

# Perbandingan metode PID dan *Fuzzy Logic Control* dalam sinkronisasi dua motor DC

Diono<sup>1</sup>, Yoga Rezky Pratama<sup>1</sup>, Bintang Kurniawan<sup>1</sup>

<sup>1</sup>Jurusan Teknik Elektro, Politeknik Negeri Batam, Batam, Indonesia

E-mail: [diono@polibatam.ac.id](mailto:diono@polibatam.ac.id), [yogarezkypratama.ytb@gmail.com](mailto:yogarezkypratama.ytb@gmail.com), [bintangkurniawans.k@gmail.com](mailto:bintangkurniawans.k@gmail.com)

## Abstrak

Sinkronisasi kecepatan dua motor DC merupakan aspek penting dalam berbagai aplikasi industri, seperti sistem konveyor, robotika, dan otomasi, karena keseragaman gerak sangat dibutuhkan. Penelitian ini membandingkan dua metode pengendali, yaitu *Proportional-Integral-Derivative* (PID) dan *Fuzzy Logic Controller* (FLC), dalam menjaga sinkronisasi kecepatan antara dua motor DC. Sistem dirancang agar motor kedua dapat mengikuti kecepatan motor pertama dengan akurasi tinggi, dan masing-masing metode diuji pada skenario perubahan kecepatan yang sama. Evaluasi dilakukan berdasarkan parameter kinerja, seperti waktu tunda (*delay time*), waktu tunak (*settling time*), serta kesalahan sinkronisasi terhadap kecepatan target dengan toleransi  $\pm 2\%$ . Hasil pengujian menunjukkan bahwa metode FLC memiliki keunggulan dalam kecepatan respons dan kemampuan adaptasi terhadap perubahan dinamis, sehingga menghasilkan waktu sinkronisasi yang lebih cepat serta kesalahan yang lebih kecil dibandingkan dengan metode PID. Sebaliknya, metode PID memberikan kinerja yang stabil pada kondisi beban tetap, tetapi kurang tanggap terhadap gangguan dan perubahan mendadak. Temuan ini membuktikan bahwa logika *fuzzy* lebih efektif digunakan pada sistem yang memerlukan respons cepat dan fleksibel terhadap kondisi variatif. Dengan demikian, metode FLC direkomendasikan untuk aplikasi pengendalian motor DC multikanal yang menuntut ketelitian sinkronisasi tinggi serta kemampuan adaptif terhadap ketidakpastian sistem.

**Kata kunci:** PID, *Fuzzy Logic Kontrol*, Sinkronisasi

## Abstract

*Synchronization of two DC motors' speed is an essential aspect of various industrial applications, such as conveyor systems, robotics, and automation, since motion uniformity is highly required. This study compares two controller methods, namely Proportional-Integral-Derivative (PID) and Fuzzy Logic Controller (FLC), in maintaining speed synchronization between two DC motors. The system is designed so that the second motor follows the speed of the first motor with high accuracy, and each method is tested under the same speed change scenarios. Evaluation is carried out based on performance parameters such as delay time, settling time, and synchronization error against the target speed within a tolerance of  $\pm 2\%$ . The experimental results show that the FLC method has advantages in response speed and adaptability to dynamic changes, resulting in faster synchronization time and smaller errors compared to the PID method. In contrast, PID provides stable performance under constant load conditions but is less responsive to disturbances and sudden changes. These findings demonstrate that fuzzy logic is more effective when applied to systems that require fast response and flexibility to variable conditions. Therefore, the FLC method is recommended for multichannel DC motor control applications that demand high synchronization accuracy and adaptive capability in the presence of system uncertainties.*

**Keywords:** PID, *Fuzzy Logic Control*, Synchronization

## 1. Pendahuluan

Motor DC banyak digunakan dalam berbagai aplikasi industri, seperti *conveyor belt*, mesin cetak, dan robotika, karena mudah diimplementasikan dan bersifat fleksibel. Namun, koordinasi beberapa motor DC secara bersamaan dengan presisi tinggi dan respons cepat masih menjadi tantangan dalam upaya

mencapai kinerja sistem yang optimal [1]. Pengembangan dan pemilihan sistem kontrol yang andal serta efisien diperlukan untuk menjamin kinerja sistem yang presisi, responsif, dan stabil dalam jangka panjang, khususnya pada aplikasi yang memerlukan koordinasi antarmotor.

Kontrol PID merupakan sistem kendali konvensional yang umum digunakan untuk mengatur kecepatan motor DC berkat struktur sederhana dan kemudahan

implementasinya. Pengendalian ini menggabungkan tiga parameter *proportional*, *integral*, dan *derivative*-untuk menghasilkan sinyal kontrol [2]. Selain memiliki struktur yang sederhana dan mudah diimplementasikan, kontrol PID juga terbukti mampu memberikan kinerja yang baik dalam berbagai kondisi. Namun, pengendali PID memiliki keterbatasan, terutama dalam menangani sistem nonlinier dan perubahan parameter lingkungan.

*Fuzzy Logic Controller* (FLC) menawarkan pendekatan alternatif yang mampu mengatasi ketidakpastian dan nonlinieritas sistem dengan menggunakan basis aturan linguistik. Berbeda dengan kontrol konvensional, FLC tidak memerlukan model matematis yang rumit dan lebih adaptif terhadap perubahan parameter [3]. Hasil penelitian menunjukkan bahwa FLC memiliki keunggulan berupa nilai *overshoot* yang lebih kecil serta waktu tunak (*settling time*) yang lebih cepat dibandingkan dengan kontrol PID [4].

Penelitian lain yang membandingkan PID dan FLC menunjukkan bahwa masing-masing pengendali memiliki karakteristik dan keunggulan pada kondisi operasi tertentu. Hasil pengujian tersebut memperlihatkan bahwa PID lebih unggul dalam menjaga stabilitas keadaan tunak (*steady-state*), sedangkan FLC lebih baik dalam merespons perubahan dinamis sistem [5]. Pada implementasi nyata, pengendali PID yang digunakan untuk mengendalikan motor DC dengan *encoder* terbukti mampu memberikan kinerja yang cukup baik [6]. Meskipun demikian, masih terdapat celah penelitian dalam penerapan kedua metode tersebut pada sinkronisasi multi-motor DC, khususnya pada kondisi yang menuntut presisi tinggi.

Sinkronisasi motor DC merupakan aspek kritis dalam aplikasi industri modern. Pada sistem *conveyor belt*, ketidaksinkronan dapat menimbulkan ketidakseimbangan beban, mempercepat keausan komponen, atau bahkan menyebabkan kerusakan sistem. Sementara itu, dalam bidang robotika, presisi sinkronisasi antar motor sangat menentukan keselamatan dan efisiensi operasi.

Berdasarkan latar belakang diatas, masih terdapat celah penelitian terkait sinkronisasi multi-motor DC. Celah ini menunjukkan adanya ruang penelitian untuk menganalisis dan membandingkan kinerja pengendali PID dan *Fuzzy* dalam aplikasi sinkronisasi multi-motor DC. Upaya tersebut penting dilakukan mengingat kebutuhan akan sistem kontrol yang presisi dan andal dalam berbagai aplikasi industri modern. Oleh karena itu, penelitian ini bertujuan untuk melakukan evaluasi komprehensif terhadap kedua metode tersebut, baik dari aspek performa maupun implementasi sinkronisasi, guna memenuhi tuntutan sistem kontrol yang lebih efektif dan adaptif.

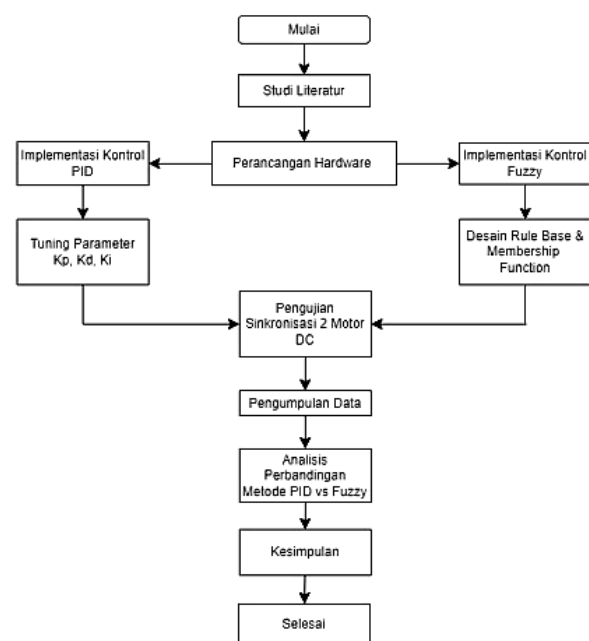
Artikel ini disusun dengan rincian sebagai berikut.

