

**PROYEK AKHIR - VI 231733**

**PENGEMBANGAN *FINAL CONTROL ELEMENT* UNTUK  
*CONTROL VALVE* BERBASIS *STEPPER MOTOR* DAN  
*GLOBE VALVE***

**RIZAL FIQRI MUTTAQIN**  
NRP. 2042211120

Dosen Pembimbing 1  
**Ir. Dwi Nur Fitriyanah, S.ST., M.T.**  
NPP. 1994202012014

Dosen Pembimbing 2  
**Muhammad Roy Ashiddiqi, S.T., M.T.**  
NIP. 199301312024061001

Program Studi Sarjana Terapan Teknologi Rekayasa Instrumentasi  
Departemen Teknik Instrumentasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2025

*Halaman ini sengaja dikosongkan*



**PROYEK AKHIR - VI 231733**

**PENGEMBANGAN *FINAL CONTROL ELEMENT* UNTUK  
*CONTROL VALVE* BERBASIS *STEPPER MOTOR* DAN  
*GLOBE VALVE***

**RIZAL FIQRI MUTTAQIN**  
NRP. 2042211120

Dosen Pembimbing 1  
**Ir. Dwi Nur Fitriyanah, S.ST., M.T.**  
NPP. 1994202012014

Dosen Pembimbing 2  
**Muhammad Roy Ashiddiqi, S.T., M.T.**  
NIP. 199301312024061001

**Program Studi Sarjana Terapan Teknologi Rekayasa Instrumentasi**  
Departemen Teknik Instrumentasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2025

*Halaman ini sengaja dikosongkan*



**FINAL PROJECT - VI 231733**

**DEVELOPMENT OF FINAL CONTROL ELEMENT FOR  
CONTROL VALVE BASED ON STEPPER MOTOR AND  
GLOBE VALVE**

**RIZAL FIQRI MUTTAQIN**  
NRP. 2042211120

*Advisor 1*

**Ir. Dwi Nur Fitriyanah, S.ST., M.T.**  
NPP. 1994202012014

*Advisor 2*

**Muhammad Roy Ashiddiqi, S.T., M.T.**  
NIP. 199301312024061001

*Applied Bachelor Program of Instrumentation Engineering Technology*  
*Department of Instrumentation Engineering*  
*Faculty of Vocation*  
*Institut Teknologi Sepuluh Nopember*  
*Surabaya*  
*2025*

*Halaman ini sengaja dikosongkan*

**LEMBAR PENGESAHAN  
PROYEK AKHIR**

**PENGEMBANGAN *FINAL CONTROL ELEMENT* UNTUK *CONTROL VALVE*  
BERBASIS *STEPPER MOTOR* DAN *GLOBE VALVE***

Oleh:



**Rizal Fiqri Muttaqin**  
NRP. 2042211120

Surabaya, 17 Juli 2025

**Menyetujui,  
Pembimbing 1**



**Ir. Dwi Nur Fitriyanah, S.ST., M.T.**  
NPP. 1994202012014

**Menyetujui,  
Pembimbing 2**



**Muhammad Roy Ashiddiqi, S.T., M.T.**  
NIP. 199301312024061001

**Mengetahui,  
Kepala Departemen  
Teknik Instrumentasi FV-ITS**



**Ir. Herry Sufyan Hadi, S.T., M.T. Ph.D.**  
NPP. 1988201711056

*Halaman ini sengaja dikosongkan*

**APPROVAL SHEET  
FINAL PROJECT**

**DEVELOPMENT OF FINAL CONTROL ELEMENT FOR CONTROL VALVE BASED  
ON STEPPER MOTOR AND GLOBE VALVE**

By:



**Rizal Figri Muttaqin**  
NRP. 2042211120

Surabaya, 17 July 2025

*Approved,*

*Advisor 1*



**Ir. Dwi Nur Fitriyanah, S.ST., M.T.**

NPP. 1994202012014

*Approved,*

*Advisor 2*



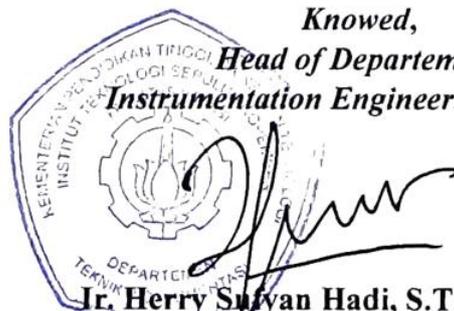
**Muhammad Roy Ashiddiqi, S.T., M.T.**

NIP. 199301312024061001

*Knowned,*

*Head of Departement*

*Instrumentation Engineering FV-ITS*



**Ir. Herry Sulwan Hadi, S.T., M.T. Ph.D.**

NPP. 1988201711056

*Halaman ini sengaja dikosongkan*

## LEMBAR PENGESAHAN

### PENGEMBANGAN *FINAL CONTROL ELEMENT* UNTUK *CONTROL VALVE* BERBASIS *STEPPER MOTOR* DAN *GLOBE VALVE*

#### TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat  
Memperoleh gelar Sarjana Terapan pada  
Program Studi D-4 Teknologi Rekayasa Instrumentasi  
Departemen Teknik Instrumentasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Oleh : **RIZAL FIQRI MUTTAQIN**

NRP. 2042211120

Disetujui oleh Tim Penguji Proposal Tugas Akhir :

1. Ir. Dwi Nur Fitriyanah, S.ST., M.T.



Pembimbing

2. Muhammad Roy Ashiddiqi, S.T., M.T.



Ko-pembimbing

3. Ir. Dwi Oktavianto Wahyu Nugroho, S.T., M.T.

Penguji

4. Ir. Herry Sufyan Hadi, S.T., M.T. Ph.D.

Penguji

**SURABAYA**  
**17 Juli, 2025**

*Halaman ini sengaja dikosongkan*

## **APPROVAL SHEET**

### **DEVELOPMENT OF FINAL CONTROL ELEMENT FOR CONTROL VALVE BASED ON STEPPER MOTOR AND GLOBE VALVE**

#### **FINAL PROJECT**

*Submitted to fulfill one of the requirements  
for obtaining a degree applied bachelor at*

*Undergraduate Study Program of D-4 Instrumentation Engineering Technology*

*Department Instrumentation Engineering*

*Faculty of Vocation*

*Institut Teknologi Sepuluh Nopember*

**By: RIZAL FIQRI MUTTAQIN**  
NRP. 2042211120

*Approved by Final Project Examiner Team:*

1. Ir. Dwi Nur Fitriyanah, S.ST., M.T.



Advisor

2. Muhammad Roy Ashiddiqi, S.T., M.T.



Co-Advisor

3. Ir. Dwi Oktavianto Wahyu Nugroho, S.T., M.T.



Examiner

4. Ir. Herry Sufyan Hadi, S.T., M.T. Ph.D.

Examiner

**SURABAYA**  
**17 july, 2025**

*Halaman ini sengaja dikosongkan*

## PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama Mahasiswa / NRP : Rizal Fiqri Muttaqin / 2042211120

Departemen : Teknik Instrumentasi

Dosen Pembimbing 1 / NPP : Ir. Dwi Nur Fitriyanah, S.ST., M.T. / 1994202012014

Dosen Pembimbing 2 / NIP : Muhammad Roy Ashiddiqi, S.T., M.T. / 199301312024061001

dengan ini menyatakan bahwa Tugas Akhir dengan judul **“PENGEMBANGAN *FINAL CONTROL ELEMENT* UNTUK *CONTROL VALVE* BERBASIS *STEPPER MOTOR* DAN *GLOBE VALVE*”** adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 17 Juli 2025

Mahasiswa,



Rizal Fiqri Muttaqin

NRP. 2042211120

Mengetahui,

Dosen Pembimbing 1,



Ir. Dwi Nur Fitriyanah, S.ST., M.T.  
NPP. 1994202012014

Dosen Pembimbing 2,



Muhammad Roy Ashiddiqi, S.T., M.T.  
NIP. 199301312024061001

*Halaman ini sengaja dikosongkan*

## **STATEMENT OF ORIGINALITY**

*The undersigned below:*

*Name of student / NRP* : Rizal Fiqri Muttaqin / 2042211120

*Department* : *Instrumentation Engineering*

*Advisor 1 / NPP* : Ir. Dwi Nur Fitriyanah, S.ST., M.T. / 1994202012014

*Advisor 2 / NIP* : Muhammad Roy Ashiddiqi, S.T., M.T. / 199301312024061001

*hereby declare that the Final Project with the title of “**DEVELOPMENT OF FINAL CONTROL ELEMENT FOR CONTROL VALVE BASED ON STEPPER MOTOR AND GLOBE VALVE**” is the result of my own work, is original, and is written by following the rules of scientific writing.*

*If in the future there is a discrepancy with this statement, then i am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.*

Surabaya, 17 July 2025

*Student,*

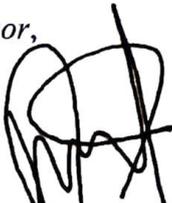


Rizal Fiqri Muttaqin

NRP. 2042211120

*Acknowledged,*

*Advisor,*



Ir. Dwi Nur Fitriyanah, S.ST., M.T.  
NPP. 1994202012014

*Co-advisor,*



Muhammad Roy Ashiddiqi, S.T., M.T.  
NIP. 199301312024061001

*Halaman ini sengaja dikosongkan*

## **PENGEMBANGAN *FINAL CONTROL ELEMENT* UNTUK *CONTROL VALVE* BERBASIS *STEPPER MOTOR* DAN *GLOBE VALVE***

**Nama Mahasiswa / NRP : Rizal Fiqri Muttaqin / 2042211120**

**Departemen : Teknik Instrumentasi**

**Dosen Pembimbing 1 : Ir. Dwi Nur Fitriyanah, S.ST., M.T.**

**Dosen Pembimbing 2 : Muhammad Roy Ashiddiqi, S.T., M.T.**

### **ABSTRAK**

Dalam sistem kontrol otomatis, *final control element* (FCE) berperan penting dalam mengatur aliran fluida secara presisi. Salah satu bentuk pengembangan FCE dilakukan melalui integrasi *globe valve* dengan *stepper motor* sebagai aktuator utama. Penelitian ini merancang dan mengimplementasikan sistem *control valve* berbasis *globe valve* yang digerakkan oleh *stepper motor* NEMA17, serta membandingkan dua *driver motor* DRV8825 dan TMC2209 untuk mengevaluasi performanya. Sistem dilengkapi dengan *rotary encoder* KY-040 untuk *input manual*, *optical rotary encoder* untuk validasi posisi, serta sensor arus INA219 dan sensor temperatur DS18B20 untuk *monitoring* kondisi *driver* secara *real-time*. Pengujian dilakukan dalam beberapa mode penggerakan (*full-step*, *half-step*, dan *micro-step*) dengan pengambilan data arus dan temperatur per detik. Analisis *Fast Fourier Transform* (FFT) dilakukan pada sinyal arus DRV8825 untuk mengevaluasi kestabilan listriknya. Selain itu, performa *control valve* dievaluasi dari akurasi posisi, responsivitas aktuator, serta karakteristik aliran fluida dan *pressure drop* berdasarkan pengujian menggunakan *valve tester*. Hasil menunjukkan bahwa *driver* TMC2209 memberikan performa yang lebih stabil dan efisien dibandingkan DRV8825, terutama dalam hal kestabilan arus dan temperatur kerja. Rata-rata *error* pada aktuasi penuh *valve* sebesar 1,4% (buka) dan 2% (tutup), dengan *deadband* sekitar 1%. Karakteristik aliran menunjukkan profil *quick-opening* sesuai tipe *disc globe valve* yang digunakan. Dengan hasil ini, sistem *control valve* yang dirancang terbukti layak untuk aplikasi yang membutuhkan kendali presisi menengah serta dapat dijadikan dasar pengembangan lebih lanjut.

**Kata Kunci:** *Control Valve, Globe Valve, Stepper Motor, Driver Motor, Final Control Element*

*Halaman ini sengaja dikosongkan*

## ***DEVELOPMENT OF FINAL CONTROL ELEMENT FOR CONTROL VALVE BASED ON STEPPER MOTOR AND GLOBE VALVE***

***Student Name / NRP*** : Rizal Fiqri Muttaqin / 2042211120  
***Department*** : *Instrumentation Engineering*  
***Advisor 1*** : Ir. Dwi Nur Fitriyanah, S.ST., M.T.  
***Advisor 2*** : Muhammad Roy Ashiddiqi, S.T., M.T.

### ***ABSTRACT***

*In an automatic control system, the final control element (FCE) plays an important role in precisely regulating fluid flow. Integration of a globe valve with a stepper motor as the main actuator is one form of FCE development. In this study, a globe valve-based kontrol system driven by a NEMA17 stepper motor is compared with two motor drivers, DRV8825 and TMC2209, to evaluate their performance. The system is equipped with a KY-040 rotary encoder for manual input, an optical rotary encoder for position validation, and an INA219 current and DS18B20 temperature sensors for real-time driver condition monitoring. Tests were conducted in several driving modes (full-step, half-step, and micro-step) with current and temperature data collected per second. Fast Fourier Transform (FFT) analysis was performed on the DRV8825 current signal to evaluate its electrical stability. In addition, the performance of the control valve is evaluated based on the positioning accuracy, actuator responsiveness, and fluid flow and pressure drop characteristics based on testing using a valve tester. The results show that the TMC2209 driver provides a more stable and efficient performance than the DRV8825, especially in terms of current stability and working temperature. The average error in full actuation of the valve is 1.4% (open) and 2% (close), with a deadband of approximately 1%. The flow characteristics show a quick-opening profile according to the type of disc globe valve used.*

***Keywords: Control Valve, Globe Valve, Stepper Motor, Driver Motor, Final Control Element***

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Segala puji dan syukur bagi kehadiran Allah SWT, atas segala berkat, rahmat, dan karunia-Nya. Sholawat serta salam kepada Nabi Muhammad SAW, beserta keluarga, sahabat, dan pengikutnya hingga akhir zaman. Penulis dapat menyelesaikan Tugas Akhir yang berjudul “PENGEMBANGAN *FINAL CONTROL ELEMENT* UNTUK *CONTROL VALVE* BERBASIS *STEPPER MOTOR* DAN *GLOBE VALVE*” dengan lancar. Dalam penyelesaian tugas akhir ini, penulis mengucapkan terima kasih banyak kepada semua orang yang telah membantu dan mendukung dalam menyelesaikan Laporan Tugas Akhir ini. Pada kesempatan ini terima kasih saya ucapkan kepada:

1. Allah SWT. Segala puji hanya bagi Allah, Pencipta alam semesta, yang telah memberikan ilham, kekuatan, dan petunjuk dalam perjalanan penelitian ini. Tanpa-Nya, tugas akhir ini tidak akan terwujud.
2. Ibu Neneng Nurhasanah dan Bapak Ayat Firmansyah selaku kedua orangtua, serta Fithriyah Rahmah Awalayah selaku kakak atas dukungan, doa, dan semangat. Kalian adalah tiang penopang dan inspirasi bagi perjalanan akademik saya.
3. Bapak Ir. Herry Sufyan Hadi, S.T., M.T., Ph.D., selaku Kepala Departemen Teknik Instrumentasi, Institut Teknologi Sepuluh Nopember.
4. Ibu Ir. Dwi Nur Fitriyanah, S.ST., M.T. dan Bapak Muhammad Roy Ashiddiqi, S.T., M.T. Sebagai dosen pembimbing 1 dan 2. Bimbingan dan masukan kalian telah membantu saya mengembangkan penelitian ini.
5. Bapak Ir. Dwi Oktavianto Wahyu Nugroho, S.T., M.T. dan Bapak Ir. Herry Sufyan Hadi, S.T., M.T. Ph.D. Sebagai penguji 1 dan 2. Kritik konstruktif dari kalian sangat berarti bagi perbaikan tugas akhir ini.
6. Semua Pihak yang Turut Membantu termasuk teman-teman, dan rekan-rekan yang telah memberikan dukungan moral dan teknis.

Penulis sadar bahwa masih terdapat kekurangan dalam penyusunan tugas akhir ini. Oleh karena itu, kritik dan saran dari pembaca sangat diharapkan untuk terus memperbaiki dan mengembangkan kualitas tugas akhir ini. Semoga tugas akhir ini bermanfaat dan menjadi bagian dari sumbangsih ilmiah. Semoga Allah senantiasa memberkahi kita semua.

Surabaya, 17 Juli 2025

Rizal Fiqri Muttaqin

NRP 204221110

*Halaman ini sengaja dikosongkan*

## DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	v
<i>APPROVAL SHEET</i>	vii
LEMBAR PENGESAHAN	ix
<i>APPROVAL SHEET</i>	xi
PERNYATAAN ORISINALITAS	xiii
<i>STATEMENT OF ORIGINALITY</i>	xv
ABSTRAK	xvii
<i>ABSTRACT</i>	xix
KATA PENGANTAR	xxi
DAFTAR ISI	xxiii
DAFTAR GAMBAR	xxv
DAFTAR TABEL	xxvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
BAB 2 TINJAUAN PUSTAKA	5
2.1 Hasil Penelitian Terdahulu	5
2.2 Dasar Teori	7
BAB 3 METODOLOGI	17
3.1 Metode yang Digunakan	17
3.2 Bahan dan Peralatan yang Digunakan	17
3.3 Urutan Pelaksanaan Penelitian	19
BAB 4 HASIL DAN PEMBAHASAN	33
4.1 Hasil Perancangan <i>Control Valve</i>	33
4.2 Hasil Validasi Sensor	34
4.3 Hasil Pengujian <i>Globe Valve</i> (Manual)	38
4.4 Hasil Pengujian <i>Stepper Motor</i> dan <i>Driver DRV8255</i>	39
4.5 Hasil Pengujian <i>Stepper Motor</i> dan <i>Driver TMC2209</i>	42
4.6 Evaluasi <i>Driver</i> dengan Analisis FFT	45
4.7 Hasil Pengujian <i>Optical Rotary Encoder</i>	47
4.8 Hasil Pengujian <i>Rotary Encoder</i>	48
4.9 Hasil Pengujian <i>Control Valve</i>	49
4.10 Hasil Pengambilan Data dengan <i>Control Valve Tester</i>	50
4.11 Hasil Analisa Performa <i>Control Valve</i>	52
BAB 5 KESIMPULAN DAN SARAN	53
5.1 Kesimpulan	53
5.2 Saran	53
DAFTAR PUSTAKA	55
LAMPIRAN	57
BIODATA PENULIS	75

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 2.1	Karakteristik Aliran untuk <i>Control Valve</i> (Baumann, 2023).	8
Gambar 2.2	<i>Globe Valve</i>	10
Gambar 2.3	<i>Stepper Motor</i>	10
Gambar 2.4	<i>Driver Stepper Motor</i>	11
Gambar 2.5	<i>Toggle Switch</i>	13
Gambar 2.6	<i>Rotary Encoder KY-040</i>	13
Gambar 2.7	<i>Optical Rotary Encoder</i>	13
Gambar 2.8	Sistem Kontrol <i>Open-loop</i>	14
Gambar 2.9	Sensor Arus dan Tegangan (INA219)	14
Gambar 2.10	Sensor Temperatur (DS18B20)	15
Gambar 2.11	<i>Pressure Sensor</i>	15
Gambar 2.12	<i>Flow Sensor</i>	16
Gambar 3.1	Urutan Pelaksanaan Penelitian	19
Gambar 3.2	Desain <i>Coupling Stepper Motor Side</i>	20
Gambar 3.3	Desain <i>Coupling Globe Valve Side</i>	21
Gambar 3.4	Mekanisme Pergerakan <i>Globe Valve</i>	22
Gambar 3.5	Desain <i>Control Valve</i>	23
Gambar 3.6	<i>Wiring Diagram PCB DRV8255</i>	27
Gambar 3.7	<i>Wiring Diagram PCB TMC2209</i>	27
Gambar 3.8	Diagram Skematik Kontrol	28
Gambar 3.9	Rancangan GUI	29
Gambar 3.10	<i>Flowchart Operasi Control Valve</i>	30
Gambar 4.1	<i>Motorized Control Valve</i>	33
Gambar 4.2	Panel Box Kontrol	34
Gambar 4.3	Grafik Validasi Sensor Arus INA219	36
Gambar 4.4	Grafik Validasi Sensor Temperatur DS18B20	38
Gambar 4.5	Grafik <i>Stepping Mode (Full-Step dan Half-Step)</i> terhadap Arus DRV8255	39
Gambar 4.6	Grafik <i>Stepping Mode (1/4, 1/8, 1/16, dan 1/32)</i> terhadap Arus DRV8255	40
Gambar 4.7	Grafik <i>Stepping Mode</i> terhadap Temperatur DRV8255	41
Gambar 4.8	Grafik <i>Stepping Mode (Full-Step dan Half-Step)</i> terhadap Arus TMC2209	42
Gambar 4.9	Grafik <i>Stepping Mode (1/4, 1/8, 1/16, dan 1/32)</i> terhadap Arus TMC2209	43
Gambar 4.10	Grafik <i>Stepping Mode</i> terhadap Temperatur TMC2209	44
Gambar 4.11	Grafik Arus <i>Fullstep Driver</i> DRV8255 dengan FFT	45
Gambar 4.12	Perbandingan Arus (a) DRV8255 dan (b) TMC2209	46
Gambar 4.13	Perbandingan Temperatur (a) DRV8255 dan (b) TMC2209	46
Gambar 4.14	Grafik Perubahan <i>Flowrate</i> Setiap 1% Bukaan <i>Valve</i>	50
Gambar 4.15	Grafik Perubahan <i>Pressure Drop</i> Setiap 1% Bukaan <i>Valve</i>	51
Gambar 4.16	Grafik Karakteristik Aliran dari Bukaan <i>Valve</i>	51
Gambar 4.17	Grafik <i>Pressure Drop</i> dari Bukaan <i>Valve</i>	52

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

<b>Tabel 2.1</b>	<i>State of The Art</i>	5
<b>Tabel 3.1</b>	Bahan yang Digunakan	17
<b>Tabel 3.2</b>	Alat yang Digunakan	18
<b>Tabel 3.3</b>	<i>Bill of Material</i> (BOM)	24
<b>Tabel 3.4</b>	<i>Bill of Quantity</i> (BOQ)	26
<b>Tabel 4.1</b>	Data Validasi Sensor Arus INA219	35
<b>Tabel 4.2</b>	Hasil Validasi Sensor Temperatur DS18B20	37
<b>Tabel 4.3</b>	Hasil Pengujian <i>Globe Valve</i>	38
<b>Tabel 4.4</b>	Hasil Pengujian <i>Optical Rotary Encoder</i>	47
<b>Tabel 4.5</b>	Hasil Pengujian <i>Rotary Encoder</i> (Sebelum Kalibrasi)	48
<b>Tabel 4.6</b>	Hasil Pengujian <i>Rotary Encoder</i> (Sesudah Kalibrasi)	48
<b>Tabel 4.7</b>	Hasil Pengujian untuk Membuka <i>Control Valve</i>	49
<b>Tabel 4.8</b>	Hasil Pengujian untuk Menutup <i>Control Valve</i>	49

*Halaman ini sengaja dikosongkan*

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Dalam sistem kontrol industri, *control valve* berperan sebagai *final control element* (FCE) yang sangat penting, yaitu komponen terakhir dalam suatu sistem kontrol otomatis yang menerima sinyal dari kontroler dan menyesuaikan aliran fluida di dalam pipa. FCE diibaratkan sebagai “tangan” dari sistem kontrol, di mana sinyal korektif dari kontroler yang merupakan otaknya dieksekusi menjadi aksi fisik melalui *valve* untuk menjaga variabel proses mendekati nilai *setpoint* (Baumann, 2023). *Control valve* banyak digunakan dalam berbagai industri seperti pembangkit listrik, *oil and gas*, dan lain sebagainya, komponen ini memiliki struktur utama yang terdiri dari *body*, *bonnet*, *stem*, dan juga aktuator. Fungsi dari *trim*, termasuk *plug* dan *seat*, juga menjadi komponen penting karena bertugas untuk mengendalikan aliran sekaligus menghindari efek yang tidak diinginkan seperti kavitasi, getaran, dan kebisingan pada *valve* itu sendiri (Asim et al., 2019).

*Control valve* memiliki berbagai jenis dan ukuran, salah satunya yaitu *globe valve* yang secara luas digunakan dalam sistem perpipaan untuk pengendalian aliran fluida. Menurut Li et al (2020), *globe valve* merupakan salah satu jenis *valve* yang umum digunakan dalam pengendalian tekanan, laju aliran, dan juga temperatur. Fokus penelitian terhadap *globe valve* umumnya diarahkan pada karakteristik alirannya serta optimalisasi struktur internal untuk mengatasi masalah seperti kavitasi.

Pengoperasian *control valve* itu sendiri memerlukan sistem aktuator yang mampu menggerakkan komponen internalnya, terutama untuk batang *valve*. Menurut Sotoodeh (2019), aktuator pada dasarnya merupakan mesin atau bisa juga disebut komponen yang dipasang di bagian atas *valve* untuk secara otomatis menggerakkan atau mengendalikan posisi dari batang *valve*. Kinerja *valve*, sebagaimana dijelaskan Sotoodeh (2019), tidak hanya tergantung pada struktur fisik *valve* itu sendiri, tetapi juga pada performa aktuator yang digunakan.

Aktuator dalam sistem industri secara umum dibagi menjadi tiga kategori utama, yakni pneumatik, hidrolik, dan elektrik (Sotoodeh, 2019). Masing-masing jenis tersebut memiliki kelebihan dan kekurangannya. Aktuator hidrolik, misalnya, cocok untuk sistem bertekanan tinggi dan ukuran besar karena mampu memberikan torsi besar dalam waktu singkat. Namun, sistem ini cenderung mahal dan kompleks, serta membutuhkan penanganan fluida bertekanan yang ketat (Ren et al., 2024; Sotoodeh, 2019). Pneumatik di satu sisi banyak digunakan pada lingkungan yang berbahaya karena aman dan cepat dalam operasional, tetapi kelemahannya adalah keterbatasan presisi akibat sifat kompresibel udara (Sotoodeh, 2019). Di sisi lain, aktuator elektrik menawarkan biaya operasional yang rendah, bentuk yang ringkas, serta kemudahan integrasi dengan sistem digital. Namun, aktuator elektrik ini tidak memiliki mekanisme *fail-safe* secara alami dan kecepatannya lebih lambat dibanding dua jenis lainnya (Sotoodeh, 2019).

Aktuator elektrik sendiri digerakkan oleh motor, salah satu jenis motor yang sangat relevan dalam sistem kontrol presisi adalah *stepper motor*. Hughes & Drury (2019) menekankan bahwa motor ini memiliki karakteristik yang unik dimana poros keluarannya berputar dalam langkah-langkah sudut diskrit berdasarkan jumlah pulsa masukan, menjadikannya sangat cocok untuk sistem kontrol *open-loop position*. Hal ini memungkinkan kontrol posisi tanpa umpan balik, yang membuatnya banyak digunakan dalam otomasi industri.

Meskipun demikian, sistem berbasis *stepper motor* tidak lepas dari tantangan dan masalah teknis. Salah satu kelemahan utama sistem kontrol *open-loop* adalah efisiensi motor yang rendah pada beban parsial serta risiko terjadinya *stalling* ketika torsi beban terlalu besar (Bernardi et al., 2022). Selain itu masalah seperti getaran dan juga kehilangan langkah pada *stepper motor* juga menjadi masalah yang sering muncul, terutama pada kecepatan tertentu. Masalah getaran ini, seperti dijelaskan oleh Kim & Ahn (2021), dapat menyebabkan kehilangan sinkronisasi dan menurunkan keakuratan posisi.

Untuk mengatasi permasalahan tersebut, salah satu pendekatan yang bisa digunakan adalah penerapan teknik *micro-stepping*. Abuowda et al (2019) melalui investigasinya menunjukkan bahwa teknik ini mampu mengurangi *friction ripples* yang muncul pada mode *full step*, sekaligus memberikan performa lebih halus dan presisi lebih tinggi. Teknik ini mengatur arus ke masing-masing fase motor secara bertahap, membagi satu langkah besar menjadi sejumlah langkah-langkah kecil. Dengan demikian, osilasi mekanik dan getaran bisa ditekan. Hal ini diperkuat oleh Kim & Ahn (2021) yang menyatakan bahwa implementasi teknik *micro-stepping* bisa digunakan untuk mengatasi permasalahan tersebut dengan teknik lain seperti pengendali arus PID dan *feed-forward compensator* juga dapat dikombinasikan untuk meningkatkan performa sistem terhadap gangguan seperti *back electromotive force* (EMF). Namun, dalam penelitian ini, teknik tambahan tersebut tidak diterapkan, dan fokus utama tetap pada pemanfaatan teknik *micro-stepping* sebagai metode utama pengurangan osilasi dan peningkatan presisi.

Selain faktor penggerak, aspek mekanis dari *control valve* juga dapat mempengaruhi kinerja sistem kontrol. Beberapa fenomena mekanis seperti *stiction* (gesekan awal sebelum gerak), *hysteresis*, *backlash*, *dead-band*, dan *dead-zone* merupakan penyebab umum ketidakakuratan dari sistem kontrol. Masalah-masalah ini membuat pengendalian dari posisi *valve* menjadi sulit, terutama jika tidak ditangani dengan baik melalui perancangan mekanik dan pemilihan aktuator yang tepat (Vázquez et al., 2023).

Hal senada juga dikemukakan oleh Yue et al (2023) bahwa friksi dan *non-linearity* yang terjadi di dalam *electric control valve*, seperti akibat dari packing *valve* yang menimbulkan hambatan dinamis terhadap gerak dari batang *valve*, menjadi penyebab dominan terhadap penurunan kinerja *Control valve* itu sendiri. Menurut statistik yang mereka sajikan, sekitar 20 - 30% osilasi dalam kontrol *loop* industri disebabkan oleh friksi dan histeresis. Oleh karena itu, upaya kompensasi friksi menjadi krusial dalam pengembangan sistem kontrol berbasis *valve* elektrik.

Selain faktor internal *valve* dan penggerak motor, aspek kontrol sistem juga tidak boleh diabaikan. Osilasi yang terjadi dalam sistem kontrol sering kali disebabkan oleh tiga faktor utama, yaitu tuning PID yang tidak tepat, gangguan dari luar, dan aksi aktuasi *valve* yang bersifat nonlinier sehingga menyebabkan terjadinya osilasi tersebut dan mengurangi kualitas produk serta meningkatkan konsumsi energi yang menyebabkan biaya operasional meningkat, sehingga berdampak pada efisiensi secara keseluruhan (Guan et al., 2024). Bahkan, Daiguji & Yamashita (2022) juga menekankan bahwa osilasi tersebut dan tuning kontrol yang buruk menjadi penyebab utama penurunan performa unit proses secara keseluruhan.

Melihat berbagai tantangan yang muncul, diperlukan sistem *final control element* yang tidak hanya memiliki presisi tinggi dalam menggerakkan *valve*, tetapi juga mampu mendeteksi *slip* atau kehilangan langkah yang dapat memengaruhi akurasi posisi aktuasi. Untuk itu, penelitian ini berfokus pada perancangan dan pengembangan *final control element* berbasis *globe valve* yang digerakkan oleh *stepper motor*. Sistem ini dilengkapi dengan pemantauan parameter operasional seperti arus dan temperatur *driver* untuk mencegah kondisi abnormal seperti *over-current* dan *over-temperature*. Selain itu, fitur pemilihan kondisi awal bukaan *valve* (*Normally Open* atau *Normally Closed*) disediakan guna menyesuaikan konfigurasi sistem dengan kebutuhan proses.

Evaluasi performa sistem akan dilakukan dengan membandingkan mode penggerakan *stepper motor* seperti *full-step*, *half-step*, dan *micro-step* untuk mengetahui pengaruhnya terhadap presisi dan kehalusan gerak aktuasi. Selain aspek kelistrikan dan mekanik, analisis karakteristik hidrolis berupa laju aliran dan *pressure drop* pada *valve* juga akan dilakukan guna menilai efektivitas kontrol yang dihasilkan. Dengan pendekatan menyeluruh ini, penelitian diharapkan mampu memberikan kontribusi nyata terhadap pengembangan teknologi *control valve* elektrik yang andal, presisi, dan adaptif terhadap berbagai kondisi operasional di lingkungan industri.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah pada tugas akhir ini adalah:

- a. Bagaimana merancang *Control valve* dengan modifikasi *globe valve* menggunakan *stepper motor*?
- b. Bagaimana performa *stepper motor* sebagai aktuator *globe valve* terhadap karakteristik aliran fluida pada *control valve*?

