

ANALISIS PENGATURAN ARAH PUTARAN CW DAN CCW MOTOR STEPPER NEMA 23 DENGAN SISTEM MANUAL DAN DIGITAL

Rian Andriana¹, Mochammad Mulia²

^{1,2}*Sekolah Tinggi Teknologi Duta Bangsa*

^{1,2}*Jl. Kalibaru Timur Kel. Kalibaru Medan Satria Kota Bekasi*

edelweis99@gmail.com¹, mohammad.mulia@sttdb.ac.id²

Abstrak— Motor stepper NEMA 23 merupakan salah satu jenis motor yang banyak digunakan dalam aplikasi kontrol presisi, seperti pada sistem otomasi dan robotika. Penelitian ini bertujuan untuk mengembangkan sistem pengaturan arah dan kecepatan motor stepper menggunakan kontrol analog dan digital, serta menganalisis keefektifan kedua metode tersebut. Pada kontrol analog, pengaturan arah dilakukan melalui saklar toggle yang secara manual mengubah logika sinyal pada pin DIR+ driver DM542. Sementara itu, pada kontrol digital, arah putaran motor diatur secara otomatis melalui program Arduino menggunakan logika HIGH dan LOW pada pin yang sama. Kecepatan motor diatur melalui frekuensi pulsa pada pin PUL+, yang diprogram untuk menghasilkan langkah motor yang stabil. Hasil penelitian menunjukkan bahwa kontrol digital memiliki tingkat presisi dan stabilitas yang lebih tinggi dibandingkan kontrol analog, terutama pada kecepatan tinggi (300 RPM). Pada kecepatan rendah hingga sedang (100–200 RPM), kedua metode memberikan performa yang relatif stabil, dengan kontrol analog lebih sederhana untuk diimplementasikan namun kurang efisien untuk aplikasi otomatisasi. Selain itu, integrasi layar LCD I2C digunakan untuk menampilkan status motor dan tombol, memberikan informasi real-time kepada pengguna. Penelitian ini membuktikan bahwa kontrol digital lebih unggul dalam aplikasi yang memerlukan presisi tinggi dan otomatisasi, sedangkan kontrol analog lebih cocok untuk pengujian atau pembelajaran dasar motor stepper. Sistem yang dirancang mampu memberikan pengaturan yang fleksibel, dengan potensi pengembangan lebih lanjut untuk aplikasi di bidang industri dan Pendidikan

Keywords—Motor stepper, NEMA 23, DM542, kontrol analog, kontrol digital, Arduino, LCD I2C.

I. PENDAHULUAN

1.1 Latar Belakang

Motor stepper adalah salah satu jenis motor listrik yang sering digunakan dalam aplikasi yang memerlukan kontrol posisi yang presisi, seperti pada mesin CNC, printer 3D, dan robotika. Salah satu keunggulan motor stepper adalah

kemampuannya untuk berputar dalam langkah-langkah yang tetap dan dapat dikendalikan, sehingga memungkinkan pengaturan posisi dan kecepatan yang sangat akurat. Motor stepper NEMA adalah salah satu jenis motor stepper yang paling populer dan banyak digunakan di berbagai aplikasi industri dan hobi. Dalam banyak kasus, motor stepper NEMA dioperasikan dengan pengontrol yang memungkinkan pengaturan arah putaran baik searah jarum jam (Clockwise/CW) maupun berlawanan arah jarum jam (Counterclockwise/CCW). Dalam penelitian ini, akan dilakukan analisis pengaturan arah putaran CW dan CCW pada motor stepper NEMA menggunakan potensiometer dan sistem kontrol digital. Potensiometer, sebagai komponen analog, sering digunakan untuk mengatur parameter seperti kecepatan atau posisi dengan memberikan sinyal tegangan yang dapat diolah oleh mikrokontroler atau rangkaian elektronik lainnya. Sementara itu, kontrol digital memungkinkan pengaturan yang lebih presisi dan fleksibel melalui pemrograman yang dapat disesuaikan dengan kebutuhan spesifik aplikasi. Pendekatan ini bertujuan untuk memahami bagaimana integrasi antara kontrol analog melalui potensiometer dan kontrol digital dapat digunakan untuk mengatur arah putaran motor stepper secara efektif. Penelitian ini juga akan mengeksplorasi manfaat dan tantangan dalam pengaturan arah putaran motor stepper NEMA menggunakan metode tersebut, serta bagaimana kombinasi dari keduanya dapat meningkatkan performa sistem secara keseluruhan.

1.2 Rumusan Masalah

1. Bagaimana cara mengatur arah putaran motor stepper NEMA menggunakan potensiometer untuk menghasilkan rotasi searah jarum jam (CW) dan berlawanan arah jarum jam (CCW)?
2. Bagaimana pengaruh variasi nilai potensiometer dan kontrol digital terhadap kecepatan dan stabilitas putaran motor stepper NEMA saat beralih antara arah CW dan CCW?
3. Bagaimana besar Arus , Tegangan , Kecepatan , Putaran , Tahanan dan frekuensi pada pengaturan motor stepper NEMA 23 sistem manual dan digital .

1.3 Batasan Masalah

Dalam penelitian ini, beberapa batasan masalah telah ditetapkan untuk menjaga fokus dan ruang lingkup

penelitian tetap terkendali, yaitu:

1. Jenis Motor Stepper: Penelitian dibatasi pada motor stepper NEMA 23, tanpa mempertimbangkan tipe motor stepper lainnya.
2. Sistem Pengaturan: Pengaturan arah putaran CW dan CCW dilakukan menggunakan potensiometer sebagai input analog dan kontrol digital berbasis mikrokontroler (seperti Arduino). Metode kontrol lain tidak dibahas.
3. Beban dan Kondisi Operasional: Fokus pada beban ringan hingga sedang, tanpa mempertimbangkan kondisi operasional ekstrem atau penggunaan jangka panjang.
4. Lingkup Pengujian: Pengujian dilakukan untuk menganalisis kinerja, stabilitas, dan akurasi arah putaran motor. Aspek seperti emisi elektromagnetik dan interferensi tidak dibahas.
5. Sumber Daya: Hanya menggunakan komponen standar yang tersedia secara komersial, tanpa modifikasi khusus pada perangkat keras atau perangkat lunak.
6. Implementasi Nyata: Penelitian dilakukan di lingkungan laboratorium, dengan aplikasi di dunia nyata dianggap sebagai potensi pengembangan lebih lanjut.

1.4 Tujuan

- 1 Merancang trainer motor stepper NEMA 23 yang dapat memfasilitasi pembelajaran konsep pengendalian motor stepper?
- 2 Menyusun modul praktek pembelajaran praktikum pada trainer motor stepper NEMA 23 .
- 3 Menguji komponen dan fitur yang perlu dimasukkan dalam trainer untuk memastikan bahwa alat ini dapat digunakan secara optimal dalam konteks pendidikan teknik.

1.5 Manfaat

- 1 Meningkatkan pemahaman dan keterampilan praktis dalam pengendalian motor stepper NEMA 23.
- 2 Menyediakan alat bantu pembelajaran yang mendukung pembelajaran mandiri.
- 3 Menyediakan alat pembelajaran yang efektif dan mendukung pengajaran konsep motor stepper.
- 4 Meningkatkan kualitas pengajaran dengan mengintegrasikan teori dan praktik.
- 5 Menghasilkan sumber daya manusia yang terlatih dalam pengoperasian motor stepper. Memperkuat hubungan antara pendidikan dan industri melalui keterampilan yang relevan.
- 6 Berkontribusi pada pengembangan media pembelajaran berbasis teknologi. Menyediakan model rancang bangun yang dapat diadaptasi untuk

aplikasi pendidikan lainnya.