Bab 2 membahas metode penelitian yang digunakan, meliputi perancangan perangkat keras, penerapan kontrol PID, penerapan kontrol *Fuzzy Logic*, serta prosedur pengujian. Bab 3 menyajikan hasil pengujian dan analisis performa sistem yang diperoleh dari kedua metode kontrol. Makalah ini ditutup dengan Bab 4 yang berisi kesimpulan serta saran untuk penelitian lebih lanjut, khususnya dalam pengembangan metode sinkronisasi motor DC yang lebih akurat dan efisien di masa depan.

## 2. Metode Penelitian

### 2.1. Flowchart Penelitian

Proses penelitian ini digambarkan dalam diagram alir yang ditunjukkan pada gambar 1.



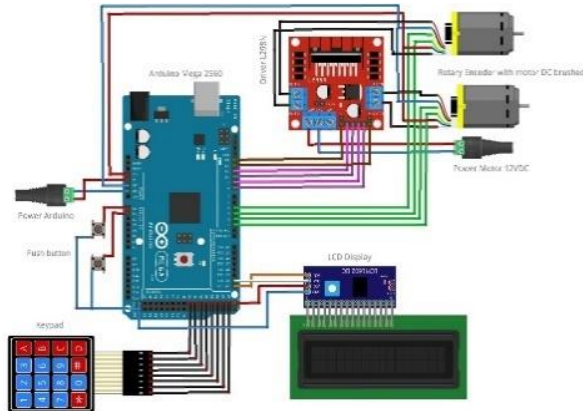
Gambar 1. Flowchart Penelitian

Penelitian ini dimulai dengan tiga proses utama. Pertama, dilakukan perancangan perangkat keras yang meliputi beberapa komponen utama yang digunakan dalam sistem. Kedua, dilakukan implementasi kontrol PID pada perangkat keras beserta proses penalaan (*tuning*) parameter. Ketiga, dilakukan implementasi kontrol *Fuzzy Logic* yang mencakup perancangan *rule base* dan fungsi keanggotaan (*membership function*). Setelah ketiga proses utama tersebut selesai, penelitian dilanjutkan dengan tahap pengujian dan pengumpulan data.

### 2.2. Desain perangkat keras

Gambar 2 menunjukkan koneksi antara Arduino Mega 2560, dua buah motor DC beserta sensor *rotary encoder*, *driver motor* (L298N), LCD (*Liquid Crystal Display*), *keypad*, dan dua buah *push button*. Rangkaian ini dirancang untuk mengontrol motor DC

dengan menggunakan umpan balik dari *rotary encoder*. Arduino berfungsi membaca masukan dari *keypad* atau *push button* untuk menerima perintah pengguna, mengatur kecepatan dan arah motor melalui *driver motor*, serta menampilkan informasi terkait status kecepatan dan nilai set RPM pada LCD.

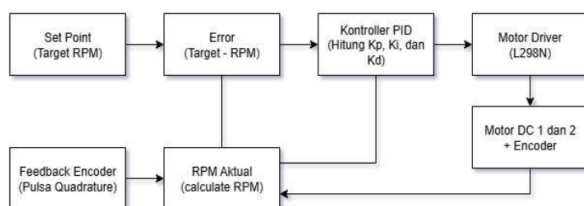


Gambar 2. Desain Hardware

### 2.3. Implementasi Metode PID

PID merupakan sebuah pengendali yang digunakan untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Komponen kontrol PID terdiri atas tiga parameter, yaitu proporsional, integral, dan derivatif. Ketiga parameter tersebut dapat digunakan secara bersamaan maupun secara terpisah, bergantung pada respons yang diinginkan terhadap sistem yang dikendalikan [10].

Gambar 3 menampilkan blok diagram sistem metode PID. Proses dimulai dari masukan berupa set point (target RPM) yang dibandingkan dengan nilai RPM aktual hasil perhitungan dari umpan balik *encoder*, sehingga menghasilkan nilai kesalahan berupa selisih antara target dan RPM aktual. Nilai kesalahan ini kemudian diproses oleh pengendali PID menggunakan parameter  $K_p$ ,  $K_i$ , dan  $K_d$  untuk menghasilkan sinyal kontrol. Sinyal kontrol tersebut dikirimkan ke *motor driver* L298N yang berfungsi mengatur putaran Motor DC 1 dan Motor DC 2. Masing-masing motor dilengkapi *encoder* yang mengirimkan sinyal pulsa kuadratur kembali ke sistem. Sinyal dari *encoder* digunakan untuk menghitung RPM aktual, yang kemudian dibandingkan kembali dengan set point dalam sistem loop tertutup guna menjaga sinkronisasi kecepatan motor secara otomatis dan presisi.



Gambar 3. Diagram Blok Sistem PID

Langkah selanjutnya adalah menetapkan parameter  $K_p$ ,  $K_i$ , dan  $K_d$  yang digunakan dalam pengendali PID. Pada implementasi awal, target kedua motor ditetapkan sebesar 2000 RPM dengan nilai  $K_p$  awal sebesar 0,1. Nilai  $K_i$  dan  $K_d$  kemudian diperoleh menggunakan metode Ziegler–Nichols serta pendekatan *trial and error*. Hasil akhir parameter PID setelah proses penyetelan ulang ditunjukkan pada Tabel 1.

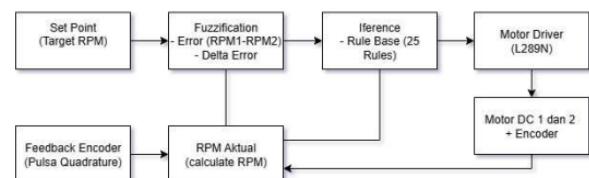
Tabel 1. Parameter PID

$K_p$	$K_i$	$K_d$
0.12	0.0238	0.00151

### 2.4. Implementasi Metode Fuzzy Logic Control

Sistem kontrol *fuzzy* merupakan salah satu metode kecerdasan buatan yang menggunakan logika *fuzzy* untuk mengendalikan sistem dan membuat keputusan. Berbeda dengan sistem kontrol konvensional yang bergantung pada model matematika dan algoritma yang akurat, sistem kontrol *fuzzy* menggunakan aturan kabur (*fuzzy rules*) untuk mengambil keputusan berdasarkan informasi yang tidak pasti atau tidak lengkap [7].

Blok diagram pada Gambar 4 menunjukkan alur sistem kontrol kecepatan dua motor DC menggunakan metode *Fuzzy Logic*. Proses diawali dengan masukan berupa set point (target RPM) yang dibandingkan dengan nilai RPM aktual hasil pembacaan umpan balik *encoder*, sehingga diperoleh nilai kesalahan (error  $RPM_1 - RPM_2$ ) dan perubahan kesalahan (*delta error*). Kedua nilai ini kemudian difuzzifikasi pada tahap *fuzzification*. Selanjutnya, nilai hasil *fuzzifikasi* diproses pada tahap *inference* dengan menggunakan *rule base* yang terdiri atas 25 aturan logika *fuzzy* untuk menghasilkan keputusan kontrol. Keputusan tersebut dikirimkan ke *motor driver* L298N guna mengatur kecepatan Motor DC 1 dan Motor DC 2. Masing-masing motor dilengkapi *encoder* yang berfungsi memberikan umpan balik berupa sinyal pulsa kuadratur, yang diolah menjadi nilai RPM aktual. Nilai tersebut kemudian digunakan kembali dalam proses kontrol berbasis loop tertutup untuk menjaga sinkronisasi kecepatan kedua motor.



