal PWM semakin lebar pulsa ON dibanding OFF dalam satu siklus, semakin tinggi tegangan rata-rata yang diterima oleh motor, dan semakin cepat motor berputar.

Dalam konteks sistem kendali PID, motor DC menjadi objek kendali yang responsnya terus-menerus dipantau dan dikoreksi. Artinya, jika kecepatan aktual motor tidak sesuai dengan nilai set point yang dimasukkan oleh pengguna, maka sistem akan menghitung error, lalu menyesuaikan sinyal PWM agar motor dapat memperbaiki kecepatannya. Proses ini berlangsung secara berulang dan cepat (dalam hitungan milidetik), mencerminkan karakteristik dari sistem kendali loop tertutup (*closed-loop control system*).

Selain itu, motor DC yang digunakan dalam sistem ini dilengkapi dengan rotary encoder, yaitu sensor yang membaca jumlah putaran poros motor dan mengirimkannya kembali ke mikrokontroler sebagai data kecepatan aktual.

Feedback ini menjadi masukan penting dalam perhitungan PID, karena tanpa informasi aktual tentang kecepatan motor, sistem tidak dapat melakukan koreksi secara efektif.

Peran motor DC tidak terbatas pada sekadar menerima sinyal PWM, tetapi juga menjadi indikator langsung dari kualitas tuning parameter PID. Jika nilai parameter terlalu rendah atau terlalu tinggi, perilaku motor akan memperlihatkan gejala seperti lambat merespons, overshoot, atau osilasi. Oleh karena itu, motor DC bukan hanya sebagai output fisik, tetapi juga sebagai penyaji data performa sistem secara nyata.

Pada blok output lainnya yaitu LCD 16x2 merupakan komponen yang berfungsi sebagai display unit yang memberikan informasi secara real time mengenai kondisi sistem kepada pengguna. LCD ini terdiri dari dua baris (row) dengan masing-masing 16 karakter (column), cukup untuk menampilkan parameter kunci seperti nilai PID (K_p , K_i , K_d), nilai set point, serta RPM aktual dari motor DC.

LCD terhubung langsung ke Arduino melalui antarmuka paralel (biasanya menggunakan 6 pin digital), dan dikendalikan melalui pustaka (library) khusus seperti `LiquidCrystal.h` dalam lingkungan pemrograman Arduino. Fungsi utama LCD dalam sistem ini adalah untuk meningkatkan interaktivitas dan keterbacaan informasi dalam pengoperasian sistem kendali.

Saat pengguna memasukkan parameter melalui keypad, nilai-nilai tersebut langsung ditampilkan pada LCD, memungkinkan konfirmasi visual terhadap input. Setelah sistem berjalan, LCD akan terus memperbarui nilai RPM motor secara real time, sehingga pengguna dapat melakukan pemantauan langsung terhadap hasil kerja sistem. Dengan demikian, pengguna tidak memerlukan komputer atau perangkat eksternal tambahan untuk mengetahui kinerja sistem.

Fungsi LCD juga penting dari sisi evaluasi sistem. Misalnya, ketika RPM aktual terlalu jauh dari set point, pengguna dapat segera menyadari adanya kesalahan dalam tuning parameter PID dan melakukan penyesuaian. Hal ini membuat LCD menjadi alat bantu diagnostik yang sangat bermanfaat dalam eksperimen dan pengujian sistem kontrol.

Secara keseluruhan, LCD 16x2 dalam sistem ini berperan sebagai jembatan

komunikasi antara sistem digital dan manusia (human-machine interface) yang memungkinkan pengawasan dan intervensi manual dilakukan secara cepat dan tepat.

4.3 Perancangan Sistem

Perancangan sistem dalam pengembangan modul pembelajaran kontrol PID ini merupakan tahap yang sangat krusial karena menjadi fondasi utama dalam membentuk keseluruhan arsitektur kendali tertutup berbasis mikrokontroler. Dalam implementasinya, sistem dirancang agar dapat dioperasikan secara mandiri oleh pengguna melalui antarmuka masukan (input) dan keluaran (output) sederhana, serta dapat diintegrasikan dalam praktik pembelajaran di laboratorium pendidikan tinggi. Perancangan sistem ini secara umum terbagi menjadi dua komponen utama, yaitu perancangan perangkat keras (hardware) dan perangkat lunak (software), yang keduanya saling mendukung dalam menjamin fungsi sistem berjalan secara optimal dan real-time.

4.3.1 Perancangan Hardware

Perancangan perangkat keras atau hardware berfokus pada penyusunan konfigurasi fisik dari berbagai komponen yang digunakan dalam sistem kontrol PID. Komponen-komponen utama yang digunakan meliputi:

1. Arduino Uno : sebagai unit mikrokontroler yang mengatur seluruh proses logika dan komunikasi antar perangkat.
2. Keypad 4x4 : sebagai alat input manual dari pengguna untuk memasukkan nilai parameter PID yang meliputi (K_p , K_i , K_d) serta nilai set point.
3. LCD 16x2 : sebagai media tampilan output untuk menyajikan informasi seperti nilai parameter PID, nilai set point, dan nilai kecepatan motor secara aktual/*real-time*.
4. Driver motor (Ardumoto) : berfungsi sebagai aktuator untuk mengatur besar kecilnya arus yang disuplai ke motor DC berdasarkan sinyal PWM dari Arduino.
5. Motor DC dengan rotary encoder : sebagai elemen aktuasi dan sumber umpan balik/*feedback* dalam bentuk sinyal pulsa.