II. LANDASAN TEORI

2.1 Arduino UNO



Gambar 2.1 Arduino UNO

Arduino Uno adalah papan mikrokontroler berbasis mikrokontroler 8-bit ATmega328P. Bersama dengan ATmega328P, terdiri dari komponen lain seperti osilator kristal, komunikasi serial, pengatur tegangan, dll untuk mendukung mikrokontroler. Arduino Uno memiliki 14 pin input/output digital (6 di antaranya dapat digunakan sebagai output PWM), 6 pin input analog, koneksi USB, soket DC, header ICSP dan tombol reset.

2.2 Arduino IDE

Arduino IDE adalah software yang digunakan untuk membuat sketch pemrograman atau dengan kata lain arduino IDE sebagai media untuk pemrograman pada board yang ingin diprogram. Arduino IDE ini berguna untuk mengedit, membuat, meng-upload ke board yang ditentukan, dan meng-coding program tertentu. Arduino IDE dibuat dari bahasa pemrograman JAVA, yang dilengkapi dengan library C/C++(wiring), yang membuat operasi input/output lebih mudah. Mengunduh software Arduino IDE dapat dilakukan pada laman web Arduino melalui link berikut <https://www.arduino.cc/en/software>. Pada laman web tersebut dapat diunduh software Arduino IDE yang sesuai dengan sistem operasi yang digunakan. Setelah mengunduh software yang tersedia pada laman tersebut langkah selanjutnya adalah melakukan proses instalasi pada komputer. Seperti teks editor pada umumnya yaitu memiliki fitur untuk cut/ paste dan untuk find /replace teks. Pada bagian keterangan aplikasi memberikan pesan balik saat menyimpan dan mengeksport dan juga sebagai tempat menampilkan kesalahan. Konsol log menampilkan output teks dari Arduino Software (IDE), termasuk pesan kesalahan yang lengkap dan informasi lainnya. Pojok kanan bawah jendela menampilkan papan dikonfigurasi dan port serial. Tombol toolbar memungkinkan Anda untuk memverifikasi dan meng-upload program, membuat, membuka, dan menyimpan sketch, dan membuka monitor serial.



Gambar 2.2 Arduino IDE

2.3 LCD 16x2 I2C

LCD atau *Liquid Crystal Display* adalah suatu jenis media display (tampilan) yang menggunakan kristal cair (*liquid crystal*) untuk menghasilkan gambar yang terlihat. Teknologi Liquid Crystal Display (LCD) atau penampil kristal cair sudah banyak digunakan pada produk-produk seperti layar laptop, layar ponsel, layar kalkulator, layar jam digital, layar multimeter, monitor komputer, televisi, layar game portabel, layar termometer digital dan produk-produk elektronik lainnya. LCD 16x2 merupakan komponen elektronika yang berfungsi untuk menampilkan suatu data dapat berupa karakter, huruf, simbol maupun grafik dengan jumlah kolom 16 karakter dan mempunyai 2 baris. Karena ukurannya yang kecil maka LCD banyak dipasang dengan Mikrokontroler. LCD tersedia dalam bentuk modul yang mempunyai pin data, kontrol catu daya, dan pengatur kontras.



Gambar 2.3 LCD 16x2 Pinout

2.4 Potensiometer

Potensiometer adalah salah satu jenis resistor yang nilai resistansinya dapat diatur sesuai dengan kebutuhan rangkaian elektronika ataupun kebutuhan pemakainya. Potensiometer merupakan keluarga resistor yang tergolong dalam kategori Variable Resistor (VR). Secara struktur, Potensiometer terdiri dari 3 kaki Terminal dengan sebuah shaft atau tuas yang berfungsi sebagai pengaturnya



Gambar 2.4 Potensiometer

2.5 Sakelar toggle

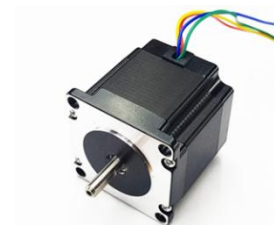
Sakelar toggle adalah sakelar mekanis yang biasa digunakan untuk mengontrol aliran listrik pada perangkat elektronik. Hal ini ditandai dengan adanya pegangan atau tuas yang dioperasikan secara manual yang mengontrol aliran arus listrik dari catu daya ke suatu perangkat (seperti peralatan rumah tangga). Ini terdiri dari tuas atau pegangan yang dapat digerakkan maju mundur antara dua posisi, oleh karena itu dinamakan “toggle”. Sakelar memiliki tiga komponen utama: tuas, kontak, dan rumah.



Gambar 2.5 Sakelar Toggle

2.6 Motor stepper NEMA 23

Motor stepper NEMA 23 adalah jenis motor stepper yang umum. Motor seperti ini biasanya digerakkan oleh dua atau empat fasa, dengan masing-masing fasa digerakkan oleh catu daya dan pengontrol. Motor stepper adalah motor presisi dengan sudut putaran diskrit, biasanya 1,8 atau 0,9 derajat. Mereka banyak digunakan dalam aplikasi yang memerlukan penentuan posisi presisi tinggi, kontrol kecepatan, dan kontrol orientasi, seperti pencetakan, tekstil, peralatan medis, dan robotika.



Gambar 2.6 Motor NEMA23

2.7 Driver Motor Stepper DM542

DM542 adalah penggerak stepper digital sepenuhnya yang dikembangkan dengan algoritma kontrol DSP canggih berdasarkan teknologi kontrol gerak terbaru. Ini telah mencapai tingkat kehalusan sistem yang unik, memberikan torsi optimal dan menghilangkan ketidakstabilan jarak menengah. Fitur identifikasi otomatis motor dan konfigurasi otomatis parameternya menawarkan pengaturan cepat ke mode optimal dengan motor berbeda. Dibandingkan dengan penggerak analog tradisional, DM542 dapat menggerakkan motor stepper dengan kebisingan yang jauh lebih rendah, pemanasan yang lebih rendah, dan pergerakan yang lebih halus. Fitur uniknya menjadikan DM542 pilihan ideal untuk aplikasi dengan kebutuhan tinggi.