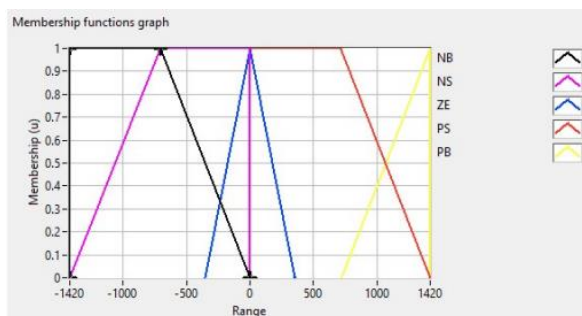
Gambar 4. Diagram blok sistem FLC

Dalam *Fuzzy Logic Control*, terdapat beberapa tahapan yang harus dilalui, yaitu sebagai berikut:

## 1) Fuzzifikasi

Tahap pertama yang dilakukan adalah *fuzzifikasi*. *Fuzzifikasi* merupakan proses mengubah nilai masukan dari bentuk tegas (*crisp*) menjadi nilai *Fuzzy* [8]. Pada tahap ini, himpunan *Fuzzy* ditentukan berdasarkan grafik fungsi keanggotaan. Dalam *Fuzzy Logic Control*, terdapat beberapa bentuk grafik representasi fungsi keanggotaan, yaitu representasi kurva linier naik, representasi kurva linier turun, representasi kurva segitiga, dan representasi kurva trapesium [9]. Grafik fungsi keanggotaan yang dirancang pada sistem ini terdiri atas empat fungsi keanggotaan berbentuk kurva trapesium dan satu fungsi keanggotaan berbentuk kurva segitiga. Kelima fungsi keanggotaan tersebut masing-masing diberi label *Negative Big* (NB), *Negative Small* (NS), *Zero* (Z), *Positive Small* (PS), dan *Positive Big* (PB). Pada sistem ini, fungsi keanggotaan dibagi menjadi tiga jenis, yaitu fungsi keanggotaan untuk input error sinkronisasi, fungsi keanggotaan untuk input *delta error*, dan fungsi keanggotaan untuk output PWM. Batasan fungsi keanggotaan pada sumbu x ditentukan berdasarkan kesalahan (*error*) maksimum yang mungkin terjadi pada sistem, sedangkan batasan fungsi keanggotaan untuk input *delta error* ditentukan berdasarkan hasil pengamatan nilai maksimum *delta error* yang dapat terjadi. Sumbu y pada grafik fungsi keanggotaan menunjukkan nilai keanggotaan, yang dalam sistem ini ditetapkan antara 0 hingga 1.

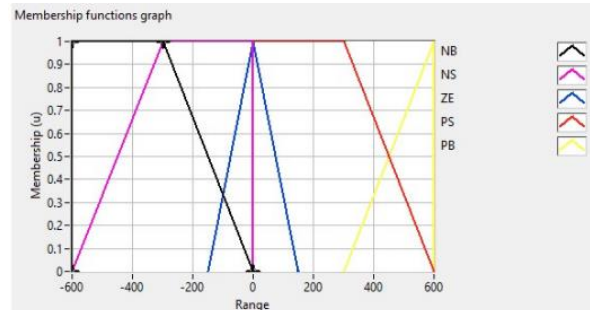
Gambar 5 menunjukkan fungsi keanggotaan untuk variabel input error pada sistem kontrol logika *Fuzzy*. Variabel ini merepresentasikan selisih antara kecepatan target dan kecepatan aktual motor DC. Nilai error memiliki rentang sekitar -1420 hingga +1420 RPM.



Gambar 5. Fungsi Keanggotaan Error

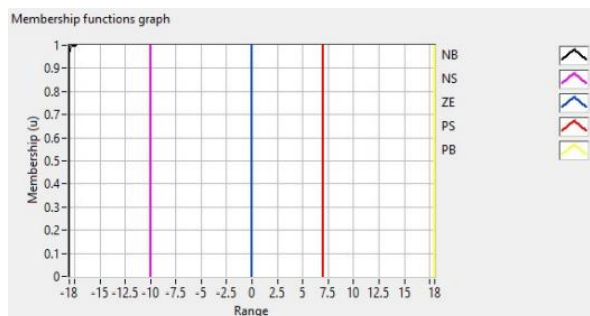
Gambar 6 menunjukkan fungsi keanggotaan untuk variabel input *delta error* pada sistem kontrol logika *Fuzzy*. *Delta error* merupakan perubahan nilai error dari satu waktu ke waktu berikutnya yang mencerminkan kecepatan perubahan kesalahan pada sistem. Rentang *delta error* berkisar antara -600 hingga +600, yang ditetapkan berdasarkan hasil pengamatan terhadap dinamika sistem motor DC. Nilai keanggotaan ( $\mu$ ) berada pada rentang 0 hingga 1 dan menunjukkan tingkat partisipasi suatu nilai *delta*

*error* dalam masing-masing himpunan *Fuzzy*. Sebagai contoh, jika *delta error* bernilai -300, maka nilai tersebut dapat memiliki keanggotaan parsial pada himpunan NB dan NS, yang selanjutnya digunakan dalam proses *inferensi Fuzzy* untuk menentukan sinyal kontrol keluaran.



Gambar 6. Fungsi Keanggotaan Delta Error

Gambar 7 menunjukkan grafik fungsi keanggotaan untuk variabel output pada sistem kontrol logika *Fuzzy*. Variabel output ini merepresentasikan besarnya perubahan sinyal PWM yang dikirim ke motor sebagai hasil dari proses inferensi berbasis error dan *delta error*.



Gambar 7. Fungsi Keanggotaan Output PWM

## 2) Rule Base

*Rule base* terdiri atas basis data dan kumpulan aturan. *Rule base* logika *Fuzzy* diperoleh melalui pendekatan analisis berdasarkan kebutuhan sistem. *Rule base* dirancang dengan mempertimbangkan pengaruh parameter, seperti error sinkronisasi dan *delta error*, terhadap respons sistem untuk menghasilkan keluaran berupa sinyal PWM yang digunakan oleh motor pengikut agar sesuai dengan kecepatan motor utama. Aturan-aturan tersebut dibuat berdasarkan logika hubungan antara variabel input dan variabel output melalui analisis perilaku motor DC serta tujuan pengendalian yang diinginkan. *Rule base Fuzzy* berfungsi sebagai penentu keluaran dari proses *fuzzifikasi*, yang selanjutnya diolah dalam proses *defuzzifikasi* untuk menghasilkan sinyal kontrol yang halus dan efisien.

Aturan *IF-THEN* digunakan untuk menentukan

keluaran berdasarkan *rule base Fuzzy* yang telah dirancang. Proses ini menghasilkan keluaran dengan menerapkan aturan *IF-THEN* pada variabel input, seperti error sinkronisasi dan *delta error*, untuk memperoleh sinyal kontrol yang sesuai. *Rule base* pada sistem ini dirancang menggunakan nilai-nilai *Fuzzy* seperti NB, NS, Z, PS, dan PB yang mewakili rentang variabel input dan output. Nilai tersebut ditetapkan berdasarkan karakteristik motor DC dan dianalisis untuk menghasilkan kontrol yang optimal. Dengan demikian, nilai keluaran diperoleh melalui pendekatan analisis kebutuhan sistem dengan tujuan mencapai respons yang lebih akurat dan stabil. *Rule base* yang digunakan dalam penelitian ini ditunjukkan pada Tabel 2.

Tabel 2. *Rule base Fuzzy*

No	Rule Base
1	If Error NB and $\Delta$ Error NB then Output NB
2	If Error NB and $\Delta$ Error NS then Output NB
3	If Error NB and $\Delta$ Error Z then Output NS
4	If Error NB and $\Delta$ Error PS then Output Z
5	If Error NB and $\Delta$ Error PB then Output Z
6	If Error NS and $\Delta$ Error NB then Output NB
7	If Error NS and $\Delta$ Error NS then Output NS
8	If Error NS and $\Delta$ Error Z then Output Z
9	If Error NS and $\Delta$ Error PS then Output PS
10	If Error NS and $\Delta$ Error PB then Output PS
11	If Error Z and $\Delta$ Error NB then Output NS
12	If Error Z and $\Delta$ Error NS then Output Z
13	If Error Z and $\Delta$ Error Z then Output Z
14	If Error Z and $\Delta$ Error PS then Output Z
15	If Error Z and $\Delta$ Error PB then Output PS
16	If Error PS and $\Delta$ Error NB then Output Z
17	If Error PS and $\Delta$ Error NS then Output PS
18	If Error PS and $\Delta$ Error Z then Output PS
19	If Error PS and $\Delta$ Error PS then Output PB
20	If Error PS and $\Delta$ Error PB then Output PB
21	If Error PB and $\Delta$ Error NB then Output Z
22	If Error PB and $\Delta$ Error NS then Output PS
23	If Error PB and $\Delta$ Error Z then Output PS
24	If Error PB and $\Delta$ Error PS then Output PB
25	If Error PB and $\Delta$ Error PB then Output PB