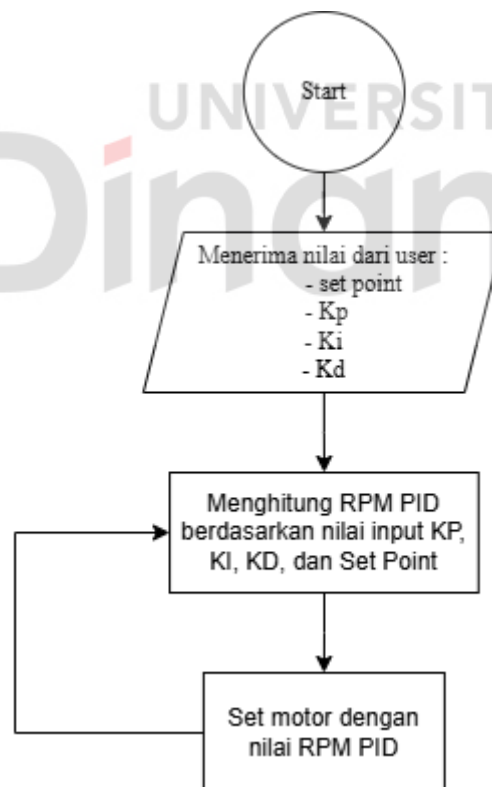
Setiap komponen yang dirangkai dalam satu sistem terpadu yang

mengutamakan efisiensi dan kemudahan dalam proses perawatan serta modifikasi. Hubungan antar komponen ditentukan berdasarkan alokasi pin yang tersedia pada Arduino Uno yang disusun sedemikian rupa untuk meminimalisir interferensi dan menjaga kestabilan sinyal. Berikut tabel konfigurasi pin.

| Komponen | Pin Arduino |
|---------------|----------------------|
| Keypad 4x4 | RowPins {8,A3,A2,A1} |
| | ColPins {12,11,10,9} |
| LCD I2C 16x2 | VCC (5v) |
| | GND (GND) |
| | SCL (A5) |
| | SDA (A4) |
| Encoder Motor | encoderPin (2) |
| Motor Driver | motorPin (3) |

Tabel 4. 1 konfigurasi pin
(Sumber : dokumentasi pribadi)

4.3.2 Perancangan Software



Gambar 4. 2 Flowchart
(Sumber :dokumentasi pribadi)

Flowchart yang ditampilkan di atas merupakan representasi visual dari alur kerja sistem kendali kecepatan motor DC menggunakan pendekatan kontrol PID (Proportional–Integral–Derivative) yang diimplementasikan melalui

mikrokontroler seperti Arduino. Diagram ini dirancang untuk menggambarkan proses kontrol yang berjalan secara berulang (looping) dan hanya berhenti ketika sistem dinonaktifkan secara manual oleh pengguna. Setiap blok dalam diagram memiliki fungsi yang sangat spesifik dalam menunjang keseluruhan kinerja sistem kendali.

Langkah pertama ialah start, proses ini terjadi ketika mikrokontroler diaktifkan, dan seluruh perangkat lunak dijalankan untuk menerima intruksi dari pengguna.

Langkah selanjutnya ialah menerima input dari pengguna yaitu proses penerimaan parameter dari pengguna untuk menginput empat nilai utama yaitu K_p , K_i , K_d dan set point. Input ini biasanya dimasukkan melalui keypad dan akan ditampilkan menggunakan LCD. Nilai-nilai tersebut akan disimpan oleh sistem dan digunakan sepanjang siklus kendali berlangsung.

Langkah ketiga ialah menghitung nilai PID setelah parameter dari pengguna diterima dan disimpan. Kemudian mikrokontroler menjalankan perhitungan PID berdasarkan nilai input tersebut. Sistem akan membaca kecepatan aktual motor melalui rotary encoder, kemudian menghitung error dari nilai selisih nilai set point dan nilai aktual. Perhitungan PID dilakukan dengan mempertimbangkan :

1. Komponen P untuk respons terhadap error saat ini.
2. Komponen I untuk memperbaiki error yang terjadi secara kumulatif.
3. Komponen D untuk mengantisipasi perubahan error yang mendadak.

Hasil perhitungan ini berupa sinyal yang umumnya berupa PWM (*Pulse Width Modulation*) yang akan dikirimkan ke driver motor untuk mengatur kecepatan.

Langkah keempat ialah mengatur kecepatan motor DC menggunakan hasil perhitungan PID. Sinyal PWM yang dihasilkan oleh mikrokontroler akan dikirim ke driver motor (Ardumoto) untuk mengatur besar tegangan rata-rata yang diterima oleh motor. Dengan demikian, kecepatan motor akan mendekati atau sesuai dengan set point.