Gambar 2.7 Driver Motor Stepper DM542

Ini dapat digunakan dalam berbagai jenis mesin, seperti mesin dispenser, printer mesin terapan 3D, pelabelan, mesin, mesin medis, mesin potong laser, dan sebagainya, fitur uniknya menjadikannya solusi ideal untuk aplikasi yang membutuhkan kebisingan rendah, getaran rendah, panas rendah dan presisi tinggi. Fitur Produk:

- 1 DSP 32bit generasi baru
- 2 kehalusan yang baik, kebisingan rendah, panas rendah
- 3 pengaturan langkah mikro build-in
- 4 pengaturan saat ini build-in
- 5 arus macet dapat diatur ke 50%
- 6 input denyut nadi diferensial opto-isolated
- 7 frekuensi respons hingga 300KHz
- 8 tegangan atas, di bawah tegangan, atas perlindungan saat ini

2.8 Power Supply

Power supply adalah perangkat elektronik yang berfungsi untuk menyediakan daya listrik ke berbagai

perangkat lain dengan mengubah energi listrik dari sumber, seperti listrik AC (arus bolak-balik), menjadi tegangan dan arus yang sesuai dengan kebutuhan perangkat. Salah satu jenis power supply yang umum digunakan adalah Switched Mode Power Supply (SMPS), yang bekerja dengan prinsip switching frekuensi tinggi untuk menghasilkan daya yang efisien dan stabil. Power supply jenis ini lebih unggul dibandingkan dengan power supply linier karena ukurannya yang lebih kecil, efisiensinya yang lebih tinggi, dan kemampuannya untuk menyediakan daya lebih besar. Prinsip kerja SMPS dimulai dengan rectifikasi dan penyaringan arus AC menjadi DC menggunakan dioda rectifier dan kapasitor untuk mengurangi ripple. Selanjutnya, arus DC tersebut diubah menjadi frekuensi tinggi melalui transistor switching, yang memungkinkan penggunaan transformator kecil untuk menaikkan atau menurunkan tegangan sesuai kebutuhan. Tegangan ini kemudian direktifikasi kembali untuk menghasilkan output DC yang stabil. Power supply jenis ini sering digunakan dalam perangkat elektronik modern seperti komputer, motor stepper, dan sistem otomatisasi lainnya, karena kemampuannya menyediakan daya stabil yang sangat penting bagi operasi perangkat tersebut



Gambar 2.8 Power Supply

2.9 Pulse Width Modulation

1. Pengertian PWM (Pulse Width Modulation)

PWM adalah kepanjangan dari Pulse Width Modulation atau dalam bahasa Indonesia dapat diterjemahkan menjadi Modulasi Lebar Pulsa. Jadi pada dasarnya, PWM adalah suatu teknik modulasi yang mengubah lebar pulsa (pulse width) dengan nilai frekuensi dan amplitudo yang tetap. PWM dapat dianggap sebagai kebalikan dari ADC (Analog to Digital Converter) yang mengkonversi sinyal Analog ke Digital, PWM atau Pulse Width Modulation ini digunakan menghasilkan sinyal analog dari perangkat Digital (contohnya dari Mikrokontroler). Untuk lebih memahami apa yang dimaksud dengan PWM atau Pulse Width Modulation ini. Kita coba melihat contoh dari sinyal yang dihasilkan oleh Mikrokontroler atau IC 555. Sinyal yang dihasilkan oleh Mikrokontroler atau IC555 ini adalah sinyal pulsa yang

umumnya berbentuk gelombang segiempat. Gelombang yang dihasilkan ini akan tinggi atau rendah pada waktu tertentu. Misalnya gelombang tinggi di 5V dan paling rendah di 0V. Durasi atau lamanya waktu dimana sinyal tetap berada di posisi tinggi disebut dengan “ON Time” atau “Waktu ON” sedangkan sinyal tetap berada di posisi rendah atau 0V disebut dengan “OFF Time” atau “Waktu OFF”. Untuk sinyal PWM, kita perlu melihat dua parameter penting yang terkait dengannya yaitu Siklus Kerja PWM (PWM Duty Cycle) dan Frekuensi PWM (PWM Frequency).

2. Siklus Kerja PWM (PWM Duty Cycle)

Seperti yang disebutkan diatas, Sinyal PWM akan tetap ON untuk waktu tertentu dan kemudian terhenti atau OFF selama sisa periodenya. Yang membuat PWM ini istimewa dan lebih bermanfaat adalah kita dapat menetapkan berapa lama kondisi ON harus bertahan dengan cara mengendalikan siklus kerja atau Duty Cycle PWM. Persentase waktu di mana sinyal PWM tetap pada kondisi TINGGI (ON Time) disebut dengan “siklus kerja” atau “Duty Cycle”. Kondisi yang sinyalnya selalu dalam kondisi ON disebut sebagai 100% Duty Cycle (Siklus Kerja 100%), sedangkan kondisi yang sinyalnya selalu dalam kondisi OFF (mati) disebut dengan 0% Duty Cycle (Siklus Kerja 0%).

Rumus untuk menghitung siklus kerja atau duty cycle dapat ditunjukkan seperti persamaan di bawah ini.

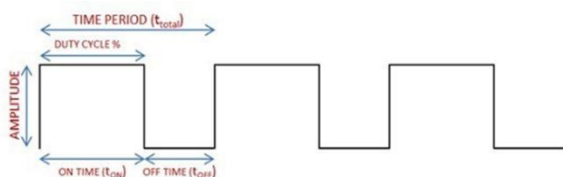
$$\text{Duty Cycle} = t_{\text{ON}} / (t_{\text{ON}} + t_{\text{OFF}}) \text{ Atau}$$

$$\text{Duty Cycle} = t_{\text{ON}} / t_{\text{total}} \text{ Dimana :}$$

t_{ON} = Waktu ON atau Waktu dimana tegangan keluaran berada pada posisi tinggi (high atau 1) t_{OFF} = Waktu OFF atau Waktu dimana tegangan keluaran berada pada posisi rendah (low atau 0) t_{total} = Waktu satu siklus atau penjumlahan antara t_{ON} dengan t_{OFF} atau disebut juga dengan “periode satu gelombang”

$$\text{Siklus Kerja} = \text{Waktu ON} / (\text{Waktu ON} + \text{Waktu OFF})$$

Gambar berikut ini mewakili sinyal PWM dengan siklus kerja 60%. Seperti yang kita lihat, dengan mempertimbangkan seluruh periode waktu (ON time + OFF time), sinyal PWM hanya ON untuk 60% dari suatu periode waktu.



3. Frekuensi PWM (PWM Frequency)

Frekuensi sinyal PWM menentukan seberapa cepat PWM menyelesaikan satu periode. Satu Periode adalah waktu ON dan OFF penuh dari sinyal PWM seperti yang ditunjukkan pada gambar di atas. Berikut ini adalah Rumus untuk menghitung Frekuensi : $\text{Frequency} = 1 / \text{Time Period}$

Keterangan : Time Periode atau Periode Waktu = Waktu ON + Waktu OFF

Biasanya sinyal PWM yang dihasilkan oleh mikrokontroler akan sekitar 500 Hz, frekuensi tinggi tersebut akan digunakan dalam perangkat switching yang berkecepatan tinggi seperti inverter atau konverter. Namun tidak semua aplikasi membutuhkan frekuensi tinggi. Sebagai contoh, untuk mengendalikan motor servo kita hanya perlu menghasilkan sinyal PWM dengan frekuensi 50Hz, frekuensi sinyal PWM ini juga dapat dikendalikan oleh program untuk semua mikrokontroler.

4. Perbedaan antara Siklus Kerja (Duty Cycle) dengan Frekuensi sinyal PWM

Siklus kerja dan frekuensi sinyal PWM sering membingungkan. Seperti yang kita ketahui bahwa sinyal PWM adalah gelombang persegi dengan waktu ON dan waktu OFF. Jumlah dari Waktu ON (ON-Time) dan Waktu OFF (OFF-Time) ini disebut sebagai satu periode waktu. Kebalikan dari satu periode waktu disebut frekuensi. Sementara jumlah waktu sinyal PWM harus tetap dalam satu periode waktu ditentukan oleh siklus kerja PWM. Sederhananya, seberapa cepat sinyal PWM harus dihidupkan (ON) dan dimatikan (OFF) ditentukan oleh frekuensi sinyal PWM dan kecepatan berapa lama sinyal PWM harus tetap ON (hidup) ditentukan oleh siklus kerja sinyal PWM.