Keterangan:

NB : *Negative Big*  
 NS : *Negative Small*  
 Z : *Zero*  
 PS : *Positive Small*  
 PB : *Positive Big*

### 3) Fuzzy Inference

Tahapan berikutnya dalam pengolahan logika *fuzzy* adalah *Fuzzy inference* atau sistem pengambilan keputusan (*decision-making*). Pada tahap ini, metode inferensi yang digunakan adalah metode Sugeno karena metode tersebut menghasilkan keluaran numerik yang dapat langsung digunakan untuk mengatur PWM motor dengan respons cepat. Proses

perolehan nilai keluaran dari himpunan *Fuzzy* dilakukan melalui mekanisme penggabungan aturan dengan menggunakan operator *AND*, di mana nilai derajat keanggotaan terkecil dari elemen-elemen yang relevan dalam aturan dipilih. Nilai derajat keanggotaan untuk setiap aturan kemudian dihitung sesuai dengan Persamaan (1).

$$\mu_{rule} = \min(\mu A[i], \mu B[j]) \quad (1)$$

Dimana :

$\mu A[i]$  : Derajat keanggotaan dari variabel input  
 $\mu B[j]$  : Derajat keanggotaan dari variabel input kedua

### 4) Defuzzifikasi

Pada tahap *defuzzifikasi*, metode yang digunakan adalah rata-rata berbobot (*weighted average*). Metode ini menghitung nilai rata-rata dari keluaran setiap aturan *Fuzzy* yang telah ditentukan sebelumnya dengan mempertimbangkan bobot berupa derajat kebenaran dari masing-masing aturan. Metode *weighted average* hanya menggunakan nilai keluaran tegas (*crisp output*) berupa konstanta dari setiap aturan serta derajat keanggotaannya sebagai bobot.

*Defuzzifier* berperan dalam mengubah informasi *Fuzzy* dari variabel  $\mu_{Error}$  dan  $\Delta Error$ , yang merupakan hasil proses inferensi dengan nilai antara 0 hingga 1, menjadi keluaran tegas (*crisp output*) berupa perubahan PWM. Proses ini dilakukan dengan menjumlahkan hasil kali antara bobot aturan dan nilai keluarannya, kemudian membaginya dengan jumlah total bobot dari seluruh aturan. Rumus perhitungan tersebut ditunjukkan pada Persamaan (2).

$$Z_{WA} = \frac{\sum_i w_i \cdot z_i}{\sum_i w_i} \quad (2)$$

Dimana nilai *output crisp* ( $Z_{WA}$ ) diperoleh dengan menjumlahkan hasil perkalian antara setiap bobot aturan *fuzzy*  $w_i$  dengan nilai *output* konstan dari aturan  $z_i$  kemudian membaginya dengan total bobot dari seluruh aturan yang aktif. Dengan kata lain, setiap *output* aturan diberi kontribusi sesuai dengan besar kecilnya bobot atau derajat keaktifannya, dan nilai akhir adalah rata-rata tertimbang dari semua *output* aturan tersebut.

Metode ini dipilih karena efisien secara komputasional dan sangat sesuai untuk sistem tertanam seperti Arduino, di mana penggunaan bilangan pecahan dan keterbatasan sumber daya menjadi pertimbangan utama. Selain itu, metode ini mampu memberikan respons sistem yang cepat dan stabil sehingga sangat penting dalam pengendalian sinkronisasi motor berbasis waktu nyata.

## 2.5. Error Sinkronisasi

Kesalahan sinkronisasi merujuk pada ketidaksesuaian atau perbedaan kecepatan antara dua motor DC yang seharusnya beroperasi secara bersamaan. Kesalahan ini merupakan faktor penting dalam menilai efektivitas sistem kendali sinkronisasi. Analisis kesalahan sinkronisasi bertujuan untuk menilai seberapa kecil nilai kesalahan yang dicapai oleh masing-masing metode, yaitu PID dan *Fuzzy*. Proses ini dilakukan dengan cara mengukur perbedaan kecepatan antara motor utama dan motor pengikut selama beroperasi, kemudian merekam data kesalahan tersebut untuk dianalisis menggunakan parameter. Parameter yang dimaksud antara lain sebagai berikut:

1) *Mean Absolute Error* (MAE):

MAE menggunakan perhitungan yang ditunjukkan pada persamaan (3).

$$MAE = \frac{1}{N} \sum_{i=1}^N |e_i| \quad (3)$$

Dimana  $e_i$  adalah *error* pada iterasi ke  $i$

## 2.6. Pengambilan Data

Pengambilan data dilakukan dengan mengamati respons pergerakan kedua motor berdasarkan waktu pencuplikan (*sampling time*) dan nilai setpoint RPM yang berbeda. Proses pengambilan data dilaksanakan melalui beberapa tahap pengujian.

Tahap pertama adalah pengujian respons terhadap berbagai setpoint, yaitu 1000, 2000, 3000, dan 4000 RPM. Parameter yang diamati pada tahap ini meliputi waktu naik (*rise time*), waktu tunak (*settling time*), lonjakan (*overshoot*), kesalahan keadaan tunak (*steady-state error*), dan kesalahan sinkronisasi. Setiap pengambilan data dilakukan sebanyak lima kali dengan durasi 30 detik untuk setiap pengujian. Hasil dari pengujian pertama selanjutnya digunakan sebagai dasar untuk analisis perbandingan antara kedua metode dalam menjaga sinkronisasi motor.

Tahap kedua adalah analisis sinkronisasi kedua motor. Data dari pengujian pertama direkam pada setiap percobaan, kemudian dihitung menggunakan rumus MAE untuk menganalisis besarnya kesalahan sinkronisasi yang terjadi pada setiap interval 0,1 detik.

Tahap ketiga adalah pengujian manual. Pengujian ini dilakukan dengan menurunkan putaran Motor 1 (RPM) secara manual.

## 3. Hasil dan Pembahasan

### 3.1. Pengujian Respon PID

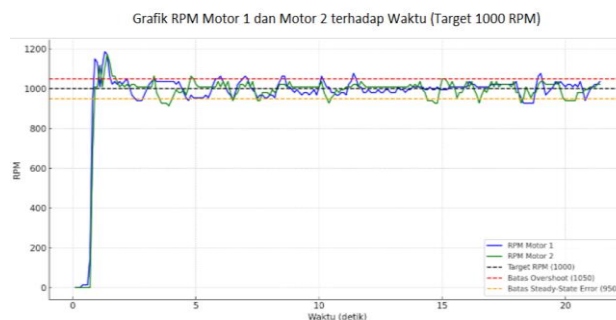
Tabel 3 menunjukkan beberapa karakteristik kinerja sistem kendali PID. Waktu naik (*rise time*) kedua motor pada setpoint 1000 RPM sangat cepat, yang mengindikasikan respons awal yang agresif dari kendali PID. Waktu tunak (*settling time*) pada setpoint 2000 RPM berada pada kisaran 2,3–2,6 detik, sehingga menunjukkan bahwa kendali PID mampu menstabilkan sistem dengan relatif cepat. Lonjakan (*overshoot*) tampak sangat tinggi pada setpoint 1000 RPM. Sementara itu, kesalahan keadaan tunak (*steady-state error*) umumnya bernilai relatif kecil pada sebagian besar setpoint.

Secara keseluruhan, hasil pengujian menunjukkan bahwa pengendali PID mampu mengatur kecepatan motor dengan respons yang bervariasi sesuai dengan setpoint. Kecepatan respons (*rise time*) cenderung menurun pada setpoint tinggi, sedangkan lonjakan (*overshoot*) yang besar terlihat pada setpoint rendah tetapi berkurang pada kecepatan menengah hingga tinggi. Tingkat ketelitian (*steady-state error*) cukup baik meskipun mengalami peningkatan pada setpoint 4000 RPM. Hasil ini menjadi dasar perbandingan lebih lanjut dengan kinerja sistem kendali berbasis logika Fuzzy untuk mengevaluasi metode yang lebih optimal dalam sinkronisasi dua motor DC.