## 1.3 Batasan Masalah

Penelitian ini memiliki beberapa batasan untuk memastikan fokus dan ketercapaian tujuan yang telah ditetapkan. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

- a. Objek utama dalam penelitian ini adalah *globe valve* berukuran ½ inci, tanpa membahas jenis atau ukuran *valve* lainnya.
- b. Sistem aktuasi menggunakan *stepper motor* tipe Nema17 yang dikendalikan oleh mikrokontroler ESP32, tanpa mempertimbangkan jenis aktuator atau mikrokontroler lain.
- c. Pengujian dilakukan pada dua jenis *driver stepper motor*, yaitu DRV8825 dan TMC2209, untuk analisis perbandingan performa.
- d. Desain mekanik menggunakan *coupling* hasil rancangan sendiri, dan tidak dibandingkan dengan jenis *coupling* lain.
- e. Evaluasi performa sistem dilakukan menggunakan air sebagai fluida uji di lingkungan laboratorium, dengan dukungan *valve tester* yang dilengkapi sensor tekanan dan sensor aliran.
- f. Parameter pengujian difokuskan pada respons aktuator, akurasi langkah motor, efektivitas aktuasi *valve*, kestabilan gerakan, serta karakteristik aliran fluida dan tekanan sebelum dan sesudah *valve*.
- g. Analisis difokuskan pada pengaruh konfigurasi perangkat keras dan pemrograman terhadap keakuratan, stabilitas, dan respons sistem.

#### **1.4 Tujuan**

Tujuan dilakukannya penelitian untuk tugas akhir ini adalah:

- a. Merancang *control valve* dengan modifikasi *globe valve* menggunakan *stepper motor*.
- b. Menganalisis performa dari *stepper motor* sebagai aktuator *globe valve* terhadap karakteristik aliran fluida pada *control valve*.

#### **1.5 Manfaat**

Penelitian tugas akhir yang diusulkan ini diharapkan memberi manfaat sebagai berikut:

- a. Menjadi sarana pembelajaran dan praktikum di laboratorium Departemen Teknik Instrumentasi ITS, khususnya dalam bidang aktuasi dan pengendalian fluida.
- b. Menyediakan solusi *final control element* berbasis aktuator elektrik yang dapat digunakan untuk mengatur laju aliran fluida dalam sistem perpipaan.
- c. Memberikan kontribusi referensi dan dasar pengembangan lebih lanjut bagi penelitian serupa di masa mendatang, khususnya dalam bidang otomasi kontrol berbasis *stepper motor*.

## BAB 2 TINJAUAN PUSTAKA

### 2.1 Hasil Penelitian Terdahulu

Tinjauan pustaka dari penelitian terdahulu yang telah dilakukan, dijadikan sebagai acuan dari pengerjaan tugas akhir ini. Hasil-hasil penelitian tersebut dipaparkan pada Tabel 2.1 berikut.

**Tabel 2.1** *State of The Art*

Judul, Penulis & Tahun	Metode	Hasil
<p><i>Model Based Driving Analysis for A novel Stepped Rotary Flow Control valve</i> (Abuowda et al., 2019)</p>	<p>Penelitian ini menguji kinerja <i>stepper motor</i> sebagai aktuator yang menggerakkan <i>rotary orifice</i> pada <i>control valve</i> dengan <i>inputan</i> dari <i>joystick</i>. Studi ini membandingkan tiga teknik penggerakan <i>stepper motor</i>: <i>full-step</i>, <i>half-step</i>, dan <i>microstepping</i>. Pengujian dilakukan untuk mengevaluasi keefektifan setiap teknik dalam mengurangi riak dan meningkatkan stabilitas gerakan pada <i>valve</i></p>	<p>Hasil penelitian menunjukkan bahwa teknik <i>microstepping</i> memberikan pengurangan riak gesekan yang signifikan dibandingkan dengan <i>full-step</i>, dengan rentang frekuensi sinyal yang lebih bervariasi. Teknik ini juga meningkatkan presisi kontrol gerakan <i>valve, stepper motor</i> yang digunakan sebagai aktuator efektif dalam menggerakkan <i>rotary orifice</i>.</p>
<p><i>Development of an Automatic Water Exchange System for Smart Freshwater Aquarium</i> (Stachowiak &amp; Hemmerling, 2022)</p>	<p>Studi ini mengembangkan sistem pertukaran air otomatis pada akuarium air tawar, yang dikendalikan oleh <i>stepper motor</i> NEMA17 JK42HS48-0406 untuk menggerakkan <i>globe valve</i> dari <i>inputan</i> sensor <i>flow</i> dan temperatur. Tiga metode kontrol <i>stepper motor</i> diuji: metode dinamis, metode minimum <i>integral squared error</i> (ISE), dan metode campuran tanpa regulator PID, untuk menentukan efisiensi dalam pengaturan aliran dan temperatur air.</p>	<p>Hasil penelitian menunjukkan bahwa <i>stepper motor</i> efektif dalam mengatur aliran dan temperatur air melalui <i>globe valve</i>. Metode dinamis memberikan respons pemanasan tercepat, sedangkan metode campuran memberikan kontrol pendinginan yang optimal. Metode minimum ISE, meskipun lebih lambat, memberikan keseimbangan terbaik antara kecepatan dan akurasi, menjadikannya pilihan yang sesuai untuk aplikasi yang memerlukan stabilitas temperatur yang konsisten.</p>

Judul, Penulis & Tahun	Metode	Hasil
<p><i>Design and internal cavitation characteristics of straight-through electric fully enclosed ultra-clean globe valve</i> (Chen et al., 2023)</p>	<p>Penelitian ini mengevaluasi karakteristik kavitasi dari <i>ultra-clean globe valve</i> yang digerakkan oleh <i>stepper motor permanen magnet</i>. Pengujian dilakukan untuk menilai performa <i>valve</i> dalam kondisi operasi tertentu, termasuk analisis kebocoran dan stabilitas gerakan buka-tutup <i>valve</i> pada tekanan tinggi.</p>	<p>Hasil yang didapat yaitu <i>ultra-clean globe valve</i> memiliki performa yang baik dimana tidak ada kebocoran di tekanan tertentu, gerakan bukaan dan menutup yang digerakkan oleh <i>stepper motor</i> juga sangat baik dan memenuhi kriteria pengujian.</p>
<p><i>Analysis of the effects of recycling on process kontrol</i> (Silva et al., 2023)</p>	<p>Studi ini meneliti pengendalian <i>level</i> dalam tangki menggunakan <i>stepper motor</i> NEMA17PM-K342B yang menggerakkan <i>globe valve 3/4"</i> dengan <i>inputan</i> dari HC-SR04 <i>ultrasonic sensor</i>, mikrokontroler yang digunakan yaitu Arduino UNO. Penelitian ini menguji dua mode kontrol, yaitu <i>Proportional-Integral</i> (PI) dan <i>Proportional-Integral-Derivative</i> (PID), untuk mengontrol aliran <i>input</i>, dengan fokus pada stabilitas dan kecepatan respons dalam kondisi operasi dengan tingkat daur ulang yang tinggi.</p>	<p>Hasil dari penelitian ini menunjukkan bahwa <i>stepper motor</i> nema 17 mampu menggerakkan <i>globe valve 3/4"</i>. Mode kontrol PI dan PID yang diuji sama-sama memberikan respon yang baik, namun pada pendekatan siklus tinggi (75%), metode PID memiliki respon yang lebih baik.</p>
<p><i>Toward optimized operation of freshwater generator using computer vision, and its economic and environmental benefits</i> (Park et al., 2023)</p>	<p>Pengoperasian <i>valve</i> outlet pompa distilat FWG secara otomatis dengan mengenali kondisi air melalui <i>sight glass</i> menggunakan teknologi <i>computer vision</i>. Tiga model deep learning (CNN-LSTM, CNN-GRU, dan 3D-CNN) dilatih menggunakan video aliran air tawar di <i>sight glass</i>. Model terbaik kemudian digunakan pada <i>stepper motor</i> dan motor elektrik untuk mengontrol <i>valve</i> outlet pompa distilat FWG secara otomatis.</p>	<p><i>Valve</i> dengan <i>stepper motor</i> menunjukkan kinerja yang lebih stabil dibandingkan <i>valve</i> motor listrik, CNN-LSTM dengan <i>stepper motor</i> memberikan hasil terbaik dalam mengontrol <i>valve</i> dengan presisi tinggi, menghasilkan operasi FWG yang lebih efisien, pengurangan konsumsi bahan bakar, dan dampak lingkungan yang lebih rendah.</p>

Judul, Penulis & Tahun	Metode	Hasil
<p><i>Implementation of An Economical Iot-Based Liquid Tank Kontrol System to Support Online Process Kontrol Practicum</i>(Anugraha et al., 2024)</p>	<p>Penelitian ini mengembangkan sistem praktikum pengendalian proses berbasis IoT yang ekonomis, menggunakan <i>stepper motor</i> sebagai aktuator untuk mengontrol aliran cairan dalam tangki. Sistem ini dikendalikan melalui Arduino Uno dan sensor ultrasonik, dengan antarmuka kontrol dan simulasi berbasis MATLAB Simulink yang dapat diakses secara <i>online</i> melalui TeamViewer.</p>	<p>Hasil penelitian menunjukkan bahwa <i>stepper motor</i> secara efektif mengontrol <i>valve</i>, menjaga level cairan sesuai dengan target yang diinginkan. Sistem ini memungkinkan pelaksanaan praktikum pengendalian proses secara <i>online</i> dengan biaya yang terjangkau dan akurasi yang cukup tinggi, menawarkan solusi yang inovatif untuk pendidikan jarak jauh dalam bidang pengendalian proses.</p>

## 2.2 Dasar Teori

Perancangan dan pengujian sistem *control valve* ini didasarkan pada sejumlah teori yang berkaitan dengan komponen dan prinsip kerja sistem. Teori-teori yang digunakan mencakup jenis dan karakteristik *valve*, prinsip kerja *stepper motor* sebagai aktuator, klasifikasi dan pemilihan *driver motor*, serta sistem kontrol *open-loop* yang digunakan dalam implementasi. Selain itu, teori mengenai sensor arus dan temperatur, *rotary encoder*, dan *switch* juga menjadi landasan dalam menyusun rancangan dan mengevaluasi performa sistem secara menyeluruh.

### 2.2.1 Fluida

Fluida merupakan zat, baik itu cairan ataupun gas yang dapat mengalir dan memiliki berbagai karakteristik fisik yang penting untuk dipahami dalam konteks dinamika fluida. Kinerja dari suatu fluida dapat diketahui melalui pengukuran laju aliran, tekanan, serta temperatur fluida tersebut dalam sistem. Fluida yang mengalir dalam sistem perpipaan akan kehilangan energi akibat gesekan saat bergerak melalui pipa, hal ini juga dapat terjadi saat fluida melewati *valve* atau mengalami perubahan arah seperti pada *elbow* ataupun *tee* pipa (Mott & Untener, 2015, p.2).

Pengukuran tekanan serta laju aliran fluida di suatu sistem sangat penting untuk memahami kinerja dari fluida tersebut. Tekanan ( $P$ ) bisa didefinisikan sebagai jumlah gaya ( $F$ ) yang diberikan pada suatu permukaan ( $A$ ) dengan satuan standar dalam sistem SI adalah  $N/m^2$ , yang bisa juga disebut *pascal (Pa)*, satuan SI lain yang mudah digunakan adalah  $kPa$ ,  $MPa$ , dan  $bar$ . *Volume flow rate (Q)* merupakan volume dari fluida yang mengalir melewati permukaan tertentu per satuan waktu yang nilainya bisa diketahui dari besarnya permukaan ( $A$ ) dikalikan dengan kecepatan rata-rata aliran ( $v$ ) dengan satuan SI dari *volume flow rate* adalah  $m^3/s$  (Mott & Untener, 2015).

### 2.2.2 Performa pada Control Valve

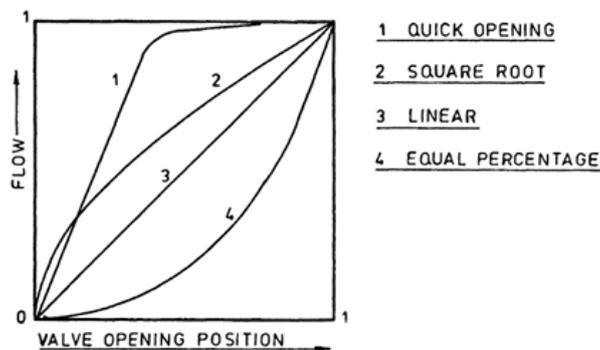
Performa *control valve* sangat dipengaruhi oleh beberapa faktor utama, seperti *dead band*, *rangeability*, dan karakteristik aliran (*flow characteristic*). *Dead band* yang rendah sangat penting untuk respons cepat dan akurat, dengan nilai ideal kurang dari 1% untuk memastikan stabilitas sistem tanpa memerlukan *valve positioner*. Namun, pada kebanyakan kasus,

*positioner* tetap diperlukan untuk mengurangi *dead band* dan meningkatkan akurasi kontrol (Baumann, 2023).

Faktor lain adalah *rangeability*, yaitu rasio maksimum hingga minimum aliran yang dapat dikendalikan oleh *valve*. *Rangeability* yang tinggi memungkinkan *valve* bekerja efektif pada berbagai kondisi aliran tanpa kehilangan presisi. Selain itu, ukuran *valve* harus sesuai dengan kebutuhan sistem untuk menghindari masalah seperti *oversizing*, yang dapat menyebabkan respons kontrol tidak stabil pada aliran rendah dan memperbesar *dead time* (Baumann, 2023).

Karakteristik aliran (*flow characteristic*) atau *installed gain* juga berperan penting, memastikan *valve* dapat mengatur aliran sesuai dengan kebutuhan sistem meskipun terjadi variasi pada penurunan tekanan (*pressure drop*). Dengan mempertimbangkan faktor-faktor tersebut, desain dan pemilihan *control valve* yang tepat sangat penting untuk memastikan kinerja yang optimal pada sistem kontrol fluida (Baumann, 2023).

### 2.2.3 Karakteristik Laju Aliran pada Control Valve



Gambar 2.1 Karakteristik Aliran untuk Control Valve (Baumann, 2023).

*Control valve* merupakan *final control element* yang paling umum digunakan dalam sistem kontrol industri. Fungsi utamanya adalah untuk memodulasi aliran fluida guna menjaga variabel proses berada pada nilai yang diinginkan. Baumann (2023) menjelaskan bahwa peran utama dari *control valve* adalah sebagai elemen yang mampu melakukan *throttling*, yaitu pengaturan aliran secara kontinu dengan mengatur posisi bukaan internal *valve*. *Control valve* dibedakan dari *automated valve* atau *on-off valve* berdasarkan kemampuannya untuk menyesuaikan posisi bukaan secara tak terbatas selama proses berlangsung.

*Control valve* memiliki tiga komponen fisik utama, yaitu *body valve* sebagai bagian tekanan utama, aktuator yang menggerakkan elemen *throttle*, dan berbagai aksesoris seperti *positioner*, *I/P transducer*, atau sensor posisi yang mendukung pengoperasiannya Baumann (2023). Dengan kemampuannya untuk menyesuaikan aliran fluida secara presisi, *control valve* menjadi elemen penting dalam sistem kontrol proses yang melibatkan pengaturan tekanan, temperatur, maupun laju aliran.

Dalam karakteristik alirannya, laju aliran fluida melalui *control valve* sangat dipengaruhi oleh besarnya hambatan yang ditimbulkan oleh *valve* tersebut. Zappe & Smith (2004) menyatakan bahwa hambatan ini diukur menggunakan koefisien resistansi  $\zeta$  (*zeta*), yang menyatakan besarnya *friction loss* dalam bentuk *headloss* atau *pressure drop*. Koefisien lain yang digunakan untuk mengkuantifikasi kinerja aliran pada *control valve* antara lain adalah  $C_v$  (*flow coefficient* dalam satuan *US gal/menit* dengan  $\Delta p = 1 \text{ psi}$ ),  $K_v$  (dalam  $\text{m}^3/\text{jam}$  dengan  $\Delta p = 1 \text{ bar}$ ), dan  $A_v$  (dalam  $\text{m}^3/\text{detik}$  dengan  $\Delta p = 1 \text{ Pa}$ ).

Hubungan antara laju aliran fluida dan kehilangan tekanan umumnya mengikuti hukum akar kuadrat, di mana kecepatan aliran ( $v$ ) berbanding lurus terhadap akar kuadrat dari tekanan diferensial ( $\Delta p$ ). Oleh karena itu, karakteristik buka-tutup *valve* akan sangat memengaruhi

respons aliran terhadap perubahan posisi *valve*. Pada posisi pembukaan tertentu, perubahan kecil pada posisi *throttle* akan memberikan perubahan yang signifikan terhadap aliran fluida, tergantung pada desain internal *valve* dan parameter seperti  $C_v$  atau  $K_v$  (Zappe & Smith, 2004).

$$C_v = Q \left( \frac{G}{\Delta p} \right)^{\frac{1}{2}} \quad (2.1)$$

$$K_v = Q \left( \frac{G}{\Delta p} \right)^{\frac{1}{2}} \quad (2.2)$$

$$A_v = Q \left( \frac{\rho}{\Delta p} \right)^{\frac{1}{2}} \quad (2.3)$$

Dimana :

$Q$  = flow rate ( $C_v = U.S. gal/min$ ) ( $K_v = m^3/hour$ ) ( $A_v = m^3/s$ )

$G$  = specific gravity

$\Delta p$  = Perbedaan tekanan ( $C_v = lb/in.^2$ ) ( $K_v = bar$ ) ( $A_v = Pa$ )

$\rho$  = Densitas fluida,  $kg/m^3$

#### 2.2.4 Valve

*Valve* merupakan komponen penting dalam sistem pengolahan fluida yang digunakan untuk mengontrol aliran dalam berbagai aplikasi industri. *Valve* manual dioperasikan secara langsung menggunakan *handwheel*, *wrench*, atau aktuator *on/off*, dan memiliki tiga fungsi utama, yaitu menghentikan dan memulai aliran, mengendalikan laju aliran, serta mengalihkan aliran (Zappe & Smith, 2004, p.47). Berdasarkan cara bagian penutupnya bergerak menuju *seat*, *valve* dapat diklasifikasikan ke dalam empat kelompok utama. Pertama, *closing-down valves*, yaitu jenis *valve* dengan penutup berbentuk *stopper* yang bergerak lurus menuju *seat* sepanjang sumbu aliran, seperti *globe valve* dan *piston valve*. Kedua, *slide valves*, yang memiliki penutup berbentuk seperti gerbang dan bergerak melintang terhadap aliran, contohnya adalah *parallel gate valve* dan *wedge gate valve*. Ketiga, *rotary valves*, yang bekerja dengan penutup berbentuk *plug* atau *disc* yang berputar pada sumbu tegak lurus terhadap arah aliran, seperti *plug valve*, *ball valve*, dan *butterfly valve*. Keempat, *flex-body valves*, yang menggunakan elemen penutup fleksibel sebagai bagian dari *body valve* itu sendiri, seperti pada *pinch valve* dan *diaphragm valve*.

Penggunaan *valve* juga disesuaikan dengan fungsinya. Untuk fungsi penghentian dan pemulaan aliran, umumnya dipilih *valve* yang memiliki resistansi aliran rendah dan jalur aliran lurus seperti tipe *slide valves*, *rotary valves*, dan *flex-body valves* yang banyak digunakan karena karakteristiknya mendukung fungsi tersebut. Untuk fungsi pengendalian laju aliran, diperlukan *valve* yang dapat diatur dengan presisi, di mana *closing-down valves* sangat cocok karena memiliki hubungan proporsional langsung antara ukuran bukaan dan pergerakan penutup, memungkinkan pengaturan laju aliran yang akurat. Sementara itu, untuk fungsi pengalihan aliran, *valve* yang memiliki tiga atau lebih *port* seperti *plug valve* dan *ball valve* merupakan pilihan ideal karena desainnya yang memungkinkan pengalihan aliran secara efisien.

Secara umum, karakteristik aliran *valve* dapat dikategorikan menjadi tiga tipe, yaitu tipe *quick opening*, *linear*, dan *equal percentage* (Nguyen et al., 2020). Tipe *quick opening* biasa digunakan pada sistem keselamatan atau pendinginan karena mampu menghasilkan lonjakan besar laju aliran pada awal pembukaan *valve*. Tipe *linear* cocok digunakan pada sistem yang membutuhkan kendali presisi terhadap fluida, seperti pada kontrol *level*, karena memiliki

hubungan linier antara laju aliran dan tingkat bukaan. Sedangkan tipe *equal percentage* banyak diterapkan dalam sistem proses pengendalian tekanan dan perpindahan panas karena memberikan laju aliran yang meningkat secara kecil pada awal pembukaan dan signifikan saat mendekati pembukaan penuh.

### 2.2.5 *Globe valve*



Gambar 2.2 *Globe Valve*

*Globe valve* merupakan jenis *valve* kontrol yang mengatur aliran fluida melalui pergerakan *disk* penutup secara tegak lurus terhadap dudukannya. Desain ini menghasilkan hubungan yang hampir *linear* antara bukaan *valve* dan aliran fluida, sehingga *globe valve* sangat cocok untuk aplikasi pengaturan laju aliran (Zappe & Smith, 2004). Mekanisme penggerakannya memanfaatkan batang ulir (*screwed stem*) yang memungkinkan kontrol tekanan penutupan secara akurat serta mengurangi gesekan saat *disk* bersentuhan dengan *seat*. Hal ini membuat *globe valve* unggul dalam kemampuan *sealing* atau penyekatan. Selain untuk pengaturan aliran, *globe valve* juga dapat digunakan dalam operasi buka-tutup cepat karena jarak tempuh *disk* relatif pendek dan desain *seat* yang tangguh terhadap gerakan frekuentif. Fleksibilitas ini menjadikan *globe valve* sebagai pilihan utama dalam berbagai sistem perpipaan fluida (Zappe & Smith, 2004).

### 2.2.6 *Stepper Motor*



Gambar 2.3 *Stepper Motor*

*Stepper motor* merupakan salah satu jenis motor listrik yang dirancang untuk bergerak dalam langkah-langkah diskrit. Setiap langkah terjadi ketika motor menerima sinyal pulsa dari kontroler, yang kemudian menggerakkan poros pada sudut tertentu. Karakteristik ini membuat *stepper motor* sangat ideal untuk aplikasi *open-loop position* kontrol, karena posisi akhir motor dapat ditentukan dengan hanya menghitung jumlah pulsa tanpa perlu sensor umpan balik (Hughes & Drury, 2019, p.376).

Terdapat dua jenis utama *stepper motor*, yaitu *variable reluctance* (VR) dan *hybrid*. Motor tipe hybrid menggabungkan keunggulan dari motor VR dan motor dengan magnet permanen. Umumnya, *hybrid stepper motor* digunakan dalam aplikasi yang membutuhkan sudut langkah kecil dan akurasi tinggi. Misalnya, motor dengan 50 gigi rotor dan empat fasa stator dapat menghasilkan 200 langkah per putaran, yang berarti satu langkahnya adalah  $1,8^\circ$  (Hughes & Drury, 2019, p.381). Persamaan (2.4) dapat digunakan untuk menghitung sudut langkah *stepper motor*.

$$\text{Sudut langkah} = \frac{360^\circ}{\text{gigi rotor} \times \text{fasa stator}} \quad (2.4)$$

Dari persamaan 2.4 kita bisa mendapat nilai sudut langkah serta jumlah langkah perputaran, nilai ini bisa digunakan untuk menghitung berapa total langkah untuk membuka atau menutup *valve* secara penuh dengan persamaannya dijabarkan pada 2.5 dan 2.6, kemudian dari 2 persamaan tersebut kita bisa mencari berapa langkah yang dibutuhkan motor untuk setiap 1% bukaan atau tutupan *valve*.

$$\text{Total Langkah} = \frac{\text{Total derajat bukaan valve}}{\text{Sudut langkah}} \quad (2.5)$$

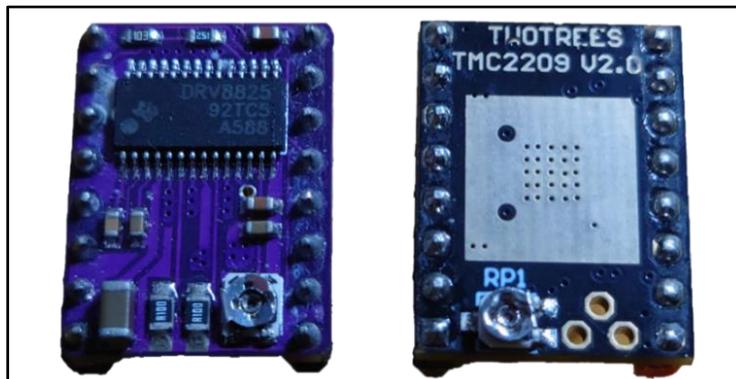
Atau

$$\text{Total Langkah} = \frac{(\text{Total derajat bukaan valve} \times \text{langkah 1 putaran motor})}{360} \quad (2.6)$$

Bernardi et al. (2022) menekankan bahwa *hybrid stepper motor* (HSM) banyak digunakan dalam berbagai aplikasi industri karena keandalan tinggi, biaya rendah, dan kemudahan integrasi dengan sistem kontrol berbasis mikrokontroler secara langsung tanpa sensor posisi. Pengendalian posisi yang sederhana namun presisi ini menjadikan *stepper motor* sebagai aktuator yang ideal dalam banyak sistem otomasi.

Selanjutnya, penelitian oleh Silva et al. (2023) menunjukkan bahwa penggunaan *stepper motor* NEMA17 pada sistem kontrol *globe valve*  $\frac{3}{4}$  inci terbukti efektif dalam membuka dan menutup *valve* secara presisi. Berdasarkan keberhasilan tersebut, penggunaan NEMA17 pada *globe valve* berukuran  $\frac{1}{2}$  inci dalam penelitian ini dapat dianggap sesuai dan layak secara teknis.

### 2.2.7 Driver Motor



Gambar 2.4 Driver Stepper Motor

*Driver stepper motor* merupakan komponen kunci dalam sistem pengendalian *stepper motor* karena berfungsi mengubah sinyal dari kontroler menjadi pulsa yang menggerakkan motor secara presisi. Pemilihan *driver* yang tepat sangat penting agar performa aktuasi sesuai dengan kebutuhan sistem. Berdasarkan catu daya dan arus keluarannya, *driver stepper motor* dapat diklasifikasikan menjadi tiga kategori utama, yaitu daya tinggi, daya menengah, dan daya rendah (Prodanov & Dankov, 2022). *Driver* daya tinggi memiliki catu daya di atas 50V dan arus keluaran lebih dari 5A. *Driver* daya menengah beroperasi pada rentang tegangan 20 - 50V dan arus keluaran antara 2 - 5A, sedangkan *driver* daya rendah memiliki catu daya hingga 35 - 40V dan arus keluaran hingga 2A.

Pemilihan *driver* yang tepat memerlukan pertimbangan beberapa faktor penting untuk memastikan kesesuaian dengan aplikasi yang dimaksud (Hughes & Drury, 2019, p.411). Pertama, *driver* harus memiliki rating daya yang sesuai, termasuk tegangan dan arus yang dapat ditangani, untuk mengakomodasi kebutuhan daya *stepper motor*. Selain itu, karakteristik *driver* dalam menghasilkan *pulse* yang presisi sangat penting. *Driver* juga harus mampu mendukung kecepatan maksimum dan rentang kecepatan yang diperlukan oleh aplikasi. Kebutuhan beban seperti karakteristik torsi-kecepatan motor harus diperhatikan agar *driver* dapat menyediakan torsi yang cukup pada berbagai kecepatan. Pertimbangan umum lainnya meliputi kemampuan *driver* dalam menangani operasi regeneratif dan pengereman, siklus kerja yang sesuai, sistem pendinginan yang memadai, standar dimensi yang sesuai dengan ruang yang tersedia, serta kemampuan *driver* untuk bekerja dengan suplai daya tanpa menyebabkan gangguan atau getaran yang merugikan.

Dalam tugas akhir ini, dua jenis *driver stepper motor* akan digunakan dan dibandingkan, yaitu DRV8255 dan TMC2209 untuk menguji keefektifan kedua *driver* dalam mengendalikan *stepper motor* NEMA 17 untuk membuka dan menutup *globe valve* 1/2". Pemilihan kedua *driver* ini didasarkan pada penelitian oleh (Prodanov & Dankov, 2022), yang menguji keandalan berbagai *low-power stepper motor driver* melalui metode pencatatan jumlah tingkat kegagalan kapasitor dan resistor *driver* serta panas yang dihasilkan *driver*. DRV8255 merupakan *driver stepper motor* daya rendah yang memiliki performa tinggi dalam aplikasi yang memerlukan torsi yang konsisten dan akurasi yang tinggi. TMC2209 juga merupakan *driver stepper motor* yang sangat efisien, kelebihan drive ini memiliki fitur-fitur canggih seperti *sensorless homing*, operasi yang sangat tenang, dan pengaturan arus otomatis.

### 2.2.8 Switch

*Switch* atau sakelar merupakan perangkat penting dalam berbagai sistem kontrol industri dan otomasi karena berfungsi untuk membuka atau menutup jalur arus listrik sebagai bentuk pengendalian. Berdasarkan cara pengoperasiannya, *switch* dibedakan menjadi dua kategori utama, yaitu *switch* yang dioperasikan secara manual (*manually operated switches*) dan *switch* yang dioperasikan secara mekanik (*mechanically operated switches*). *Switch* manual dikendalikan langsung oleh pengguna melalui gerakan tangan. Beberapa contoh umum dari jenis ini meliputi *pushbutton switch*, *toggle switch*, *selector switch*, dan *knife switch*. Di antara semua jenis tersebut, *pushbutton switch* merupakan salah satu yang paling sering digunakan karena kesederhanaan dan keandalannya.

*Pushbutton switch* bekerja dengan membuka atau menutup kontak listrik saat tombol ditekan, dan kembali ke posisi awal ketika dilepaskan. Berdasarkan karakteristik kontakannya, *pushbutton switch* dapat dibedakan menjadi beberapa jenis (Petruzella, 2023, p.93). Jenis yang pertama adalah *Normally Open* (NO), di mana jalur arus hanya terhubung saat tombol ditekan dan kembali terbuka saat dilepas. Jenis kedua adalah *Normally Closed* (NC), yang justru memutus jalur saat ditekan dan kembali terhubung saat tombol dilepaskan. Selain itu, terdapat juga jenis *Break-Before-Make pushbutton switch*, yang memiliki dua buah kontak: kontak atas yang *Normally Closed* dan kontak bawah yang *Normally Open*. Saat tombol ditekan, kontak

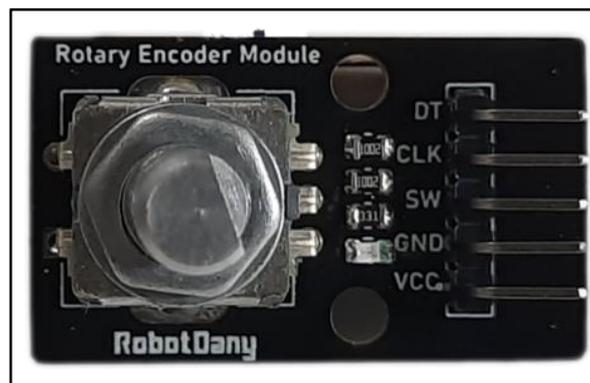
atas akan terbuka terlebih dahulu sebelum kontak bawah tertutup, sehingga tidak terjadi hubungan singkat antar jalur saat transisi berlangsung.



Gambar 2.5 Toggle Switch

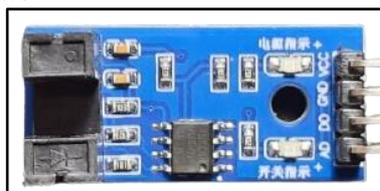
Dengan adanya berbagai jenis *switch* ini, *toggle switch* digunakan untuk mengendalikan operasi *globe valve* yang digerakkan oleh *stepper motor* NEMA17. *Toggle switch* berfungsi mengubah mode operasi *valve* antara *Normally open* (NO) dan *Normally closed* (NC), memberikan fleksibilitas dalam pengendalian. Implementasi *switch* ini meningkatkan keandalan, dan fleksibilitas sistem.

### 2.2.9 Rotary encoder



Gambar 2.6 Rotary Encoder KY-040

*Incremental encoder* merupakan perangkat yang digunakan untuk mendeteksi gerakan. *Encoder* ini memberikan sejumlah pulsa yang sama dan merata setiap putaran, setiap inci ataupun setiap milimeter gerakan linier. *Encoder incremental* ini menghasilkan pulsa setiap kali dan hanya saat poros *encoder* berputar, sehingga memungkinkan pelacakan gerakan dengan presisi tinggi (Petruzella, 2023, p.160). Modul KY-040 pada gambar 2.6 merupakan salah satu *rotary encoder* yang memanfaatkan prinsip kerja *incremental rotary encoder*, modul ini juga dilengkapi dengan *push button switch*.