5. Cara menghitung tegangan output sinyal PWM

Tegangan output sinyal PWM yang telah diubah menjadi analog akan menjadi persentase dari siklus kerja (Duty Cycle). Misalnya jika tegangan operasi 5V maka sinyal PWM juga akan memiliki 5V ketika tinggi. Apabila Duty Cycle atau siklus kerja adalah 100%, maka tegangan output akan menjadi 5V. Sedangkan untuk siklus kerja 50% akan menjadi 2.5V.

Demikian juga apabila siklus kerja 60% maka Tegangan Output analognya akan menjadi 3V.

Rumus perhitungan tegangan output sinyal PWM ini dapat dilihat seperti persamaan dibawah ini :

$$V_{\text{out}} = \text{Duty Cycle} \times V_{\text{in}}$$

6. Perhitungan PWM

Desain PWM dengan siklus kerja 60% dengan frekuensi 50Hz dan Tegangan Input 5V.

Penyelesaiannya :

Diketahui:

Duty Cycle : 60% Frequency : 50Hz Vin : 5V

Mencari Time Period atau Periode Waktu :

Time Period = $1 / 50\text{Hz}$

Time Period = 0,02 detik atau 20 milidetik

Mencari Waktu ON (ON-Time) dengan siklus kerja 60% (0,6) Duty Cycle = $t_{\text{ON}} / (t_{\text{ON}} + t_{\text{OFF}})$

$0,6 = t_{\text{ON}} / (t_{\text{ON}} + t_{\text{OFF}})$

$0,6 = t_{\text{ON}} / 20 \text{ milidetik}$

$= 0,6 \times 20 \text{ milidetik}$ $t_{\text{ON}} = 12 \text{ milidetik}$

Mencari Waktu OFF (OFF- Time) $t_{\text{OFF}} = t_{\text{total}} - t_{\text{ON}}$

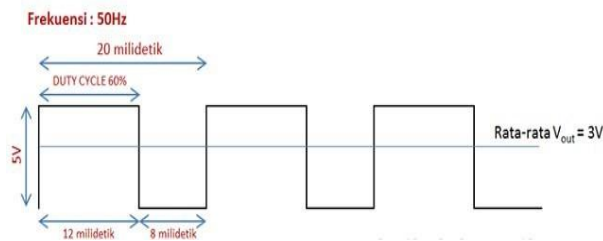
$= 20 - 12$ $t_{\text{OFF}} = 8 \text{ milidetik}$

Mencari Tegangan Output $V_{\text{out}} = \text{Duty Cycle} \times V_{\text{in}}$

$V_{\text{out}} = 60\% \times 5\text{V}$

$V_{\text{out}} = 3\text{V}$

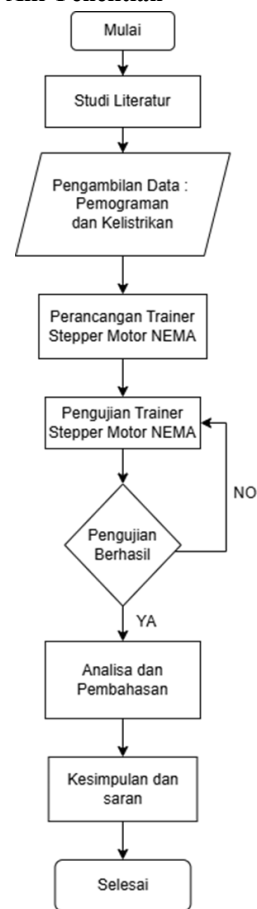
Hasil dari Perhitungan diatas dapat digambarkan menjadi seperti grafik dibawah ini



Gambar 2.9 Power Supply

III . GAMBARAN UMUM / KERANGKA SKRIPSI

3.1 Diagram Alir Penelitian



Gambar 3.1 Digram Alir Penelitian

- 1) Mulai: Penelitian dimulai dengan menetapkan tujuan dan ruang lingkup penelitian.
- 2) Studi Literatur: Peneliti melakukan studi literatur untuk mengumpulkan informasi dari berbagai sumber terkait topik yang akan diteliti, khususnya tentang stepper motor NEMA dan konsep-konsep yang relevan dengan pemrograman dan kelistrikan.
- 3) Pengambilan Data: Pemrograman dan Kelistrikan: Data dikumpulkan melalui kegiatan pemrograman dan pengujian kelistrikan yang berkaitan dengan stepper motor NEMA. Proses ini melibatkan pengambilan data teknis yang dibutuhkan untuk perancangan dan implementasi sistem.
- 4) Perancangan Trainer Stepper Motor NEMA: Berdasarkan data yang telah dikumpulkan, peneliti merancang trainer (alat pelatihan) untuk stepper motor NEMA. Desain ini meliputi aspek mekanik, elektronik, dan pemrograman untuk memastikan bahwa trainer dapat digunakan sesuai kebutuhan.

- 5) Pengujian Trainer Stepper Motor NEMA: Setelah trainer dirancang, tahap selanjutnya adalah melakukan pengujian untuk memastikan bahwa trainer bekerja sesuai dengan spesifikasi yang telah ditetapkan. Pengujian ini dilakukan untuk mengevaluasi kinerja dan keandalan sistem.
- 6) Pengujian Berhasil: Hasil dari pengujian dianalisis untuk menentukan apakah pengujian berhasil atau tidak. Jika pengujian berhasil, penelitian dapat dilanjutkan ke tahap berikutnya. Jika tidak, perlu dilakukan revisi atau perbaikan pada desain trainer.
- 7) Analisa dan Pembahasan: Pada tahap ini, peneliti melakukan analisis mendalam terhadap lima pengujian wiring diagram yang dilakukan selama praktikum. Setiap wiring diagram diuji untuk memastikan kesesuaiannya dengan spesifikasi teknis dan fungsionalitas yang diharapkan. Peneliti membahas hasil yang diperoleh dari masing-masing pengujian, mengidentifikasi potensi masalah, serta memberikan solusi atau perbaikan yang diperlukan. Analisis ini sangat penting untuk memastikan bahwa seluruh sistem bekerja dengan baik dan dapat digunakan secara efektif dalam praktik.
- 8) Kesimpulan dan Saran: Berdasarkan analisis, peneliti menarik kesimpulan dari penelitian ini, termasuk efektivitas dan efisiensi dari wiring diagram yang telah diuji. Saran untuk penelitian lebih lanjut atau perbaikan pada sistem yang ada juga disampaikan.