Tabel 3. Pengujian Respon PID

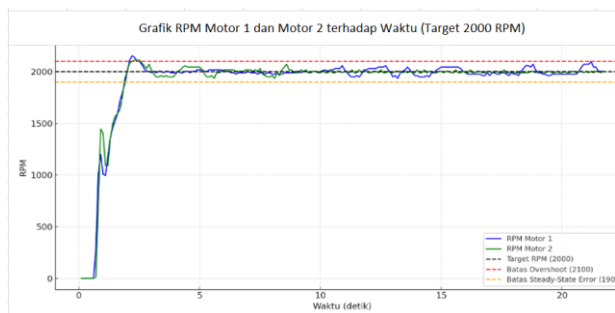
Set RPM	Rise time (ms)		Settling time (ms)		Overshoot (%)		Steady state error(%)	
	M1	M2	M1	M2	M1	M2	M1	M2
1000	100	100	5825	7075	14	15	0.30	0.32
2000	1440	1300	2360	2640	7.15	6.34	0.93	0.42
3000	2280	2320	3020	3320	5.35	5.62	0.43	0.17
4000	2720	3340	3980	3880	4.37	4.58	0.27	0.28

Gambar 8 menunjukkan bahwa kecepatan Motor 1 dan Motor 2 meningkat dengan cepat menuju setpoint 1000 RPM. Kedua motor mengalami *overshoot* hingga sekitar 1200 RPM pada awal respons, kemudian berosilasi ringan sebelum stabil di sekitar nilai target. Setelah beberapa detik, kecepatan kedua motor berada dalam rentang  $\pm 5\%$  (950–1050 RPM)



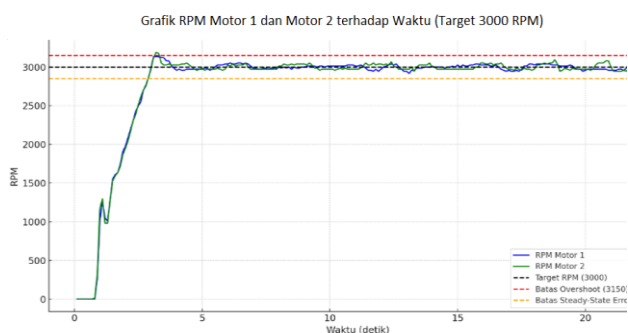
Gambar 8. Grafik Respon PID Target 1000 RPM

Gambar 9 menunjukkan bahwa kecepatan Motor 1 dan Motor 2 meningkat dengan cepat menuju setpoint 2000 RPM. Kedua motor mengalami *overshoot* ringan pada awal respon, kemudian stabil di sekitar nilai target. Setelah beberapa detik, kecepatan kedua motor berada dalam rentang  $\pm 5\%$  (1900–2100 RPM)



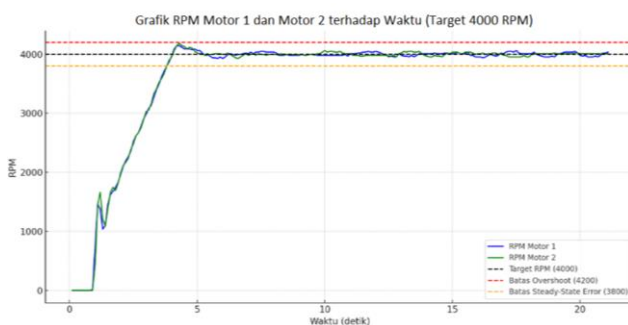
Gambar 9. Grafik Respon PID Target 2000 RPM

Gambar 10 menunjukkan bahwa kecepatan Motor 1 dan Motor 2 meningkat dengan cepat menuju setpoint 3000 RPM. Kedua motor mengalami *overshoot* ringan pada awal respon, kemudian stabil di sekitar nilai target. Setelah beberapa detik, kecepatan kedua motor berada dalam rentang  $\pm 5\%$  (2850–3150 RPM), sehingga sistem mencapai kondisi *steady-state* dengan error yang kecil.



Gambar 10. Grafik Respon PID Target 3000 RPM

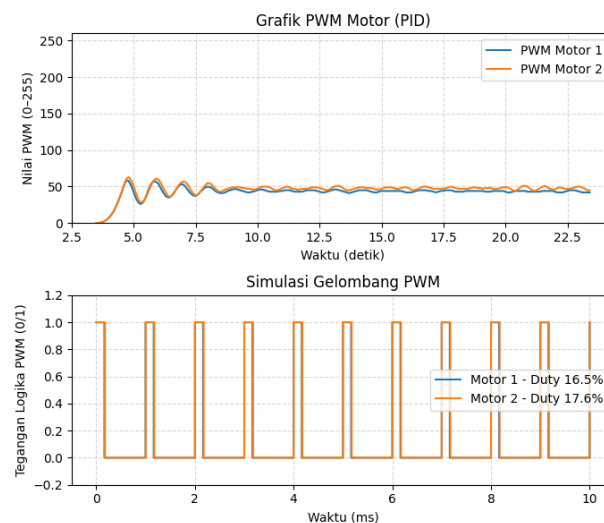
Gambar 11 menunjukkan bahwa kecepatan Motor 1 (biru) dan Motor 2 (hijau) meningkat dengan cepat menuju setpoint 4000 RPM. Kedua motor mengalami *overshoot* ringan pada awal respon, kemudian stabil di sekitar nilai target.



Gambar 11. Grafik Respon PID Target 4000 RPM

Gambar 12 menunjukkan validasi respons sistem kendali PID dalam menyinkronkan dua motor DC melalui dua grafik sinyal PWM. Grafik pertama menunjukkan nilai PWM motor terhadap waktu. Grafik tersebut memperlihatkan bahwa Motor 1 dan Motor 2 memiliki pola nilai PWM yang relatif serupa, tetapi tetap menunjukkan sedikit perbedaan. Kondisi ini menandakan bahwa kendali PID secara aktif mengoreksi kesalahan kecepatan untuk mempertahankan sinkronisasi. Fluktuasi kecil pada nilai PWM setelah mencapai setpoint menunjukkan bahwa proses penyesuaian berlangsung secara terus menerus, yang merupakan karakteristik kendali PID dalam menjaga kestabilan sistem.

Grafik kedua, yaitu simulasi gelombang PWM digital, menunjukkan bahwa Motor 1 memiliki *duty cycle* sebesar 16,5% dan Motor 2 memiliki *duty cycle* sebesar 17,6%. Perbedaan *duty cycle* tersebut menunjukkan bahwa sinyal PWM yang diberikan ke masing-masing motor disesuaikan secara presisi oleh kendali PID untuk mengatasi ketidakseimbangan kecepatan, sehingga kedua motor tetap sinkron. Dengan demikian, kedua grafik ini memvalidasi bahwa respons sistem PID tidak hanya dapat diamati secara numerik melalui parameter seperti *rise time*, *settling time*, *steady-state error*, dan *overshoot*, tetapi juga dapat diamati secara visual melalui pola sinyal PWM yang dihasilkan selama proses kontrol berlangsung.



Gambar 12. GUI gelombang PWM PID

### 3.2. Analisis Sinkronisasi PID

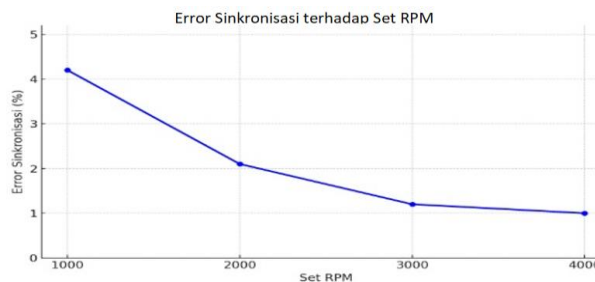
Pengujian pada tahap ini mengukur kesalahan sinkronisasi yang terjadi pada kedua motor selama masa *steady-state*. Nilai kesalahan sinkronisasi diperoleh dari selisih absolut antara RPM Motor 1 dan RPM Motor 2

Tabel 4 menyajikan hasil pengukuran *Mean Absolute Error* (MAE) dan persentase error sinkronisasi pada

berbagai nilai set point RPM. Nilai error tertinggi tercatat pada kecepatan 1000 RPM sebesar 4,2%, sedangkan nilai terendah tercatat pada kecepatan 4000 RPM sebesar 1,0%. Secara keseluruhan, rata-rata error sinkronisasi sistem sebesar 2,1%. Hasil ini menunjukkan bahwa sistem kendali PID memiliki kinerja yang lebih stabil dan presisi pada kecepatan tinggi dibandingkan pada kecepatan rendah.