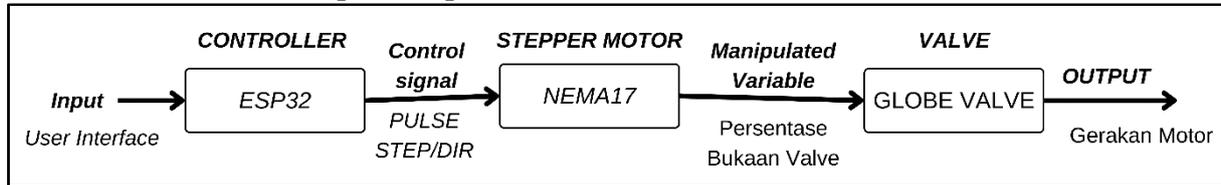


Gambar 2.7 Optical Rotary Encoder

*Optical rotary encoder* juga merupakan salah satu modul yang menerapkan prinsip *incremental encoder* yang memberikan pulsa *high* ketika sensor terhalang objek dan *low* ketika tidak ada objek yang menghalangi sensornya. Sehingga pada penelitian ini, modul KY-040

digunakan untuk mengontrol bukaan *globe valve* yang digerakkan oleh *stepper motor* NEMA17, *encoder* ini berfungsi sebagai *input* manual persentase *valve* untuk mengontrol posisi *valve* dengan menghasilkan pulsa yang dihitung oleh mikrokontroler (ESP32). Sedangkan *Optical Rotary encoder* digunakan sebagai sistem *safety* untuk mendeteksi *slip* pada *valve*.

### 2.2.10 Sistem Kontrol *Open Loop*



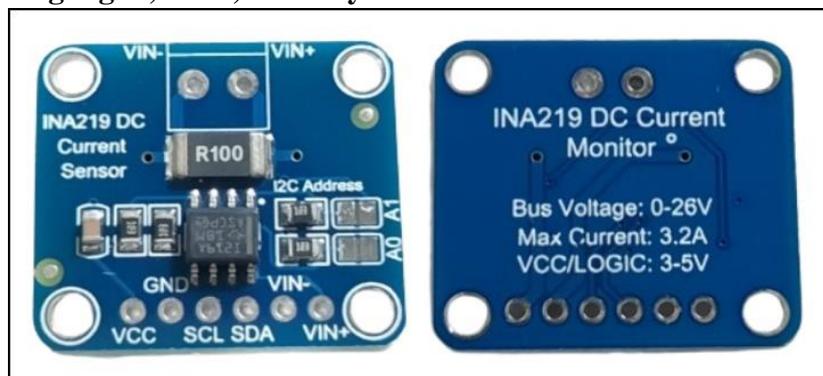
Gambar 2.8 Sistem Kontrol *Open-loop*

Sistem kontrol *open-loop* merupakan jenis sistem kontrol di mana *output* tidak memengaruhi aksi kontrol, karena tidak terdapat proses umpan balik. Dalam sistem ini, *output* tidak diukur maupun dibandingkan dengan *input*, sehingga sistem beroperasi berdasarkan perintah awal yang telah ditentukan tanpa mempertimbangkan kondisi aktual dari hasil keluaran. Sistem kontrol *open-loop* hanya dapat digunakan secara efektif apabila hubungan antara *input* dan *output* diketahui dengan baik serta tidak terdapat gangguan internal maupun eksternal yang dapat memengaruhi hasil (Ogata, 2010).

Keuntungan utama dari sistem ini adalah desainnya yang lebih sederhana dibandingkan dengan sistem kontrol *closed-loop*, sehingga lebih mudah dalam proses perancangan dan perawatan. Selain itu, sistem *open-loop* umumnya lebih ekonomis karena tidak memerlukan sensor umpan balik tambahan. Tidak adanya *loop* umpan balik juga membuat sistem ini bebas dari permasalahan kestabilan yang umum terjadi pada sistem *closed-loop*. Sistem ini sangat cocok untuk digunakan dalam kondisi di mana *output* sulit diukur atau pengukuran tersebut tidak efisien secara biaya.

Namun, sistem kontrol *open-loop* juga memiliki beberapa keterbatasan signifikan. Sistem ini cenderung rentan terhadap gangguan dan perubahan parameter sistem yang menyebabkan *output* tidak sesuai dengan nilai yang diinginkan. Oleh karena itu, sistem ini memerlukan kalibrasi ulang secara berkala untuk menjaga konsistensi performa. Dengan demikian, meskipun sistem kontrol *open-loop* menawarkan kemudahan implementasi dan biaya yang lebih rendah, penggunaannya tetap harus disesuaikan dengan kondisi dan karakteristik aplikasi yang tidak terlalu bergantung pada akurasi keluaran atau ketahanan terhadap gangguan.

### 2.2.11 Sensor Tegangan, Arus, dan Daya

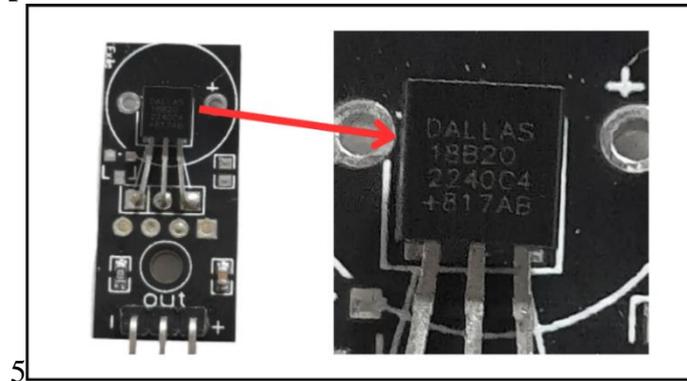


Gambar 2.9 Sensor Arus dan Tegangan (INA219)

Sensor tegangan, arus, dan daya seperti INA219 bekerja berdasarkan prinsip dasar hukum *Ohm* dan aturan *Kirchhoff* dalam analisis rangkaian. Hukum *Ohm* menyatakan bahwa tegangan

yang melintasi sebuah resistor sebanding dengan hasil kali arus yang mengalir melaluinya dan nilai resistansi resistor tersebut. Selain itu, aturan *Kirchhoff* menjelaskan bahwa jumlah arus yang masuk ke suatu simpul dalam rangkaian harus sama dengan jumlah arus yang keluar dari simpul tersebut. Dengan menggunakan prinsip-prinsip ini, INA219 mampu mengukur arus dan tegangan pada suatu rangkaian, serta menghitung daya dengan mengalikan nilai tegangan dan arus yang terukur (Wilson, 2005).

### 2.2.12 Sensor Temperatur



Gambar 2.10 Sensor Temperatur (DS18B20)

Temperatur merupakan variabel yang paling banyak diamati karena memiliki efek yang signifikan terhadap sebuah proses bahkan sampai tingkat molekuler, temperatur sendiri merupakan jumlah energi panas didalam suatu objek ataupun sistem. Sensor Temperatur mendeteksi perubahan parameter seperti resistansi ataupun tegangan *output* yang sesuai dengan perubahan temperatur, ada dua tipe dasar dari sensor ini yaitu kontak yang mendeteksi panas melalui kontak dengan media atau objek yang diindera, kemudian ada non-kontak yang menginterpretasikan energi radiasi sumber panas dalam bentuk energi yang dipancarkan di bagian inframerah spektrum elektromagnetik. Sensor temperature juga terdiri dari tiga kelompok, pertama yaitu elektro-mekanis seperti thermostat, kedua ada elektronik seperti silicon sensor, *infrared (IR) pyrometry* dan termokopel, ketiga ada resistif seperti *Resistance Temperature Detector (RTD)* (Wilson, 2005). DS18B20 sendiri merupakan sensor temperatur yang bekerja secara kontak dan termasuk dalam kelompok sensor temperatur elektronik yang menggunakan silicon sensor berbasis semikonduktor dengan *output* digital.

### 2.2.13 Sensor Tekanan



Gambar 2.11 Pressure Sensor

Sensor tekanan adalah perangkat yang digunakan untuk mengukur tekanan, baik tekanan relatif maupun absolut. Tekanan relatif diukur sebagai perbedaan terhadap tekanan sekitar, sedangkan tekanan absolut diukur terhadap vakum. Tekanan, sebagai konsep mekanis, dinyatakan dalam satuan *pascal* (Pa), yang setara dengan satu *newton* per meter persegi, atau dalam beberapa kasus, atmosfer (atm) (Fraden, 2016)

Sensor tekanan bekerja dengan mendeteksi deformasi elemen yang berubah bentuk ketika terpapar tekanan. Elemen deformasi ini dapat berupa tabung Bourdon, diafragma, kapsul, atau bellow, yang mengubah bentuk di bawah tekanan dan mengonversi perubahan ini menjadi sinyal listrik untuk menentukan nilai tekanan (Fraden, 2016).

Dalam penelitian ini, sensor tekanan digunakan untuk mengumpulkan data performa *control valve*. Meskipun bukan fokus utama, sensor ini penting untuk mengevaluasi efektivitas dan efisiensi *control valve* dalam mengatur aliran fluida, dengan menyediakan data tekanan yang diperlukan (Fraden, 2016).

#### 2.2.14 Sensor Aliran



Gambar 2.12 *Flow Sensor*

*Flow sensor* adalah perangkat yang digunakan untuk mengukur laju aliran media bergerak seperti cairan, udara, dan gas. Laju aliran mengacu pada kecepatan pergerakan volume dalam fluida, yang dalam aliran stabil tetap konstan seiring waktu. Unit SI untuk laju aliran massa adalah kilogram per detik (kg/s), sedangkan satuan umum AS adalah *pound* per detik (lb/s). Untuk gas, laju aliran massa atau volume harus dispesifikasikan pada tekanan tertentu (Fraden, 2016).

Salah satu jenis *flow sensor* yang populer dalam industri adalah *flowmeter* tipe turbin. *Flowmeter* ini berfungsi seperti kincir angin, menggunakan kecepatan rotasi untuk menentukan kecepatan aliran. Konstruksinya terdiri dari rotor turbin berbilah yang terpasang dalam tabung aliran. Bilah rotor berputar sebanding dengan laju aliran cairan melalui tabung, dan sensor pickup magnetik mendeteksi gerakan ini. Fluida yang mengalir menyebabkan rotor berputar, menghasilkan pulsa pada koil pickup yang diterjemahkan menjadi frekuensi. Frekuensi ini kemudian dikonversi menjadi unit pengukuran seperti galon per menit (Petruzella, 2023).

Dalam tugas akhir ini, *flow sensor* digunakan sebagai alat penunjang untuk mengambil data performa dari *control valve*. Data aliran yang diukur oleh sensor ini akan digunakan untuk menganalisis efektivitas dan efisiensi *control valve* dalam mengatur aliran fluida. Oleh karena itu, meskipun *flow sensor* bukan fokus utama penelitian, perannya sangat penting dalam menyediakan data yang diperlukan untuk evaluasi performa *control valve*.

## BAB 3 METODOLOGI

### 3.1 Metode yang Digunakan

Dalam penelitian ini menggunakan pendekatan campuran yang melibatkan metode kualitatif dan kuantitatif untuk memastikan pengembangan sistem *control valve* yang akurat dan andal. Metode yang digunakan mencakup beberapa tahapan yang sistematis, yaitu perancangan, implementasi, dan pengujian dari sistem *control valve* yang dikendalikan oleh *stepper motor* dan *rotary encoder KY-040*.

Metode kualitatif digunakan dalam tahap perancangan, di mana desain sistem dan pemilihan komponen dilakukan berdasarkan studi literatur dan analisis kebutuhan sistem. Metode kuantitatif diterapkan pada tahap pengujian, di mana berbagai data performa sistem dikumpulkan dan dianalisis secara statistik. Data yang dihasilkan dari pengujian ini akan digunakan untuk mengevaluasi keefektifan sistem dan melakukan perbaikan jika diperlukan. Dengan kombinasi pendekatan kualitatif dan kuantitatif, penelitian ini diharapkan dapat menghasilkan sistem *control valve* yang fungsional, efisien, dan sesuai dengan aplikasi yang diinginkan.

### 3.2 Bahan dan Peralatan yang Digunakan

Dalam pengerjaan tugas akhir ini, bahan dan peralatan yang digunakan sangat berperan penting dalam memastikan keberhasilan perancangan dan implementasi sistem *control valve*. Bahan dan peralatan yang dipilih harus sesuai dengan spesifikasi dan kebutuhan dari desain sistem yang direncanakan. Berikut adalah rincian dari bahan dan peralatan yang digunakan dalam penelitian ini.

#### 3.2.1 Bahan

Bahan yang digunakan dalam penelitian ini meliputi berbagai komponen elektronik dan mekanik yang diperlukan untuk merancang dan membangun sistem *control valve*. Beberapa bahan utama yang digunakan adalah sebagai berikut.

**Tabel 3.1** Bahan yang Digunakan

No.	Bahan	Fungsi
1.	<i>Globe valve</i>	<i>Valve</i> utama
2.	<i>Stepper motor</i>	Penggerak <i>valve</i> utama
3.	<i>Driver Stepper motor</i>	Mengendalikan <i>stepper motor</i> dengan arus yang sesuai
4.	<i>Rotary encoder KY-040</i>	Sebagai <i>input</i> penggerak bukaan/tutup <i>valve</i>
5.	<i>Optical Rotary Encoder</i>	Sebagai <i>safety</i> pergerakan <i>valve</i>
6.	<i>Toggle switch</i>	Untuk mengubah mode operasi <i>Normally close/ Normally open</i>
7.	ESP32	Kontroler utama sistem
8.	Sensor Temperatur (DS18B20)	Sebagai monitoring temperatur dari <i>driver stepper motor</i>
9.	Sensor Arus (INA219)	Sebagai monitoring daya yang dipakai oleh sistem

No.	Bahan	Fungsi
10.	PSU 12v 3A	Catu daya utama
11.	<i>Electrolytic Capacitor</i> (Elco) 16v 100uf	Filter <i>driver stepper motor</i>
12.	<i>Step Down 5V</i>	Catu daya untuk esp32
13.	LCD OLED 128x64	Tampilan GUI
14.	Saklar On/Off	Untuk menyalakan atau mematikan sistem
15.	Fuse 3A	<i>Safety</i> antara sumber AC dan PSU
16.	Kabel AWG 20	Koneksi dari PSU ke PCB utama dan <i>StepDown</i> .
17.	Kabel 3 x 0,75	Koneksi sumber listrik AC untuk PSU
18.	<i>Printed Circuit Board</i> (PCB)	Tempat koneksi <i>microkontroller</i> dengan komponen lainnya
19.	Filamen PLA	Bahan untuk <i>control valve</i>

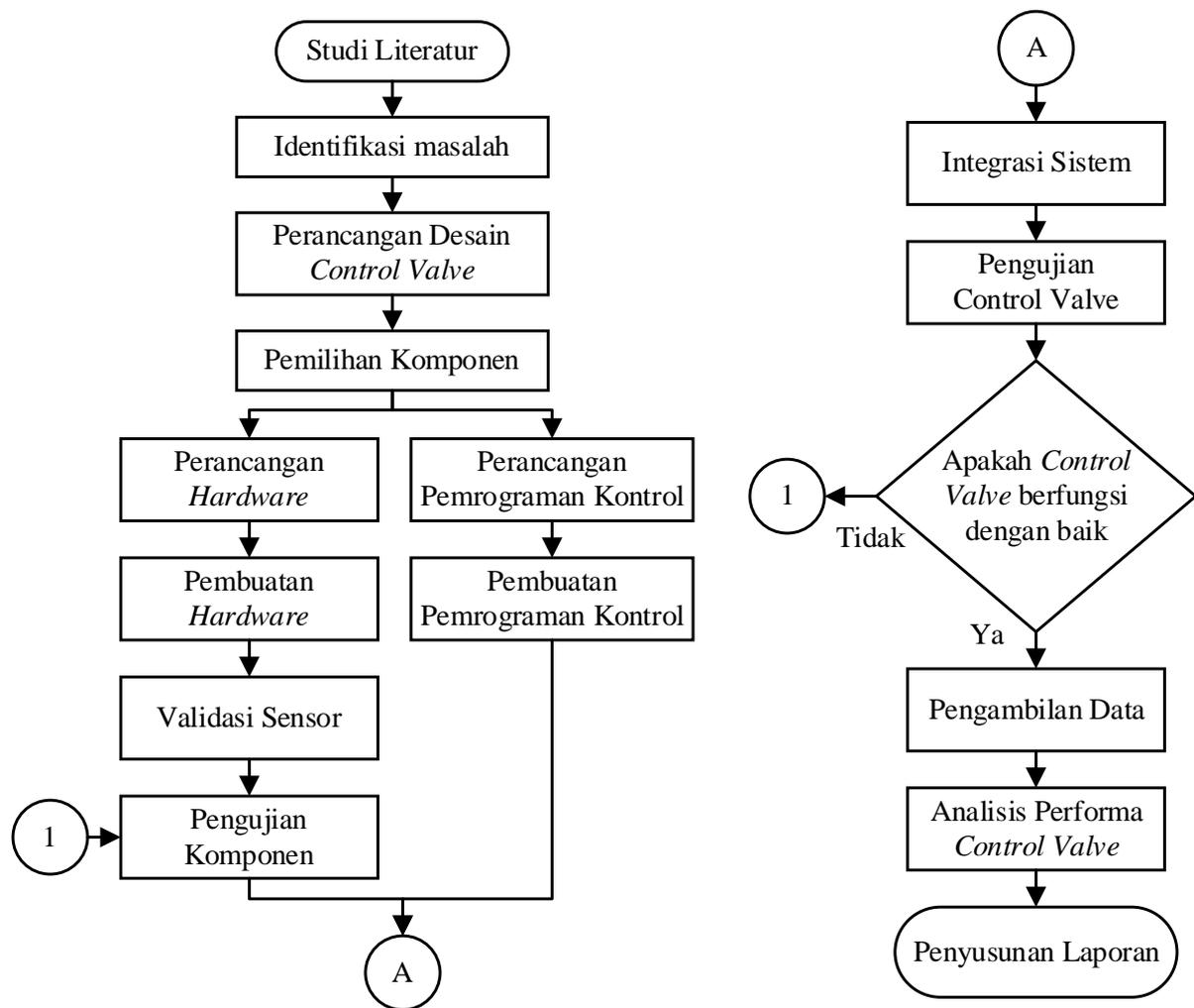
### 3.2.2 Alat

Peralatan yang digunakan dalam penelitian ini mencakup berbagai alat yang diperlukan untuk merakit, menguji, dan menganalisis sistem *control valve*. Berikut adalah peralatan utama yang digunakan.

**Tabel 3.2** Alat yang digunakan

No.	Alat	Fungsi
1.	Set Obeng	Mengencangkan atau melonggarkan sekrup
2.	3D Print	Mencetak bagian <i>control valve</i>
3.	Jangka Sorong	Alat ukur
4.	Penggaris / Meteran	Alat ukur
5.	Busur	Alat ukur
5.	Kunci pas	Mengencangkan/melonggarkan <i>valve</i>
6.	Tang Potong	Memotong kabel
7.	Solder	Melelehkan timah untuk menyambungkan komponen di pcb
9.	<i>Multimeter</i>	Alat ukur kelistrikan
10.	<i>Thermometer Digital</i> FY-10	Sebagai validator untuk sensor temperatur
11.	Tang Ampere / <i>Amperemeter</i>	Sebagai validator untuk sensor ampere
12.	<i>Control valve Tester</i>	Sebagai monitoring <i>flowrate</i> dan <i>pressure drop</i> pada <i>control valve</i>

### 3.3 Urutan Pelaksanaan Penelitian



Gambar 3.1 Urutan Pelaksanaan Penelitian

#### 3.3.1 Studi Literatur

Studi literatur dalam tugas akhir ini mencakup berbagai aspek yang relevan dengan perancangan dan pembuatan *control valve*. Beberapa topik utama yang dikaji meliputi prinsip dasar fluida dan karakteristik aliran pada *control valve*, berbagai jenis *valve* termasuk *globe valve*, serta penggunaan *stepper motor* dan *driver* untuk menggerakkan *valve*. Selain itu, sistem kontrol *open loop* dan respon dinamik juga dianalisis untuk memastikan stabilitas dan akurasi sistem yang dirancang.

Literatur lain yang ditinjau mencakup penggunaan komponen lain seperti *rotary encoder KY-040*, *optical rotary encoder*, sensor temperatur (DS18B20), sensor arus (INA219), *pressure sensor*, dan *flow sensor*. Referensi utama yang digunakan dalam studi literatur ini antara lain buku karya Nise (2019), Ogata (2010), Petruzella (2023), Fraden (2016), serta berbagai artikel dan jurnal yang relevan. Studi literatur ini membantu memahami teknologi dan teori yang diperlukan untuk merancang *Control valve* yang lebih terjangkau dan efektif untuk aplikasi laboratorium dan industri.

#### 3.3.2 Identifikasi Masalah

*Control valve* memainkan peran penting dalam mengatur aliran fluida dalam berbagai aplikasi industri dan komersial. Proses kerja *control valve* melibatkan pengaturan posisi *valve* untuk mengontrol laju aliran fluida melalui sistem. Mekanisme ini biasanya melibatkan

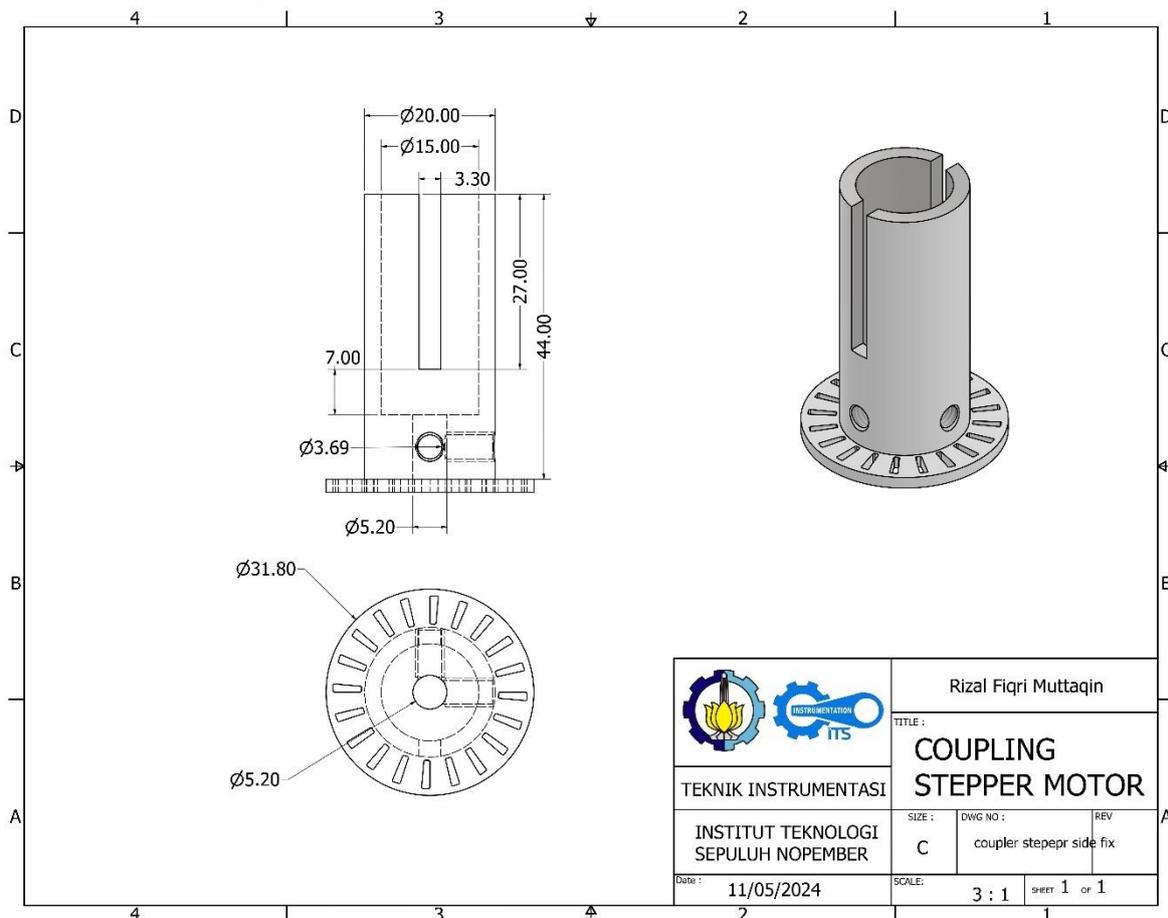
aktuator seperti *stepper motor* yang digerakkan oleh sinyal kontrol untuk membuka atau menutup *valve* secara presisi. Komponen sensor juga digunakan untuk memastikan posisi dan aliran yang tepat.

Masalah utama yang dihadapi adalah tingginya biaya *control valve* yang tersedia di pasaran serta membutuhkan infrastruktur tambahan yang mahal, seperti kompresor untuk *valve* pneumatik, yang semakin meningkatkan biaya keseluruhan sistem sehingga membuatnya sulit diakses oleh banyak institusi pendidikan dan industri kecil yang memiliki anggaran terbatas. Masalah lainnya termasuk kebutuhan untuk sistem yang mudah diintegrasikan dan dioperasikan dalam lingkungan yang mungkin memiliki keterbatasan ruang dan daya. Oleh karena itu, dilakukan pengembangan *control valve* yang lebih ekonomis namun tetap menawarkan performa yang baik dan mudah diintegrasikan ke dalam sistem yang ada. Hal ini akan memungkinkan penggunaan yang lebih luas dan efisien dalam berbagai aplikasi, baik di laboratorium pendidikan maupun dalam industri.

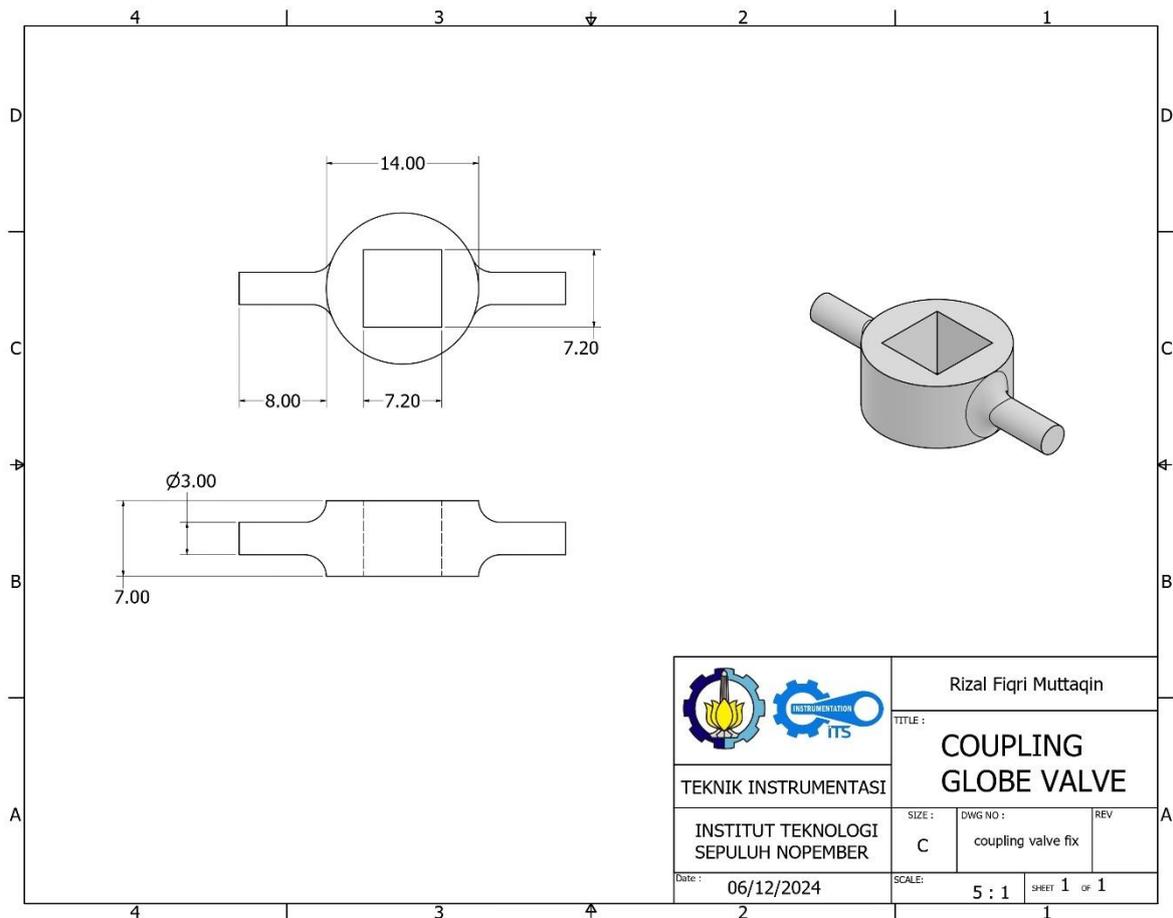
### 3.3.3 Perancangan Desain *Control valve*

Perancangan *control valve* bertujuan untuk menciptakan sebuah sistem yang efisien, handal, dan ekonomis untuk mengontrol aliran fluida dalam berbagai aplikasi industri dan pendidikan. Pada tahap perancangan ini, fokus utama adalah pada pengembangan komponen utama, termasuk desain coupler dan desain *control valve* itu sendiri. Perancangan ini harus memperhatikan aspek fungsionalitas, kemudahan integrasi, dan biaya produksi yang rendah.

#### 3.3.3.1 Desain Coupling



Gambar 3.2 Desain *Coupling Stepper Motor Side*



Gambar 3.3 Desain *Coupling Globe Valve Side*

*Coupling* merupakan komponen penting yang berfungsi menghubungkan *stepper motor* dengan batang *control valve* dalam sistem aktuasi. Desain *coupling* yang dirancang mempertimbangkan kekuatan mekanis, akurasi, dan kemudahan pemasangan. Dari segi kekuatan, *coupling* harus mampu mentransmisikan torsi dari *stepper motor* ke batang *valve* secara andal tanpa mengalami deformasi atau kerusakan akibat beban kerja. Selain itu, *coupling* juga harus menjaga akurasi posisi batang *valve* agar sesuai dengan pergerakan *stepper motor*, guna memastikan sistem kontrol dapat mengatur aliran fluida secara presisi. Kemudahan pemasangan turut menjadi pertimbangan dalam desain, agar komponen ini dapat dipasang dan dilepas dengan mudah, memungkinkan proses perawatan dan penggantian dilakukan secara efisien.

Desain *coupling* ditampilkan pada Gambar 3.2 untuk sisi *stepper motor* dan Gambar 3.3 untuk sisi batang *globe valve*. Dalam rencana pembuatannya, bahan yang digunakan adalah PLA (*Polylactic Acid*), dipilih karena memiliki kekuatan mekanik yang cukup untuk aplikasi ini serta lebih ekonomis dari sisi biaya produksi. Cara kerja dari custom *coupling* ini adalah bagian yang terhubung langsung dengan *stepper motor* bersifat tetap (*fixed*), sementara bagian yang terkoneksi dengan batang *valve* dirancang agar dapat bergerak naik-turun, mengikuti arah gerak linier *valve*. Mekanisme kerja *coupling* ini secara umum digambarkan dalam Gambar 3.4.



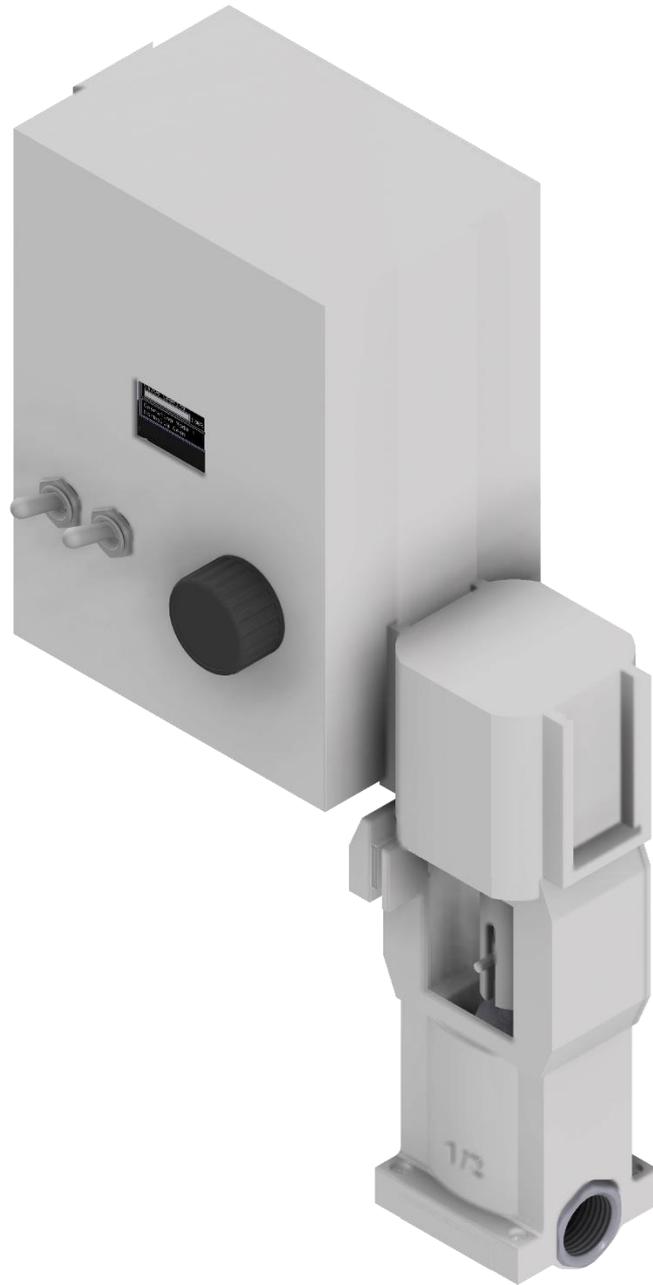
Gambar 3.4 Mekanisme Pergerakan *Globe Valve*

### 3.3.3.2 Desain *Control Valve*

Desain *control valve* mencakup pemilihan jenis *valve*, aktuator, serta integrasi berbagai komponen pendukung untuk memastikan fungsi pengaturan aliran fluida dapat berjalan secara optimal. Dalam sistem ini, *globe valve* dipilih sebagai jenis *valve* utama karena karakteristiknya yang mampu memberikan pengaturan aliran yang akurat dan handal, terutama dalam aplikasi yang memerlukan kendali presisi. *Valve* yang digunakan merupakan produk *globe valve* yang telah dirancang dan dipasarkan oleh merek dagang tertentu.

