Pengendali PID pada Motor DC dan Tuning Menggunakan Metode Differential Evolution

PID Controllers on DC Motors and Tuning Using the Differential Evolution Method

Siti Fatimah Anggraini[1*], Alfian Ma'arif [2*], Riky Dwi Puriyanto[3]
[1,2,3]Department of Electrical Engineering, Universitas Ahmad Dahlan
siti1600022051@webmail.uad.ac.id[1], alfianmaarif@ee.uad.ac.id[2], rikydp@ee.uad.ac.id[3]

Abstrak – *Penggunaan PID controller pada sistem industri sangat umum digunakan dalam menentukan kepresisian suatu feedback yang diberikan dengan keluaran sesuai dengan keinginan pengguna. Namun pada umumnya optimalisasi masih menggunakan cara manual yaitu trial dan error pada tuning gain terhadap nilai Kp, Ki, dan Kd. Kendala tersebut dapat diatasi dengan melakukan pendekatan fungsi dan mengaplikasikan nilai tuning dengan metode Diferensial Evolusi. Penggunaaan tuning PID pada implementasinya menggunakan motor DC untuk menentukan kecepatan putar motor dengan keluaran yang stabil. Hasil yang didapatkan berupa sinyal keluaran atau respon sistem terhadap putaran motor DC dengan menampilkan nilai risetime, settlingtime, dan overshoot. Pengoptimalan nilai dilakukan dengan pencarian nilai terbaik pada setiap iterasi dan diaplikasikan sebagai tuning Kp, Ki, dan Kd tanpa harus melakukan cara manual. Hasil pengujian parameter Diferensial Evolusi menunjukkan efesiensi pengoptimalan tuning dan memberikan hasil keluaran yang baik dengan implementasinya terhadap plan hardware motor DC.*

Kata Kunci*: PID controller, Motor DC, Pemyetelan, Diferensial Evolusi.*

Abstract – *The use of PID in industrial systems is very commonly used in determining the optimal accuracy of the output. However, several studies still uses manual methods for the optimization, namely trial and error on tuning gain with values of Kp, Ki, and Kd. This problem could be overcome applying the tuning value with the Evolution Differential method. The use of PID tuning in its implementation in this study uses a DC motor to determine the rotational speed of the motor with a stable output. Based on the results, we can see the value of rise time, settling time, and overshoot based on an output signals. The optimization was done by finding the best value in iteration and applying it as Kp, Ki, and Kd tuning without having to do the manual method. The results of testing the Differential Evolution parameters show the efficiency of tuning optimization and provide good output with the implementation of the DC motor hardware plan.*

Keywords*: PID(Proportional–Integral–Derivative controller), Motor DC, Tuning, Differential Evolution.*

1. Pendahuluan

Proportional–Integral–Derivative (PID) *controller* merupakan mekanisme umpan balik yang biasa digunakan pada sistem kendali industri. Fungsi PID adalah untuk mengetahui ketepatan presisi suatu instrumentasi pada umpan balik atau biasa disebut sebagai *feedback*. PID

tuning dilakukan untuk menentukan nilai *gain* pada sebuah mikrokontroler yang akan diimplementasikan untuk mendapatkan respons sistem yang diinginkan. Masalah yang dialami pada proses penyelesaiannya terletak pada penentuan nilai yang akan digunakan sebagai *gain* untuk mendapatkan respons sistem dari *tuning* yang dilakukan.