Didalam flowchart diatas memperlihatkan adanya panah yang kembali dari blok terakhir menuju ke proses perhitungan PID. Hal ini menandakan bahwa sistem masuk ke dalam loop tak terbatas yang berjalan terus-menerus:

1. Membaca kecepatan motor.
2. Menghitung ulang PID berdasarkan error baru.
3. Menyesuaikan kembali kecepatan motor.

Loop ini akan terus berjalan hingga sistem dimatikan oleh pengguna. Konsep ini sesuai dengan prinsip kontrol loop tertutup (*closed-loop control*), di mana output sistem digunakan sebagai masukan untuk proses pengendalian berikutnya guna mencapai kestabilan dan akurasi.

4.4 Implementasi dan Pengujian

Setelah proses perancangan perangkat keras dan lunak selesai, maka selanjutnya ialah implementasi sistem secara keseluruhan. Implementasi sistem ini mencakup integrasi seluruh komponen untuk membentuk sistem kontrol PID yang utuh dan siap uji coba. Implementasi ini bertujuan untuk mengamati secara langsung bagaimana sistem bekerja dalam kondisi sebenarnya dan mengevaluasi efektivitas pengendalian kecepatan motor berdasarkan parameter yang dimasukkan oleh pengguna.

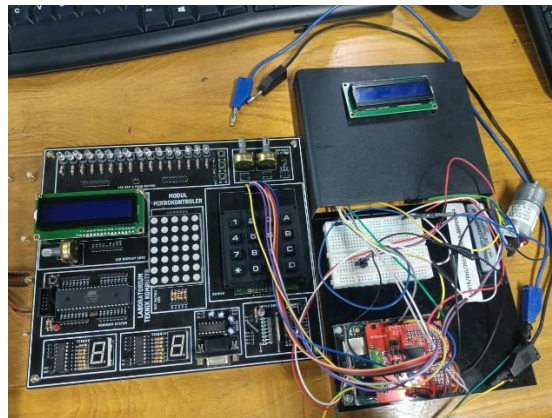
Saat sistem diaktifkan dengan memasukkan daya ke rangkaian Arduino dan komponen lainnya. Pada saat rangkaian dinyalakan, seluruh komponen akan menjalani proses inisialisasi, kemudian pengguna akan diminta memasukkan parameter K_p, K_i, K_d , dan set point melalui keypad, setelah itu Arduino akan mulai menjalankan algoritma PID untuk mengontrol kecepatan motor berdasarkan nilai-nilai tersebut.

Dalam implementasi nyata, sistem berhasil merespons input pengguna dengan baik. Nilai set point yang dimasukkan sebesar 100 RPM, akan diproses dan digunakan sebagai acuan kontrol. Arduino kemudian membaca kecepatan motor aktual melalui sinyal dari rotary encoder dan menghitung error yang kemudian menghasilkan sinyal PWM sebagai koreksi. Sinyal PWM dikirimkan ke driver motor Ardumoto, yang kemudian mengatur tegangan ke motor DC. LCD 16x2 secara *real-time* menampilkan nilai set point, parameter PID, nilai aktual RPM, sehingga pengguna dapat memantau kondisi sistem setiap saat.

Dokumentasi selama pengujian berguna agar komponen yang dirangkai dapat dikonfirmasi berjalan dengan baik. Berikut dokumentasi selama pengujian:

1. proses input PID melalui keypad.

2. Tampilan LCD yang menampilkan set point dan nilai aktual
3. Rangkaian lengkap



Gambar 4. 3 Rangkaian lengkap
(Sumber : dokumentasi pribadi)

4. Tampilan motor DC saat berjalan

Implementasi ini membuktikan bahwa sistem kendali dapat beroperasi dengan baik untuk keperluan pembelajaran. Hasil pengujian menunjukkan bahwa sistem dapat dikendalikan secara adaptif berdasarkan parameter yang diinputkan, serta memberikan pengalaman kepada pengguna dalam memahami prinsip kendali umpan balik (*feedback control*).

4.5 Karakteristik Motor

Motor yang digunakan dalam proyek ini adalah motor DC tipe JGA25-370, yaitu jenis motor DC dengan gearbox yang dirancang untuk menghasilkan torsi tinggi pada putaran rendah. Motor ini cocok untuk berbagai aplikasi sistem kendali karena memiliki karakteristik mekanik yang stabil serta mudah diintegrasikan dengan mikrokontroler dan sensor umpan balik. Beberapa spesifikasi umum dari motor JGA25-370 ditunjukkan pada Tabel berikut:

| Spesifikasi | Nilai |
|-----------------------------|---|
| Tegangan Operasional | 6V – 12V DC |
| Kecepatan Tanpa Beban (12V) | $\pm 100 - 1000$ RPM (tergantung rasio gearbox) |
| Torsi Maksimum | Hingga ± 10 kg·cm |
| Rasio Gearbox | 1:10 hingga 1:150 |
| Diameter Motor | 25 mm |
| Panjang Motor + Gearbox | ± 65 mm |
| Diameter Poros | 6 mm |

Tabel 4. 2 spesifikasi motor JGA25-370

Motor ini juga dilengkapi dengan rotary encoder yang terpasang di bagian belakang, dengan resolusi 360 pulsa per putaran. Encoder ini berfungsi sebagai sensor umpan balik untuk mengukur kecepatan rotasi aktual motor secara real-time, sehingga sangat mendukung implementasi sistem kontrol berbasis PID.