Sebagai aktuator, *stepper motor* digunakan untuk menggerakkan batang *valve*. Pemilihan *stepper motor* didasarkan pada kemampuannya menghasilkan gerakan presisi dalam kontrol posisi serta respons yang cepat terhadap perubahan *input*. Untuk kebutuhan *input* manual dan navigasi sistem, digunakan *rotary encoder incremental KY-040*, sementara *optical rotary encoder* difungsikan sebagai bagian dari sistem *safety* guna memverifikasi posisi aktuator dan mendeteksi kesalahan posisi.

Desain *control valve* yang direncanakan ditunjukkan pada Gambar 3.5 dan Gambar 3.6. Desain ini mengintegrasikan seluruh komponen utama dalam satu kesatuan yang kompak, dengan mempertimbangkan aspek kemudahan instalasi, aksesibilitas dalam perawatan, serta keandalan operasional dalam berbagai kondisi aplikasi.



Gambar 3.5 Desain *Control Valve*

Desain *control valve* ini mengintegrasikan aktuator berbasis *stepper motor* dengan sistem pengendali elektronik yang dikemas dalam sebuah panel kontrol. Tujuan utama dari desain ini adalah menciptakan solusi pengendalian aliran fluida yang fungsional namun tetap ekonomis, sehingga dapat menjadi alternatif bagi *control valve* komersial yang harganya relatif tinggi. Desain ini dirancang agar mudah direplikasi, terutama untuk kebutuhan edukasi dan penelitian, serta memungkinkan pengembangan fitur tambahan pada tahap pengujian berikutnya.

### 3.3.4 Pemilihan Komponen

Dalam tugas akhir ini, pemilihan komponen secara tepat dilakukan untuk memastikan sistem *control valve* berfungsi optimal. Komponen utama yang dipilih meliputi mikrokontroler ESP32 WROOM 38-pin, *globe valve*, *stepper motor* NEMA 17, *driver stepper motor* DRV8255 dan TMC2209, *rotary encoder* KY-040, *optical rotary encoder*, *toggle switch*, display LCD OLED, sensor temperatur DS18B20 dan sensor arus INA219, dan catu daya yang diperlukan.

#### 3.3.4.1 Bill of Material (BOM)

*Bill of Materials* (BOM) merupakan daftar lengkap material atau komponen yang digunakan dalam proyek ini. Daftar ini mencakup setiap komponen elektronik, mekanik, dan pendukung lainnya yang dibutuhkan untuk merancang sistem *control valve* berbasis *stepper motor* dan *globe valve*. Informasi pada tabel BOM disusun untuk mempermudah proses identifikasi dan perencanaan material selama penelitian berlangsung. Setiap komponen yang tercantum dilengkapi dengan detail seperti nama komponen, tipe, spesifikasi teknis, serta fungsi utamanya dalam sistem.

**Tabel 3.3** *Bill of Material* (BOM)

No.	Komponen	Spesifikasi	Fungsi dalam Sistem
1.	ESP32 WROOM 38-pin	<ul style="list-style-type: none"> <li>• <i>Processor</i>: Dual-core Tensilica LX6 (up to 240MHz)</li> <li>• <i>Memory</i> : 520KB SRAM, 448KB ROM, 4MB Flash (<i>program storage</i>)</li> <li>• Wi-Fi: 802.11 b/g/n (2.4GHz)</li> <li>• <i>Operating Voltage</i> : 2.2V to 3.6V</li> <li>• Pin GPIO : 38</li> <li>• <i>Power supply</i> : 5V via USB or external power source</li> </ul>	Kontrol utama sistem dan komunikasi dengan komponen lain
2.	<i>Globe valve</i>	<ul style="list-style-type: none"> <li>• <i>Material</i> : SS 316</li> <li>• Ukuran : 1/2"</li> <li>• <i>Disc type</i> : Plug</li> </ul>	Mengatur aliran fluida sesuai <i>setpoint</i>
3.	<i>Stepper motor</i> NEMA17	<ul style="list-style-type: none"> <li>• PxLxT : 42mm x 42mm x 40mm</li> <li>• Panjang As : 20mm</li> <li>• <i>Voltage</i>: 12V</li> <li>• <i>Phase current</i>: 1.2 A</li> <li>• Torsi : 4000 g.cm pada 1.2 A</li> <li>• <i>Step angle</i> : 1.8°/step</li> <li>• <i>Step/revolution</i> : 200</li> </ul>	Menggerakkan bukaan dan penutupan <i>globe valve</i>
4.	<i>Driver</i> DRV8255	<ul style="list-style-type: none"> <li>• <i>Voltage Range</i> : 8.2 - 45V DC</li> <li>• Arus maksimal : 2.5A pada 24V</li> <li>• <i>Stepping mode</i> : 1 - 32</li> </ul>	Mengontrol pergerakan <i>stepper motor</i>

No.	Komponen	Spesifikasi	Fungsi dalam Sistem
5.	<i>Driver TMC2209</i>	<ul style="list-style-type: none"> <li>• <i>Voltage Range : 4.75 - 29V DC</i></li> <li>• <i>Arus maksimal : 2.8A</i></li> <li>• <i>Stepping mode : 1 - 256</i></li> </ul>	<i>Mengontrol pergerakan stepper motor</i>
6.	<i>Rotary encoder KY-040</i>	<ul style="list-style-type: none"> <li>• <i>Voltage : 5V (arduino) dan 3,3V</i></li> <li>• <i>Encode 2 arah maju dan mundur</i></li> <li>• <i>Push button</i></li> <li>• <i>Step/revolution : 20</i></li> </ul>	<i>Input persentase bukaan valve</i>
7.	<i>Optical Rotary Encoder</i>	<ul style="list-style-type: none"> <li>• <i>Operating Voltage : 3.3V to 5 V DC</i></li> <li>• <i>Slot width : 5mm</i></li> <li>• <i>Output Signal : Analog and Digital output</i></li> <li>• <i>Digital output will be HIGH when something is within the Slot, blocking the sensor. Of course LOW when there is nothing in the slot.</i></li> </ul>	<i>Deteksi batas bukaan dan penutupan valve</i>
8.	<i>Toggle Switch</i>	<ul style="list-style-type: none"> <li>• <i>Rating Voltage &amp; Current : 250 V 3 A // 125V 6 A</i></li> </ul>	<i>Penentu mode awal Normally Open/Normally Close</i>
9.	<i>Display LCD OLED</i>	<ul style="list-style-type: none"> <li>• <i>OLED 0.96" Display</i></li> <li>• <i>Resolusi layar : 128 x 32 pixel</i></li> <li>• <i>Komunikasi : i2C /IIC</i></li> <li>• <i>VCC : 3.3 - 5V</i></li> </ul>	<i>Menampilkan data dan status sistem</i>
10.	<i>Sensor Temperatur DS18B20</i>	<ul style="list-style-type: none"> <li>• <i>0.5°C Accuracy from -10°C to +85°C</i></li> <li>• <i>Power supply Range is 3.0V to 5.5V</i></li> <li>• <i>Converts Temperature to 12-Bit Digital Word in 750ms (Max)</i></li> </ul>	<i>Monitoring temperatur pada driver</i>
11.	<i>Sensor Arus INA219</i>	<ul style="list-style-type: none"> <li>• <i>Up to +26V target voltage</i></li> <li>• <i>Up to 3.2A current measurement, with 0.8mA resolution</i></li> <li>• <i>VCC : DC 3 - 5V</i></li> </ul>	<i>Monitoring arus, tegangan, dan daya</i>
12.	<i>Printed Circuit Board (PCB)</i>	<ul style="list-style-type: none"> <li>• <i>P x L = 7 cm x 9 cm</i></li> </ul>	<i>Integrasi seluruh komponen</i>
13.	<i>Power Supply Unit (PSU)</i>	<ul style="list-style-type: none"> <li>• <i>Tegangan : 12V</i></li> <li>• <i>Arus : 3A</i></li> </ul>	<i>Catu daya utama sistem</i>
14.	<i>StepDown converter</i>	<ul style="list-style-type: none"> <li>• <i>Adjustable Version Output Voltage Range: 1.25V to 37V</i></li> <li>• <i>Maximum output current : 3A, normal and stable working current is 2A</i></li> <li>• <i>Input Voltage Range Up to 40 V</i></li> </ul>	<i>Mengurangi tegangan untuk ESP32</i>

### 3.3.4.2 Bill of Quantity (BOQ)

*Bill of Quantity* (BOQ) adalah daftar yang memuat jumlah dan estimasi biaya dari setiap komponen yang digunakan dalam proyek ini. BOQ disusun berdasarkan informasi pada tabel BOM dengan tambahan kolom yang mencakup harga satuan, dan total harga komponen. Tabel ini bertujuan untuk memberikan gambaran anggaran keseluruhan yang diperlukan dalam penelitian, sekaligus menjadi acuan untuk memastikan bahwa proyek dapat diselesaikan sesuai dengan alokasi biaya yang tersedia.

**Tabel 3.4** *Bill Of Quantity* (BOQ)

No.	Komponen	Jumlah	Harga Satuan (Rp)	Total Harga (Rp)
1.	ESP32 WROOM 38-pin	2	68000	136000
2.	<i>Globe valve</i>	1	82600	82600
3.	<i>Stepper motor</i> NEMA17	1	110000	110000
4.	<i>Driver</i> DRV8255	1	20.000	20000
5.	<i>Driver</i> TMC2209	1	43700	43700
6.	<i>Rotary encoder</i> KY-040	1	20000	20000
7.	<i>Optical Rotary Encoder</i>	1	8000	8000
8.	<i>Toggle Switch</i>	1	2700	2700
9.	Display LCD OLED	1	45000	45000
10.	Sensor Temperatur DS18B20	1	9000	9000
11.	Sensor Arus INA219	1	22000	22000
12.	PCB dan Konektor	2	23480	46960
13.	PSU 12V 3A	1	29000	29000
14.	<i>StepDown converter</i>	1	15.900	15900
15	3D print filamen (/gram)	370	165	61050
Total				651910

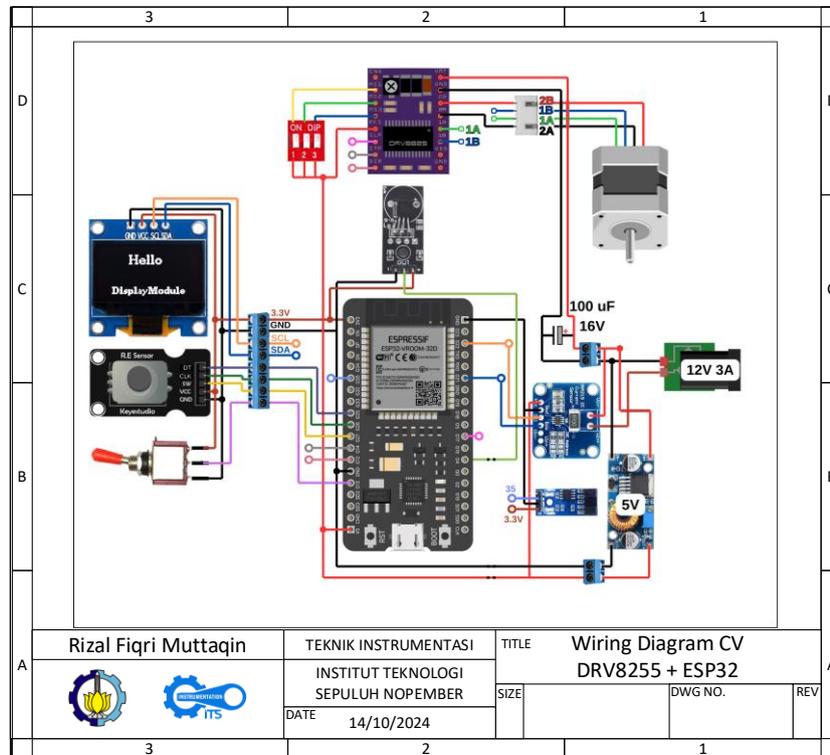
### 3.3.5 Perancangan *Hardware* dan Pemrograman Kontrol

Pada tahap ini, perancangan *hardware* dan pemrograman kontrol menjadi bagian krusial dalam pengembangan sistem *control valve*. *Hardware* yang digunakan harus diintegrasikan secara tepat untuk memastikan sistem bekerja sesuai dengan spesifikasi. Selain itu, pemrograman kontrol diperlukan untuk mengatur dan mengontrol semua komponen yang terhubung dalam sistem.

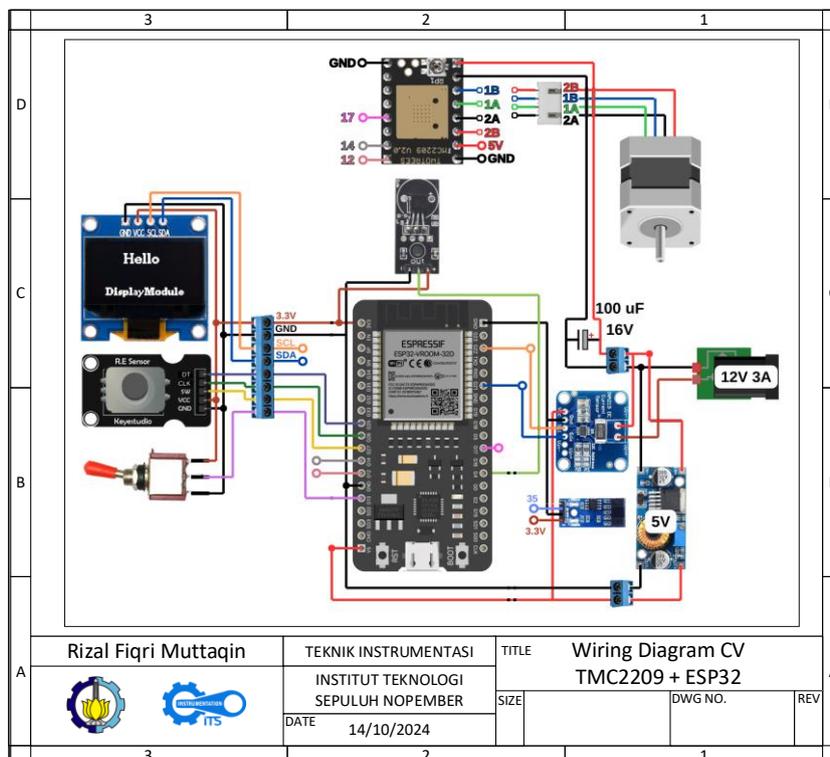
#### 3.3.5.1 Perancangan *Hardware*

Perancangan *hardware* untuk sistem *control valve* mencakup beberapa tahapan, yaitu penyusunan *wiring diagram*, penempatan komponen pada panel kontrol, serta pembuatan diagram blok kontrol. *Power supply Unit* (PSU) 12V 3.3A dipilih sebagai sumber daya utama

untuk sistem ini. PSU ini memberikan daya yang dibutuhkan oleh *driver stepper motor* DRV8255 dan TMC2209. Untuk menstabilkan tegangan sebelum masuk ke *driver*, digunakan kapasitor elektrolit 100uF 16V. PSU 12V 3.3A juga terhubung ke *step-down converter* 5V yang menyediakan tegangan yang diperlukan untuk ESP32. *Wiring diagram* dapat dilihat pada Gambar 3.7 dan 3.8, kemudian terdapat diagram skematik kontrol pada Gambar 3.9.



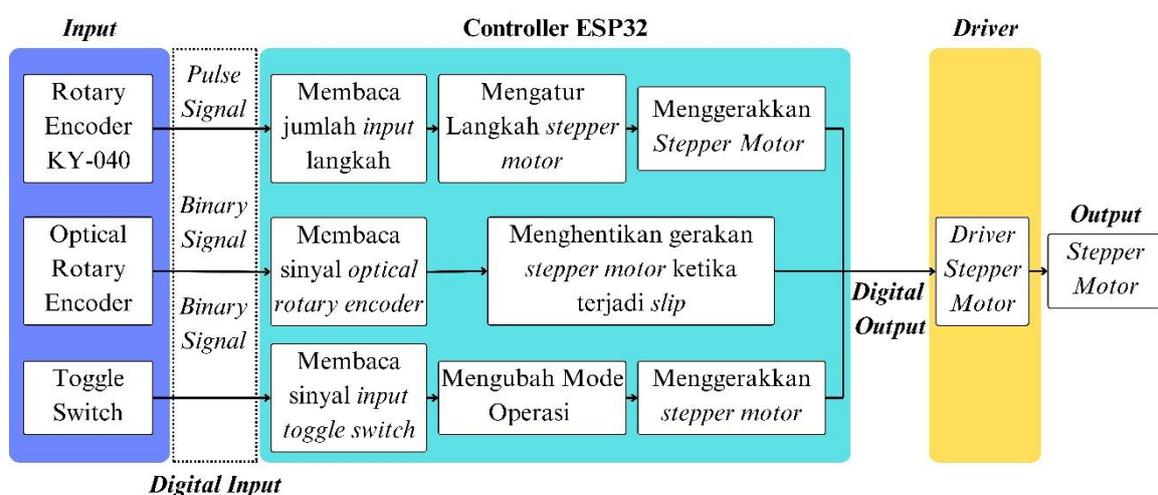
Gambar 3.6 *Wiring Diagram* PCB DRV8255



Gambar 3.7 *Wiring Diagram* PCB TMC2209

Pada panel kontrol, beberapa komponen utama ditempatkan untuk memudahkan pengoperasian sistem seperti LCD, ESP32, *rotary encoder*, *driver stepper motor*, *toggle switch*, PSU, dan *step down*. LCD menampilkan informasi operasional seperti persentase bukaan *valve*, mode operasi, dan *monitoring* dari sensor. *Rotary encoder* berfungsi sebagai *input* manual untuk mengatur posisi *valve* dan *driver stepper motor* yang akan menggerakkan *globe valve* berdasarkan sinyal kontrol dari ESP32. *Toggle switch* berguna untuk mengubah mode operasi *valve* antara *Normally Open* (NO) dan *Normally Closed* (NC).

Fitur tambahan yaitu sistem *safety* dari *slip* atau deteksi buka dan menutup penuh, *optical rotary encoder* dipasang pada *body control valve* dan terhubung ke ESP32 di panel kontrol untuk mendeteksi batas atas dan bawah dari pergerakan *valve*, memastikan keamanan dan akurasi posisi dalam sistem kontrol. Semua komponen yang terletak di bagian *housing control valve* meliputi *stepper motor*, *globe valve*, dan *optical rotary encoder*.



Gambar 3.8 Diagram Skematik Kontrol

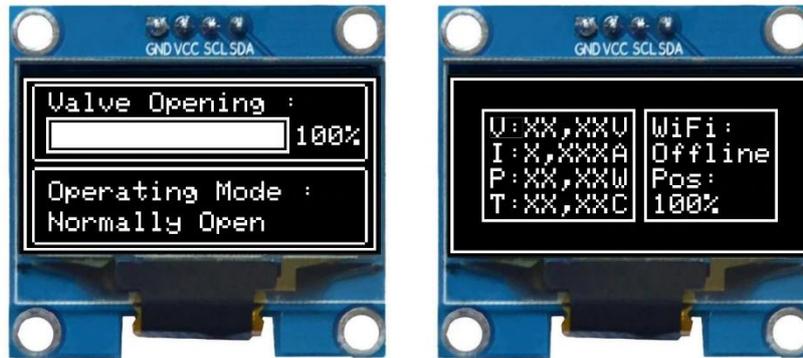
Pada konfigurasi ini, *rotary encoder* bertanggung jawab mengubah *input* rotasi manual pengguna menjadi sinyal digital yang kemudian diteruskan ke kontroler ESP32. Kontroler tersebut menginterpretasikan sinyal ini untuk menghasilkan sinyal kontrol yang diperlukan bagi *driver stepper motor* DRV8825. *Driver* tersebut kemudian memberikan daya ke *stepper motor* NEMA17, mengubah sinyal listrik menjadi gerakan mekanis, yang mengatur posisi *globe valve* untuk mengontrol aliran fluida.

*Toggle switch* memungkinkan pengguna untuk mengatur kondisi awal *globe valve*, memilih antara *Normally Open* (NO) atau *Normally Closed* (NC). Ketika pengguna menggeser saklar, saklar tersebut menghasilkan sinyal biner yang dikirimkan ke kontroler ESP32. Kontroler ini menginterpretasikan sinyal ini untuk menghasilkan sinyal kontrol awal yang dikirim ke *driver* DRV8825. Akibatnya, *driver* menggerakkan *stepper motor* NEMA17 untuk memosisikan *valve* ke posisi awal yang diinginkan. Proses ini memastikan bahwa *globe valve* dimulai pada posisi yang dipilih oleh pengguna, baik sepenuhnya terbuka atau tertutup.

Diagram blok kontrol tersebut menggambarkan hubungan antara setiap komponen dan bagaimana data dan sinyal dikomunikasikan di dalam sistem. Diagram blok kontrol ini akan mempermudah dalam memahami alur kerja sistem secara keseluruhan. Dengan perancangan *hardware* ini, sistem *control valve* diharapkan dapat berfungsi dengan baik dan siap untuk dilakukan pengujian lebih lanjut. Integrasi antara komponen dilakukan dengan memperhatikan konektivitas dan stabilitas tegangan untuk memastikan setiap komponen bekerja optimal.

### 3.3.5.2 Perancangan Pemrograman Kontrol

Perancangan pemrograman kontrol pada tugas akhir ini dilakukan menggunakan ESP32 yang akan mengendalikan pergerakan *stepper motor* berdasarkan *input* dari *rotary encoder* KY-040, dengan tambahan fitur untuk menampilkan informasi pada LCD OLED dan menggunakan *optical rotary encoder* untuk sistem keamanan serta *toggle switch* yang akan menentukan mode operasi *control valve*. Berikut rancangan *Graphical User Interface* (GUI) pada LCD OLED 128x64 yang akan digunakan :



Gambar 3.9 Rancangan GUI

Berikut adalah logika operasi dan langkah-langkah dalam perancangan pemrograman kontrol:

#### 1. *Initial Setup*:

- Pembacaan *Toggle switch*: *Toggle switch* digunakan untuk menentukan mode operasi *valve* (*Normally Open* atau *Normally Closed*). Jika *switch* dalam keadaan low maka mode operasi NC, jika high maka mode operasi NO.
- Inisialisasi Posisi Awal: Untuk mode NC, *stepper motor* akan memutar *valve* ke posisi tertutup penuh hingga *optical rotary encoder* mendeteksi *slip* yang menandakan *valve* sudah menutup sempurna, kemudian mendefinisikan posisi awal (0%). Untuk mode NO, *stepper motor* akan memutar *valve* ke posisi terbuka penuh hingga *optical rotary encoder* mendeteksi sudah membuka penuh dan mendefinisikan posisi awal (0%).

#### 2. Pembacaan *rotary encoder*:

- Rotary encoder* digunakan untuk mengatur persentase bukaan *valve*. Setiap putaran searah jarum jam (positif) menambah 1%, sedangkan putaran berlawanan arah jarum jam mengurangi 1%.
- Pembacaan nilai *rotary encoder* akan diubah menjadi pergerakan *stepper motor*.

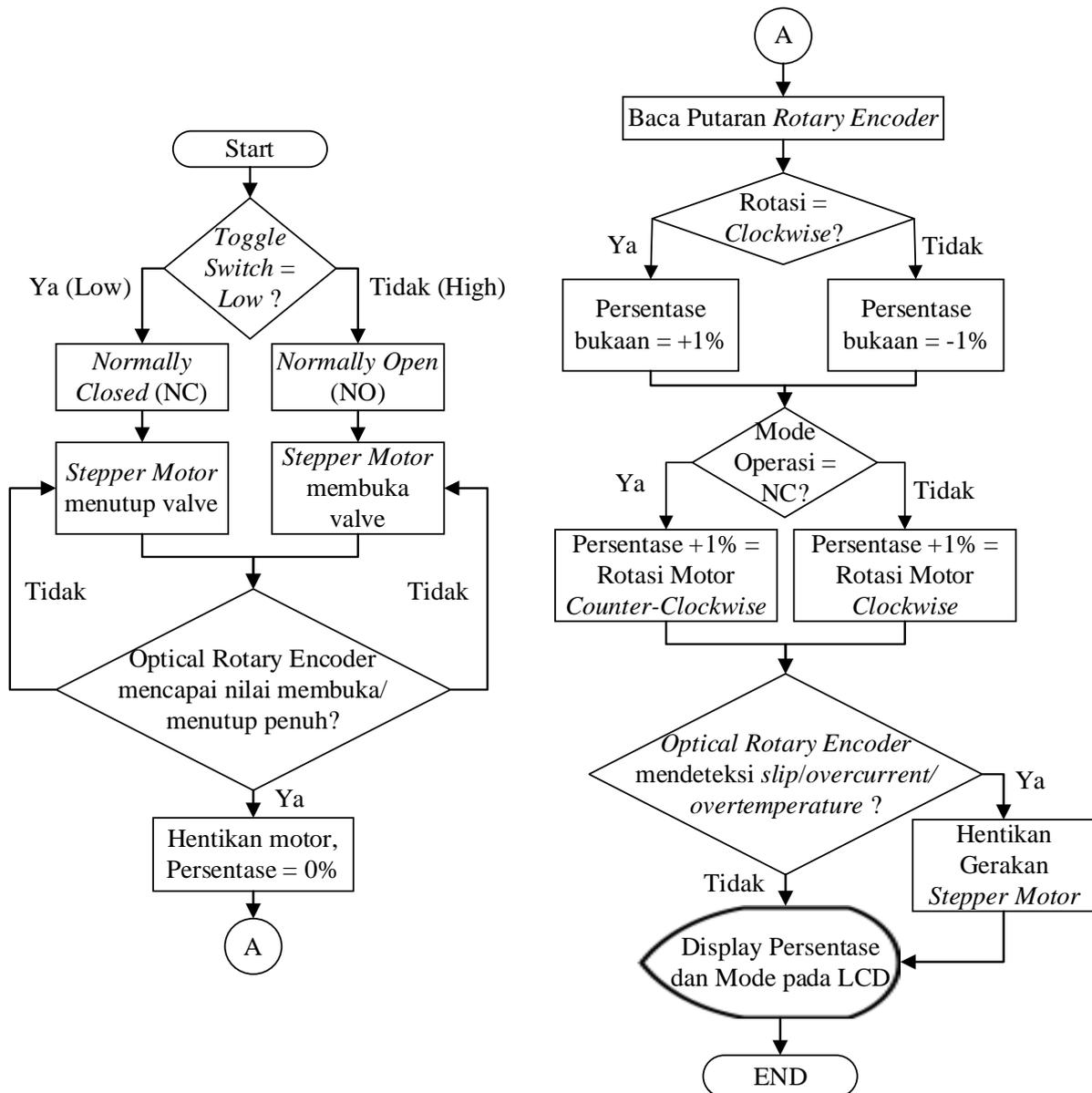
#### 3. Penggunaan *optical rotary encoder*:

- optical rotary encoder* digunakan sebagai *safety* untuk mencegah pergerakan *stepper motor* melebihi batas buka atau tutup penuh. Saat *optical rotary encoder* mendeteksi nilai anomali yang menandakan adanya *slip* pada motor, pergerakan *stepper motor* akan dihentikan.

#### 4. Pengaturan Persentase Bukaan:

- Persentase bukaan *valve* ditampilkan pada LCD. Mode operasi (NC atau NO) serta nilai pembacaan sensor juga ditampilkan untuk memudahkan monitoring.

Rincian alur programnya dijelaskan melalui *flowchart* pada Gambar 3.11.



Gambar 3.10 Flowchart Operasi Control Valve

### 3.3.6 Validasi Sensor

Sebelum sistem dijalankan secara menyeluruh, dilakukan proses validasi terhadap dua sensor utama yang digunakan, yaitu sensor temperatur DS18B20 dan sensor arus INA219. Tujuan dari validasi ini adalah untuk memastikan bahwa pembacaan sensor-sensor tersebut akurat dan dapat diandalkan dalam memantau kondisi sistem secara *real-time*. Proses validasi dilakukan dengan membandingkan pembacaan sensor terhadap alat ukur standar yang telah dikalibrasi.

Validasi sensor temperatur DS18B20 dilakukan dengan membandingkan hasil pembacaan sensor dengan *thermometer* FY-010. Pengambilan data dilakukan ketika sistem dalam kondisi aktif, terutama saat *driver stepper motor* sedang bekerja, karena pada kondisi tersebut biasanya terjadi kenaikan temperatur akibat beban kerja. Pembacaan temperatur dari DS18B20 kemudian dibandingkan dengan nilai dari *thermometer* FY-010 untuk mengetahui selisih atau *error* pengukuran.

Sementara itu, validasi sensor INA219 dilakukan untuk memastikan keakuratan pembacaan arus dan tegangan selama pengoperasian motor. Proses validasi dilakukan dengan cara mengaktifkan *stepper motor* tanpa beban tambahan, lalu mencatat pembacaan arus dan tegangan dari INA219 secara periodik setiap detik. Hasil pencatatan tersebut kemudian dibandingkan dengan nilai pembacaan dari multimeter digital sebagai alat ukur pembanding.

### 3.3.7 Pengujian Komponen

Setelah sensor divalidasi, tahap selanjutnya adalah pengujian terhadap komponen-komponen utama yang digunakan dalam sistem. Tujuan dari pengujian ini adalah untuk memastikan bahwa setiap komponen bekerja sesuai spesifikasi dan dapat terintegrasi dengan baik dalam sistem kontrol. Komponen yang diuji meliputi *globe valve*, *stepper motor* NEMA17, dua jenis *driver* yaitu DRV8825 dan TMC2209, *rotary encoder* KY-040, serta *optical rotary encoder* yang digunakan untuk verifikasi posisi.

Pengujian manual *globe valve* dilakukan untuk mengetahui jumlah rotasi yang dibutuhkan untuk membuka dan menutup *valve* secara penuh. Selain itu, dilakukan pengukuran torsi minimum yang diperlukan untuk memutar *valve* menggunakan pemberat bandul yang dikaitkan pada ujung tuas. Pengujian *stepper motor* dilakukan dengan menghubungkan motor pada masing-masing *driver* dan mengoperasikan dalam beberapa mode, seperti *full-step*, *half-step*, dan *microstep*. Parameter yang diamati meliputi stabilitas gerakan, konsumsi arus, dan temperatur *driver*. Selain itu, dilakukan pengamatan terhadap fenomena *bounce-back* atau gerakan balik setelah berhenti. *Optical rotary encoder* diuji untuk memastikan keakuratan pembacaan jumlah langkah motor. Pengujian ini dilakukan dengan membandingkan hasil pembacaan *encoder* terhadap jumlah pulsa yang seharusnya dihasilkan dalam satu putaran penuh. *Encoder* juga diuji dalam skenario tertentu untuk mendeteksi kemungkinan terjadinya *slip* saat pengoperasian motor. *Rotary encoder* KY-040 diuji untuk mengetahui konsistensi *input* posisi manual terhadap *output* gerakan motor. Setiap langkah rotasi *encoder* harus memberikan *input* yang konsisten terhadap sistem.

### 3.3.8 Pembuatan dan Integrasi Sistem

Setelah semua komponen diuji secara individu, langkah selanjutnya adalah pembuatan dan integrasi sistem. Tahapan ini melibatkan proses perakitan seluruh perangkat keras, mulai dari *globe valve*, *stepper motor* NEMA17, *driver* DRV8825 dan TMC2209, *rotary encoder* KY-040, *optical rotary encoder*, saklar *toggle*, ESP32, hingga komponen pendukung seperti *power supply* dan *step-down converter*.