Tabel 4. Analisis MAE pada sinkronisasi PID

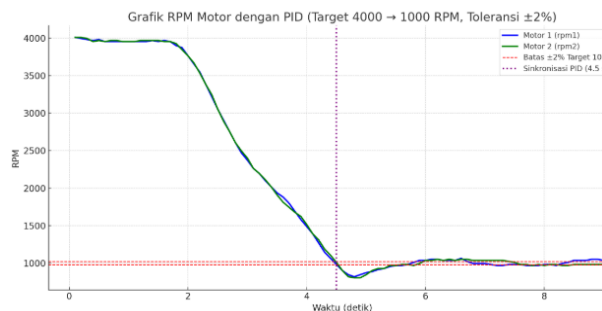
Set RPM	Pengukuran MAE ke -					Rata - Rata	
	1	2	3	4	5	MAE	Error (%)
1000	45	42	49	37	35	42	4,2 %
2000	34	48	41	39	49	42	2,1 %
3000	35	28	36	47	39	37	1,2 %
4000	36	35	42	43	46	40	1 %
Rata - Rata Error (%) Keseluruhan							2,1 %



Gambar 13. Grafik Sinkronisasi PID

### 3.3. Pengujian Respon Manual

Pengujian ini dilakukan dengan menurunkan kecepatan Motor 1 secara manual menggunakan *push button*, dari kecepatan awal 4000 RPM hingga 1000 RPM. Pengujian ini bertujuan untuk menganalisis akurasi dan kecepatan pelacakan Motor 2 terhadap Motor 1. Gambar 14 menunjukkan bahwa ketika kecepatan Motor 1 diturunkan, Motor 2 secara langsung mengikuti perubahan kecepatan tersebut, sehingga sinkronisasi antara kedua motor tetap terjaga.



Gambar 14. Respon sinkronisasi PID Motor 2 terhadap Motor 1

### 3.4. Hasil Pengujian Fuzzy Logic Control

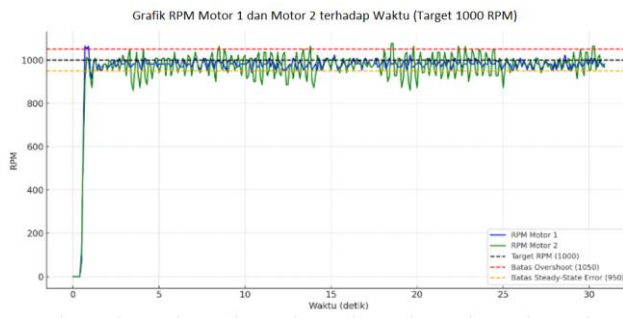
Tabel 5 menunjukkan bahwa parameter *rise time* cenderung meningkat seiring dengan bertambahnya nilai setpoint, dari 140–180 ms pada 1000 RPM hingga mencapai 1433–1462 ms pada 4000 RPM. Pada setpoint rendah, seperti 1000 RPM, Motor 2 (motor *follower*) memiliki *settling time* yang cukup tinggi, yaitu 4480 ms, dibandingkan Motor 1 (motor utama) yang hanya 1080 ms. Untuk parameter *overshoot*, sistem FLC mampu menjaga nilai osilasi pada level yang rendah. Nilai *overshoot* tertinggi tercatat sebesar 5,68% pada Motor 2 di setpoint 1000 RPM, kemudian menurun hingga hanya 0,7% pada 4000 RPM. Terakhir, parameter *steady-state error* menunjukkan hasil yang positif. Nilai error maksimum tercatat pada 1000 RPM, yaitu sebesar 1,61% pada Motor 1 dan 1,57% pada Motor 2, kemudian menurun menjadi hanya 0,4–0,5% pada setpoint 4000 RPM.

Secara keseluruhan, metode *Fuzzy Logic Control* menunjukkan kemampuan adaptasi yang baik terhadap variasi setpoint. Metode ini memberikan respons yang halus, *overshoot* yang minimal, dan error yang rendah, terutama pada kecepatan tinggi.

Tabel 5. Pengujian performa respon sistem FLC

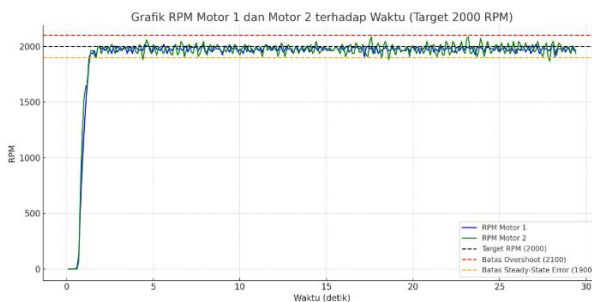
Set RPM	Rise time (ms)		Settling time (ms)		Overshoot (%)		Steady state error(%)	
	M1	M2	M1	M2	M1	M2	M1	M2
1000	180	140	1080	4480	3.88	5.68	1.61	1.57
2000	540	480	1360	2062	1	2.9	1.1	1.1
3000	880	1100	1900	1820	0.3	0.3	0.7	0.7
4000	1462	1433	6580	8868	0.5	0.7	0.4	0.5

Grafik pada Gambar 15 menunjukkan respon kecepatan Motor 1 dan Motor 2 terhadap waktu ketika sistem *Fuzzy Logic Controller* (FLC) diberikan target 1000 RPM. Kedua motor dapat mencapai kecepatan target dengan cepat, namun hasil pengujian memperlihatkan adanya fluktuasi nilai RPM di sekitar garis referensi. Walaupun nilai RPM masih berada dalam batas toleransi *overshoot* dan *steady-state error* yang ditentukan, penyimpangan yang muncul menunjukkan bahwa respon sistem belum sepenuhnya stabil.



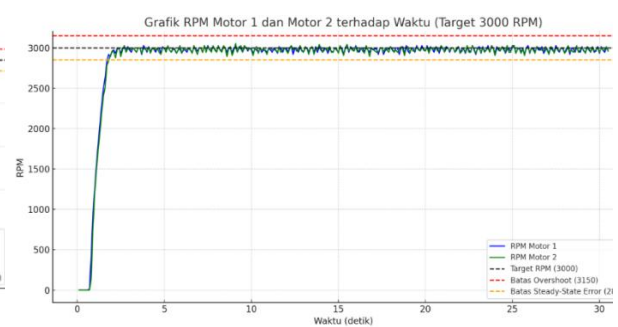
Gambar 15. Grafik Respon FLC Target 1000 RPM

Grafik pada Gambar 16 memperlihatkan respon kecepatan Motor 1 dan Motor 2 terhadap waktu ketika sistem *Fuzzy Logic Controller* (FLC) diberikan target 2000 RPM. Kedua motor dapat mencapai kecepatan target dengan cepat dan kurva respon mendekati garis referensi 2000 RPM. Hasil pengujian menunjukkan bahwa nilai *overshoot* dan *steady-state error* tetap berada dalam batas toleransi, sehingga sistem mampu menjaga kestabilan kecepatan serta sinkronisasi kedua motor dengan lebih baik dibandingkan pada target 1000 RPM.



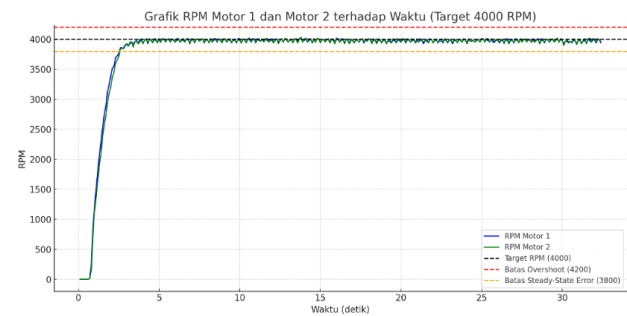
Gambar 16. Grafik Respon FLC Target 2000 RPM

Grafik pada Gambar 17 menunjukkan respon kecepatan Motor 1 dan Motor 2 terhadap waktu ketika sistem *Fuzzy Logic Controller* (FLC) diberikan target 3000 RPM. Kedua motor dapat mencapai kecepatan target dengan sangat cepat dan stabil, ditandai oleh kurva respon yang mendekati garis referensi 3000 RPM. Hasil pengujian memperlihatkan bahwa nilai *overshoot* dan *steady-state error* berada dalam batas toleransi yang telah ditentukan, sehingga sistem mampu menjaga kestabilan kecepatan serta sinkronisasi kedua motor secara konsisten pada kecepatan tinggi.