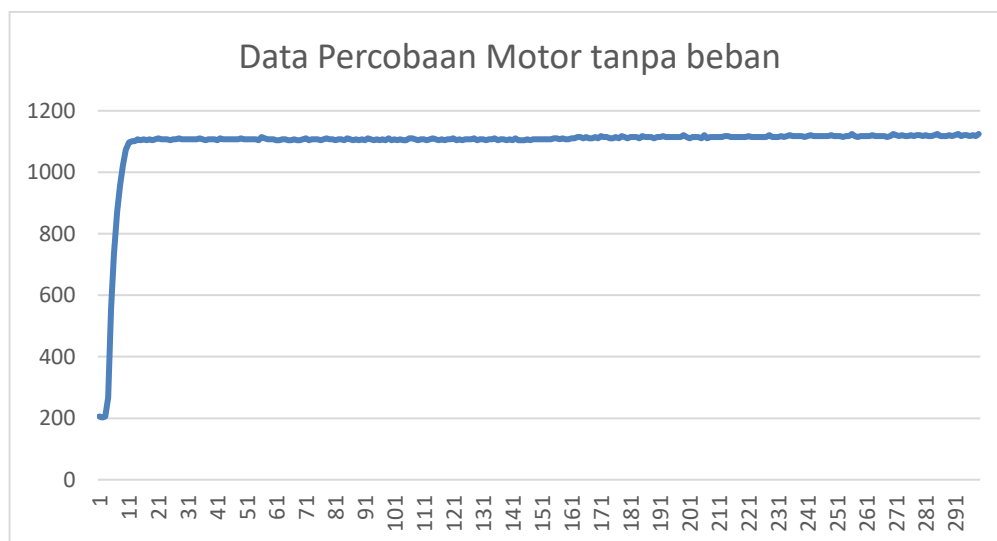
4.5.1 Uji Respons Dinamis Tanpa Beban

Sebelum implementasi sistem kendali PID, dilakukan pengujian awal untuk memperoleh karakteristik respons waktu alami motor tanpa beban. Pengujian ini bertujuan untuk mengidentifikasi tipe sistem berdasarkan tanggapan motor terhadap masukan konstan (PWM duty cycle tetap).

Metode:

- Pengambilan data dilakukan selama 75 detik dengan interval 0,25 detik, sehingga diperoleh 300 data sampel.
- Data diambil dari output kecepatan motor dalam satuan pulsa per interval waktu menggunakan encoder.
- Pengolahan dan visualisasi data dilakukan dengan perangkat lunak Microsoft Excel.

Program simulasi yang digunakan untuk menguji motor tanpa beban dapat dilihat pada Lampiran 6. Gambar dibawah ini memperlihatkan kurva tanggapan waktu motor berdasarkan hasil pengukuran:



Gambar 4. 4 Grafik tanggapan motor tanpa beban

4.5.2 Analisis Grafik dan Parameter Sistem

Dari hasil grafik diperoleh beberapa parameter penting dari sistem motor, yaitu:

- Nilai setpoint (K): 1111
- Waktu tunak (settling time, 2%): 2.5 detik
- Steady-State Error (SSE): 0.43%
- Overshoot: 1.22%
- Waktu naik (rising time, 10%–90%): 0 – 2.5 detik

Nilai-nilai ini menunjukkan bahwa motor memiliki respons yang cepat, stabil, dan minim osilasi, sesuai dengan karakteristik sistem orde satu.

4.5.3 Pemodelan Sistem Orde Satu

Respons sistem pertama kali dimodelkan sebagai sistem orde satu berdasarkan bentuk kurva eksponensial:

$$c(t) = 1 - e^{-\frac{t}{T}}$$

Diketahui:

- Nilai steady-state: K=1111
- Waktu tunak 98% (4T): 2.5 detik, T = 0.625

Maka:

$$c(t) = 1111(1 - e^{-\frac{t}{0.625}})$$

$$c(t) = 1111(1 - e^{-1.6t})$$

Atau:

$$c(t) = 1111 - 1111e^{-1.6t}$$

4.5.4 Fungsi Alih (Transfer Function)

Transformasi Laplace dari model di atas menghasilkan fungsi alih sistem motor:

$$C(s) = \frac{1111}{s+1.6}$$

Model ini menggambarkan bahwa sistem memiliki satu kutub real negatif di -1.6 , tanpa nol, sehingga menunjukkan karakteristik stabil dan respons eksponensial klasik. Fungsi alih ini akan digunakan dalam perancangan simulasi dan tuning parameter PID pada bab selanjutnya.

4.6 Pengujian Transef Function Pada Motor

Untuk mengevaluasi performa pengendalian PID terhadap model matematis motor DC JGA25-370 yang telah diperoleh sebelumnya, dilakukan serangkaian simulasi berbasis perangkat lunak MATLAB dengan berbagai kombinasi nilai konstanta K_p , K_i , dan K_d . Tujuan dari pengujian ini adalah untuk memperoleh konfigurasi parameter PID yang menghasilkan respons sistem paling optimal berdasarkan sejumlah indikator performa, yakni *settling time*, *steady-state error*, *overshoot*, dan *rise time*.