Perakitan dilakukan dengan mengacu pada diagram rangkaian dan desain mekanik yang telah disiapkan sebelumnya. Semua sambungan kabel dan konektor diverifikasi untuk menghindari kesalahan sambungan yang dapat mempengaruhi kinerja sistem. Setelah perakitan selesai, perangkat keras ditempatkan dalam panel kontrol untuk memastikan kepraktisan dan keamanan saat pengoperasian. Proses integrasi dilanjutkan dengan pemrograman mikrokontroler ESP32. Program yang ditanamkan mencakup logika pembacaan *input* dari *rotary encoder* dan saklar *toggle*, pengendalian posisi *valve* berdasarkan perintah dari *encoder*, serta pembacaan sensor arus dan temperatur. Data dari sensor kemudian ditampilkan pada layar OLED sebagai antarmuka pengguna. Selain itu, sistem juga dilengkapi dengan fitur proteksi apabila temperatur *driver* melebihi batas aman atau terjadi lonjakan arus.

### 3.3.9 Pengujian Sistem Control Valve

Setelah sistem terintegrasi, dilakukan pengujian terhadap sistem *control valve* secara keseluruhan. Pengujian ini bertujuan untuk mengevaluasi performa aktuasi *globe valve* oleh *stepper motor* dalam berbagai konfigurasi *driver* dan mode penggerakan. Sistem diuji dalam

mode manual dan otomatis untuk menilai keakuratan posisi *valve*, respon terhadap *input* dari *rotary encoder*, dan validitas deteksi posisi dari *optical encoder*.

Dalam pengujian ini, masing-masing kombinasi *driver* dan mode penggerakan diuji secara berulang untuk mengetahui keandalan sistem dalam mencapai posisi yang diinginkan. Selain itu, *optical rotary encoder* digunakan sebagai referensi utama untuk mengidentifikasi adanya selisih posisi atau *slip* antara perintah aktuasi dan gerakan aktual.

### 3.3.10 Pengambilan Data

Pengambilan data dilakukan untuk mendokumentasikan performa sistem selama pengujian berlangsung. Data yang diambil meliputi arus dan temperatur *driver*, posisi *valve* berdasarkan *optical encoder*, serta data aliran dan tekanan sebelum dan sesudah *valve* menggunakan *flow sensor* dan *pressure sensor*. Seluruh data ini direkam secara otomatis dan disimpan dalam database untuk kemudian dianalisis. Pengambilan data dilakukan dengan interval setiap detik, terutama untuk parameter kelistrikan seperti arus dan temperatur, guna mendeteksi perubahan kecil selama pengoperasian.

### 3.3.11 Analisis Performa

Analisis performa dilakukan untuk mengevaluasi efektivitas dan keandalan sistem *control valve* berdasarkan data hasil pengujian. Aspek utama evaluasi ini yaitu akurasi aktuasi posisi *valve*, stabilitas gerakan motor, respons sistem terhadap *input*, serta kemampuan kontrol aliran dan tekanan. Akurasi aktuasi dinilai dari perbandingan antara posisi target *input* dengan data yang diperoleh dari pembacaan *optical rotary encoder*. Selanjutnya dilakukan analisis terhadap nilai *deadband*, yaitu rentang perubahan *input* posisi terkecil yang mampu menghasilkan perubahan signifikan pada output *flowrate*.

Performa sistem dalam mengendalikan fluida diuji melalui karakteristik hubungan antara persentase bukaan *valve* dengan *flowrate* yang dihasilkan. Hasil pengujian menunjukkan pola karakteristik aliran, seperti *quick opening*, *linear*, atau *equal percentage*, yang menentukan efektivitas sistem dalam berbagai kondisi operasi. Di samping itu, analisis *pressure drop* dilakukan untuk menilai kemampuan *valve* dalam mempertahankan tekanan sistem di berbagai tingkat bukaan. Data ini memberikan gambaran mengenai performa *valve* yang dihasilkan dari desain sistem.

Dari sisi kelistrikan, arus pada *driver stepper motor*, khususnya DRV8825, dianalisis menggunakan metode *Fast Fourier Transform* (FFT) untuk mengidentifikasi adanya *noise* atau fluktuasi osilasi pada domain frekuensi rendah. Hasil ini digunakan sebagai dasar untuk proses penyaringan sinyal arus dengan *low-pass filter*, sehingga grafik perbandingan arus antara DRV8825 dan TMC2209 menjadi lebih informatif. Selain itu, temperatur *driver* juga diamati untuk mengevaluasi efisiensi termal sistem dan potensi risiko *overheating* dalam penggunaan jangka panjang.

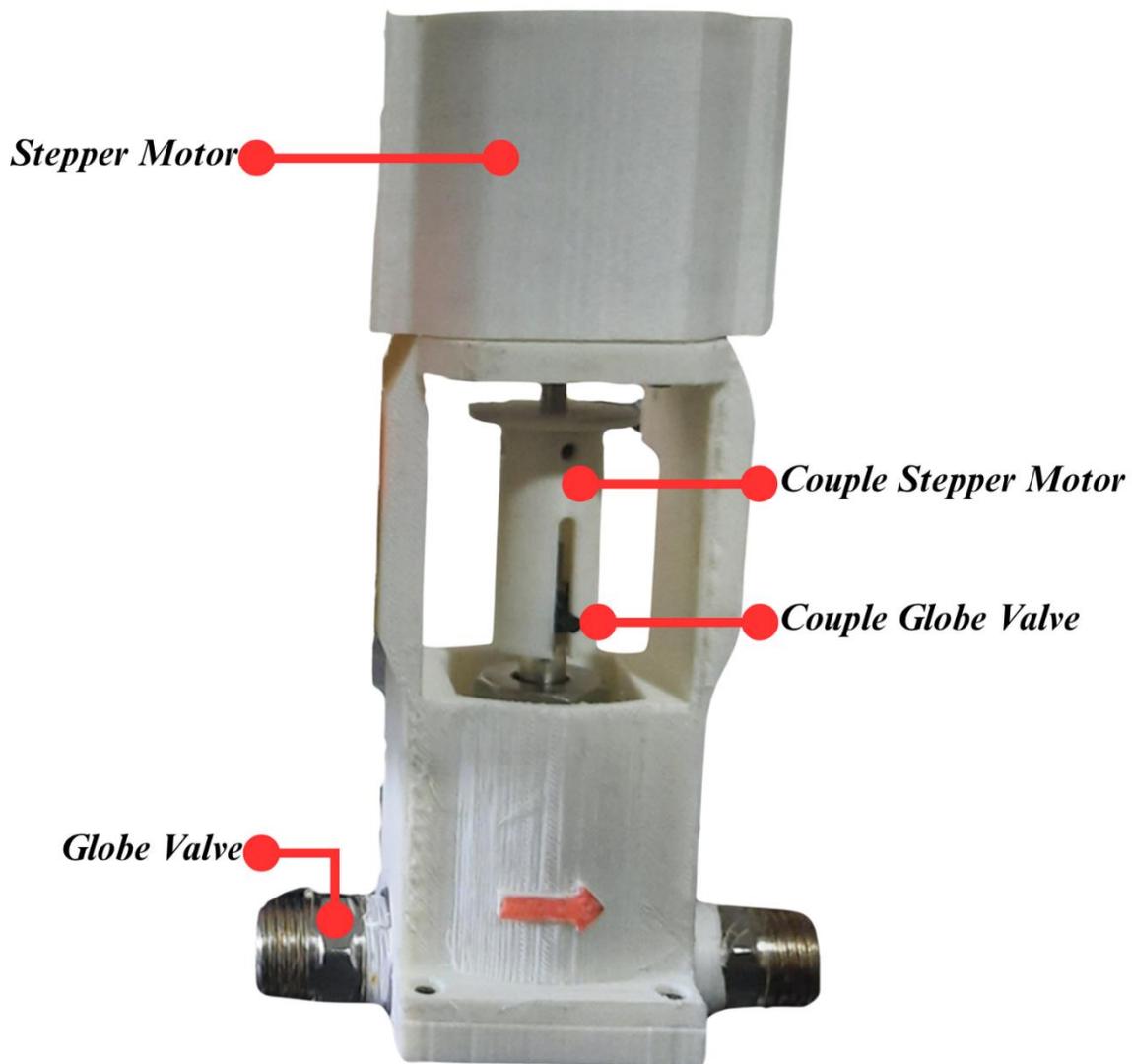
### 3.3.12 Penyusunan laporan

Tahap akhir dari proses ini adalah penyusunan laporan tugas akhir yang mencakup seluruh tahapan mulai dari perancangan sistem, validasi sensor, pengujian komponen, pembuatan dan integrasi sistem, hingga analisis performa. Laporan ini disusun secara sistematis dan komprehensif untuk menjelaskan seluruh proses dan hasil yang telah diperoleh. Informasi yang dicantumkan dalam laporan tidak hanya berguna untuk dokumentasi, tetapi juga sebagai referensi untuk penelitian lanjutan atau pengembangan sistem serupa di masa depan.

## BAB 4 HASIL DAN PEMBAHASAN

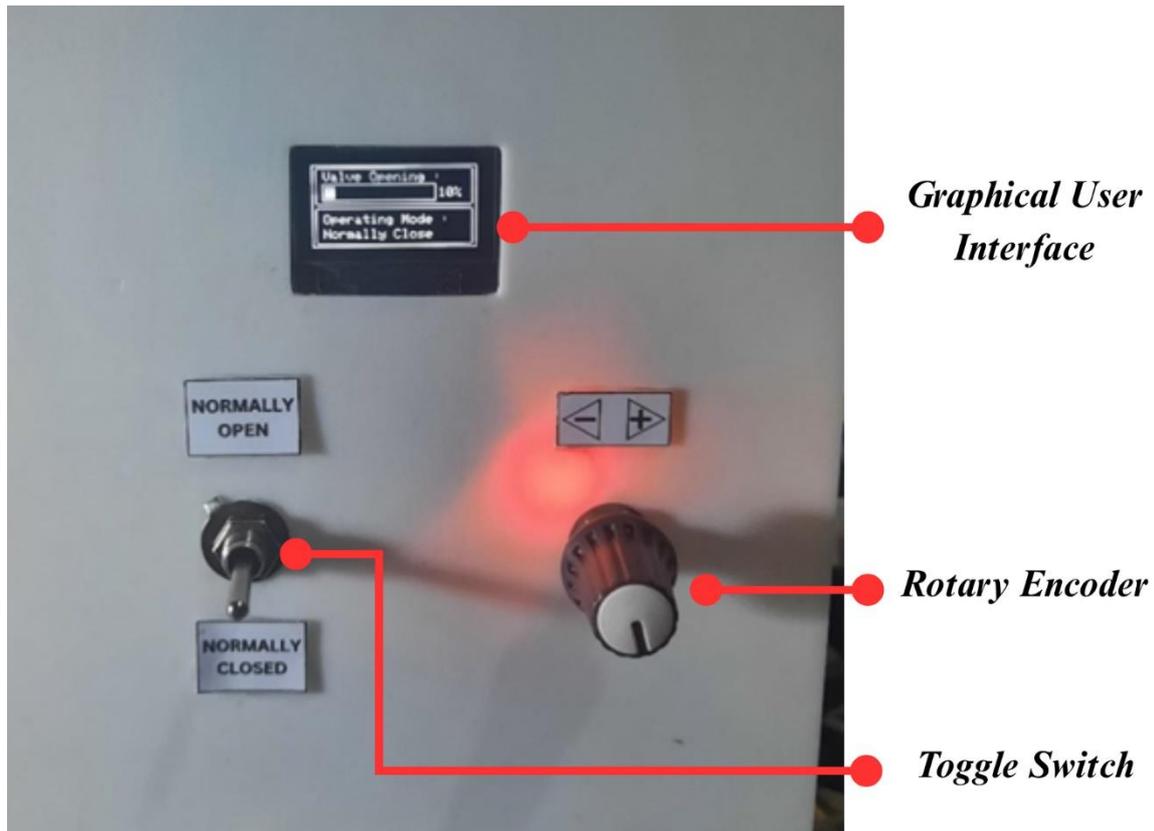
### 4.1 Hasil Perancangan *Control Valve*

*Control valve* yang dirancang pada proyek ini telah selesai diintegrasikan dengan seluruh komponen utama, termasuk *globe valve*, *stepper motor*, *rotary encoder KY-040*, *toggle switch*, dan PCB. Komponen-komponen utama seperti *globe valve*, *stepper motor*, limit switch dirakit dalam housing yang sudah dirancang, desain ini juga telah melalui serangkaian proses penyesuaian untuk memastikan stabilitas serta respons yang optimal dalam berbagai kondisi operasional. *Control valve* yang terintegrasi dapat dilihat pada Gambar 4.1 dan komponen lain yang sudah terintegrasi pada panel kontrol dapat dilihat pada Gambar 4.2.



Gambar 4.1 *Motorized Control Valve*

Pada Gambar 4.1 terdapat beberapa komponen utama yang terintegrasi seperti *Stepper motor* sebagai penggerak utama, *Globe valve* sebagai *valve* utama untuk mengendalikan laju aliran, kemudian ada bagian *body Control valve* termasuk dengan *cover* dari *stepper motor*, *coupling custom* dibagi menjadi 2 yaitu pada bagian *stepper motor* dan bagian *valve*. Komponen lain pada bagian ini yaitu ada *optical rotary encoder* dengan plat atau piringannya menyatu dengan *couple stepper motor*.



Gambar 4.2 *Panel Box Kontrol*

Gambar 4.2 menunjukkan seluruh komponen yang telah terpasang pada *panel box* kontrol, didalamnya termasuk penempatan PCB dan koneksi dengan catu daya utama yaitu 12V 3A dan juga stepdown 5V dari catu daya utama untuk ESP32. Komponen-komponen ini dirakit dalam sistem yang terhubung untuk mengontrol bukaan *valve* secara akurat berdasarkan masukan dari *rotary encoder* dan sinyal kontrol yang diberikan. Pada LCD, selain status persentase bukaan *valve* dan mode operasi, ditampilkan juga monitoring atau data pembacaan sensor temperatur dari *driver stepper motor* dan kelistrikan seperti arus, tegangan, dan daya pada PCB.

## 4.2 Hasil Validasi Sensor

Sensor yang digunakan pada 2 PCB utama yaitu PCB CV DRV8255 dan PCB CV TMC2209 adalah sensor temperatur DS18B20 dan sensor arus INA219 yang juga bisa membaca tegangan serta daya pada kelistrikan. Sensor INA219 divalidasi dengan menggunakan multimeter dan sensor DS18B20 menggunakan termokopel, hasil validasi setiap sensor dengan masing masing validatornya tertera pada sub-bab validasi masing masing sensor.

### 4.2.1 Validasi INA219

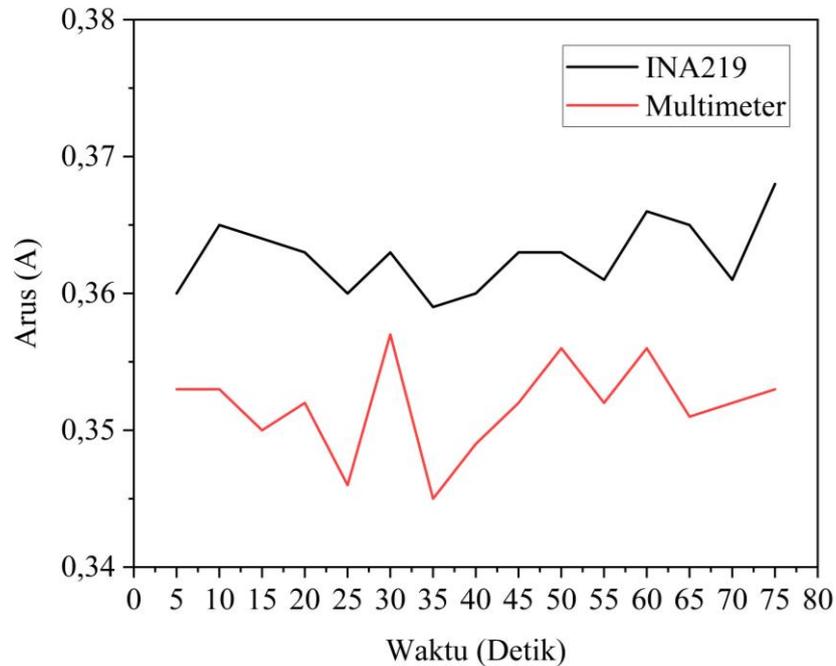
Pengujian validasi sensor arus INA219 dilakukan dengan menggunakan multimeter sebagai alat referensi untuk memastikan akurasi pengukuran sensor. Pengujian ini dilakukan dalam kondisi *stepper motor* diputar secara kontinu. Data pengukuran diambil setiap 5 detik, sehingga poin 1 merepresentasikan data pada 5 detik pertama, poin 2 pada 10 detik, dan seterusnya hingga poin ke-15 (75 detik). Multimeter yang digunakan memiliki spesifikasi akurasi pembacaan arus DC sebesar  $\pm(1,2\% + 3)$ , yang menunjukkan batas ketidakpastian pengukuran.

Validasi ini bertujuan untuk menganalisis perbedaan pembacaan antara sensor INA219 dengan multimeter serta mengevaluasi kinerja sensor dalam mengukur arus listrik.

**Tabel 4.1** Data Validasi Sensor Arus INA219

No.	Arus Sensor	Arus Multimeter	Selisih	<i>Error</i>
1	0,360	0,353	0,007	2,0%
2	0,365	0,353	0,012	3,4%
3	0,364	0,350	0,014	4,0%
4	0,363	0,352	0,011	3,1%
5	0,360	0,346	0,014	4,0%
6	0,363	0,357	0,006	1,7%
7	0,359	0,345	0,014	4,1%
8	0,360	0,349	0,011	3,2%
9	0,363	0,352	0,011	3,1%
10	0,363	0,356	0,007	2,0%
11	0,361	0,352	0,009	2,6%
12	0,366	0,356	0,010	2,8%
13	0,365	0,351	0,014	4,0%
14	0,361	0,352	0,009	2,6%
15	0,368	0,353	0,015	4,2%
Rata-Rata			0,011	3,1%

Berdasarkan tabel hasil pengukuran, sensor INA219 memiliki rata-rata pembacaan arus yang sedikit lebih tinggi dibandingkan multimeter. Selisih rata-rata antara kedua perangkat adalah 0,011 A, dengan nilai *error* rata-rata 3,1%. Nilai *error* tertinggi terjadi pada poin ke-15, yaitu sebesar 4,2%, di mana selisih antara kedua alat mencapai 0,019 A. Sementara itu, pada poin ke-6, terjadi selisih terkecil sebesar 0,006 A dengan *error* hanya 1,7%, menandakan performa sensor cukup baik pada titik tersebut. Secara umum, perbedaan yang terjadi antara sensor dan multimeter konsisten di sebagian besar titik pengukuran, meskipun terdapat sedikit fluktuasi pada beberapa poin. Nilai *error* yang diperoleh masih berada dalam batas wajar dan menunjukkan bahwa sensor INA219 memiliki performa yang cukup akurat untuk pengukuran arus listrik.



Gambar 4.3 Grafik Validasi Sensor Arus INA219

Gambar 4.3 menunjukkan perbandingan antara hasil pengukuran arus oleh sensor INA219 dan multimeter dari waktu ke waktu. Pada grafik tersebut, garis hitam mewakili hasil pembacaan sensor INA219, sedangkan garis merah menunjukkan hasil dari multimeter. Data diambil setiap 5 detik, sehingga grafik menggambarkan perubahan arus selama periode 75 detik. Pada sebagian besar titik, sensor INA219 cenderung membaca nilai arus yang lebih tinggi dibandingkan multimeter. Hal ini terlihat dari garis hitam yang berada di atas garis merah hampir di setiap poin pengukuran.

Perbedaan yang kecil antara kedua alat juga terlihat pada poin-poin awal dan akhir, dengan selisih yang relatif konstan di kisaran 0,009 A hingga 0,014 A. Grafik ini menunjukkan bahwa sensor INA219 memiliki tren pengukuran yang sejalan dengan multimeter, namun terdapat sedikit bias positif yang menyebabkan pembacaan sensor cenderung lebih tinggi. Secara keseluruhan, performa sensor INA219 dalam pengukuran arus listrik cukup baik, dengan perbedaan pembacaan yang masih dalam batas wajar. Meskipun terdapat fluktuasi kecil, pola pengukuran antara INA219 dan multimeter tetap konsisten, menunjukkan bahwa sensor ini dapat diandalkan untuk pemantauan arus listrik pada kondisi yang serupa.

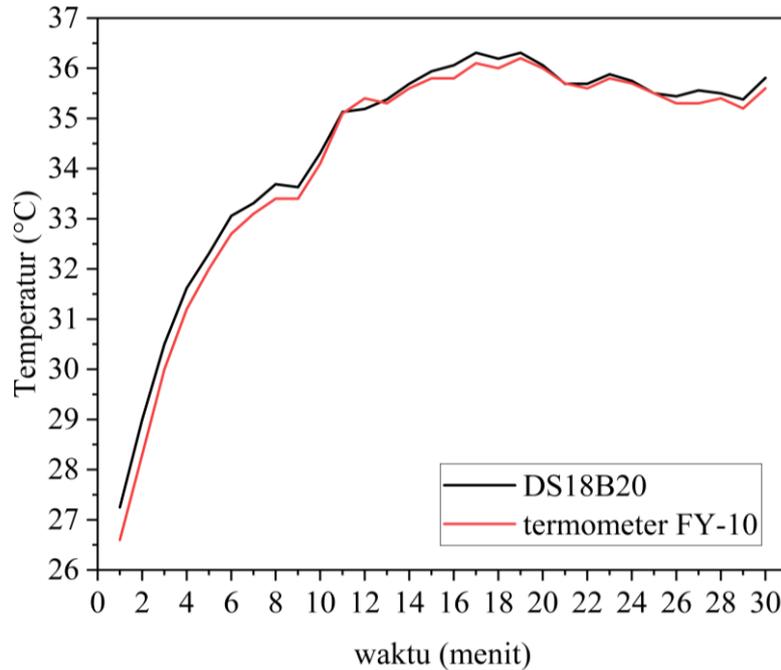
#### 4.2.2 Validasi DS18B20

Validasi dilakukan untuk memastikan keakuratan sensor DS18B20 dalam mendeteksi temperatur pada *heatsink driver stepper motor* yang berperan penting dalam menjaga performa sistem. Pengujian dilakukan dengan membandingkan hasil pembacaan sensor DS18B20 dengan termometer FY-10 yang memiliki probe, di mana keduanya direkatkan berdampingan menggunakan *thermal tape* pada *heatsink*. Data temperatur dicatat setiap 1 menit selama 30 menit dalam kondisi ruangan yang diset pada temperatur 35°C, dengan sistem memutar *stepper motor* secara kontinu.

**Tabel 4.2** Hasil Validasi Sensor Temperatur DS18B20

<b>Waktu (menit)</b>	<b>Temperatur Sensor</b>	<b>Temperatur Termometer FY-10</b>	<b>Selisih</b>	<b>Error</b>
1	27,25	26,6	0,65	2%
2	29,00	28,3	0,70	2%
3	30,50	30	0,50	2%
4	31,62	31,2	0,42	1%
5	32,31	32	0,31	1%
6	33,06	32,7	0,36	1%
7	33,31	33,1	0,21	1%
8	33,69	33,4	0,29	1%
9	33,63	33,4	0,23	1%
10	34,31	34,1	0,21	1%
11	35,13	35,1	0,03	0%
12	35,19	35,4	0,21	1%
13	35,38	35,3	0,08	0%
14	35,69	35,6	0,09	0%
15	35,94	35,8	0,14	0%
16	36,06	35,8	0,26	1%
17	36,31	36,1	0,21	1%
18	36,19	36	0,19	1%
19	36,31	36,2	0,11	0%
20	36,06	36	0,06	0%
21	35,69	35,7	0,01	0%
22	35,69	35,6	0,09	0%
23	35,88	35,8	0,08	0%
24	35,75	35,7	0,05	0%
25	35,5	35,5	0,00	0%
26	35,44	35,3	0,14	0%
27	35,56	35,3	0,26	1%
28	35,5	35,4	0,10	0%
29	35,38	35,2	0,18	1%
30	35,81	35,6	0,21	1%
Rata-Rata			0,21	1%

Berdasarkan data pengujian, sensor DS18B20 menunjukkan kinerja yang cukup akurat. Selisih rata-rata pembacaan antara kedua alat hanya sebesar 0,21°C dengan tingkat *error* rata-rata 1%. Pada awal pengujian, terdapat perbedaan yang lebih besar, yaitu hingga 0,70°C atau 2%, namun selisih ini semakin kecil seiring waktu, hingga mendekati 0°C di beberapa titik pengujian. Hal ini menunjukkan bahwa sensor DS18B20 mampu menyesuaikan pembacaannya dengan lingkungan termal yang berubah dan memberikan hasil yang konsisten.



Gambar 4.4 Grafik Validasi Sensor Temperatur DS18B20

Gambar 4.4 menunjukkan bahwa sensor DS18B20 dan termometer FY-10 memiliki tren pembacaan yang sangat serupa. Kenaikan temperatur terjadi secara bertahap hingga mencapai kestabilan pada menit ke-10, di mana perbedaan nilai antara kedua alat semakin kecil. Pola ini menunjukkan bahwa DS18B20 dapat diandalkan untuk mencatat perubahan temperatur secara *real-time* pada *heatsink driver stepper motor*. Setelah kestabilan tercapai, selisih antara pembacaan kedua alat menjadi sangat kecil, dengan beberapa menit terakhir menunjukkan selisih kurang dari 0,1°C.

### 4.3 Hasil Pengujian *Globe Valve* (Manual)

Pengujian manual dilakukan untuk menentukan karakteristik fisik dari *globe valve*, khususnya jumlah putaran yang dibutuhkan untuk mencapai posisi buka penuh dan tutup penuh. Pengujian dilakukan dengan memutar *valve* secara manual dan menghitung sudut rotasi total dari posisi tertutup hingga terbuka sepenuhnya. Pengujian juga melibatkan pencatatan waktu yang dibutuhkan untuk operasi manual ini.

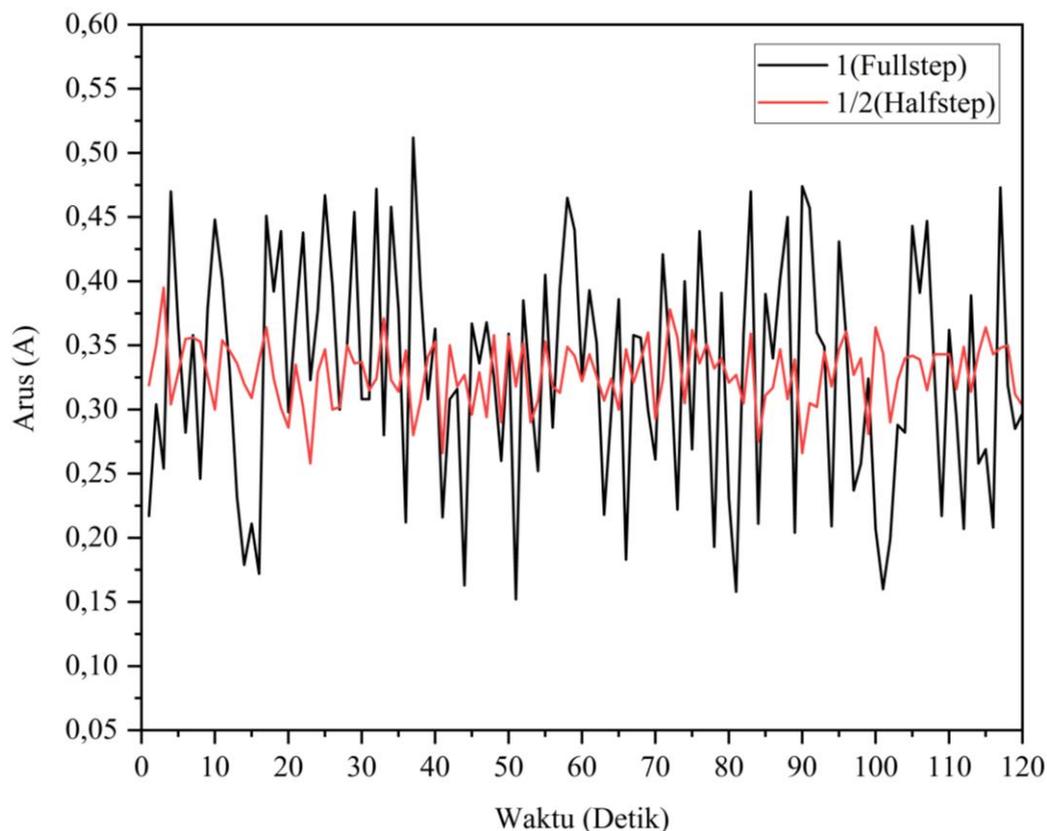
Tabel 4.3 Hasil Pengujian *Globe valve*

No.	Parameter	Hasil
1.	Jumlah Putaran untuk Membuka	5 putaran + 200° (2000°)
2.	Jumlah Putaran untuk Menutup	5 putaran + 200° (2000°)
3.	Beban untuk menggerakkan <i>valve</i>	500g

Hasil pengujian *globe valve* manual pada tabel 4.5 akan digunakan sebagai referensi untuk *step* yang diperlukan setiap 1% pergerakan *valve* yang akan digerakkan dengan *stepper motor* nantinya. Dimana total 100% bukaan atau tutupan *valve* harus memenuhi sejumlah 5 putaran lebih 200° atau total 2000°. Beban untuk menggerakkan *valve* bernilai 500g yang berarti *stepper motor* yang dibutuhkan harus memiliki torsi yang lebih dari nilai ini, untuk nema17 yang digunakan pada tugas akhir ini sudah memenuhi karena memiliki torsi sebesar 4000 g.cm pada 1.2 A.

#### 4.4 Hasil Pengujian Stepper Motor dan Driver DRV8255

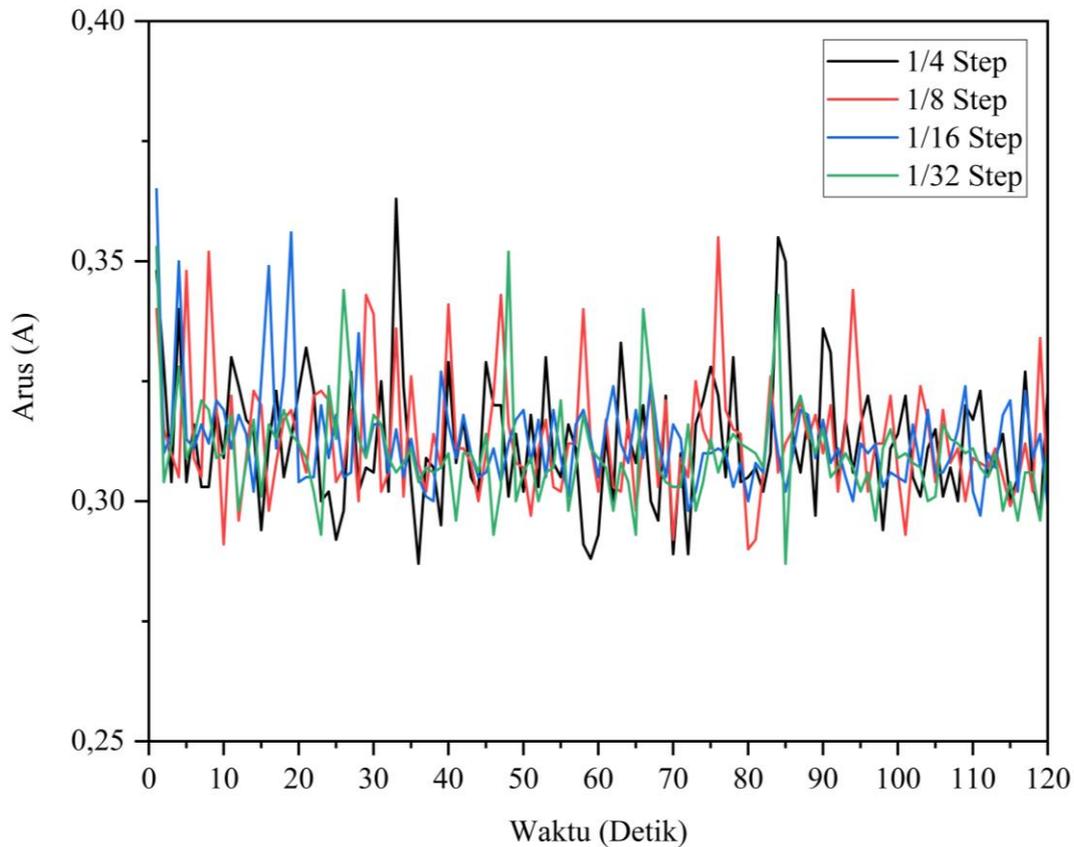
Pengujian ini dilakukan untuk mengevaluasi karakteristik konsumsi arus dan kestabilan temperatur *stepper motor* NEMA17 yang dikendalikan oleh *driver* DRV8255 dengan berbagai mode *stepping*, yakni *full-step*, *half-step* (1/2), 1/4, 1/8, 1/16, dan 1/32 *step*. Arus motor diamati menggunakan sensor INA219, sementara temperatur *driver* dipantau menggunakan DS18B20. Masing-masing data dicatat selama 18 menit (1080 detik) untuk pengamatan temperatur, dan disesuaikan menjadi 120 detik untuk keperluan analisis data arus. Pencatatan data dilakukan secara otomatis ke dalam basis data lokal untuk menjaga akurasi dan kontinuitas data.