Selesai: Penelitian diakhiri setelah kesimpulan dan saran diberikan, menandai selesainya seluruh rangkaian proses penelitian

3.2 Perancangan Trainer Motor Stepper

Berikut merupakan bahan dan alat yang diperlukan pada penelitian ini

a. Perancangan Program:

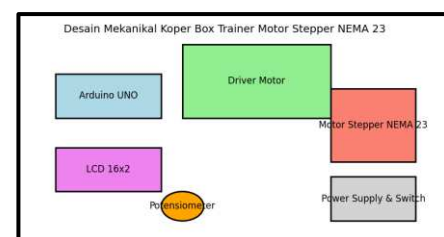


Gambar 3.2 Perancangan Program

Pada tahap ini, peneliti fokus pada pengembangan perangkat lunak yang akan mengendalikan stepper motor NEMA. Program ini harus dirancang untuk mengontrol gerakan motor dengan presisi tinggi, termasuk pengaturan kecepatan, arah, dan jumlah langkah yang diambil oleh motor. Bahasa pemrograman dan platform yang digunakan dipilih berdasarkan kemampuan mereka dalam mendukung komunikasi dengan perangkat keras yang ada. Program ini juga harus dilengkapi dengan fitur-fitur seperti pemantauan real-time dan kemampuan untuk merespons perubahan kondisi atau input dari pengguna. Perancangan program ini akan diuji secara iteratif untuk memastikan fungsionalitas dan keandalannya.

b. Perancangan Mekanikal:

Pada tahap ini, desain mekanikal dari trainer stepper motor NEMA dibuat. Desain ini melibatkan pembuatan struktur fisik yang mendukung motor, termasuk pemilihan bahan, dimensi, dan cara pemasangan motor pada trainer. Selain itu, desain mekanikal juga harus memperhatikan aspek-aspek seperti kestabilan, durabilitas, dan kemampuan untuk menahan beban serta gaya yang dihasilkan oleh motor selama operasi. Semua elemen mekanikal harus dirancang agar kompatibel dengan komponen elektronik dan sistem kontrol yang telah direncanakan, serta memenuhi standar keselamatan dan ergonomi.



Gambar 3.3 Desain Perancangan Mekanikal Trainer Motor Stepper NEMA23

Gambar di atas merupakan desain mekanikal 2D dari koper box untuk trainer motor stepper NEMA 23, yang menampilkan tata letak komponen-komponen utama seperti:

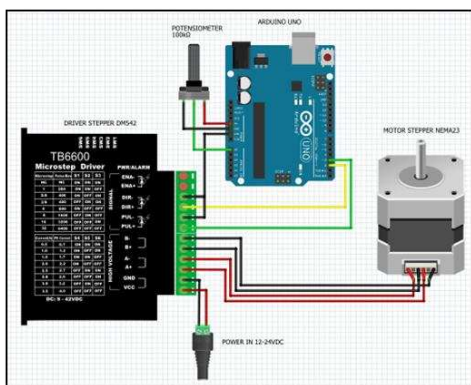
1. Arduino UNO: Terletak di sisi kiri atas.

2. Driver Motor: Terletak di tengah, berfungsi sebagai pengendali motor stepper.
3. Motor Stepper NEMA 23: Terletak di sisi kanan, dihubungkan langsung ke driver motor.
4. LCD 16x2: Diletakkan di sisi kiri bawah, menampilkan informasi selama pengoperasian.
5. Potensiometer: Digunakan untuk mengatur kecepatan motor atau parameter lainnya.
6. Konektor (Con-1 hingga Con-5): Simbolis untuk berbagai konektor yang mungkin ada.
7. Power Supply & Switch: Terletak di kanan bawah untuk memasok daya dan mengontrol sistem.

Desain ini memberikan gambaran umum tata letak komponen di dalam koper box. Jika Anda memerlukan penyesuaian atau penambahan komponen lain, saya siap membantu.

c. Perancangan Kelistrikan:

Tahap ini melibatkan perancangan sistem kelistrikan yang akan menghubungkan motor dengan sumber daya serta komponen kontrol lainnya. Perancangan kelistrikan mencakup pembuatan wiring diagram yang mengatur bagaimana komponen seperti power supply, driver motor, dan kontroler dihubungkan satu sama lain. Sistem kelistrikan juga harus dirancang untuk meminimalkan risiko korsleting, interferensi, dan memastikan distribusi daya yang efisien ke seluruh komponen. Selain itu, langkah-langkah pengamanan, seperti pemasangan fuse atau sirkuit proteksi, juga direncanakan untuk melindungi komponen dari kerusakan akibat kelebihan arus atau tegangan.



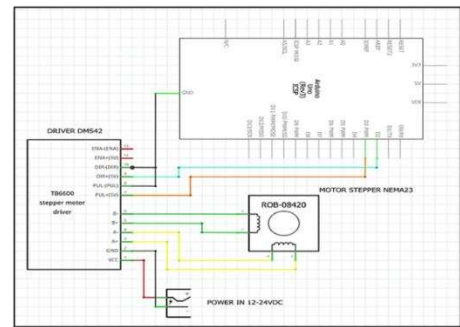
Gambar 3.4 Desain Kelistrikan Trainer Motor Stepper NEMA23

3.3 Pengujian Trainer Stepper Motor NEMA23

Dalam pengujian trainer stepper motor NEMA 23, wiring diagram berfungsi sebagai panduan untuk merangkai komponen elektronik dengan tepat dan aman. Diagram ini menunjukkan hubungan antara microcontroller, driver motor, motor stepper, dan sumber daya, serta jalur koneksi yang diperlukan. Setiap komponen diberi label atau kode

warna untuk memudahkan pemahaman dan menghindari kesalahan. Wiring diagram juga mencakup pengaturan pin microcontroller yang terhubung ke driver, memastikan sinyal kontrol diterjemahkan dengan benar. Selain itu, diagram ini memberikan panduan pengaturan sumber daya yang sesuai, mencegah kerusakan akibat suplai daya yang tidak tepat.

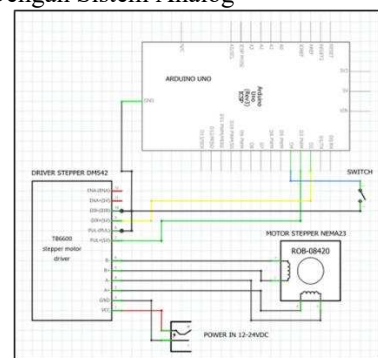
1. Wiring Diagram : Pengaturan Arah Putaran Motor Dengan system Digital



Gambar 3.5 schematic Diagram Sistem Digital

Gambar ini menampilkan percobaan di mana Arduino Uno digunakan untuk mengakses dan mengontrol driver stepper DM542. Dalam percobaan ini, Arduino Uno mengirimkan sinyal STEP dan DIR ke driver untuk mengatur kecepatan dan arah putaran motor stepper NEMA 23. Kode yang diunggah ke Arduino memungkinkan pengguna untuk mengatur parameter gerakan motor secara presisi, menjadikan percobaan ini sebagai langkah penting dalam memahami interaksi antara microcontroller dan driver motor stepper.

2. Wiring Diagram : Pengujian Arah Putaran Motor Stepper Dengan Sistem Analog



Gambar 3.6 schematic Diagram Sistem Analog

Gambar ini menunjukkan percobaan di mana Arduino Uno digunakan untuk mengontrol driver stepper DM542. Pada pengujian ini menggunakan Switch untuk menentukan arah putaran motor stepper dengan menggunakan driver DM542 yang diprogram melalui aplikasi Arduino IDE. Jika posisi Switch berubah, maka arah putaran motor stepper juga akan berubah. Berikut adalah wiringnya

IV . HASIL DAN PEMBAHASAN

4.1 Hasil penelitian

Penelitian ini bertujuan untuk merancang dan mengimplementasikan system monitoring suhu pada mesin holding furnace menggunakan ESP32 yang berbasis web. Sistem ini memanfaatkan termokopel tipe K yang dikombinasikan dengan modul MAX6675 untuk mendeteksi dan mengkonversi suhu, yang kemudian ditampilkan secara real-time di layar OLED 0,96 inci dan disimpan ke dalam basis data berbasis PHP serta MySQL. Memiliki sistem cronjob untuk backup data dan hapus data pada database. Sistem ini juga dilengkapi dengan fitur penyimpanan data sementara di *microSD* card saat koneksi internet terputus.