Lampiran 7 memuat program simulasi yang digunakan dalam proses pengujian untuk menentukan nilai parameter PID terbaik. Visualisasi grafik dari hasil simulasi setiap kombinasi nilai K_p , K_i , dan K_d juga disajikan secara lengkap dalam lampiran tersebut sebagai bagian dari dokumentasi pendukung analisis numerik.

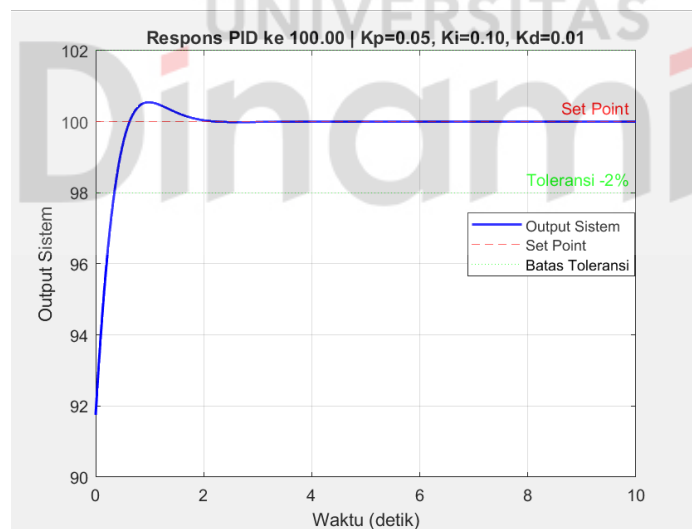
| Percobaan | K_p | K_i | K_d | Settling Time | Steady State Error | Overshoot | Rise Time |
|-----------|-------|-------|-------|----------------------|--------------------|---------------------|----------------------|
| 1 | 0.001 | 0 | 0 | Tidak dapat dihitung | 59.02% | Tidak ada overshoot | Tidak dapat dihitung |
| 2 | 0.01 | 0 | 0 | Tidak dapat dihitung | 12.59% | Tidak ada overshoot | Tidak dapat dihitung |
| 3 | 0.1 | 0 | 0 | 0.05 detik | 1.42% | Tidak ada overshoot | 0.02 detik |
| 4 | 0.05 | 0 | 0 | Tidak dapat dihitung | 2.80% | Tidak ada overshoot | 0.04 detik |
| 5 | 0.05 | 0 | 0.01 | Tidak | 2.80% | Tidak ada | 0 detik |

| | | | | dapat dihitung | | overshoot | |
|---|------|------|------|-------------------|-------|------------------------|---------|
| 6 | 0.05 | 0.01 | 0.01 | 1.46 detik | 0.37% | Tidak ada overshoot | 0 detik |
| 7 | 0.05 | 0.1 | 0.01 | 0.35 detik | 0% | 0.55% | 0 detik |

Tabel 4.3 Percobaan untuk mencari nilai PID terbaik dengan MATLAB

Berdasarkan hasil ketujuh percobaan dan analisa performa sistem secara keseluruhan, dapat disimpulkan bahwa kombinasi nilai parameter kontrol K_p , K_i , dan K_d sangat memengaruhi karakteristik respons sistem kontrol, khususnya dalam hal kecepatan, akurasi, dan kestabilan sistem.

Dari ketiga komponen tersebut, pengaruh terbesar terhadap penurunan steady-state error berasal dari penambahan komponen integral (K_i), sedangkan peningkatan kestabilan dan kecepatan respon sistem umumnya dicapai melalui kombinasi parameter proporsional dan integral yang seimbang.



Gambar 4.5 Grafik menggunakan MATLAB dengan nilai PID terbaik

Hal ini terlihat jelas pada percobaan ke-7, yang menunjukkan performa optimal dengan settling time 0.35 detik, steady-state error sebesar 0%, overshoot minimum sebesar 0.55% dan rising time 0 detik, dengan nilai $K_p = 0.05$, $K_i = 0.1$, $K_d = 0.01$. Oleh karena itu, konfigurasi parameter PID pada percobaan ini dapat dijadikan sebagai acuan awal untuk implementasi pengendali pada sistem fisik yang

sesungguhnya.

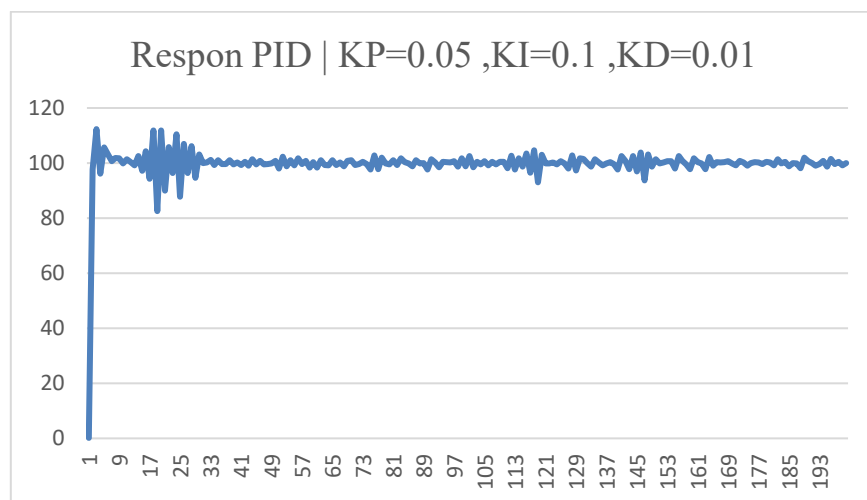
4.7 Hasil Tuning PID Terbaik untuk Implementasi pada Arduino