Gambar 4.5 Grafik *Stepping Mode* (*Full-Step* dan *Half-Step*) terhadap Arus DRV8255

Gambar 4.5 menunjukkan perbandingan pola arus antara mode *full-step* (garis hitam) dan *half-step* (garis merah). Secara umum, terlihat bahwa mode *full-step* menghasilkan fluktuasi arus yang sangat tinggi, dengan rentang berkisar antara 0,15 A hingga 0,50 A. Lonjakan tajam yang terjadi secara berulang menunjukkan bahwa gaya gerak motor lebih impulsif dan tidak seragam, sehingga berpotensi menghasilkan getaran mekanik.

Sebaliknya, pada mode *half-step*, fluktuasi arus cenderung lebih terkendali dengan rentang antara 0,25 A hingga 0,40 A, menunjukkan bahwa mode ini lebih stabil dalam memberikan daya ke kumparan motor. Grafik memperlihatkan kontur gelombang yang lebih halus, menandakan bahwa transisi tiap langkah lebih ringan dan efisien dalam penggunaan daya. Hasil ini memperkuat bahwa mode *half-step* memberikan kinerja yang lebih stabil dibandingkan *full-step*, baik dari segi arus maupun potensi keausan mekanik jangka panjang.



Gambar 4.6 Grafik *Stepping Mode* (1/4, 1/8, 1/16, dan 1/32) terhadap Arus DRV8255

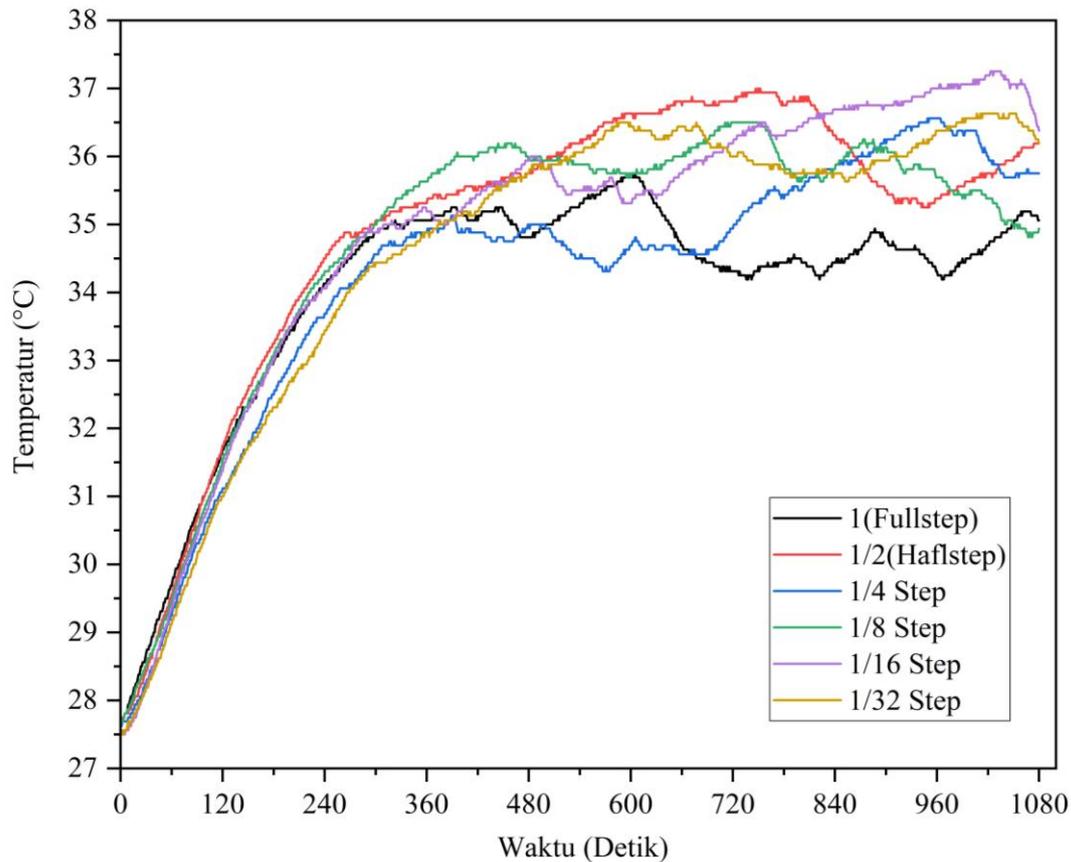
Pada Gambar 4.6, ditampilkan perbandingan konsumsi arus dari mode *microstepping* yang lebih halus, yaitu 1/4 step, 1/8 step, 1/16 step, dan 1/32 step. Terlihat bahwa semakin kecil langkah per gerakan motor (semakin tinggi tingkat *microstepping*), maka fluktuasi arus semakin kecil. Mode 1/4 step (garis hitam) masih menunjukkan lonjakan-lonjakan kecil dengan rentang arus antara 0,275 A hingga 0,375 A, namun sudah jauh lebih halus dibandingkan *full-step*.

Sementara itu, mode 1/8, 1/16, dan 1/32 step (garis merah, biru, dan hijau) memperlihatkan grafik arus yang semakin rata dan stabil, dengan fluktuasi berkisar sangat dekat di sekitar 0,3 A. Hal ini menunjukkan bahwa mode *microstepping* tinggi tidak hanya memberikan gerakan yang lebih presisi secara mekanis, tetapi juga mengurangi konsumsi arus secara lebih efisien dan merata, yang sangat penting untuk sistem aktuator yang digunakan dalam waktu lama.

Jika dibandingkan dengan Gambar 4.5, grafik pada Gambar 4.6 secara umum menunjukkan kestabilan arus yang jauh lebih baik. Mode *full-step* memiliki puncak fluktuasi hingga 0,5 A, sementara mode *microstepping* maksimum (1/32 step) hanya menunjukkan sedikit variasi di sekitar 0,3 A. Hal ini memberikan indikasi bahwa penggunaan *microstepping* seperti 1/16 dan

1/32 tidak hanya meningkatkan kehalusan gerak, namun juga berkontribusi terhadap efisiensi daya pada *driver* dan motor.

Dengan demikian, dalam aplikasi seperti *control valve* berbasis *stepper motor*, mode *stepping* yang lebih tinggi (misalnya 1/16 atau 1/32) lebih disarankan karena mampu memberikan pengendalian yang lebih presisi dengan konsumsi daya yang lebih stabil, serta mengurangi kemungkinan terjadinya lonjakan arus yang dapat mempercepat kerusakan sistem.



Gambar 4.7 Grafik *Stepping Mode* terhadap Temperatur DRV8255

Gambar 4.7 menunjukkan pengaruh variasi mode *stepping* terhadap temperatur kerja *driver* DRV8255 selama pengoperasian *stepper motor* dalam rentang waktu 18 menit. Seluruh mode *stepping* menunjukkan kecenderungan kenaikan temperatur yang serupa. Temperatur meningkat secara bertahap dari sekitar 27,5°C hingga mencapai kisaran 34 - 35°C dalam 5 - 6 menit pertama. Kenaikan ini disebabkan oleh akumulasi panas akibat aliran arus yang terus-menerus melalui kumparan motor dan *switching* internal pada *driver*.

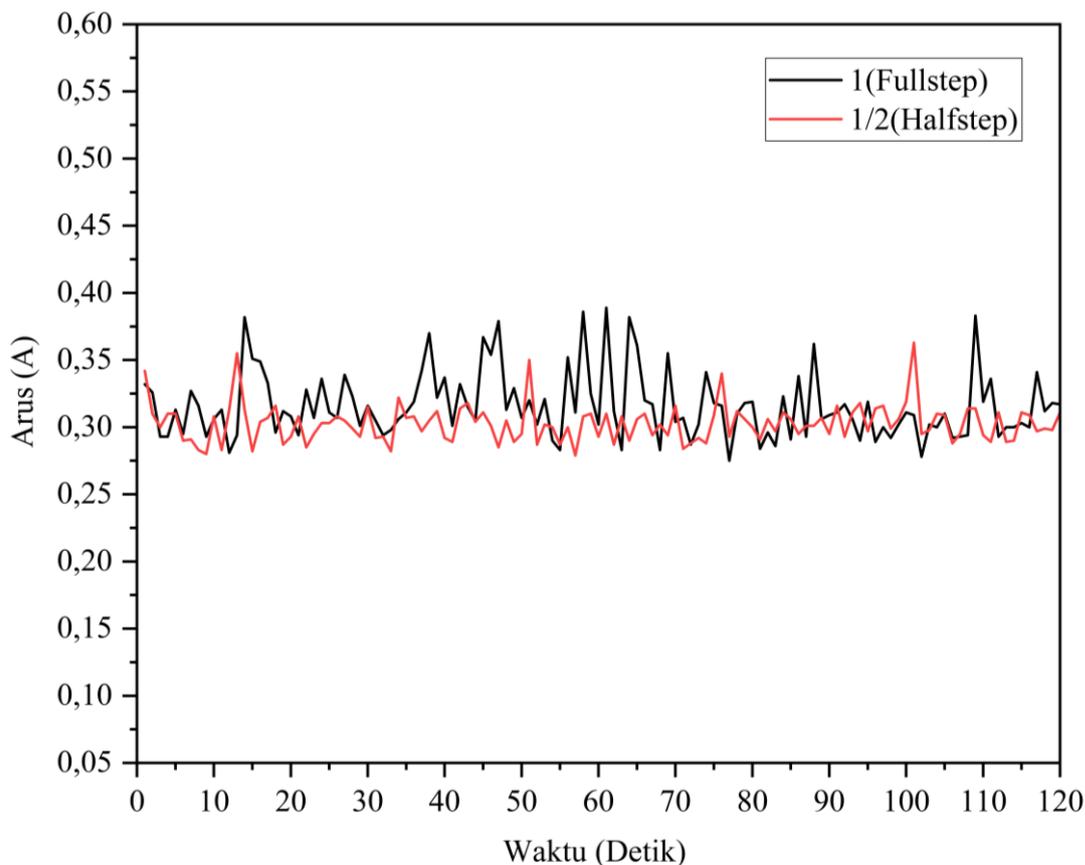
Setelah fase awal, grafik menunjukkan adanya perbedaan nilai temperatur stabil untuk masing-masing mode. Mode *full-step* (garis hitam) menunjukkan temperatur yang paling rendah dibanding mode lainnya, dengan temperatur stabil berkisar antara 34°C hingga 36°C. Mode *half-step* (garis merah) sedikit lebih tinggi, mendekati 37°C, mode 1/4 dan 1/8 step (biru dan hijau) menunjukkan pola yang sedikit lebih bervariasi, dengan fluktuasi kecil namun tetap menjaga temperatur rata-rata pada kisaran 35,5°C hingga 36°C sepanjang pengujian. Sementara itu, pada mode *microstepping* yang lebih tinggi seperti 1/16 step (garis ungu) dan 1/32 step (garis oranye), temperatur tercatat paling tinggi dibandingkan mode lainnya. Grafik menunjukkan temperatur yang terus meningkat dan mencapai titik tertinggi pada sekitar 37,5°C. Hal ini dapat diinterpretasikan sebagai konsekuensi dari kompleksitas kontrol arus dalam mode

*microstepping* yang lebih tinggi, di mana *driver* harus melakukan *switching* yang lebih sering dan presisi untuk menjaga kehalusan gerakan motor. Akumulasi beban *switching* ini menyebabkan peningkatan temperatur yang lebih signifikan dibandingkan mode *stepping* yang lebih kasar.

Secara keseluruhan, meskipun *microstepping* memberikan keuntungan dari sisi kestabilan arus dan kehalusan pergerakan, data temperatur ini menunjukkan adanya kompromi dalam bentuk peningkatan panas pada *driver*. Oleh karena itu, dalam aplikasi jangka panjang atau lingkungan tertutup, penting untuk memperhatikan aspek pendinginan atau batas temperatur maksimum *driver*. Temuan ini memperkuat pentingnya pemilihan mode *stepping* tidak hanya berdasarkan aspek elektrik dan mekanikal, tetapi juga memperhitungkan manajemen termal sistem secara keseluruhan.

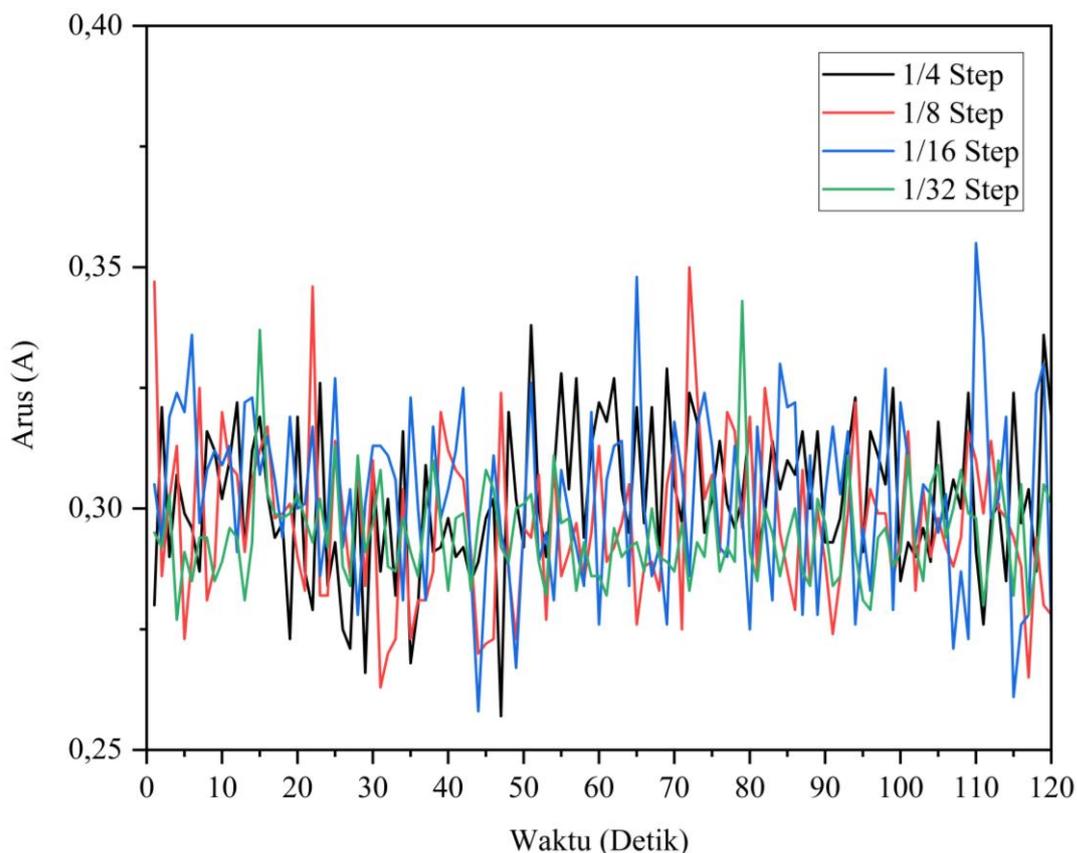
#### 4.5 Hasil Pengujian Stepper Motor dan Driver TMC2209

*Stepper motor* diuji menggunakan *driver* TMC2209 dengan berbagai variasi mode *stepping*, yaitu *fullstep*, *halfstep* (1/2), serta mode *microstepping* 1/4, 1/8, 1/16, dan 1/32. Pengujian ini dilakukan menggunakan sensor INA219 untuk memantau arus secara real time, serta sensor temperatur DS18B20 yang sebelumnya telah dikalibrasi untuk memantau perubahan temperatur akibat beban kerja motor. Perekaman data temperatur dilakukan selama 1080 detik, sementara data arus difokuskan selama durasi 120 detik. Seluruh data yang diperoleh dicatat secara lokal dan divisualisasikan dalam bentuk grafik yang ditampilkan pada Gambar 4.8 dan Gambar 4.9.



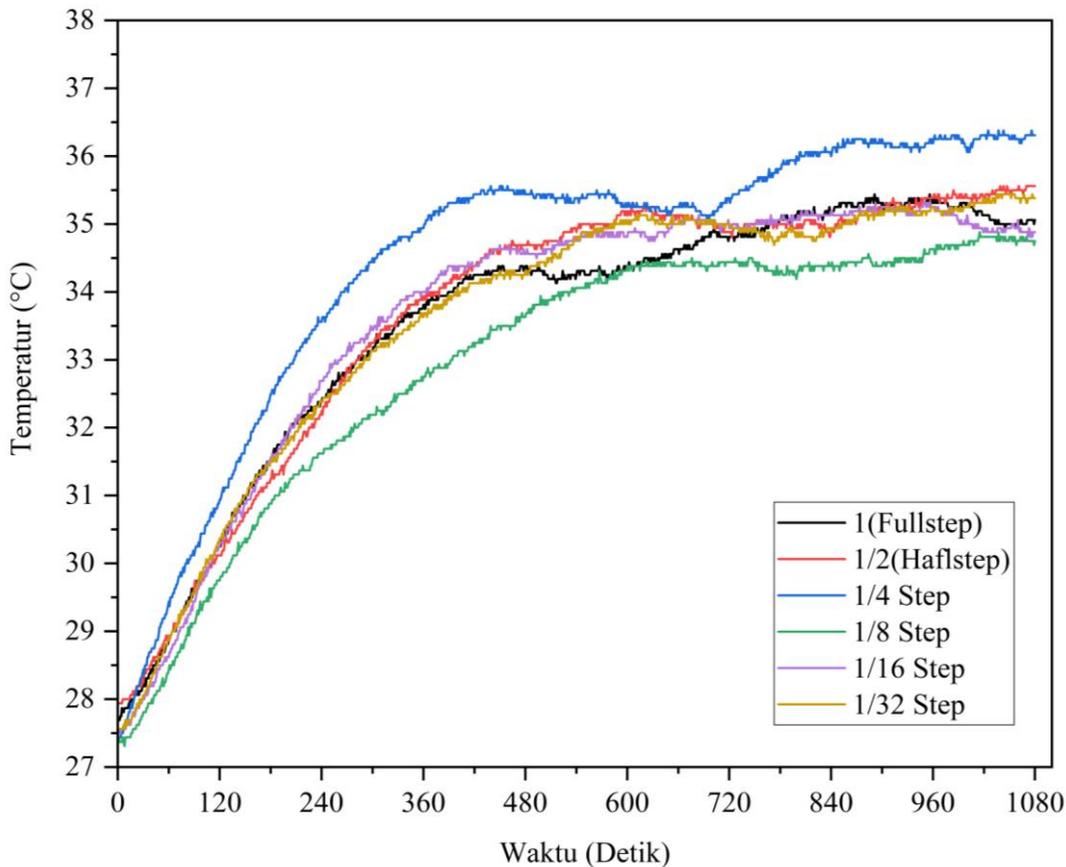
Gambar 4.8 Grafik *Stepping Mode* (*Full-Step* dan *Half-Step*) terhadap Arus TMC2209

Berdasarkan grafik Gambar 4.8, terlihat bahwa arus untuk mode *fullstep* (garis hitam) dan *halfstep* (garis merah) memiliki kisaran nilai yang stabil, berada antara 0,28 A hingga 0,38 A. Fluktuasi arus pada mode *fullstep* memang sedikit lebih besar dibandingkan *halfstep*, tetapi keduanya tetap berada dalam rentang yang aman dan tidak menunjukkan lonjakan ekstrem. Mode *halfstep* tampak menghasilkan arus yang lebih halus dan cenderung rata, yang bisa menunjukkan bahwa torsi dihasilkan dengan lebih halus dan minim getaran. Kondisi ini membuktikan bahwa TMC2209 cukup andal menjalankan kedua mode ini dengan efisiensi yang baik, tanpa indikasi adanya *bouncing* atau getaran keras pada motor, terutama pada mode *fullstep* yang sebelumnya cenderung bermasalah saat menggunakan driver lain seperti DRV8255.



Gambar 4.9 Grafik *Stepping Mode* (1/4, 1/8, 1/16, dan 1/32) terhadap Arus TMC2209

Untuk pengujian *microstepping* seperti yang ditampilkan pada Gambar 4.9, keempat mode memperlihatkan pola fluktuasi arus yang cukup rapat dan dinamis, namun tetap terjaga dalam kisaran aman antara 0,27 A hingga 0,35 A. Tidak terdapat nilai negatif maupun *spike* ekstrem yang menandakan kestabilan *driver* dalam mengatur pembagian langkah. Meskipun terdapat sedikit *noise* atau variasi pada tiap mode (misalnya 1/8 dan 1/16 *step*), perbedaan ini masih dalam batas wajar dan tidak menimbulkan gangguan pada performa motor secara umum. Secara keseluruhan, TMC2209 terbukti mampu menangani mode-mode *microstepping* dengan baik, memberikan kontrol yang lebih presisi pada gerakan motor dengan konsumsi arus yang tetap efisien.



Gambar 4.10 Grafik *Stepping Mode* terhadap Temperatur TMC2209

Gambar 4.10 menunjukkan respons termal driver TMC2209 terhadap berbagai mode *stepping* selama proses pengujian. Pada fase awal, semua mode *stepping* mengalami kenaikan temperatur yang serupa, dimulai dari suhu sekitar 27 °C dan terus meningkat seiring waktu. Setelah melewati 600 detik, grafik mulai menunjukkan kecenderungan stabil, menandakan bahwa sistem telah mencapai kondisi termal tunak. Mode 1/8 *step* menunjukkan performa paling efisien secara termal, dengan suhu stabil berkisar antara 34 °C hingga 35 °C, sedangkan mode 1/4 *step* mencatat temperatur tertinggi, mencapai sekitar 36 °C hingga 37 °C. Hal ini mengindikasikan bahwa pada mode 1/4 *step*, driver bekerja dengan beban *switching* dan arus yang menghasilkan disipasi daya lebih besar dibandingkan mode lainnya.

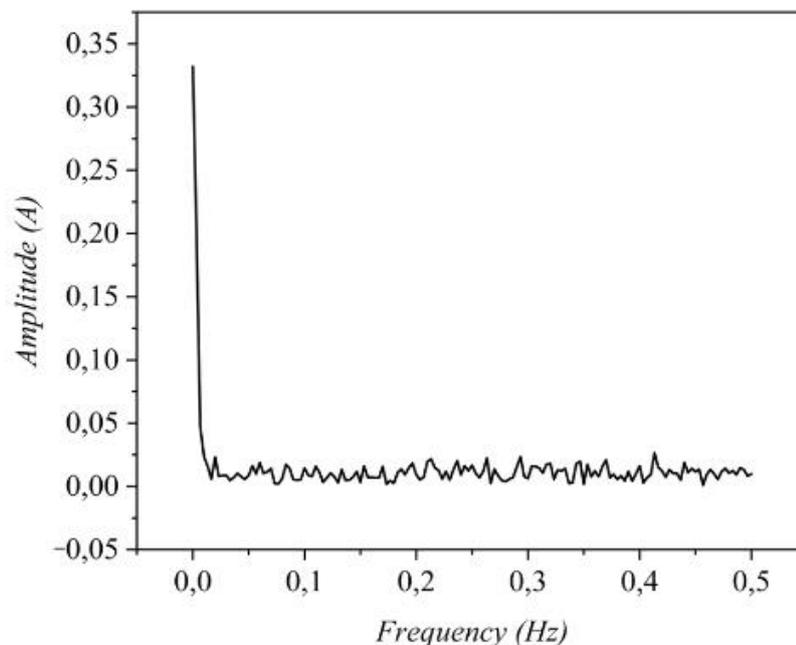
Performa TMC2209 secara keseluruhan tampak lebih unggul dibandingkan DRV8825 dalam hal pengendalian panas. Temperatur akhir yang dicapai oleh TMC2209 lebih rendah dan cenderung stabil untuk semua mode, mencerminkan efisiensi internal serta kemampuan manajemen termal yang lebih baik dari sisi desain rangkaian maupun teknik *switching*-nya. Perbedaan temperatur yang tidak terlalu ekstrem antar mode juga menunjukkan bahwa TMC2209 mampu mempertahankan kestabilan kerja tanpa fluktuasi termal yang berlebihan, meskipun bekerja pada mode *microstepping* tinggi seperti 1/16 dan 1/32 *step*.

Secara keseluruhan, hasil ini menguatkan posisi TMC2209 sebagai driver yang cocok untuk aplikasi presisi tinggi tanpa mengorbankan kestabilan termal. Penggunaan *microstepping* dalam sistem kendali aktuator dapat dimaksimalkan dengan memanfaatkan efisiensi termal *driver* ini, sehingga sistem tidak hanya presisi dalam kendali posisi, tetapi juga andal untuk pengoperasian jangka panjang.

#### 4.6 Evaluasi *Driver* dengan Analisis FFT

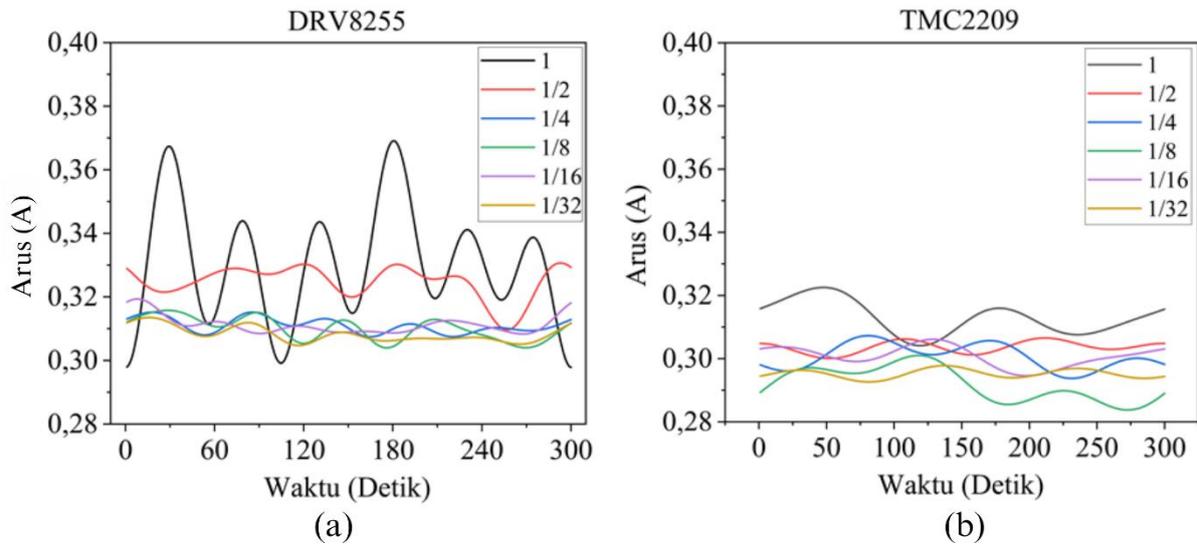
Evaluasi lebih lanjut dilakukan terhadap hasil pengujian *driver* DRV8255 dan TMC2209 dengan pendekatan analisis spektrum frekuensi terhadap data arus motor yang dihasilkan. Analisis ini ditambahkan karena grafik arus awal tampak kurang jelas untuk dibaca akibat fluktuasi sinyal yang tinggi. Oleh karena itu, dilakukan *Fast Fourier Transform* (FFT) untuk melihat lebih dalam karakteristik sinyal arus secara frekuensi.

Pengolahan FFT dilakukan pada data arus *full-step driver* DRV8255, dan hasilnya ditampilkan pada Gambar 4.11. Terlihat bahwa sinyal memiliki puncak amplitudo besar di frekuensi 0 Hz, yang menunjukkan adanya komponen arus DC dimana kondisi ini memang sesuai dengan karakteristik *stepper motor* DC yang digunakan. Selain itu, terdapat amplitudo kecil di antara frekuensi 0,1 Hz hingga 0,5 Hz, yang kemungkinan besar merupakan *noise* atau komponen harmonik kecil akibat variasi mekanis seperti getaran atau akibat *switching internal driver*. Dengan melihat hasil spektrum ini, dapat dilihat bahwa sinyal arus mengandung *noise* frekuensi rendah yang bisa memengaruhi interpretasi data dalam bentuk domain waktu.



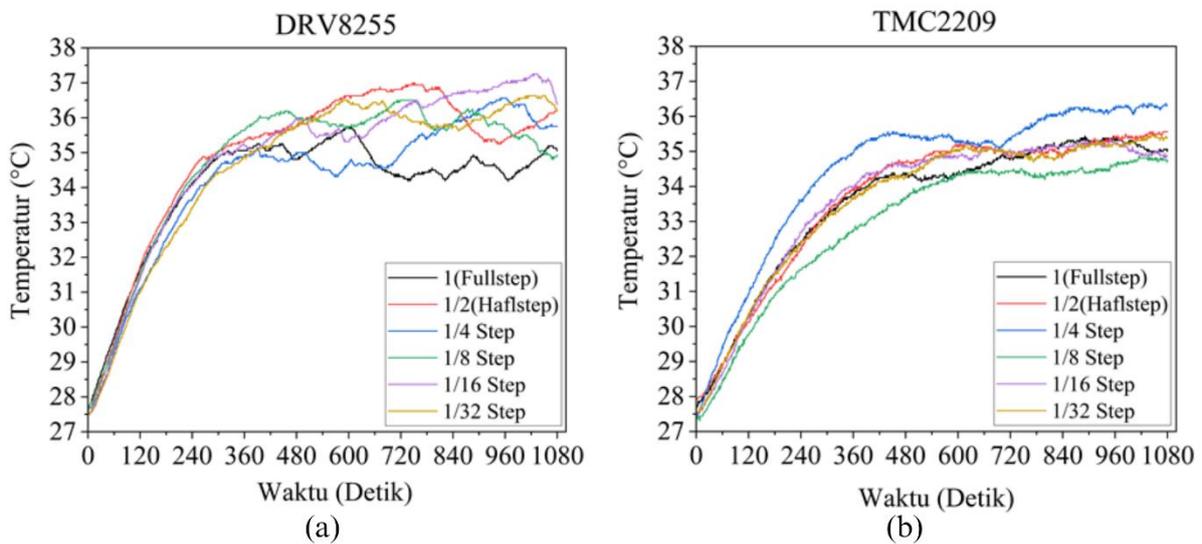
Gambar 4.11 Grafik Arus *Fullstep Driver* DRV8255 dengan FFT

Berdasarkan hasil FFT tersebut, dilakukan filtering sinyal menggunakan *low-pass filter* dengan *cutoff* di bawah 0,1 Hz, bertujuan untuk menghilangkan *noise* pada frekuensi yang lebih tinggi. Hasil dari proses penyaringan ini digunakan untuk menghasilkan grafik perbandingan arus antar kedua *driver*. Grafik tersebut ditampilkan pada Gambar 4.12, di mana terlihat dengan lebih jelas bahwa DRV8255 menunjukkan fluktuasi arus yang lebih besar pada mode *full-step*, sedangkan TMC2209 menghasilkan arus yang lebih halus dan stabil di semua mode *stepping*. Penerapan filter ini membantu mempertegas perbedaan performa masing-masing *driver*, yang sebelumnya tertutup oleh *noise* sinyal.



Gambar 4.12 Perbandingan Arus (a) DRV8255 dan (b) TMC2209

Selain arus, temperatur operasi masing-masing *driver* juga diamati sebagai parameter penting dalam menentukan efisiensi sistem. Hasil pengukuran temperatur dapat dilihat pada Gambar 4.13, yang menunjukkan bahwa DRV8255 mengalami kenaikan temperatur lebih cepat dan lebih tinggi dibanding TMC2209. Pada beberapa mode *microstepping*, temperatur DRV8255 bahkan mendekati 38 °C, sementara TMC2209 tetap berada di kisaran temperatur lebih rendah. Hal ini mendukung kesimpulan bahwa TMC2209 memiliki efisiensi termal yang lebih baik.



Gambar 4.13 Perbandingan Temperatur (a) DRV8255 dan (b) TMC2209

Dari hasil evaluasi ini, dapat disimpulkan bahwa TMC2209 memiliki keunggulan dalam kestabilan arus, kejernihan sinyal listrik, dan efisiensi panas, sehingga menjadikannya lebih layak digunakan dalam sistem kendali presisi berbasis *stepper motor* dibandingkan DRV8255. Hasil pengujian dari kedua *driver* dibandingkan untuk melihat *driver* mana yang sebaiknya digunakan antara *driver* DRV8255 atau *driver* TMC2209 yang membutuhkan biaya yang lebih mahal dibandingkan dengan *driver* DRV8255.