Gambar 17. Grafik Respon FLC Target 3000 RPM

Grafik pada Gambar 18 menunjukkan respon kecepatan Motor 1 dan Motor 2 terhadap waktu ketika sistem *Fuzzy Logic Controller* (FLC) diberikan target 4000 RPM. Kedua motor dapat mencapai kecepatan target dengan sangat cepat dan kurva respon mendekati garis referensi 4000 RPM. Hasil pengujian memperlihatkan bahwa nilai *overshoot* dan *steady-state error* berada dalam batas toleransi.

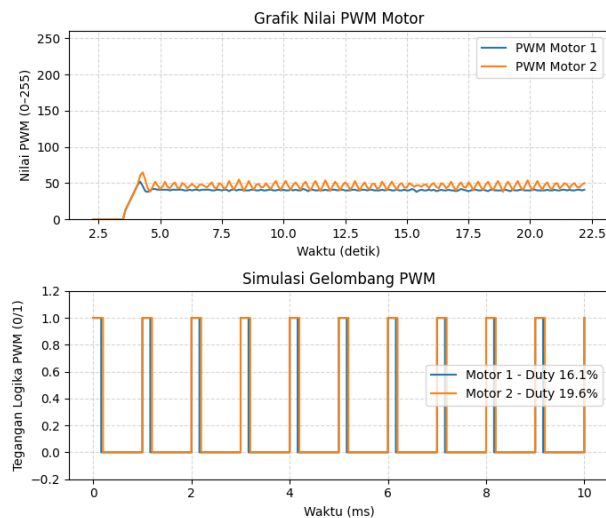


Gambar 18. Grafik Respon FLC Target 4000 RPM

Gambar 19 menunjukkan hasil validasi respon sistem kontrol *Fuzzy Logic* dalam menyinkronkan dua motor DC berdasarkan grafik sinyal PWM.

Pada grafik pertama, sistem memperlihatkan hubungan nilai PWM motor terhadap waktu. Setelah motor mencapai kondisi stabil, nilai PWM Motor 1 dan Motor 2 menunjukkan kecenderungan yang sangat dekat dengan fluktuasi yang halus serta terkontrol.

Pada grafik kedua, sistem menampilkan simulasi gelombang PWM digital. *Duty cycle* Motor 1 tercatat sebesar 16,1% sedangkan *duty cycle* Motor 2 sebesar 19,6%. Bentuk gelombang PWM yang dihasilkan menunjukkan pola yang stabil dan konsisten. Perbedaan *duty cycle* tersebut memperlihatkan bahwa sistem *Fuzzy Logic* menghasilkan output PWM yang disesuaikan secara halus untuk setiap motor guna mempertahankan keserempakan kecepatan.



Gambar 19. Grafik GUI gelombang PWM FLC

### 3.5. Analisis Sinkronisasi Fuzzy Logic Control

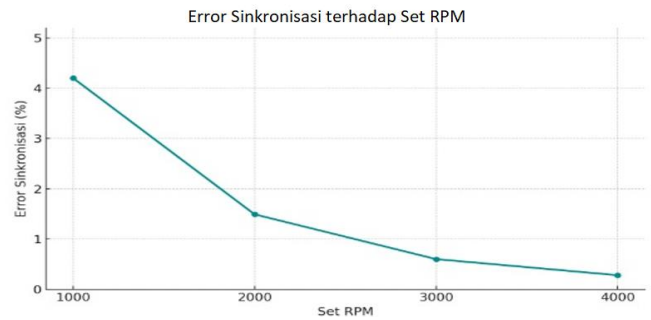
Tabel 6 menunjukkan bahwa pada setpoint 1000 RPM, nilai error sinkronisasi mencapai titik tertinggi sebesar 4,2%. Kondisi tersebut menunjukkan bahwa pada kecepatan rendah, sistem memerlukan waktu lebih lama untuk menyamakan respon kedua motor. Seiring dengan meningkatnya nilai setpoint, nilai error mengalami penurunan secara signifikan hingga mencapai 0,28% pada setpoint 4000 RPM.

Penurunan nilai error menunjukkan bahwa metode *Fuzzy Logic Control* bekerja semakin optimal pada kecepatan tinggi. Pada kondisi tersebut, motor pengikut dapat mengikuti motor utama dengan perbedaan kecepatan yang sangat kecil. Hal tersebut juga menunjukkan bahwa kendali berbasis *Fuzzy* memiliki karakteristik adaptif yang baik terhadap dinamika sistem.

Secara keseluruhan, rata-rata error sinkronisasi untuk seluruh setpoint mencapai 1,64%. Nilai tersebut mengindikasikan bahwa sistem memiliki tingkat presisi yang baik dalam mempertahankan keserempakan pergerakan antara dua motor DC.

Tabel 6. Analisis MAE sinkronisasi FLC

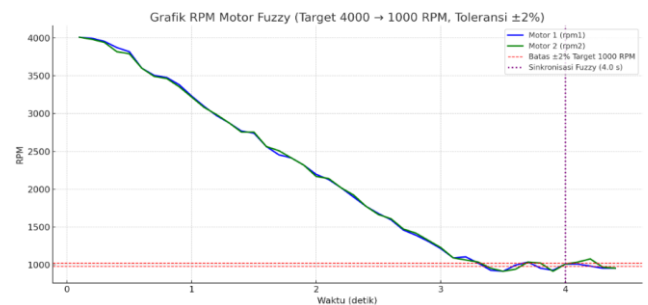
Set RPM	Pengukuran ke -					Rata - Rata	
	1	2	3	4	5	MAE	Error (%)
1000	44.65	37.46	43.02	42.81	42.90	42.16	4.2%
2000	29.13	25.20	29.71	30.08	34.9	29.80	1.49%
3000	18.11	18.91	16.55	20.29	16.73	18.11	0.6%
4000	11.58	10.56	12.09	11.25	10.89	11.27	0.28%
Rata - Rata total Error (%) Keseluruhan							1.64%



Gambar 20. Grafik Sinkronisasi FLC

### 3.6. Pengujian Respon Manual

Pengujian ini dilakukan dengan menurunkan RPM Motor 1 secara manual menggunakan *push button*, dari RPM awal 4000 hingga 1000. Tujuan pengujian adalah untuk melihat akurasi dan kecepatan pelacakan Motor 2 terhadap Motor 1. Gambar 21 menunjukkan bahwa ketika kecepatan Motor 1 diturunkan, Motor 2 langsung mengikuti kecepatan Motor 1.



Gambar 21. Respon sinkronisasi Motor 2 terhadap Motor 1 dengan FLC

### 3.7. Perbandingan metode PID dan *Fuzzy Logic Control* (FLC)

Parameter yang dibandingkan meliputi *rise time*, *settling time*, *overshoot*, *steady-state error*, dan error sinkronisasi. Nilai setiap parameter diperoleh melalui pengujian eksperimental pada empat variasi setpoint, yaitu 1000, 2000, 3000, dan 4000 RPM. Tabel 7 menyajikan rata-rata nilai setiap parameter yang dihitung dari gabungan Motor 1 (M1) dan Motor 2 (M2) untuk masing-masing metode kontrol, serta selisihnya sebagai indikator keunggulan relatif.

Berdasarkan hasil perbandingan pada Tabel 7, metode *Fuzzy Logic Control* menunjukkan keunggulan signifikan pada *rise time* dan *overshoot*. Rata-rata *rise time* pada FLC tercatat sebesar 776,88 ms, lebih cepat dibandingkan dengan PID yang memiliki rata-rata 1762,5 ms. Perbedaan tersebut menunjukkan bahwa sistem dengan kendali FLC merespons lebih cepat terhadap perubahan setpoint.

Dari sisi *settling time*, FLC mencatat waktu pemantapan sistem yang lebih rendah, yaitu 3518,75 ms, dibandingkan dengan PID yang memiliki waktu

rata-rata 4012,5 ms. Perbedaan tersebut menunjukkan bahwa sistem dengan kendali FLC mencapai kestabilan lebih cepat dibandingkan dengan sistem yang menggunakan PID.

Dalam hal *overshoot*, metode PID menghasilkan nilai rata-rata sebesar 13,39%, sedangkan metode FLC hanya menghasilkan 1,91%. Perbedaan tersebut membuktikan bahwa FLC mampu meredam lonjakan awal secara signifikan, sehingga metode ini sangat penting dalam aplikasi presisi seperti sinkronisasi motor.