Setelah dilakukan proses tuning parameter PID secara simulatif menggunakan perangkat lunak MATLAB, nilai konstanta K_p , K_i , dan K_d yang dianggap paling optimal kemudian diimplementasikan secara langsung pada sistem fisik berbasis Arduino. Tujuan dari pengujian ini adalah untuk mengevaluasi apakah hasil tuning dari simulasi dapat direalisasikan dengan baik pada perangkat keras dan menghasilkan respons sistem yang stabil dan sesuai dengan ekspektasi.

Pengujian dilakukan dengan memprogram Arduino menggunakan nilai parameter PID hasil tuning terbaik yang diperoleh dari percobaan ke-7 pada simulasi MATLAB, yaitu $K_p=0,05$, $K_i=0,1$ dan $K_d=0,01$. Selama proses eksekusi, data keluaran sistem dicatat melalui fitur *serial monitor* Arduino sebanyak 200 sampel data yang mewakili respon sistem terhadap perubahan setpoint.

Data hasil *serial print* tersebut kemudian diekspor dan dimasukkan ke dalam perangkat lunak Microsoft Excel untuk dianalisis lebih lanjut. Melalui Excel, dilakukan pemrosesan dan visualisasi data dalam bentuk grafik respon sistem terhadap waktu, sehingga dapat diperoleh gambaran karakteristik dinamis dari sistem kendali PID saat dijalankan pada perangkat keras secara nyata.

Lampiran 8 memuat program simulasi yang digunakan dalam proses secara real time, berikut ialah grafik dari hasil percobaan secara real-time dengan nilai K_p , K_i dan K_d yang sama :



Gambar 4. 6 Grafik secara Real Time

| Settling Time | Steady State Error | Overshoot | Rising Time |
|---------------|--------------------|-----------|-------------|
| 1.45 | 1.1% | 12% | 0.04 |

Tabel 4. 4 Hasil secara Real Time

Setelah parameter PID hasil tuning terbaik diimplementasikan pada sistem fisik berbasis Arduino, dilakukan proses pengamatan terhadap respons sistem secara real-time guna mengevaluasi performa kendali secara kuantitatif. Pengambilan data dilakukan melalui fitur *serial monitor* Arduino dan kemudian diolah menggunakan perangkat lunak Excel untuk memperoleh gambaran karakteristik dinamis sistem secara lebih mendalam.

Berdasarkan pengukuran manual terhadap 200 sampel data yang telah direkam, diperoleh sejumlah indikator performa utama, sebagai berikut:

- Settling Time tercatat sebesar 1,45 detik, yang menunjukkan bahwa sistem memerlukan waktu tersebut untuk masuk dan bertahan dalam rentang toleransi $\pm 2\%$ terhadap nilai setpoint. Durasi ini menandakan sistem memiliki kemampuan stabilisasi yang cukup baik.
- Steady State Error terukur sebesar 1,1%, yang merefleksikan deviasi kecil antara nilai akhir output sistem dan target setpoint. Nilai ini menunjukkan bahwa sistem telah berhasil mencapai kestabilan akhir dengan akurasi yang tinggi.
- Overshoot sebesar 12% mengindikasikan bahwa respons sistem sempat melebihi nilai setpoint sebelum akhirnya stabil. Meskipun masih dalam batas yang dapat ditoleransi, besarnya overshoot ini memberikan indikasi bahwa komponen derivatif (Kd) masih dapat disempurnakan untuk meredam lonjakan awal secara lebih optimal.
- Waktu naik (rise time) sebesar 0,04 detik mencerminkan kemampuan sistem dalam memberikan respons cepat terhadap perubahan input dari 10% hingga 90% nilai akhir. Nilai ini memperlihatkan bahwa sistem memiliki kecepatan respons yang tinggi.

Secara keseluruhan, hasil implementasi kendali PID pada sistem nyata menunjukkan performa yang cukup baik, dengan keseimbangan antara kecepatan respons, akurasi, dan kestabilan. Keberhasilan sistem dalam mencapai setpoint dengan error kecil dan waktu respon yang cepat mengindikasikan bahwa parameter

PID yang digunakan telah cukup optimal untuk diimplementasikan dalam sistem fisik secara langsung

4.8 Analisa dan Pembahasan Matlab dengan Arduino

Setelah diperoleh parameter PID terbaik melalui simulasi MATLAB, langkah selanjutnya adalah mengimplementasikannya secara langsung pada sistem fisik berbasis Arduino. Untuk mengevaluasi performa sistem secara menyeluruh, dilakukan perbandingan antara hasil simulasi dan hasil real-time melalui pengolahan data di Excel. Kedua metode menggunakan parameter yang sama, yaitu

$$K_p=0.05, K_i=0.1, \text{ dan } K_d=0.01.$$

Hasil simulasi MATLAB ditunjukkan pada Gambar 4.5, yang memperlihatkan karakteristik sistem yang sangat responsif dan stabil. Berdasarkan grafik tersebut, sistem mampu mencapai setpoint sebesar 100 dengan rise time yang hampir instan (~ 0 detik), overshoot yang sangat kecil (0,55%), dan settling time hanya sekitar 0,35 detik. Selain itu, steady-state error tercatat 0%, menandakan sistem simulasi benar-benar mencapai titik target tanpa deviasi.