4.2 Hasil uji coba Alat dan Pembahasan

4.2.1 Hasil Pembuatan Prototipe



Gambar 4.1 Hasil Pembuatan Prototipe

Alat yang telah berhasil dibuat adalah sebuah Trainer Motor Stepper NEMA 23 yang dirancang untuk mendemonstrasikan pengaturan arah dan kecepatan motor stepper secara praktis. Alat ini dilengkapi dengan berbagai komponen utama, termasuk driver DM542 sebagai pengontrol motor stepper, sebuah motor stepper tipe NEMA 23, Arduino Uno sebagai pusat pengolahan data, LCD 16x2 berbasis I2C untuk menampilkan informasi, potensiometer untuk pengaturan kecepatan, serta saklar toggle untuk mengubah arah putaran motor. Sistem ini dirancang dengan tata letak yang terorganisir, memungkinkan pengguna untuk mempelajari dan memahami cara kerja motor stepper melalui kontrol manual dan digital. Transformator disediakan untuk suplai daya AC ke DC, sementara terminal sambungan yang jelas dan terstruktur memudahkan penghubungan antar komponen. Dengan potensiometer, pengguna dapat mengatur kecepatan motor secara bertahap, sedangkan saklar toggle memberikan kontrol arah putaran dengan mengubah sinyal logika pada driver. LCD berfungsi untuk menampilkan informasi seperti kecepatan motor dalam RPM, memberikan umpan balik langsung kepada pengguna. Trainer ini sangat sesuai untuk digunakan dalam pendidikan atau pelatihan, baik di bidang teknik elektro maupun mekatronika, karena

memberikan pengalaman langsung dalam mengendalikan motor stepper. Dengan sistem modular dan fleksibel, alat ini juga memungkinkan penyesuaian atau pengembangan lebih lanjut sesuai kebutuhan pengguna. Secara keseluruhan, pembuatan alat ini berhasil menciptakan sebuah perangkat yang fungsional, praktis, dan dapat digunakan secara luas untuk aplikasi pembelajaran maupun praktikum.

1. Pengujian Pengaturan Arah Putaran Motor Dengan Sistem Analog

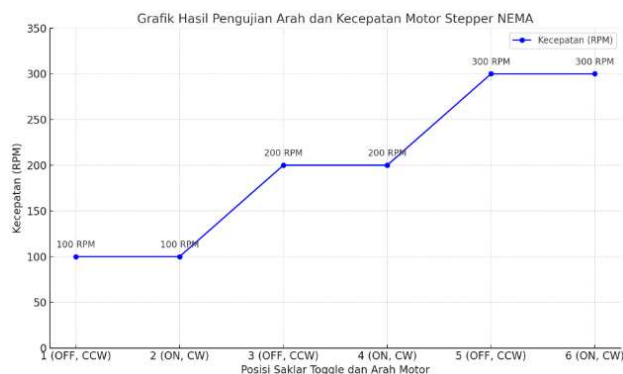
Pada praktikum ini, kita akan menggunakan Switch untuk menentukan arah putaran motor stepper dengan menggunakan driver DM542 yang diprogram melalui aplikasi Arduino IDE. Jika posisi Switch berubah, maka arah putaran motor stepper juga akan berubah. Berikut adalah wiringnya

Tabel 4.1 Hasil Pengujian Pengaturan Arah Putaran Motor Stepper Dengan Sistem Analog

N o.	Kond isi Saklar Toggl e	Status Sinyal DIR+ (Ardui no Pin 4)	Arah Putaran Motor	Kecepatan (RPM)	Catatan
1	OFF	LOW	Berlawanan arah jarum jam (CCW)	100	Motor bergerak stabil
2	ON	HIGH	Searah jarum jam (CW)	100	Motor bergerak stabil
3	OFF	LOW	Berlawanan arah jarum jam (CCW)	200	Kecepatan meningkat
4	ON	HIGH	Searah jarum jam (CW)	200	Kecepatan meningkat
5	OFF	LOW	Berlawanan arah jarum jam (CCW)	300	Motor sedikit bergetar pada RPM tinggi
6	ON	HIGH	Searah jarum jam (CW)	300	Motor sedikit bergetar pada

					RPM tinggi
--	--	--	--	--	------------

Hasil pengujian menunjukkan bahwa saklar toggle berhasil mengontrol arah putaran motor stepper sesuai dengan sinyal logika pada pin DIR+. Ketika saklar dalam posisi OFF (LOW), motor berputar berlawanan arah jarum jam (CCW), sedangkan pada posisi ON (HIGH), motor berputar searah jarum jam (CW). Kecepatan motor juga diamati meningkat secara bertahap dari 100 RPM hingga 300 RPM. Pada kecepatan rendah (100 RPM), motor bergerak stabil tanpa getaran, namun pada kecepatan tinggi (300 RPM), motor mulai menunjukkan sedikit getaran, yang dapat diatasi dengan penyesuaian driver atau beban torsi. Secara keseluruhan, pengujian menunjukkan bahwa sistem bekerja dengan baik untuk mengontrol arah dan kecepatan motor stepper.



Gambar 4.2 Grafik Pengaturan Arah Putaran Motor Stepper NEMA23 Dengan Sistem Digital

Hasil pengujian arah putaran motor stepper NEMA menunjukkan bahwa saklar toggle dapat digunakan sebagai pengontrol analog untuk mengatur arah putaran motor. Saklar toggle yang terhubung ke pin DIR+ pada driver DM542 bekerja dengan baik dalam mengubah arah putaran motor. Ketika saklar dalam posisi OFF, sinyal logika pada pin DIR+ adalah LOW, sehingga motor berputar berlawanan arah jarum jam (CCW). Sebaliknya, ketika saklar dalam posisi ON, sinyal logika pada pin DIR+ berubah menjadi HIGH, menyebabkan motor berputar searah jarum jam (CW). Perubahan arah ini sepenuhnya sesuai dengan konfigurasi standar driver DM542, yang mengatur arah rotasi motor berdasarkan sinyal HIGH atau LOW. Selain arah putaran, pengujian juga menunjukkan bahwa kecepatan motor stepper dapat ditingkatkan dengan mengubah frekuensi pulsa yang dikirimkan ke pin PUL+ pada driver. Pada kecepatan rendah (100 RPM), motor bergerak stabil tanpa gangguan, baik dalam arah CW maupun CCW. Ketika kecepatan ditingkatkan ke 200 RPM, motor tetap stabil dengan sedikit peningkatan suara, yang merupakan hal normal pada motor stepper saat bekerja dengan frekuensi pulsa yang lebih

tinggi. Namun, pada kecepatan tinggi (300 RPM), motor mulai menunjukkan sedikit getaran, terutama jika beban pada motor cukup besar atau jika mekanisme torsi tidak seimbang. Hal ini menunjukkan bahwa motor stepper bekerja paling optimal pada kecepatan rendah hingga sedang, sementara pada kecepatan tinggi memerlukan penyesuaian pada driver atau mekanisme pendukung untuk mengurangi getaran.