#### 4.7 Hasil Pengujian *Optical Rotary Encoder*

Pada pengujian *optical rotary encoder* dilakukan dengan *stepper motor* yang berputar 1 putaran dengan plat *rotary encoder* yang berada di *coupling* untuk mendapatkan nilai *HIGH* dan *LOW*. *Output* digital akan *HIGH* ketika ada yang menghalangi sensor dan *LOW* ketika tidak ada yang menghalangi dalam *slot* atau ketika melewati bolongan plat. Bolongan pada plat berjumlah 20 jadi seharusnya LED akan mati dan nilai akan terbaca *LOW* sebanyak 20 kali, begitu juga dengan bagian yang tidak bolong untuk menghalangi sensor dan membuat pembacaan menjadi *HIGH*. Namun pada pembacaan sinyalnya tidak mengambil ketika berada di *HIGH* ataupun *LOW* melainkan menggunakan pendekatan *edge* dimana *RisingEdge* membaca transisi dari *LOW* ke *HIGH* dan *FallingEdge* membaca transisi dari *HIGH* ke *LOW*. hasil datanya ditampilkan pada tabel 4.6 dan kode yang digunakan yaitu :

```
// Deteksi rising edge (LOW ke HIGH)
if (lastEncoderState == LOW && currentEncoderState == HIGH) {
    RisingEdgeCount++;
}
// Deteksi falling edge (HIGH ke LOW)
if (lastEncoderState == HIGH && currentEncoderState == LOW) {
    FallingEdgeCount++;
}
```

**Tabel 4.4** Hasil Pengujian *Optical Rotary Encoder*

<i>Optical Rotary Encoder</i>				
Percobaan Ke-	Jumlah <i>RisingEdge</i>	Jumlah <i>FallingEdge</i>	LED OFF ( <i>HIGH</i> )	LED ON ( <i>LOW</i> )
1	20	20	20	20
2	20	20	20	20
3	20	20	20	20
4	20	20	20	20
5	20	20	20	20

Hasil pengujian dapat dilihat bahwa LED sudah menyala dan mati sebagaimana mestinya ketika motor berputar 1 putaran yaitu berjumlah 20 *LOW* dan 20 *HIGH*. Nilai pembacaan dari program pengujiannya juga sudah menghasilkan nilai yang sama dengan pembacaan transisi yaitu *RisingEdge* dan *FallingEdge*, pendekatan ini menghasilkan pembacaan yang lebih akurat dibandingkan dengan pembacaan nilai *LOW* dan *HIGH* secara langsung. Dari hasil pengujian ini, *optical rotary encoder* sudah siap untuk digunakan sebagai *positioner* ataupun sistem *safety* untuk *control valvenya*

#### 4.8 Hasil Pengujian *Rotary Encoder*

Pertama dalam pengujian *Rotary encoder* KY-040 sebagai komponen utama untuk *input* bukaan/tutupan persentase *valve*, maka langkah pertama diuji dahulu pembacaan atau nilai yang dihasilkan setiap langkahnya untuk 1 putaran *rotary encoder*, 1 putarannya memiliki 20 langkah jadi seharusnya nilai yang dihasilkan untuk 1 putaran adalah 20.

**Tabel 4.5** Hasil Pengujian *Rotary Encoder* (Sebelum Kalibrasi)

Pengujian <i>Rotary encoder</i> (Sebelum Kalibrasi)		
Percobaan ke-	Nilai LCD	Step 1 putaran
1	40	20
2	39	20
3	40	20
4	40	20
5	39	20

Data pada Tabel 4.5 pengujian *rotary* sebelum kalibrasi menunjukkan bahwa nilai yang dihasilkan setiap *step*nya berjumlah 2 sehingga nilai pada LCD atau yang terbaca adalah 40, kemudian juga pada percobaan ke 2 dan ke 5 ada *step* yang menghasilkan 1 nilai sehingga nilainya untuk 1 putaran adalah 39, sehingga dibutuhkan penyesuaian dalam pembacaannya.

**Tabel 4.6** Hasil Pengujian *Rotary Encoder* (Sesudah Kalibrasi)

Pengujian <i>Rotary encoder</i> (Sesudah Kalibrasi)		
Percobaan ke-	Nilai LCD	Step 1 putaran
1	20	20
2	20	20
3	20	20
4	20	20
5	20	20

Data pada Tabel 4.8 merupakan data *rotary encoder* ky-040 yang sudah disesuaikan pada programnya dengan pembacaan dibagi 2, pada kodenya digunakan *handler* “*rotaryEncoder.readEncoder() / 2;*” agar nilai yang terbaca tidak lagi bernilai ganda dan dapat digunakan pada sistem utama yang terintegrasi untuk menambah persentase *valve*.

#### 4.9 Hasil Pengujian Control Valve

Pengujian *Control valve* dilakukan sebelum pemasangan pada *Control valve tester*. Dalam pengujiannya melibatkan *optical rotary encoder* untuk melihat nilai yang terbaca apakah sudah sesuai dengan nilai teoretisnya atautkah ada selisih dari bukaan ataupun tutupan *control valve*. Berdasarkan nilai derajat yang dibutuhkan untuk membuka/menutup *valve*, kemudian jumlah *step* perputaran pada *stepper motor*, jumlah *RisingEdge* dan *FallingEdge* pada *optical rotary encoder*, maka seharusnya untuk membuka dan menutup *valve* secara penuh nilai yang terbaca untuk *RisingEdge* dan *FallingEdge* masing-masing harus berjumlah 111. Hasil pengujian saat membuka *valve* dapat dilihat pada Tabel 4.7 dan keadaan saat menutup dapat dilihat pada Tabel 4.8.

**Tabel 4.7** Hasil Pengujian untuk Membuka *Control Valve*

Percobaan ke-	Nilai <i>Optical Rotary encoder (RisingEdge)</i>	Nilai Seharusnya (Teoretis)	Selisih	Persentase <i>Error (%)</i>
1	109	111	2	1,8%
2	109	111	2	1,8%
3	112	111	1	0,9%
4	110	111	1	0,9%
5	109	111	2	1,8%
Rata-Rata			1,6	1,4%

Hasil pengujian untuk membuka *valve* menunjukkan bahwa untuk membuka *valve* secara penuh dari kondisi menutup tidak menghasilkan nilai yang sesuai, rata-rata dari 5 pengujiannya memiliki selisih hingga 1,6 dan memiliki rata-rata *error* di 1,4%. Nilai rata-rata *error* ini masih kurang ideal untuk sistem yang membutuhkan *Control valve* yang sangat akurat dimana *error* harus berada dibawah 1%. Namun untuk pengendalian aliran yang tidak begitu membutuhkan akurasi yang sangat tinggi maka *Control valve* ini bisa digunakan karena nilai *error* masih dibawah 2%

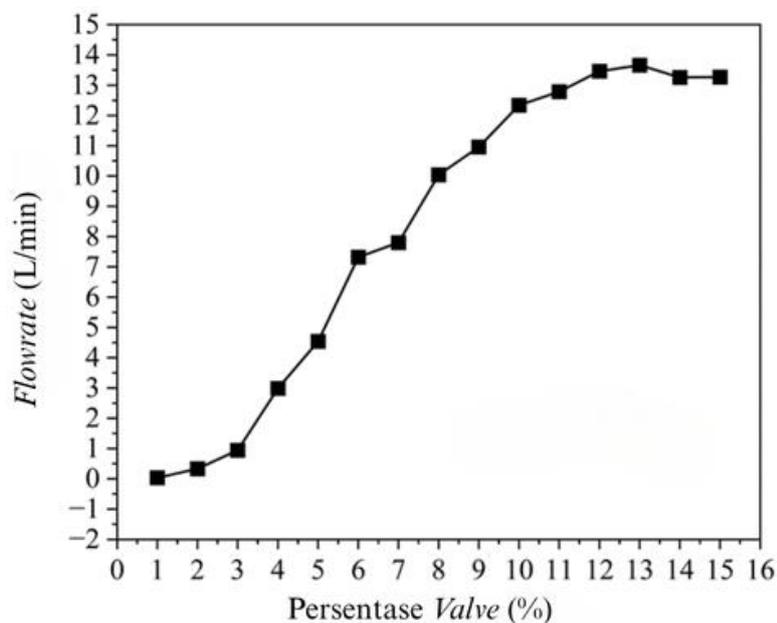
**Tabel 4.8** Hasil Pengujian untuk Menutup *Control Valve*

Percobaan ke-	Nilai <i>Optical Rotary encoder (FallingEdge)</i>	Nilai Seharusnya (Teoretis)	Selisih	Persentase <i>Error (%)</i>
1	108	111	3	2,7%
2	109	111	2	1,8%
3	109	111	2	1,8%
4	113	111	2	1,8%
5	109	111	2	1,8%
Rata-Rata			2,2	2,0%

Pengujian untuk menutup *Control valve* juga tidak menunjukkan nilai yang sesuai dari 5 kali pengujiannya. Selisih rata-rata untuk menutup *valve* ini bernilai 2,2 dengan rata-rata nilai *error* sebesar 2%. Berdasarkan nilai *error* saat membuka dan menutup *valve* seperti yang sudah disebutkan sebelumnya bahwa *Control valve* ini masih belum ideal untuk pengendalian yang membutuhkan akurasi yang sangat tinggi. *Control valve* ini masih membutuhkan penyesuaian untuk mendapatkan hasil yang sesuai untuk membuka atau menutup secara penuh dan juga disetiap 1% nya.

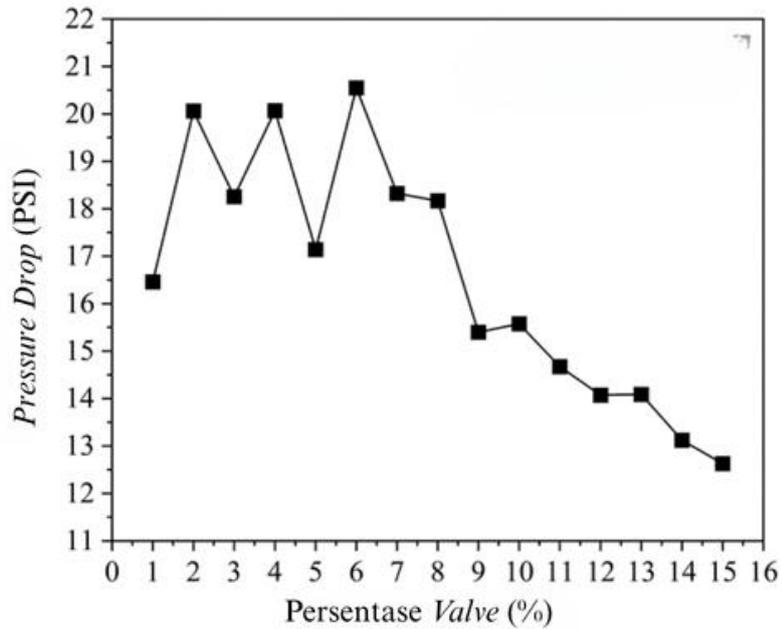
#### 4.10 Hasil Pengambilan Data dengan *Control Valve Tester*

Pengambilan data pada *control valve* dilakukan untuk mengetahui seberapa besar *deadband* yang dimiliki oleh sistem serta bagaimana karakteristik aliran yang terjadi saat *valve* dibuka secara bertahap. Pengujian dimulai dari bukaan terkecil sebesar 1%, kemudian dilanjutkan hingga mencapai 15%, dengan mencatat secara rinci respons yang muncul pada tiap langkah. Pada saat *valve* diberi input bukaan 1%, dilakukan pengamatan untuk memastikan apakah ada kenaikan nilai *flowrate* serta *pressure drop*. Grafik yang ditampilkan pada Gambar 4.14 memperlihatkan bahwa seiring meningkatnya persentase bukaan, *flowrate* yang dihasilkan pun meningkat secara bertahap. Hal ini menunjukkan bahwa *valve* memberikan respons yang cukup baik bahkan pada perubahan input kecil, sehingga bisa dikatakan bahwa *deadband*-nya relatif sempit.



Gambar 4.14 Grafik Perubahan *Flowrate* Setiap 1% Bukaan *Valve*

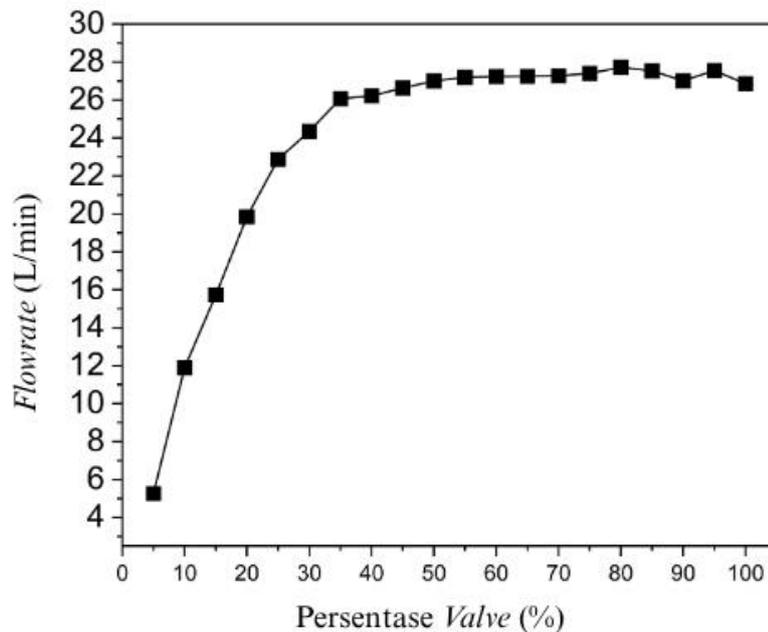
Selain itu, evaluasi terhadap *pressure drop* juga dilakukan sebagai pelengkap analisis. Penurunan tekanan menjadi indikator penting dalam memahami resistansi aliran, terutama pada saat bukaan *valve* masih rendah. Dengan demikian, hasil pengujian ini memberi gambaran bahwa kontrol terhadap aliran dapat dilakukan secara presisi bahkan dengan perubahan kecil pada bukaan, menjadikan data ini penting untuk menentukan performa keseluruhan dari sistem kendali yang digunakan.



Gambar 4.15 Grafik Perubahan *Pressure Drop* Setiap 1% Buka-an *Valve*

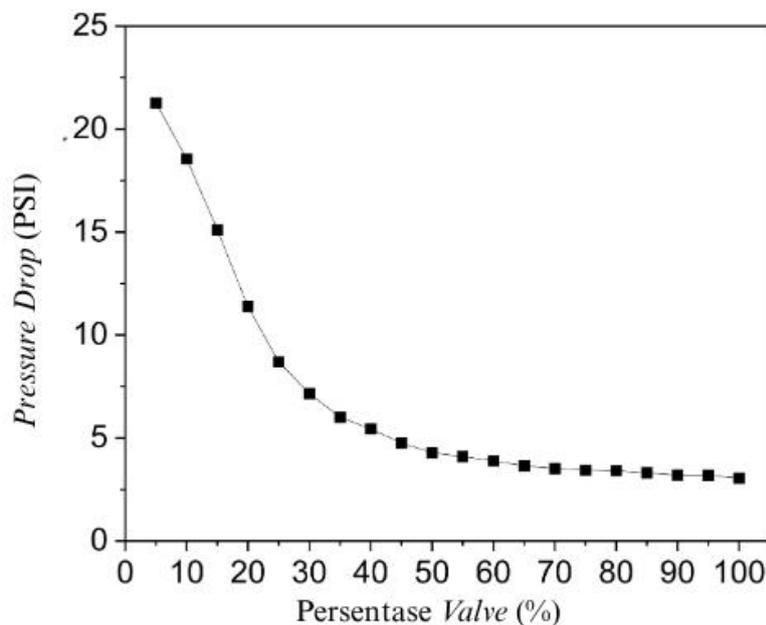
Grafik perubahan *pressure drop* setiap *input* 1% pada *valve* juga menunjukkan adanya perubahan, dimana semakin tinggi bukaan maka semakin turun juga nilai dari *pressure drop*. Dari kedua grafik pengujian terhadap *flowrate* dan *pressure drop* ini bisa dikatakan bahwa *deadband* sudah berada di nilai 1% dan sudah bisa dikatakan ideal untuk performa *Control valve* ini sesuai dengan referensi pada dasar teori performa pada *control valve*.

Selanjutnya dilakukan pengambilan data untuk melihat grafik karakteristik dari aliran fluida yang melewati *control valve*. Kemudian juga dicatat *pressure drop* dari *Control valve* tersebut. Pengujiannya dilakukan dengan membuka *valve* sebesar 5% dari 0% hingga 100%, grafik hasil pengujian terhadap *flowratanya* dapat dilihat pada Gambar 4.16 dan 4.17.



Gambar 4.16 Grafik Karakteristik Aliran dari Buka-an *Valve*

Dari grafik karakteristik aliran fluida di Gambar 4.16 dapat dilihat bahwa karakteristik aliran dari *Control valve* ini sudah sesuai dengan jenis *disc* pada *globe valve* yaitu *plug type* dan karakteristik aliran ini berjenis *quick opening* sesuai dengan Gambar 2.1 pada dasar teori. Selanjutnya hal yang perlu diperhatikan yaitu *pressure drop* pada *control valve* tersebut.



Gambar 4.17 Grafik *Pressure Drop* dari Bukaannya *Valve*

Dari grafik pada Gambar 4.16 menunjukkan bahwa nilai dari *pressure drop* semakin turun seiring dengan bertambahnya bukaan *valve*, hal ini sudah sesuai teori dimana jika *valve* menutup maka *pressure* sebelum *valve* akan lebih besar dibanding *pressure* sesudah *valve*, begitu juga ketika *valve* membuka penuh maka *pressure drop* akan semakin rendah.

#### 4.11 Hasil Analisa Performa *Control Valve*

Berdasarkan hasil pengujian pada sub-bab (4.9) dan hasil pengambilan data menggunakan *valve tester* pada sub-bab (4.10), performa *control valve* dapat dianalisis dari dua aspek utama, yaitu akurasi bukaan *valve* serta karakteristik aliran dan *pressure drop*. Pada pengujian bukaan dan penutupan penuh *valve*, rata-rata *error* masing-masing mencapai 1,4% dan 2%. Nilai ini masih berada dalam batas toleransi untuk aplikasi yang tidak memerlukan akurasi tinggi, performa ini kurang ideal untuk sistem yang membutuhkan presisi di bawah 1%, *control valve* ini masih memerlukan penyesuaian lebih lanjut agar dapat memenuhi standar akurasi untuk aplikasi presisi tinggi.

Hasil pengujian menggunakan *control valve tester* menunjukkan performa yang cukup memuaskan dalam aspek karakteristik aliran dan *pressure drop*. Perubahan *flowrate* pada setiap peningkatan bukaan sebesar 1% menunjukkan adanya respon aliran yang konsisten, dengan *deadband* yang sudah berada pada kisaran 1%. Hal ini mengindikasikan bahwa *control valve* ini telah memiliki performa yang ideal untuk kontrol aliran fluida. Selain itu, karakteristik aliran yang dihasilkan sesuai dengan jenis *plug type disc* pada *globe valve*, yaitu *quick opening*. Grafik *pressure drop* juga menunjukkan hubungan yang sesuai dengan teori, di mana *pressure drop* menurun seiring dengan bertambahnya bukaan *valve*. Hal ini mengkonfirmasi bahwa *control valve* berfungsi sesuai prinsip dasar pengendalian aliran fluida.

## BAB 5 KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan, dapat disimpulkan beberapa hal berikut:

- Perancangan *control valve* berbasis *globe valve* dan *stepper motor* berhasil direalisasikan melalui integrasi perangkat keras dan lunak, termasuk penggunaan sensor arus dan temperatur untuk monitoring, *rotary encoder* untuk *input* manual, serta *optical encoder* sebagai sistem verifikasi posisi. Sistem ini juga memungkinkan pemilihan mode awal (NO/NC) melalui *toggle switch*. Pengujian validasi dan integrasi membuktikan bahwa keseluruhan sistem dapat beroperasi sesuai perintah *input*.
- Performa *stepper motor* sebagai aktuator dalam mengatur aliran fluida menunjukkan hasil yang cukup baik. Pengujian akurasi bukaan dan penutupan *valve* menghasilkan rata-rata *error* masing-masing sebesar 1,4% dan 2%. Karakteristik aliran menunjukkan respons yang konsisten, dengan *deadband* sebesar 1% dan karakteristik *flowrate* bertipe *quick opening*. Analisis arus juga menunjukkan bahwa *driver* TMC2209 memberikan performa kelistrikan yang lebih stabil dan efisien dibandingkan DRV8825. Secara keseluruhan, sistem mampu menjalankan fungsi pengaturan aliran fluida secara presisi dan responsif sesuai tujuan awal perancangan.

### 5.2 Saran

Untuk meningkatkan performa sistem *control valve*, beberapa saran pengembangan yang dapat dilakukan yaitu :

- Optimasi pada sistem penggerak, untuk mengurangi *error* bukaan *valve* agar sesuai dengan standar akurasi yang lebih tinggi.
- Pengujian lebih lanjut pada karakteristik aliran fluida untuk berbagai jenis fluida dengan viskositas berbeda guna memahami performa sistem dalam kondisi yang lebih beragam.
- Pengujian lebih lanjut dengan tipe *valve* lainnya.
- Pertimbangkan untuk mengintegrasikan sistem tambahan seperti koneksi protokol komunikasi seperti I2C, UART, SPI ataupun IoT untuk kontrol secara otomatis.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- Abuowda, K., Noroozi, S., Dupac, M., & Godfrey, P. (2019). Model based driving analysis for a novel stepped rotary flow control valve. *IFAC-PapersOnLine*, 52(12), 549–554. <https://doi.org/10.1016/j.ifacol.2019.11.301>
- Anugraha, R. P., Juwari, Renanto, Fachrudya, S. P., & Daoed, Y. E. (2024). IMPLEMENTATION OF AN ECONOMICAL IoT-BASED LIQUID TANK CONTROL SYSTEM TO SUPPORT ONLINE PROCESS CONTROL PRACTICUM. *Journal of Chemical Technology and Metallurgy*, 59(1), 105–112. <https://doi.org/10.59957/jctm.v59.i1.2024.12>
- Asim, T., Oliveira, A., Charlton, M., & Mishra, R. (2019). Improved design of a multi-stage continuous-resistance trim for minimum energy loss in control valves. *Energy*, 174, 954–971. <https://doi.org/10.1016/j.energy.2019.03.041>
- Baumann, H. D. . (2023). *Control valve primer : a user's guide*. ISA.
- Bernardi, F., Carfagna, E., Migliazza, G., Buticchi, G., Immovilli, F., & Lorenzani, E. (2022). Performance Analysis of Current Control Strategies for Hybrid Stepper Motors. *IEEE Open Journal of the Industrial Electronics Society*, 3, 460–472. <https://doi.org/10.1109/OJIES.2022.3185659>
- Chen, C., Zhou, R., & Sun, W. (2023). Design and internal cavitation characteristics of straight-through electric fully enclosed ultra-clean globe valve. *Journal of Mechanical Science and Technology*, 37(7), 3543–3554. <https://doi.org/10.1007/s12206-023-0620-5>
- Daiguji, M., & Yamashita, Y. (2022). An approach for stiction compensation in industrial process control valves. *Computers and Chemical Engineering*, 158. <https://doi.org/10.1016/j.compchemeng.2021.107641>
- Fraden, J. (2016). *Handbook of Modern Sensors Physics, Designs, and Applications Fifth Edition*. <https://doi.org/10.1007/978-3-319-19303-8>
- Guan, A. qi, Xiang, F. na, Lin, Z. hao, Liu, P., Jin, Z. jiang, & Qian, J. yuan. (2024). Experimental and modeling investigation on dynamic response of sticky control valves. *Control Engineering Practice*, 148. <https://doi.org/10.1016/j.conengprac.2024.105953>
- Hughes, A., & Drury, B. (2019). *Electric Motors and Drives Fundamentals, Types and Applications* (5th ed.).
- Kim, S. K., & Ahn, C. K. (2021). Variable-performance positioning law for hybrid-type stepper motors via active damping injection and disturbance observer. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(4), 1308–1312. <https://doi.org/10.1109/TCSII.2020.3020224>
- Li, J. Y., Gao, Z. X., Wu, H., & Jin, Z. J. (2020). Numerical investigation of methodologies for cavitation suppression inside globe valves. *Applied Sciences (Switzerland)*, 10(16). <https://doi.org/10.3390/app10165541>
- Mott, R. L., & Untener, J. A. (2015). *APPLIED FLUID MECHANICS* (7th ed.). PEARSON.

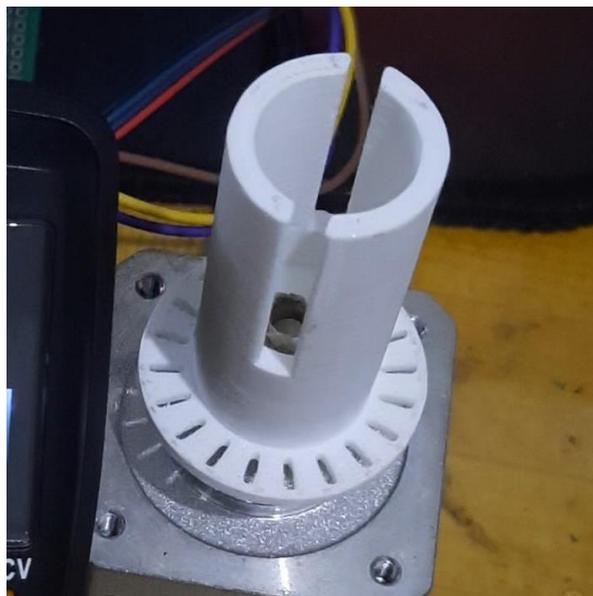
- Nguyen, Q. K., Jung, K. H., Lee, G. N., Suh, S. B., & To, P. (2020). Experimental study on pressure distribution and flow coefficient of globe valve. *Processes*, 8(7). <https://doi.org/10.3390/pr8070875>
- Ogata, Katsuhiko. (2010). *Modern control engineering*. Prentice-Hall.
- Park, M. H., Park, J. S., & Lee, W. J. (2023). Toward optimized operation of freshwater generator using computer vision, and its economic and environmental benefits. *Desalination*, 573. <https://doi.org/10.1016/j.desal.2023.117214>
- Petruzella, F. D. . (2023). *Programmable logic controllers*. McGraw Hill.
- Prodanov, P., & Dankov, D. (2022). Reliability of low-power stepper motor drivers. *2022 31st International Scientific Conference Electronics, ET 2022 - Proceedings*. <https://doi.org/10.1109/ET55967.2022.9920214>
- Ren, W., Wu, L., Zhang, W., Jiang, P., Wang, Z., Luo, C., Guo, J., Liu, C., Wei, Y., Chen, Z., He, Z., Liu, Y., Yu, T., Song, Y., & Yu, B. (2024). Design and Parametric Analysis of the Constant Force Characteristics of the Electromagnet for Hydraulic Valves. *Processes*, 12(12). <https://doi.org/10.3390/pr12122632>
- Silva, C. R. A., Panuci, V. C., Coutinho, M. R., Conceição, W. A. S., & Andrade, C. M. G. (2023). Analysis of the effects of recycling on process control. *Polish Journal of Chemical Technology*, 25(2), 43–55. <https://doi.org/10.2478/pjct-2023-0016>
- Sotoodeh, K. (2019). Actuator selection and sizing for valves. In *SN Applied Sciences* (Vol. 1, Issue 10). Springer Nature. <https://doi.org/10.1007/s42452-019-1248-z>
- Stachowiak, D., & Hemmerling, P. (2022). Development of an Automatic Water Exchange System for Smart Freshwater Aquarium. *Electronics (Switzerland)*, 11(17). <https://doi.org/10.3390/electronics11172705>
- Vázquez, F., Garrido, J., Ruz, M., & Jiménez, J. (2023). Stiction compensation for low-cost electric valves. *Control Engineering Practice*, 134. <https://doi.org/10.1016/j.conengprac.2023.105482>
- Wilson, J. S. (2005). *Sensor Technology Handbook*.
- Yue, Y., Bai, X., & Zuo, X. (2023). FRICTION COMPENSATION OF ELECTRIC CONTROL VALVE BASED ON SLIDING MODE WITH IMPROVED VARIABLE RATE POWER REACHING LAW. *International Journal of Innovative Computing, Information and Control*, 19(2), 403–418. <https://doi.org/10.24507/ijicic.19.02.403>
- Zappe, R. W., & Smith, P. (2004). *Valve Selection Handbook, Engineering Fundamentals for Selecting the Right Valve Design for Every Industrial Flow Application* (5th ed.). Gulf Professional Publishing.