Sebaliknya, hasil pengujian pada perangkat keras ditampilkan pada Gambar 4.6, yang dihasilkan dari data real-time Arduino dan diolah menggunakan Excel. Meskipun menggunakan parameter yang sama, terlihat adanya perbedaan yang cukup signifikan. Respons awal sistem menunjukkan adanya overshoot yang relatif tinggi ($\sim 12\%$) serta osilasi yang berlangsung beberapa saat sebelum sistem akhirnya mendekati keadaan tunak. Settling time tercatat sekitar 1,45 detik, sedangkan steady-state error berada pada 1,1%, menunjukkan bahwa output sistem tidak sepenuhnya berhimpit dengan nilai setpoint. Selain itu, rise time terukur secara manual sebesar 0,04 detik.

Perbedaan ini menunjukkan adanya pengaruh faktor-faktor dunia nyata yang tidak termodelkan dalam simulasi MATLAB, seperti delay sensor, noise, inersia mekanik, dan ketidaksempurnaan aktuator (motor). Fluktuasi data pada grafik real-time mengindikasikan bahwa sistem fisik lebih rentan terhadap gangguan, sehingga memerlukan peredaman tambahan, baik melalui penyempurnaan tuning PID maupun filter digital pada sinyal sensor.

Meskipun demikian, pola respons sistem riil tetap mengikuti karakteristik umum dari hasil simulasi, yaitu menuju kestabilan dan mencapai nilai mendekati setpoint. Dengan demikian, parameter hasil tuning dari simulasi dapat dijadikan referensi awal yang cukup baik dalam implementasi nyata, meskipun tetap dibutuhkan penyesuaian untuk mengakomodasi ketidakidealan sistem fisik.



UNIVERSITAS
Dinamika

BAB V

Kesimpulan dan Saran

5.1 Kesimpulan

Berdasarkan hasil kerja praktik yang telah dilakukan di Laboratorium Fakultas Teknologi dan Informatika Universitas Dinamika, serta analisis yang dijelaskan pada Bab IV, diperoleh beberapa kesimpulan sebagai berikut:

1. Perancangan Modul Pembelajaran Sistem Kontrol PID
Modul pembelajaran sistem kontrol berbasis PID berhasil dirancang dan diimplementasikan menggunakan mikrokontroler Arduino UNO. Modul ini memungkinkan pengguna, khususnya mahasiswa, untuk memahami konsep dasar sistem kontrol PID secara praktis dan interaktif. Parameter K_p , K_i , dan K_d dapat diinput secara langsung melalui keypad 4x4, sementara hasil kendali ditampilkan secara real-time pada LCD 16x2 dan dapat diamati melalui grafik hasil pengolahan data.
2. Pengaruh Parameter PID terhadap kecepatan motor hasil percobaan baik secara simulasi di MATLAB maupun implementasi nyata pada perangkat keras menunjukkan bahwa kombinasi parameter PID sangat memengaruhi karakteristik sistem. Pengaturan nilai K_p , K_i , dan K_d memberikan dampak signifikan terhadap performa sistem, seperti *settling time*, *steady-state error*, *overshoot*, dan *rise time*. Konfigurasi terbaik ditemukan pada nilai $K_p = 0.05$, $K_i = 0.1$, dan $K_d = 0.01$. Pada simulasi MATLAB, kombinasi ini menghasilkan *settling time* 0,35 detik, *steady-state error* 0%, *overshoot* 0,55%, dan *rise time* mendekati 0 detik. Sementara pada implementasi nyata, diperoleh *settling time* 1,45 detik, *steady-state error* 1,1%, *overshoot* 12%, dan *rise time* sebesar 0,04 detik.
3. Visualisasi dan Pemantauan Hasil PID modul mampu menampilkan hasil perhitungan PID dan nilai RPM motor secara real-time melalui LCD 16x2. Selain itu, pengambilan data melalui serial monitor Arduino sebanyak 200 sampel telah diolah dalam Excel untuk menghasilkan grafik respons sistem yang membantu pengguna dalam menganalisis kestabilan dan kecepatan

sistem secara kuantitatif. Hal ini mendukung proses pembelajaran dan eksperimentasi yang lebih mendalam terhadap sistem kendali otomatis.

Secara keseluruhan, modul ini memberikan pengalaman belajar yang komprehensif tentang tuning PID dan pengaruhnya terhadap sistem dinamis. Pengujian menunjukkan bahwa hasil tuning melalui simulasi dapat dijadikan referensi awal yang baik untuk implementasi sistem nyata, dengan tetap memperhatikan faktor ketidaksempurnaan pada perangkat keras.