Kecepatan motor stepper bertambah seiring dengan perubahan parameter pulsa (frekuensi sinyal PUL+), yang diatur melalui nilai delay. Berikut pengamatan berdasarkan tabel:

- Pada RPM **100**, motor bergerak stabil baik untuk arah CW maupun CCW.
- Pada RPM **200**, motor menunjukkan peningkatan kecepatan tanpa adanya masalah atau gangguan.
- Pada RPM **300**, motor mengalami sedikit getaran pada torsi tinggi, yang biasa terjadi pada motor stepper saat mencapai kecepatan tinggi karena inersia.

Kecepatan Putaran motor dihitung berdasarkan frekuensi pulsa yang dikirim ke pin PUL+ melalui Arduino menggunakan rumus. Secara keseluruhan, pengujian ini berhasil menunjukkan bahwa saklar toggle dapat digunakan sebagai kontrol analog sederhana untuk mengatur arah motor stepper. Sistem bekerja dengan baik dalam kondisi operasional normal, di mana kecepatan motor dapat disesuaikan melalui perubahan frekuensi pulsa, sementara arah putaran dapat diubah dengan mudah menggunakan saklar toggle. Motor memberikan performa yang optimal pada RPM rendah hingga sedang, dengan stabilitas yang baik dan hanya sedikit gangguan pada kecepatan tinggi. Observasi ini menegaskan bahwa konfigurasi menggunakan Arduino, saklar toggle, dan driver DM542 efektif untuk mengendalikan motor stepper secara manual.

2. Pengujian Pengaturan Arah Putaran Motor Dengan Sistem Digital

Pada percobaan ini kita akan membuat sebuah rangkaian pemutar motor stepper dengan driver DM542 yang dikendalikan dengan microcontroller Arduino Uno, tentu hal ini lebih kompleks dari sebelumnya karena dengan adanya Arduino Uno kita juga harus melakukan pemrograman untuk mengakses driver DM542. Program yang akan dibuat meliputi jumlah step atau putaran pada motor stepper dan arah putaran motor stepper (cw/ccw).

No	Nilai Digital Sinyal DIR+ (HIGH/LOW)	Arah Putaran Motor	Kecepatan (RPM)	Catatan
1	LOW	Berlawanan arah	100	Motor bergerak stabil

		jarum jam (CCW)		
2	HIGH	Searah jarum jam (CW)	100	Motor bergerak stabil
3	LOW	Berlawanan arah jarum jam (CCW)	200	Kecepatan meningkat
4	HIGH	Searah jarum jam (CW)	200	Kecepatan meningkat
5	LOW	Berlawanan arah jarum jam (CCW)	300	Motor sedikit bergetar pada RPM tinggi
6	HIGH	Searah jarum jam (CW)	300	Motor sedikit bergetar pada RPM tinggi

Pada praktikum ini, kita akan menggunakan Switch untuk menentukan arah putaran motor stepper dengan menggunakan driver DM542 yang diprogram melalui aplikasi Arduino IDE. Jika posisi Switch berubah, maka arah putaran motor stepper juga akan berubah.

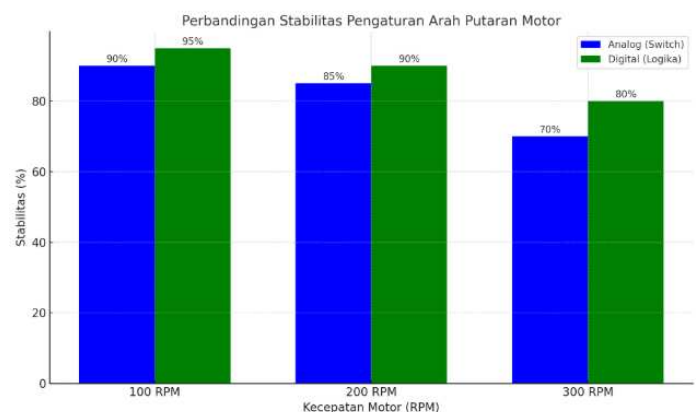


Gambar 4.3 Grafik Pengaturan Arah Putaran Motor Stepper NEMA23 Dengan Sistem Digital

Pengujian pengaturan arah dan kecepatan motor stepper NEMA secara digital menunjukkan bahwa arah putaran motor dapat dikontrol dengan presisi melalui sinyal digital pada pin DIR+

yang dikirimkan dari Arduino ke driver DM542. Ketika sinyal DIR+ berada pada kondisi LOW, motor berputar berlawanan arah jarum jam (CCW), sedangkan ketika sinyal DIR+ berada pada kondisi HIGH, motor berputar searah jarum jam (CW). Pengaturan arah motor ini bekerja secara konsisten sesuai dengan konfigurasi standar driver DM542 yang menggunakan logika digital untuk mengontrol arah motor. Hasil pengujian juga menunjukkan bahwa kecepatan motor dapat diatur dengan mengontrol frekuensi pulsa pada pin PUL+. Kecepatan motor tercatat meningkat dari 100 RPM hingga 300 RPM. Pada kecepatan rendah (100 RPM), motor bergerak stabil tanpa gangguan atau getaran. Ketika kecepatan ditingkatkan ke 200 RPM, motor tetap stabil dengan sedikit peningkatan suara, yang merupakan karakteristik normal motor stepper saat bekerja pada frekuensi lebih tinggi. Namun, pada kecepatan tinggi (300 RPM), motor mulai menunjukkan sedikit getaran, terutama jika beban pada motor cukup besar atau jika inersia dari motor tidak terkontrol sepenuhnya. Hal ini menunjukkan bahwa motor bekerja paling optimal pada kecepatan rendah hingga sedang, sementara pada kecepatan tinggi perlu dilakukan penyesuaian driver atau mekanisme untuk mengurangi getaran. Secara keseluruhan, pengujian ini menunjukkan bahwa kontrol digital memberikan keunggulan dalam hal presisi pengaturan arah dan kecepatan motor dibandingkan kontrol analog. Dengan memanfaatkan logika digital HIGH dan LOW, arah motor dapat dikontrol secara otomatis tanpa intervensi manual, menjadikannya lebih praktis dan efisien untuk aplikasi yang memerlukan pengaturan arah dan kecepatan yang dinamis. Pengujian ini juga membuktikan bahwa sistem berbasis Arduino dan driver DM542 mampu memberikan kendali yang andal untuk motor stepper dalam berbagai kondisi operasional.

4.2.2 Perbandingan Pengaturan Arah Motor Stepper Secara Analog dan Digital



Grafik perbandingan menunjukkan stabilitas pengaturan arah putaran motor stepper antara kontrol analog (switch) dan digital (logika) pada tiga kecepatan yang berbeda: 100 RPM, 200 RPM, dan 300 RPM. Dari grafik tersebut, beberapa poin penting dapat dianalisis:

1. Pada Kecepatan Rendah (100 RPM):

a) Stabilitas kontrol analog tercatat sebesar 90%, sementara kontrol digital mencapai 95%.

b) Perbedaan stabilitas kecil menunjukkan bahwa pada kecepatan rendah, kedua metode kontrol sama-sama mampu menjaga arah dan kecepatan motor dengan baik. Namun, kontrol digital sedikit lebih unggul karena tidak memerlukan intervensi manual, sehingga mengurangi kemungkinan kesalahan.