## LAMPIRAN

### Lampiran 1



Gambar 1. Control Valve



Gambar 2. Coupling

## Lampiran 2 program DRV8255

```
#include <AiEsp32RotaryEncoder.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_INA219.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <AccelStepper.h>
// Stepper motor pin
#define DIR_PIN 12
#define STEP_PIN 14
#define SLEEP_PIN 17
// Optical Rotary Encoder Pin
#define ENCODER_PIN 35
// Stepper motor constanta
const int MICROSTEP_MODE = 32; //Adjust base on drive stepping mode
const int BASE_STEPS_PER_REV = 200; // Nema17 total step/rev, 1.8 degree/step
const float TOTAL_DEGREES = 2000.0; // 5 full rotations + 200 degrees (base on globe valve)
const long TOTAL_STEPS = (TOTAL_DEGREES * BASE_STEPS_PER_REV * MICROSTEP_MODE) / 360; // Total
steps for full range
const int STEPS_PER_PERCENT = TOTAL_STEPS / 100; // Steps per 1% movement
AccelStepper stepper(AccelStepper::DRIVER, STEP_PIN, DIR_PIN);
// Variables for position tracking
long targetPosition = 0;
int lastProgress = 0;
bool isInitialized = false;
// Pinout rotary encoder
#define ROTARY_ENCODER_A_PIN 25
#define ROTARY_ENCODER_B_PIN 26
#define ROTARY_ENCODER_BUTTON_PIN 27
#define ROTARY_ENCODER_VCC_PIN -1
// Optical Rotary Encoder Variables
volatile int RisingEdgeCount = 0;
volatile int FallingEdgeCount = 0;
volatile int lastEncoderState = LOW;
// Rotary Encoder Setup
AiEsp32RotaryEncoder rotaryEncoder(ROTARY_ENCODER_A_PIN, ROTARY_ENCODER_B_PIN,
ROTARY_ENCODER_BUTTON_PIN, ROTARY_ENCODER_VCC_PIN);
int progress = 0; // Variable progress untuk progress bar
int progressPixels = 0; // Progress dalam ukuran piksel (untuk progress bar di OLED)
int lastPosition = 0; // Menyimpan posisi terakhir rotary encoder
int switchState = HIGH; // Status awal toggle switch NO/NC
// Pin toggle switch SPDT
const int togglePin = 13; // Toggle switch pin
void IRAM_ATTR readEncoderISR() {
    rotaryEncoder.readEncoder_ISR();
}
bool displayValve = true; // Mode display awal adalah status valve
// Konfigurasi DS18B20
const int oneWireBus = 4;
OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);
float correctionFactor = 5.02; // Koreksi rata-rata
// Konfigurasi INA219 dan OLED
Adafruit_INA219 ina219;
Adafruit_SSD1306 display(128, 64, &Wire);
float temperatureC = 0;
// Variabel untuk menyimpan data INA219
float busvoltage = 0;
float current_mA = 0;
float power_mW = 0;
// Task handles
TaskHandle_t motorTaskHandle = NULL;
TaskHandle_t valveTaskHandle = NULL;
TaskHandle_t monitorTaskHandle = NULL;
```

```

TaskHandle_t displayswitchTaskHandle = NULL;
TaskHandle_t opticalEncoderTaskHandle = NULL;
// Function declarations
void motorTask(void * parameter);
void valveTask(void * parameter);
void monitorTask(void * parameter);
void displayswitchTask(void * parameter);
void opticalEncoderTask(void * parameter);
void measureValues();
void displayMonitoring();
//function proteksi
bool overcurrent = false;
bool overtemp = false;

void setup() {
  Serial.begin(115200);
  // Setup rotary encoder
  rotaryEncoder.begin();
  rotaryEncoder.setup(readEncoderISR);
  rotaryEncoder.setBoundaries(0, 100, true); // Set batas 0 hingga 100 (progress range)
  rotaryEncoder.setAcceleration(0); // 0 berarti tanpa akselerasi
  // Get initial rotary encoder position
  lastPosition = rotaryEncoder.readEncoder() / 2;
  // Setup Optical Rotary Encoder pin
  pinMode(ENCODER_PIN, INPUT_PULLUP);
  // Setup toggle pin
  pinMode(togglePin, INPUT_PULLUP);
  // Inisialisasi sensor DS18B20
  sensors.begin();
  // Inisialisasi INA219
  if (!ina219.begin()) {
    Serial.println("Gagal menemukan modul INA219!");
    while (1) delay(10);
  }
  ina219.setCalibration_32V_2A();
  // Inisialisasi OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("Gagal menemukan OLED"));
    while (1);
  }
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.display();
  // Membuat task untuk motor pada core 0
  xTaskCreatePinnedToCore(
    motorTask, // Function to implement the task
    "motorTask", // Name of the task
    10000, // Stack size in words
    NULL, // Task input parameter
    2, // Priority of the task
    NULL, // Task handle
    0 // Core where the task should run
  );
  // Membuat task untuk valve pada core 0
  xTaskCreatePinnedToCore(
    valveTask, // Function to implement the task
    "valveTask", // Name of the task
    10000, // Stack size in words
    NULL, // Task input parameter
    1, // Priority of the task
    &valveTaskHandle, // Task handle
    0 // Core where the task should run
  );
  // Membuat task untuk optical encoder pada core 0
  xTaskCreatePinnedToCore(
    opticalEncoderTask, // Function to implement the task

```

```

"opticalEncoderTask", // Name of the task
10000,                // Stack size in words
NULL,                 // Task input parameter
1,                    // Priority of the task
&opticalEncoderTaskHandle, // Task handle
0                      // Core where the task should run
);
// Membuat task untuk monitoring pada core 1
xTaskCreatePinnedToCore(
monitorTask,          // Function to implement the task
"monitorTask",        // Name of the task
10000,                // Stack size in words
NULL,                 // Task input parameter
1,                    // Priority of the task
&monitorTaskHandle, // Task handle
1                      // Core where the task should run
);
// Membuat task untuk switch display pada core 1
xTaskCreatePinnedToCore(
displayswitchTask,    // Function to implement the task
"displayswitchTask", // Name of the task
10000,                // Stack size in words
NULL,                 // Task input parameter
1,                    // Priority of the task
&monitorTaskHandle, // Task handle
1                      // Core where the task should run
);
}
void loop() {
vTaskDelete(NULL);
}

void resetOpticalEncoderCounter() {
RisingEdgeCount = 0;
FallingEdgeCount = 0;
}

void setupMotor() {
// Configure motor pins
pinMode(SLEEP_PIN, OUTPUT);
digitalWrite(SLEEP_PIN, HIGH);
// Configure stepper
stepper.setMaxSpeed(16000);
stepper.setAcceleration(8000);
stepper.setCurrentPosition(0);
}

void motorTask(void * parameter) {
setupMotor();
int lastSwitchState = digitalRead(togglePin);
long currentPosition = 0;
bool modeChanging = false;
while(1) {
switchState = digitalRead(togglePin);
// Proteksi overcurrent atau overtemperature
if (overcurrent || overtemp) {
stepper.stop();
stepper.setCurrentPosition(stepper.currentPosition());
digitalWrite(SLEEP_PIN, LOW); // Matikan driver untuk pendinginan
vTaskDelay(pdMS_TO_TICKS(100));
continue;
}
// deteksi perubahan mode operasional
if (switchState != lastSwitchState) {
modeChanging = true;
resetOpticalEncoderCounter(); // Reset counters at start of mode change
// NC ke NO
}
}
}

```

```

if (lastSwitchState == LOW && switchState == HIGH) {
    currentPosition += TOTAL_STEPS;
    stepper.moveTo(currentPosition);
}
// NO ke NC
else if (lastSwitchState == HIGH && switchState == LOW) {
    currentPosition -= TOTAL_STEPS;
    stepper.moveTo(currentPosition);
}
lastSwitchState = switchState;
}
// Proteksi berdasarkan optical encoder
if (modeChanging && (RisingEdgeCount >= 111 || FallingEdgeCount >= 111)) {
    stepper.stop();
    stepper.setCurrentPosition(stepper.currentPosition());
    digitalWrite(SLEEP_PIN, LOW);
    currentPosition = stepper.currentPosition();
    resetOpticalEncoderCounter();
    modeChanging = false;
    vTaskDelay(pdMS_TO_TICKS(100));
    continue;
}
// Mode normal
if (!modeChanging) {
    if (switchState == LOW) {
        if (progress > lastProgress) {
            currentPosition += STEPS_PER_PERCENT;
            stepper.moveTo(currentPosition);
        } else if (progress < lastProgress) {
            currentPosition -= STEPS_PER_PERCENT;
            stepper.moveTo(currentPosition);
        }
    } else {
        if (progress > lastProgress) {
            currentPosition -= STEPS_PER_PERCENT;
            stepper.moveTo(currentPosition);
        } else if (progress < lastProgress) {
            currentPosition += STEPS_PER_PERCENT;
            stepper.moveTo(currentPosition);
        }
    }
}
// Jalankan motor
if (stepper.distanceToGo() != 0) {
    digitalWrite(SLEEP_PIN, HIGH);
    stepper.run();
} else {
    digitalWrite(SLEEP_PIN, LOW);
    if (modeChanging) {
        modeChanging = false;
    }
}
lastProgress = progress;
vTaskDelay(pdMS_TO_TICKS(1));
}

void valveTask(void * parameter) {
    while(1) {
        // Rotary Encoder Handling
        int newPosition = rotaryEncoder.readEncoder() / 2;
        // Periksa apakah posisi rotary encoder telah berubah
        if (newPosition != lastPosition) {
            if (newPosition > lastPosition && progress < 100) {
                progress++; // Tambah progress
            } else if (newPosition < lastPosition && progress > 0) {
                progress--; // Kurangi progress
            }
        }
    }
}

```

```

    }
    lastPosition = newPosition; // Update posisi terakhir
    // Hitung panjang progress bar dalam piksel
    progressPixels = map(progress, 0, 100, 0, 83); // 0-100% mapped to 0-86 pixels
    }
    vTaskDelay(pdMS_TO_TICKS(10));
}
}

void monitorTask(void * parameter) {
    while(1) {
        measureValues();
        vTaskDelay(pdMS_TO_TICKS(1000));
    }
}

void displayswitchTask(void * parameter) {
    while(1) {
        // Handle displayswitch (valvestatus / monitoring) dari Rotary Encoder
        if (rotaryEncoder.isEncoderButtonClicked()) {
            displayValve = !displayValve; // Ubah mode saat tombol rotary encoder ditekan
        }
        // Tampilkan informasi di OLED
        display.clearDisplay();
        if (displayValve) {
            displayStatusValve();
        } else {
            displayMonitoring();
        }
        display.display();
        vTaskDelay(pdMS_TO_TICKS(10));
    }
}

void measureValues() {
    busvoltage = ina219.getBusVoltage_V();
    current_mA = ina219.getCurrent_mA();
    power_mW = ina219.getPower_mW();
    sensors.requestTemperatures();
    temperatureC = sensors.getTempCByIndex(0) + correctionFactor;
    // Proteksi arus lebih (1.2 A = 1200 mA)
    if (abs(current_mA) >= 1200.0) {
        overcurrent = true;
    } else {
        overcurrent = false;
    }
    // Proteksi temperatur lebih (100°C)
    if (temperatureC >= 100.0) {
        overtemp = true;
    } else {
        overtemp = false;
    }
}

void opticalEncoderTask(void * parameter) {
    int currentEncoderState = LOW;
    int lastLocalEncoderState = LOW;
    while(1) {
        currentEncoderState = digitalRead(ENCODER_PIN);
        // Deteksi rising edge (LOW ke HIGH)
        if (lastLocalEncoderState == LOW && currentEncoderState == HIGH) {
            RisingEdgeCount++;
        }
        // Deteksi falling edge (HIGH ke LOW)
        if (lastLocalEncoderState == HIGH && currentEncoderState == LOW) {
            FallingEdgeCount++;
        }
    }
}

```

```

    lastLocalEncoderState = currentEncoderState;
    vTaskDelay(pdMS_TO_TICKS(10));
}
}

void displayStatusValve(){
// Tampilkan informasi di OLED
display.clearDisplay();
//Box Utama
display.drawRect(1, 1, 126, 62, 1);
display.setTextWrap(false);
//Box Status Valve
display.drawRoundRect(4, 3, 120, 28, 1, 1);
display.setCursor(10, 6);
display.print("Valve Opening :");
display.setCursor(98, 18);
display.print(progress);
display.print("%");
//ProgressBar
display.drawRect(9, 15, 87, 13, 1);//outline
display.fillRect(11, 17, progressPixels, 9, 1);//Bar progress
//Box Operating mode
display.drawRoundRect(4, 33, 120, 28, 1, 1);
//Status Operating Mode : Normally Open (NO)/ Normally Close (NC)
display.setCursor(10, 38);
display.print("Operating Mode :");
display.setCursor(10, 49);
switchState = digitalRead(togglePin);
// Tampilkan status toggle switch (NO atau NC)
if (switchState == HIGH) {
    display.print("Normally Open");
} else {
    display.print("Normally Close");
}
display.display(); // Update OLED display
}

void displayMonitoring() {
display.clearDisplay();
if (overcurrent || overtemp) {
    display.setTextColor(SSD1306_BLACK, SSD1306_WHITE); // Inverse warna
    display.setCursor(5, 0);
    if (overcurrent) {
        display.print("Overcurrent! ");
    }
    if (overtemp) {
        display.print("Overtemp! ");
    }
}
display.setTextColor(SSD1306_WHITE); // Kembalikan warna normal
}

display.drawRect(1, 1, 126, 62, 1);
display.setTextWrap(false);
display.drawRect(12, 12, 54, 40, 1);
// Voltage, Current, Power, Temperature display
display.setCursor(15, 24);
display.print("I:");
display.setCursor(15, 15);
display.print("V:");
display.setCursor(15, 33);
display.print("P:");
display.setCursor(15, 42);
display.print("T:");
display.setCursor(27, 15);
display.print(busvoltage, 2);
float current_A = abs(current_mA) / 1000.0;
display.setCursor(27, 24);

```

```
display.print(current_A, 3);
display.setCursor(27, 33);
display.print(power_mW / 1000.0, 2);
display.setCursor(27, 42);
display.print(temperatureC, 2);
display.drawRect(1, 1, 126, 62, 1);
// Units
display.setCursor(58, 15);
display.print("V");
display.setCursor(58, 24);
display.print("A");
display.setCursor(58, 33);
display.print("W");
display.setCursor(58, 42);
display.print("C");
// Optical Rotary Encoder display
display.drawRect(69, 12, 47, 40, 1);
display.setCursor(71, 15);
display.print("WiFi:");
display.setCursor(71, 24);
display.print("Offline");
display.setCursor(72, 33);
display.print("Re:");
display.setCursor(95, 33);
display.print(RisingEdgeCount);
display.setCursor(72, 42);
display.print("Fe:");
display.setCursor(95, 42);
display.print(FallingEdgeCount);
display.display();
}
```

### Lampiran 3 program TMC2209

```
#include <AiEsp32RotaryEncoder.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_INA219.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <AccelStepper.h>
#include <TMCStepper.h>

// Pin definitions for TMC2209
#define TXD_PIN 17 // UART TX pin for TMC2209
#define DIR_PIN 12 // Direction pin
#define STEP_PIN 14 // Step pin
#define STEPMODE 32 // Microstepping mode: 0,2,4,8,16,32,64,128,256
#define MAXCURRENT 1200 // Maximum current in mA
#define R_SENSE 0.11f // Current sense resistor value
// Optical Rotary Encoder Pin
#define ENCODER_PIN 35
// Konstanta stepper motor
const int BASE_STEPS_PER_REV = 200;
const float TOTAL_DEGREES = 2000.0;
const long TOTAL_STEPS = (TOTAL_DEGREES * BASE_STEPS_PER_REV * STEPMODE) / 360;
const int STEPS_PER_PERCENT = TOTAL_STEPS / 100;
// Variables for position tracking
long targetPosition = 0;
int lastProgress = 0;
bool isInitialized = false;
// Pinout rotary encoder
#define ROTARY_ENCODER_A_PIN 25
#define ROTARY_ENCODER_B_PIN 26
#define ROTARY_ENCODER_BUTTON_PIN 27
#define ROTARY_ENCODER_VCC_PIN -1 // Not used, powered directly
// Optical Rotary Encoder Variables
volatile int RisingEdgeCount = 0;
volatile int FallingEdgeCount = 0;
volatile int lastEncoderState = LOW;
// Rotary Encoder Setup
AiEsp32RotaryEncoder rotaryEncoder(ROTARY_ENCODER_A_PIN, ROTARY_ENCODER_B_PIN,
ROTARY_ENCODER_BUTTON_PIN, ROTARY_ENCODER_VCC_PIN);
int progress = 0; // Variable progress untuk progress bar
int progressPixels = 0; // Progress dalam ukuran piksel (untuk progress bar di OLED)
int lastPosition = 0; // Menyimpan posisi terakhir rotary encoder
int switchState = HIGH; // Status awal toggle switch NO/NC
// Pin toggle switch SPDT
const int togglePin = 13; // Toggle switch pin
void IRAM_ATTR readEncoderISR() {
    rotaryEncoder.readEncoder_ISR();
}
bool displayValve = true; // Mode display awal adalah status valve
// Konfigurasi DS18B20
const int oneWireBus = 4;
OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);
float correctionFactor = 5.02; // Koreksi rata-rata
// Konfigurasi INA219 dan OLED
Adafruit_INA219 ina219;
Adafruit_SSD1306 display(128, 64, &Wire);
float temperatureC = 0;
// Variabel untuk menyimpan data INA219
float busvoltage = 0;
float current_mA = 0;
float power_mW = 0;
// Initialize TMC2209 driver
TMC2209Stepper driver(&Serial2, R_SENSE, 0x00);
```

```

AccelStepper stepper(AccelStepper::DRIVER, STEP_PIN, DIR_PIN);
// Task handles
TaskHandle_t motorTaskHandle = NULL;
TaskHandle_t valveTaskHandle = NULL;
TaskHandle_t monitorTaskHandle = NULL;
TaskHandle_t displayswitchTaskHandle = NULL;
TaskHandle_t opticalEncoderTaskHandle = NULL;
// Function declarations
void motorTask(void * parameter);
void valveTask(void * parameter);
void monitorTask(void * parameter);
void displayswitchTask(void * parameter);
void opticalEncoderTask(void * parameter);
void measureValues();
void displayMonitoring();
//proteksi
bool overcurrent = false;
bool overtemp = false;
void setup() {
  Serial.begin(115200);
  // Initialize UART for TMC2209
  Serial2.begin(115200, SERIAL_8N1, TXD_PIN, TXD_PIN);
  // Configure TMC2209
  driver.begin();
  driver.toff(5); // Enable driver
  driver.blank_time(24); // Blank time
  driver.rms_current(MAXCURRENT); // Set motor current
  driver.microsteps(STEPMODE); // Set microsteps
  driver.pwm_autoscale(true); // Enable automatic current scaling
  driver.en_spreadCycle(false); // Enable stealthChop
  // Setup rotary encoder
  rotaryEncoder.begin();
  rotaryEncoder.setup(readEncoderISR);
  rotaryEncoder.setBoundaries(0, 100, true); // Set batas 0 hingga 100 (progress range)
  rotaryEncoder.setAcceleration(0); // 0 berarti tanpa akselerasi
  // Get initial rotary encoder position
  lastPosition = rotaryEncoder.readEncoder() / 2;
  // Setup Optical Rotary Encoder pin
  pinMode(ENCODER_PIN, INPUT_PULLUP);
  // Setup toggle pin
  pinMode(togglePin, INPUT_PULLUP);
  // Inisialisasi sensor DS18B20
  sensors.begin();
  // Inisialisasi INA219
  if (!ina219.begin()) {
    Serial.println("Gagal menemukan modul INA219!");
    while (1) delay(10);
  }
  ina219.setCalibration_32V_2A();
  // Inisialisasi OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("Gagal menemukan OLED"));
    while (1);
  }
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.display();
  // Membuat task untuk motor pada core 0
  xTaskCreatePinnedToCore(
    motorTask, // Function to implement the task
    "motorTask", // Name of the task
    10000, // Stack size in words
    NULL, // Task input parameter
    2, // Priority of the task
    NULL, // Task handle
    0 // Core where the task should run

```

```

);
// Membuat task untuk valve pada core 0
xTaskCreatePinnedToCore(
    valveTask,          // Function to implement the task
    "valveTask",       // Name of the task
    10000,             // Stack size in words
    NULL,              // Task input parameter
    1,                // Priority of the task
    &valveTaskHandle, // Task handle
    0                 // Core where the task should run
);
// Membuat task untuk optical encoder pada core 0
xTaskCreatePinnedToCore(
    opticalEncoderTask, // Function to implement the task
    "opticalEncoderTask", // Name of the task
    10000,             // Stack size in words
    NULL,              // Task input parameter
    1,                // Priority of the task
    &opticalEncoderTaskHandle, // Task handle
    0                 // Core where the task should run
);
// Membuat task untuk monitoring pada core 1
xTaskCreatePinnedToCore(
    monitorTask,       // Function to implement the task
    "monitorTask",    // Name of the task
    10000,            // Stack size in words
    NULL,             // Task input parameter
    1,               // Priority of the task
    &monitorTaskHandle, // Task handle
    1                // Core where the task should run
);
// Membuat task untuk switch display pada core 1
xTaskCreatePinnedToCore(
    displayswitchTask, // Function to implement the task
    "displayswitchTask", // Name of the task
    10000,            // Stack size in words
    NULL,             // Task input parameter
    1,               // Priority of the task
    &monitorTaskHandle, // Task handle
    1                // Core where the task should run
);
}
void loop() {
    vTaskDelete(NULL);
}
void resetOpticalEncoderCounter() {
    RisingEdgeCount = 0;
    FallingEdgeCount = 0;
}
void setMotorSleep(bool sleep) {
    if (sleep) {
        driver.ihold(0); // Set hold current to 0
        driver.TPOWERDOWN(255); // Max power down time
        driver.toff(0); // Disable driver
    } else {
        driver.ihold(16); // Set hold current to 50% of run current
        driver.TPOWERDOWN(10); // Quick power down time
        driver.toff(5); // Enable driver
    }
}
void setupMotor() {
    // Configure stepper
    stepper.setMaxSpeed(16000);
    stepper.setAcceleration(8000);
    stepper.setCurrentPosition(0);
}
void motorTask(void * parameter) {

```

```

setupMotor();
int lastSwitchState = digitalRead(togglePin);
long currentPosition = 0;
bool modeChanging = false;
while(1) {
  switchState = digitalRead(togglePin);
  // Mode change detection
  if (switchState != lastSwitchState) {
    modeChanging = true;
    resetOpticalEncoderCounter();
    setMotorSleep(false); // Wake up motor

    if (lastSwitchState == LOW && switchState == HIGH) {
      currentPosition += TOTAL_STEPS;
      stepper.moveTo(currentPosition);
    } else if (lastSwitchState == HIGH && switchState == LOW) {
      currentPosition -= TOTAL_STEPS;
      stepper.moveTo(currentPosition);
    }
    lastSwitchState = switchState;
  }
  // Safety check
  if (modeChanging && (RisingEdgeCount >= 111 || FallingEdgeCount >= 111)) {
    stepper.stop();
    stepper.setCurrentPosition(stepper.currentPosition());
    setMotorSleep(true); // Put motor to sleep
    currentPosition = stepper.currentPosition();
    resetOpticalEncoderCounter();
    modeChanging = false;
    vTaskDelay(pdMS_TO_TICKS(100));
    continue;
  }
  // Normal operation
  if (!modeChanging) {
    if (switchState == LOW) { // NC mode
      if (progress > lastProgress) {
        setMotorSleep(false);
        currentPosition += STEPS_PER_PERCENT;
        stepper.moveTo(currentPosition);
      } else if (progress < lastProgress) {
        setMotorSleep(false);
        currentPosition -= STEPS_PER_PERCENT;
        stepper.moveTo(currentPosition);
      }
    } else { // NO mode
      if (progress > lastProgress) {
        setMotorSleep(false);
        currentPosition -= STEPS_PER_PERCENT;
        stepper.moveTo(currentPosition);
      } else if (progress < lastProgress) {
        setMotorSleep(false);
        currentPosition += STEPS_PER_PERCENT;
        stepper.moveTo(currentPosition);
      }
    }
  }
  // Run motor
  if (stepper.distanceToGo() != 0) {
    stepper.run();
  } else {
    setMotorSleep(true); // Put motor to sleep when not moving
    if (modeChanging) {
      modeChanging = false;
    }
  }
  lastProgress = progress;
  vTaskDelay(pdMS_TO_TICKS(1));
}

```

```

}
}
void valveTask(void * parameter) {
  while(1) {
    // Rotary Encoder Handling
    int newPosition = rotaryEncoder.readEncoder() / 2;
    // Periksa apakah posisi rotary encoder telah berubah
    if (newPosition != lastPosition) {
      if (newPosition > lastPosition && progress < 100) {
        progress++; // Tambah progress
      } else if (newPosition < lastPosition && progress > 0) {
        progress--; // Kurangi progress
      }
      lastPosition = newPosition; // Update posisi terakhir
      // Hitung panjang progress bar dalam piksel
      progressPixels = map(progress, 0, 100, 0, 83); // 0-100% mapped to 0-86 pixels
    }
    vTaskDelay(pdMS_TO_TICKS(10));
  }
}
void monitorTask(void * parameter) {
  while(1) {
    measureValues();
    vTaskDelay(pdMS_TO_TICKS(1000));
  }
}
void displayswitchTask(void * parameter) {
  while(1) {
    // Handle displayswitch (valvestatus / monitoring) dari Rotary Encoder
    if (rotaryEncoder.isEncoderButtonClicked()) {
      displayValve = !displayValve; // Ubah mode saat tombol rotary encoder ditekan
    }
    // Tampilkan informasi di OLED
    display.clearDisplay();
    if (displayValve) {
      displayStatusValve();
    } else {
      displayMonitoring();
    }
    display.display();
    vTaskDelay(pdMS_TO_TICKS(10));
  }
}
void measureValues() {
  busvoltage = ina219.getBusVoltage_V();
  current_mA = ina219.getCurrent_mA();
  power_mW = ina219.getPower_mW();
  sensors.requestTemperatures();
  temperatureC = sensors.getTempCByIndex(0) + correctionFactor;
  overcurrent = (abs(current_mA) >= 1200); // Proteksi jika >= 1.2A
  overtemp = (temperatureC >= 100.0); // Proteksi jika temperatur >= 100C
}
void opticalEncoderTask(void * parameter) {
  int currentEncoderState = LOW;
  int lastLocalEncoderState = LOW;
  while(1) {
    currentEncoderState = digitalRead(ENCODER_PIN);
    // Deteksi rising edge (LOW ke HIGH)
    if (lastLocalEncoderState == LOW && currentEncoderState == HIGH) {
      RisingEdgeCount++;
    }
    // Deteksi falling edge (HIGH ke LOW)
    if (lastLocalEncoderState == HIGH && currentEncoderState == LOW) {
      FallingEdgeCount++;
    }
    lastLocalEncoderState = currentEncoderState;
    vTaskDelay(pdMS_TO_TICKS(10));
  }
}

```

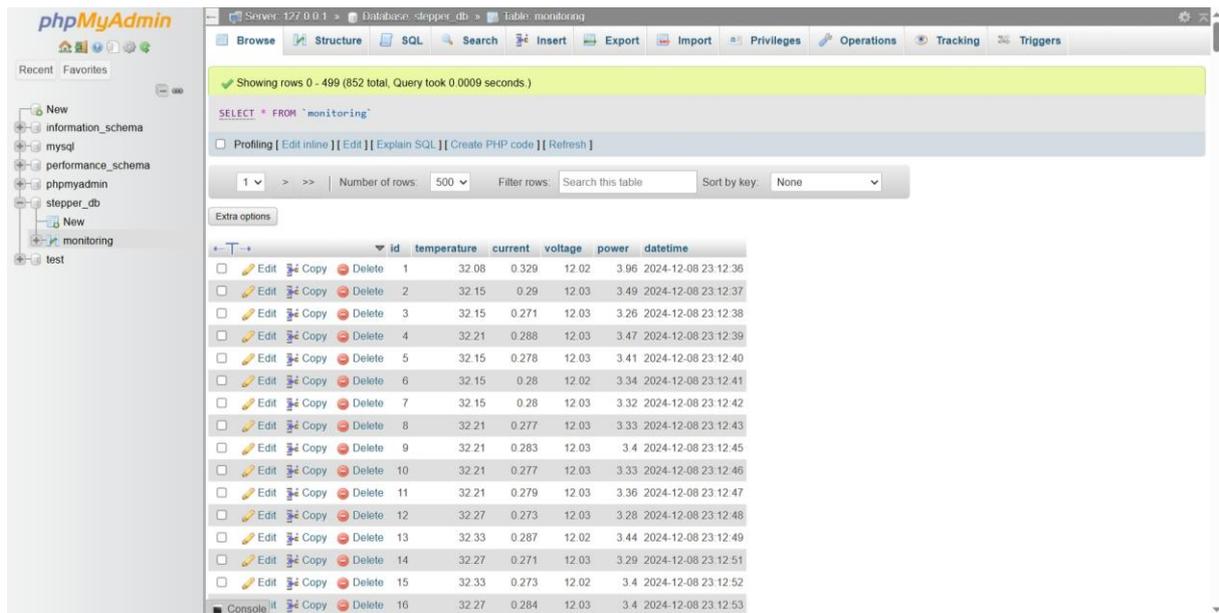
```

}
}
void displayStatusValve(){
// Tampilkan informasi di OLED
display.clearDisplay();
//Box Utama
display.drawRect(1, 1, 126, 62, 1);
display.setTextWrap(false);
//Box Status Valve
display.drawRoundRect(4, 3, 120, 28, 1, 1);
display.setCursor(10, 6);
display.print("Valve Opening :");
display.setCursor(98, 18);
display.print(progress);
display.print("%");
//ProgressBar
display.drawRect(9, 15, 87, 13, 1);//outline
display.fillRect(11, 17, progressPixels, 9, 1);//Bar progress
//Box Operating mode
display.drawRoundRect(4, 33, 120, 28, 1, 1);
//Status Operating Mode : Normally Open (NO)/ Normally Close (NC)
display.setCursor(10, 38);
display.print("Operating Mode :");
display.setCursor(10, 49);
switchState = digitalRead(togglePin);
// Tampilkan status toggle switch (NO atau NC)
if (switchState == HIGH) {
display.print("Normally Open");
} else {
display.print("Normally Close");
}
}
display.display(); // Update OLED display
}
void displayMonitoring() {
display.clearDisplay();
display.drawRect(1, 1, 126, 62, 1);
display.setTextWrap(false);
display.drawRect(12, 12, 54, 40, 1);
// Voltage, Current, Power, Temperature display
display.setCursor(15, 24);
display.print("I:");
display.setCursor(15, 15);
display.print("V:");
display.setCursor(15, 33);
display.print("P:");
display.setCursor(15, 42);
display.print("T:");
display.setCursor(27, 15);
display.print(busvoltage, 2);
float current_A = abs(current_mA) / 1000.0;
display.setCursor(27, 24);
display.print(current_A, 3);
display.setCursor(27, 33);
display.print(power_mW / 1000.0, 2);
display.setCursor(27, 42);
display.print(temperatureC, 2);
display.drawRect(1, 1, 126, 62, 1);
// Units
display.setCursor(58, 15);
display.print("V");
display.setCursor(58, 24);
display.print("A");
display.setCursor(58, 33);
display.print("W");
display.setCursor(58, 42);
display.print("C");
// Optical Rotary Encoder display

```

```
display.drawRect(69, 12, 47, 40, 1);
display.setCursor(71, 15);
display.print("WiFi:");
display.setCursor(71, 24);
display.print("Offline");
display.setCursor(72, 33);
display.print("Re:");
display.setCursor(95, 33);
display.print(RisingEdgeCount);
display.setCursor(72, 42);
display.print("Fe:");
display.setCursor(95, 42);
display.print(FallingEdgeCount);
display.display();
}
```

## Lampiran 4 Database lokal pengujian driver



Server: 127.0.0.1 » Database: stepper\_db » Table: monitoring

Showing rows 0 - 499 (852 total, Query took 0.0009 seconds.)

```
SELECT * FROM `monitoring`
```

Number of rows: 500 Filter rows: Search this table Sort by key: None

	id	temperature	current	voltage	power	datetime
<input type="checkbox"/>	1	32.08	0.329	12.02	3.96	2024-12-08 23:12:36
<input type="checkbox"/>	2	32.15	0.29	12.03	3.49	2024-12-08 23:12:37
<input type="checkbox"/>	3	32.15	0.271	12.03	3.26	2024-12-08 23:12:38
<input type="checkbox"/>	4	32.21	0.288	12.03	3.47	2024-12-08 23:12:39
<input type="checkbox"/>	5	32.15	0.278	12.03	3.41	2024-12-08 23:12:40
<input type="checkbox"/>	6	32.15	0.28	12.02	3.34	2024-12-08 23:12:41
<input type="checkbox"/>	7	32.15	0.28	12.03	3.32	2024-12-08 23:12:42
<input type="checkbox"/>	8	32.21	0.277	12.03	3.33	2024-12-08 23:12:43
<input type="checkbox"/>	9	32.21	0.283	12.03	3.4	2024-12-08 23:12:45
<input type="checkbox"/>	10	32.21	0.277	12.03	3.33	2024-12-08 23:12:46
<input type="checkbox"/>	11	32.21	0.279	12.03	3.36	2024-12-08 23:12:47
<input type="checkbox"/>	12	32.27	0.273	12.03	3.28	2024-12-08 23:12:48
<input type="checkbox"/>	13	32.33	0.287	12.02	3.44	2024-12-08 23:12:49
<input type="checkbox"/>	14	32.27	0.271	12.03	3.29	2024-12-08 23:12:51
<input type="checkbox"/>	15	32.33	0.273	12.02	3.4	2024-12-08 23:12:52
<input type="checkbox"/>	16	32.27	0.284	12.03	3.4	2024-12-08 23:12:53

**Lampiran 5** Tabel perubahan *flowrate* dan *pressure drop* setiap 1% bukaan *valve*

<i>Valve (%)</i>	<i>Flowrate (L/min)</i>	P2 (PSI)	P3 (PSI)	<i>Pressure drop (PSI)</i>
1	0,036	19,2086755	2,75053187	16,4581434
2	0,331	23,1131577	3,05217053	20,0609871
3	0,943	21,3428516	3,08849937	18,2543521
4	2,986	22,9483112	2,88155448	20,0667565
5	4,544	19,7881194	2,64617008	17,1419491
6	7,322	23,2094025	2,65915906	20,5502433
7	7,801	20,9608007	2,63834368	18,322457
8	10,041	20,8628551	2,69448057	18,1683745
9	10,961	18,0458912	2,64909214	15,3967993
10	12,339	18,250138	2,67532208	15,5748159
11	12,788	17,3568663	2,68339422	14,6734721
12	13,459	16,7826996	2,71046721	14,0722324
13	13,664	16,7947511	2,70728682	14,0874643
14	13,257	15,8847884	2,76064407	13,1241442
15	13,267	15,3587214	2,73040313	12,6283182

**Lampiran 6** Tabel aliran dan *pressure drop* dari bukaan *valve* penuh

<i>Valve</i> (%)	<i>Flowrate</i> (L/min)	P2 (PSI)	P3 (PSI)	<i>Pressure drop</i> (PSI)
5	5,26059	24,869	3,59373	21,2753
10	11,8983	22,2005	3,63989	18,5606
15	15,7324	18,63	3,5163	15,1137
20	19,8438	14,9453	3,56588	11,3794
25	22,8707	12,329	3,62733	8,70163
30	24,3438	10,8205	3,65387	7,16668
35	26,0733	9,74318	3,72983	6,01335
40	26,2204	9,1717	3,72609	5,44561
45	26,6272	8,46387	3,70334	4,76054
50	27,006	7,99125	3,70345	4,2878
55	27,1907	7,77377	3,66736	4,10641
60	27,2379	7,56717	3,6765	3,89067
65	27,2476	7,29339	3,64124	3,65214
70	27,2698	7,16042	3,64778	3,51264
75	27,3981	7,10913	3,66608	3,44305
80	27,7183	7,07674	3,65788	3,41886
85	27,5334	6,97669	3,66643	3,31026
90	27,0132	6,81493	3,61535	3,19958
95	27,5454	6,94724	3,75742	3,18982
100	26,8558	6,8141	3,76667	3,04743

## BIODATA PENULIS



Rizal Fiqri Muttaqin lahir di Bekasi pada tanggal 3 Juni 2003. Penulis berasal dari Bekasi, Jawa Barat, dan merupakan mahasiswa Departemen Teknik Instrumentasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan NRP 2042211120. Penulis memulai studi pada tahun 2021 dan menyelesaikan tugas akhir dengan judul "*Pengembangan Final Control Element untuk Control Valve Berbasis Stepper Motor dan Globe Valve*". Sejak awal perkuliahan, penulis memiliki minat besar pada bidang sistem kontrol, instrumentasi, elektrikal, dan elektronika. Ketertarikan ini juga tercermin dari aktivitas pribadi penulis yang gemar membongkar, menganalisis, dan memperbaiki perangkat komputer, khususnya *Graphics Processing Unit* dan *motherboard*. Pengalaman tersebut dikembangkan menjadi kegiatan sampingan berupa perbaikan dan penjualan kembali komponen komputer sebagai bentuk pengembangan keahlian teknis sekaligus usaha mandiri. Penulis dapat dihubungi melalui email: [rizalfiqri03@gmail.com](mailto:rizalfiqri03@gmail.com).