5.2 Saran

Untuk pengembangan lebih lanjut, disarankan agar sistem dilengkapi dengan kemampuan auto-tuning parameter PID secara otomatis, seperti dengan metode fuzzy logic atau algoritma adaptif, agar modul dapat menyesuaikan diri terhadap kondisi sistem secara dinamis tanpa mengandalkan input manual. Selain itu, integrasi fitur pencatatan data seperti penyimpanan ke SD card atau koneksi serial ke komputer akan sangat bermanfaat untuk dokumentasi dan analisis data eksperimen oleh mahasiswa. Pengembangan antarmuka berbasis grafis atau platform web menggunakan modul komunikasi seperti ESP8266 juga dapat meningkatkan fleksibilitas penggunaan, terutama untuk pengendalian jarak jauh. Dalam aspek keamanan, penting untuk menambahkan sistem proteksi terhadap arus berlebih atau gangguan lainnya guna melindungi perangkat dari kerusakan saat digunakan dalam waktu lama. Terakhir, penyelenggaraan pelatihan penggunaan modul bagi dosen dan mahasiswa sangat disarankan agar perangkat ini dapat dimanfaatkan secara maksimal, baik dalam kegiatan praktikum terstruktur maupun dalam proyek penelitian dan tugas akhir secara mandiri.

Daftar Pustaka

- Di, B. (2023, August 10). Kontroller PID: Prinsip Kerja, Aplikasinya dalam Pengendalian Proses, Keuntungan Kontroller PID. KMTek. <https://www.kmtech.id/post/kontroller-pid-prinsip-kerja-aplikasinya-dalam-pengendalian-proses-keuntungan-kontroller-pid>
- Bolton, W. (2015). *Mechatronics: Electronic control systems in mechanical and electrical engineering* (6th ed.). Pearson Education.
- Dorf, R. C., & Bishop, R. H. (2017). *Modern control systems* (13th ed.). Pearson Education.
- Ogata, K. (2010). *Modern control engineering* (5th ed.). Prentice Hall.
- Rajamani, R. (2012). *Vehicle dynamics and control* (2nd ed.). Springer.
- Banzy, M., & Shiloh, M. (2014). *Getting started with Arduino* (3rd ed.). Maker Media.
- Geddes, R. (2011). *Arduino project handbook*. Elector International Media.
- Geddes, R. (2011). *Arduino project handbook: 25 practical projects to get you started*. No Starch Press.
- Kumar, S., & Singh, S. P. (2019). A review on Arduino-based applications for the development of embedded systems. *International Journal of Engineering and Advanced Technology (IJEAT)*, 8(6), 1564–1571.
- Margolis, M. (2011). *Arduino cookbook* (2nd ed.). O'Reilly Media.
- McRoberts, M. (2013). *Beginning Arduino* (2nd ed.). Apress.
- Monk, S. (2016). *Programming Arduino: Getting started with sketches* (2nd ed.). McGraw-Hill Education.
- Kumar, P. (2014). *Beginning Arduino Programming*. Apress.
- Mazidi, M. A., Naimi, S., & Naimi, S. (2008). *AVR microcontroller and embedded systems: Using assembly and C*. Pearson Education.
- Norris, T. (2011). Arduino-controlled keypad security system. In *Arduino Projects for Dummies* (pp. 151–170). Wiley.
- Pillai, S. (2011). *Programming and Customizing the AVR Microcontroller*. McGraw-Hill Education.
- Purdum, J. (2012). *Beginning C for Arduino: Learn C Programming for the Arduino*. Apress.

- Sloan, T. (2010). *Getting Started with Arduino and LCDs*. Maker Media, Inc.
- Astrom, K. J., & Murray, R. M. (2010). *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press.
- Jones, B. (2015). *Practical Robotics in C++: Build Robotic Systems Using ROS and Modern C++*. Apress.
- Jones, M. (2015). *Practical Arduino engineering* (2nd ed.). Apress.
- Mitchell, S. (2012). *Arduino: A Quick-Start Guide* (2nd ed.). Pragmatic Bookshelf.
- Mitchell, M. (2012). *Arduino: A quick-start guide* (2nd ed.). Pragmatic Bookshelf.
- Cook, D. (2010). *Intermediate Robot Building* (2nd ed.). Apress.
- Electronics-Salon. (2020). 25mm DC Geared Motor with Encoder – Product Datasheet. Retrieved from <https://www.electronics-salon.com>
- Singh, R., & Sharma, V. (2019). *Embedded systems design using microcontrollers*. Oxford University Press.
- Kumar, R., & Singh, A. (2019). *Microcontroller and Embedded Systems*. Oxford University Press.
- Anggraini, R. R., Suryani, I., & Budiharto, W. (2020). Optimization of PID controller using differential evolution for DC motor speed control. *TELKA: Jurnal Teknik Elektro*, 6(2), 147–159. <https://telka.ee.uinsgd.ac.id/index.php/TELKA/article/view/telka.v6n2.147-159>
- Fajuke, R. A., & Raji, W. O. (2023). Equilibrium optimizer-based PID tuning technique for performance improvement of DC motor speed control system. *International Journal of Electrical and Computer Engineering (IJECE)*, 13(2), 1760–1768. <https://ijeecs.iaescore.com/index.php/IJECS/article/view/30462>
- Yuliawan, A., Nugraha, A. S., & Saepuloh, D. (2021). Improving PID performance on DC motor control system using Kalman filter. *International Journal of Informatics and Information Systems*, 4(1), 33–42. <https://jurnal.ugm.ac.id/ijitee/article/view/6451>