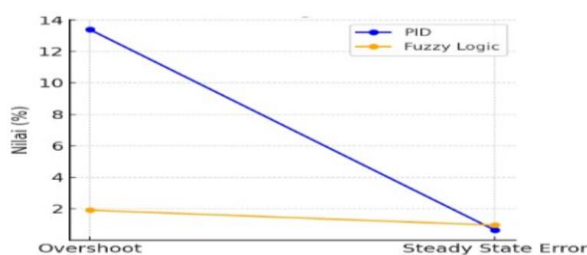
Satu-satunya keunggulan metode PID terlihat pada parameter *steady-state error*. Metode PID mencatat error lebih rendah sebesar 0,64% dibandingkan dengan metode FLC yang mencatat 0,96%. Perbedaan tersebut tidak terlalu signifikan dan masih berada dalam batas toleransi sistem kendali.

Secara keseluruhan, metode *Fuzzy Logic Control* terbukti lebih unggul dibandingkan metode PID dalam menangani sinkronisasi dua motor DC, terutama pada aspek respons awal dan pengendalian lonjakan. Hasil penelitian ini mendukung temuan pada literatur sebelumnya yang menyatakan bahwa FLC lebih adaptif dan efektif untuk sistem nonlinier.

Dengan mempertimbangkan kecepatan respon, kestabilan, *overshoot*, dan error sinkronisasi, metode *Fuzzy Logic Control* lebih direkomendasikan untuk implementasi kontrol sinkronisasi motor DC pada aplikasi industri yang membutuhkan presisi dan waktu respon cepat.

Tabel 7. Perbandingan parameter respon hasil pengujian metode PID dan FLC

Parameter	PID	FLC	Selisih PID-FLC
Rise Time(ms)	1762.5	776.88	+985.62
Settling time(ms)	4012.5	3518.75	+493.75
Overshoot(%)	13.39	1.91	+11.48
Steady state error	0.64	0.96	-0.32



Gambar 22. Grafik Perbandingan Respon PID dan FLC



Gambar 23. Grafik Perbandingan Waktu Respons PID dan FLC

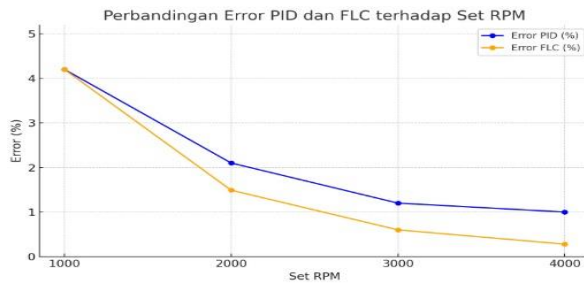
Berdasarkan hasil pengujian sinkronisasi dua motor pada berbagai nilai setpoint RPM, metode *Fuzzy Logic Controller* (FLC) menunjukkan performa yang lebih baik dibandingkan dengan metode PID. Keunggulan tersebut ditunjukkan oleh rata-rata error sinkronisasi FLC sebesar 1,64%, lebih rendah dibandingkan dengan rata-rata error PID sebesar 2,1% sebagaimana ditampilkan pada Tabel 8.

Pada kecepatan rendah (1000 RPM), kedua metode menghasilkan tingkat error yang sama, yaitu 4,2%. Kondisi tersebut mengindikasikan bahwa keduanya memiliki keterbatasan serupa dalam mengatasi kecepatan rendah, kemungkinan karena pengaruh beban awal atau friksi sistem. Seiring dengan meningkatnya nilai RPM, metode *Fuzzy Logic Controller* (FLC) menunjukkan kinerja yang lebih konsisten dan presisi dengan penurunan error yang signifikan, yaitu 1,49% pada 2000 RPM, 0,6% pada 3000 RPM, dan 0,28% pada 4000 RPM. Sementara itu, metode PID menunjukkan fluktuasi error yang lebih tinggi pada nilai RPM yang sama.

Kinerja FLC yang lebih unggul disebabkan oleh kemampuannya dalam menangani ketidaklinieran dan dinamika sistem secara lebih adaptif melalui aturan logika fuzzy. Dengan demikian, penelitian ini menyimpulkan bahwa metode FLC lebih efektif dan akurat dalam menjaga sinkronisasi kecepatan dua motor, terutama pada rentang kecepatan menengah hingga tinggi.

Tabel 8. Perbandingan Error Sinkronisasi

Set RPM	Error PID (%)	Error FLC (%)	Metode lebih baik
1000	4.20%	4.20%	Setara
2000	2.10%	1.49%	FLC
3000	1.20%	0.60%	FLC
4000	1.00%	0.28%	FLC



Gambar 24. Perbandingan Sinkronisasi Metode PID dan FLC

#### 4. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan pada sistem sinkronisasi dua motor DC, dapat disimpulkan bahwa metode *Fuzzy Logic Control* (FLC) menunjukkan performa yang lebih unggul dibandingkan *Proportional-Integral-Derivative* (PID) dalam menjaga keserempakan kecepatan kedua motor. FLC memiliki kecepatan respon lebih tinggi, waktu tunak lebih singkat, serta *overshoot* dan error sinkronisasi yang lebih kecil dibandingkan PID, dengan rata-rata error sinkronisasi sebesar 1,64% (lebih rendah dari PID sebesar 2,1%). Selain itu, FLC mampu beradaptasi lebih baik terhadap perubahan dinamis pada kecepatan motor, menjadikannya lebih efektif untuk sistem nonlinier. Namun, metode PID tetap memiliki keunggulan pada kestabilan keadaan tunak (*steady-state*) dengan error akhir yang sedikit lebih kecil. Dengan demikian, FLC lebih direkomendasikan untuk aplikasi sinkronisasi multi-motor DC yang memerlukan respons cepat dan kemampuan adaptif tinggi, sedangkan PID cocok untuk kondisi beban tetap dengan tuntutan kestabilan jangka panjang.

#### References

- [1] A. Shahzad, M. Kashif, T. Munir, and M. He, "DC Motor Synchronization Speed Controller Based on Microcontroller." [Online].
- [2] B. Subarta, H. Eteruddin, and D. Setiawan, "Analisis Perbandingan Sistem Kendali PID dan Sistem Kendali Fuzzy PID pada Pengontrolan Kecepatan Motor DC" *TELKA*, vol. 10, no. 2, pp. 178–192, 2024.
- [3] M. Z. F. B. M. Zawawi, I. Elamvazuthi, A. B. A. Aziz, and S. A. Daud, "Comparison of PID and Fuzzy Logic controller for DC servo motor in the development of lower extremity exoskeleton for rehabilitation," in *Proc. 2017 IEEE 3rd International Symposium in Robotics and Manufacturing Automation, ROMA 2017*, Institute of Electrical and Electronics Engineers Inc., Dec. 2017, pp. 1–6. doi: 10.1109/ROMA.2017.8231822.
- [4] A. Viantika, "Studi perbandingan kinerja kontroler Proporsional Integral Derivative (PID) dan Fuzzy Logic Control (FLC) pada pengontrolan kecepatan motor DC," *Sigma Teknika*, vol. 2, no. 2, pp. 282–295, 2019.
- [5] A. L. Shuraiji and S. W. Shneen, "Fuzzy Logic Control and PID Controller for Brushless Permanent Magnetic Direct Current Motor: A Comparative Study," *Journal of Robotics and Control (JRC)*, vol. 3, no. 6, pp. 762–768, Nov. 2022, doi: 10.18196/jrc.v3i6.15974.
- [6] M. Reza, A. N. Putera, R. Hidayat, and S. Karawang, "Kendali kecepatan motor DC menggunakan pengendali PID dengan encoder sebagai feedback." STRING (Satuan Tulisan Riset dan Inovasi Teknologi), [Online].
- [7] S. Rahmawati, M. T. I. J. St, M. T. I. N. Syarif, and M. T. Juhan, "Pengantar desain sistem kendali otomatis dengan Fuzzy Logic Controller." [Online].
- [8] E. W. Saputra, "Optimasi fungsi keanggotaan fuzzy Mamdani menggunakan algoritma genetika untuk penentuan penerima beasiswa," *SIMADA: Jurnal Sistem Informasi dan Manajemen Basis Data*, vol. 2, no. 2, pp. 160–175, Jan. 2020, doi: 10.30873/simada.v2i2.1789.
- [9] S. Sasongko and U. Teknologi Yogyakarta, "Implementasi Fuzzy Logic Controller sebagai Pengendali Posisi Motor Servo." [Online].
- [10] N. Budi Hartono, B. Sumantri, and A. Wijayanto, "Pengaturan posisi motor servo DC dengan metode P, PI, dan PID." [Online].