2. Pada Kecepatan Sedang (200 RPM):

a) Stabilitas kontrol analog menurun menjadi 85%, sedangkan kontrol digital tetap lebih tinggi pada 90%.

b) Penurunan stabilitas pada kontrol analog disebabkan oleh ketergantungan pada respon manual pengguna, yang mungkin kurang responsif pada kecepatan lebih tinggi. Kontrol digital tetap lebih konsisten karena pengaturan dilakukan secara otomatis melalui logika sinyal pada Arduino.

3. Pada Kecepatan Tinggi (300 RPM):

a) Stabilitas kontrol analog menurun signifikan menjadi 70%, sementara kontrol digital masih mampu mencapai 80%.

b) Pada kecepatan tinggi, kontrol analog mengalami lebih banyak getaran dan ketidakseimbangan torsi, yang mengurangi stabilitas motor. Di sisi lain, kontrol digital tetap lebih stabil karena pengaturan arah dilakukan secara presisi melalui program yang mengatur sinyal logika pada driver motor.

Dari grafik, dapat disimpulkan bahwa kontrol digital memiliki keunggulan stabilitas dibandingkan kontrol analog pada semua tingkat kecepatan. Perbedaan stabilitas paling terlihat pada kecepatan tinggi (300 RPM), di mana kontrol digital mampu mempertahankan performa lebih baik dibandingkan kontrol analog. Hal ini menunjukkan bahwa kontrol digital lebih cocok untuk aplikasi yang membutuhkan kecepatan tinggi, presisi, dan efisiensi, sedangkan kontrol analog lebih sesuai untuk aplikasi sederhana pada kecepatan rendah hingga sedang.

V. PENUTUP

5.1 Kesimpulan

Hasil pengujian pengaturan arah putaran motor stepper NEMA 23 dengan kontrol analog dan digital menunjukkan bahwa kedua metode memiliki keunggulan dan kekurangan masing-masing. Pada kontrol analog, arah motor diatur menggunakan saklar toggle yang mengubah sinyal logika secara manual. Metode ini sederhana dan mudah digunakan, terutama untuk demonstrasi atau pengaturan manual. Motor berputar stabil pada kecepatan rendah hingga sedang (100–200 RPM), namun pada kecepatan tinggi (300 RPM), muncul sedikit getaran akibat ketidakseimbangan torsi atau beban motor. Sementara itu, pada kontrol digital, pengaturan

arah motor dilakukan secara otomatis melalui logika HIGH dan LOW pada pin DIR+ yang dikendalikan oleh program Arduino. Kontrol digital memberikan presisi yang lebih baik, efisiensi waktu, dan fleksibilitas dalam pengaturan arah motor tanpa intervensi manual. Kecepatan motor juga dapat ditingkatkan secara terprogram melalui pengaturan frekuensi pulsa pada pin PUL+. Pada kecepatan rendah hingga sedang, motor bergerak stabil seperti pada kontrol analog. Namun, pada kecepatan tinggi, getaran juga masih terjadi, meskipun lebih dapat dikurangi dengan penyesuaian pada program. Secara keseluruhan, kontrol digital lebih unggul untuk aplikasi yang membutuhkan otomatisasi dan presisi tinggi, sedangkan kontrol analog lebih cocok untuk aplikasi sederhana atau demonstrasi.

5.2 Saran

1. Stabilitas Kecepatan Tinggi: Lakukan penyesuaian pada driver DM542, seperti konfigurasi microstepping atau pengurangan beban, untuk mengurangi getaran pada kecepatan tinggi (300 RPM).

2. Penggunaan Praktikum: Gunakan kontrol analog untuk pengenalan dasar dan kontrol digital untuk aplikasi yang memerlukan presisi dan otomatisasi lebih tinggi.

3. Pengembangan Sistem: Tambahkan fitur monitoring real-time, seperti tampilan torsi atau daya, untuk analisis kinerja motor yang lebih detail.

4. Efisiensi Pengujian: Kombinasikan kontrol analog dan digital dalam satu sistem untuk fleksibilitas pengujian yang lebih baik.

DAFTAR PUSTAKA

- [1] Agus Sudarmanto, S. I. (2011). Rancang Bangun Pengendali Motor Stepper untuk Deteksi Jumlah Objek Putar dengan Menggunakan Komputer. *Berkala FISIKA UNDIP*, 1-6.
- [2] Arief Wisnu Wardhana, D. T. (2018). Pengontrolan Motor Stepper Menggunakan Driver DRV 8825 Berbasis Signal Square Wave dari Timer Mikrokontroler AVR. *Jurnal Nasional Teknik Elektro*, 1-10.
- [3] Dfrobot. (2014, January 3). *TB6600 Stepper Motor Driver SKU DRI0043*. Retrieved from [wiki.dfrobot.com: https://wiki.dfrobot.com/TB6600_stepper_motor_driver_sku_dri004](https://wiki.dfrobot.com/TB6600_stepper_motor_driver_sku_dri004)
- [4] Elektro, Z. (2015, Mei Selasa). *Referensi Belajar Elektronika Online*. Retrieved from Zona

Elektro: <http://zoniaelektro.net/motor-stepper/>

[5] Febrianto. (2018, November 29). *ndoware.com*.

Retrieved from apa itu arduino uno:
<https://ndoware.com/apa-itu-arduino-uno.html>

[6] Hutahaean, R. Y. (2013). *Teknik Kontrol Automatik*. Yogyakarta: Penerbit andi.

[7] Iswanto. (2011, Maret). *Aplikasi Motor-Stepper*. Retrieved from
iswanto.staff.umy.ac.id:<http://iswanto.staff.umy.ac.id/files/2011/03/APLIKASI-MOTORSTEPPER.doc>

[8] Julianto, D. (2017). *Media Pembelajaran Trainer Motor DC, Brushletss, Servo, dan Stepper dengan Kendali Mikrokontroler Arduino UNO pada Mata Pelajaran Teknik Mikroprosesor di SMK NEGERI 2 DEPOK YOGYAKARTA*. Yogyakarta: Fakultas Teknik Universitas Negeri Yogyakarta.

[9] Kurniawan, A. (2018, oktober 10). *project author*. Retrieved from www.semesin.com:
<https://www.semesin.com/project/author/asepkurniawan/>

[10] Leselektronika. (2012, juni 27). *liguid crystal display lcd 16* Retrieved
From www.leselektronika.com:
<http://www.leselektronika.com/2012/06/liguid-crystal-display-lcd-16-x-2.html>

[11] Ogata, K. (1997). *Teknik Kontrol Automatik* (Kedua ed., Vol. Jilid 1). Jakarta: Penerbit Erlangga.

[12] Ogata, K. (1997). *Teknik Kontrol Automatik* (Kedua ed., Vol. Jilid 2).
Jakarta: Penerbit Erlangga.

[13] Syifaul Fuada, C. N. (2015). *Perancangan dan Pembuatan Tool Kit*

Motor Stepper jenis Bipolar untuk
Pembelajaran Sistem Kontrol .

ISBN ITB, 1-4.11