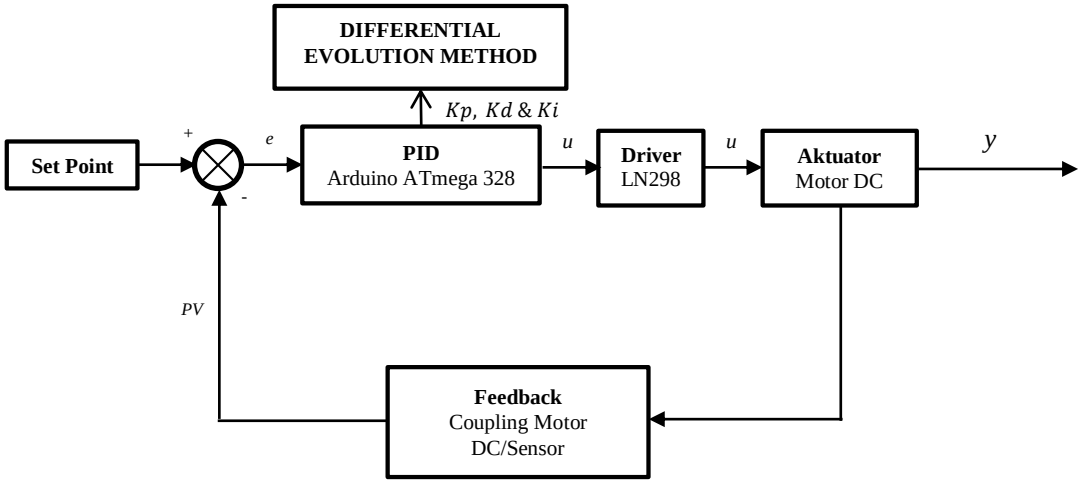
Penyetelan/*tuning* biasanya dilakukan menggunakan metode *trial* dan *error* untuk menentukan nilai *tuning* yang akan di masukkan ke dalam *gain* Kp, Ki dan Kd. Pada permasalahan *tuning* manual PID, solusi Diferensial Evolusi (DE) dapat digunakan untuk pencarian nilai gain [1]. DE digunakan untuk mendapatkan penyelesaian dengan evolusi baru sebagai salah satu langkah pada suatu masalah. Pada algoritma DE, vektor sidang baru diciptakan secara acak dengan menggabungkan tiga strategi percobaan DE dan tiga pengaturan parameter di setiap generasi [2]. DE merupakan sebuah model stokastik yang mesimulasikan *evolusi biologis* [3]. DE dilengkapi dengan tiga komponen algoritmik yang diaktifkan dengan cara probabilistik dengan perbedaan berbasis strategi gangguan pada langkah mutasi [4]. Pada tahap seleksi acak masalah ini adalah bahwa F dan CR harus dipilih secara acak dari distribusi seragam setiap vektor sidang yang dihasilkan dengan menggunakan skema sendiri yang dipilih secara acak juga [5]. Mutasi, Rekombinasi, dan seleksi Diferensial Evolusi menunjukkan bahwa algoritma memberikan kinerja yang cukup baik dan cukup mudah diterapkan dalam kinerja PID. Metode desain *PID controller* menghasilkan kelemahan dan kekurangan PID sendiri [6]. Untuk mengatasi kelemahan kinerja dalam mengurangi overshoot, dapat dilakukan *tuning* untuk penyetelan Kp, Ki dan Kd dengan metode Diferensial Evolusi [7]. Selain itu, efektivitas keakuratan sistem kontrol motor dc dapat ditingkatkan dengan cara menyesuaikan parameternya secara optimal [8]. Berdasarkan pemaparan sebelumnya, pada makalah ini dipaparkan sebuah desain pengendali PID pada motor DC dan *tuning* menggunakan metode DE. Penambahan *tuning* dengan metode DE dimaksudkan untuk meningkatkan kinerja sistem pengendali yang dibuat.

2. Metode Penelitian

Pada penelitian ini dilakukan penentuan parameter PID menggunakan DE dalam upaya untuk menghasilkan nilai Kp, Ki dan Kd yang optimal serta mendapatkan nilai pengujian pada motor DC. Pada bab metode penelitian ini akan dipaparkan tahapan-tahapan yang dilakukan pada penelitian ini, meliputi desain sistem, desain sistem sambungan, perancangan perangkat keras, pengujian sistem, dan algoritma DE yang digunakan pada penelitian ini.

2.1. Desain Sistem

Pada penelitian ini, desain sistem ditunjukkan seperti pada Gambar 1. Pada desain sistem ini, keluaran metode Evolusi Diferensial diproses dan selanjutnya diimplementasikan pada perangkat keras.



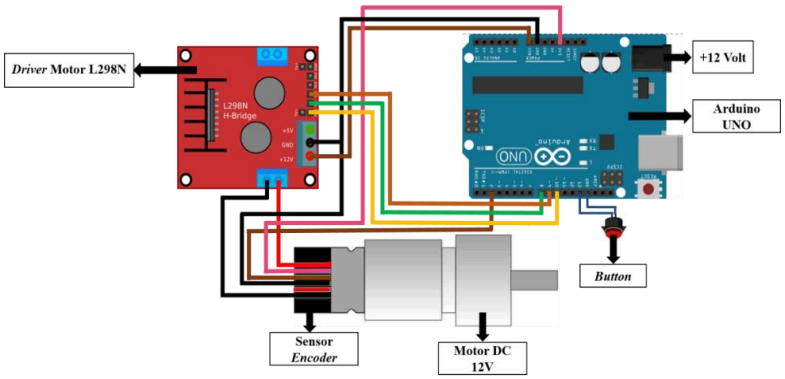
Gambar 1. Plan pengujian atau desain sistem.

ISSN (e): 2540-9123
ISSN (p): 2502-1982

Sistem akan memprogram masukan pada PID mikrokontroler dan akan memberikan nilai terbaik pada *gain Kp, Ki* dan *Kd* melalui hasil *tuning* degan metode Diferensial Evolusi sebelumnya. Pada Gambar 1, setting point (SP) digunakan sebagai acuan utama untuk menjalankan putaran motor/nilai referensi, *e* adalah nilai error, PID mikrokontroler digunakan untuk mengatur *gain Kp, Ki* dan *Kd*, *u* adalah sinyal kendali digital dan analog, driver mengatur presisi motor DC, akuator motor DC saat berputar, *feedback* nilai yang dihasilkan dari *driver* dan *akuator*, *present value (PV)* adalah nilai *feedback*, dan *y* output keluaran atau hasil.

2.2. Desain Sistem Sambungan

Pada Gambar 2, ditunjukkan sambungan diagram sistem pengujian pada plan. Pada desain sistem sambungan terdapat driver motor L298N yang berfungsi sebagai feedback dan menentukan presisi pada motor dc, Arduino sebagai PID mikrokontroler digunakan untuk melakukan tuning agar memberikan gain Kp, Ki dan Kd pada motor dc, dan motor dc akan menghitung nilai putaran dari gain PID mokrokontroler terhadap suatu respons sistem yang akan dihasilkan.



Gambar 2. Desain sistem sambungan.

Masukan dan keluaran dari desain sistem sambungan PID adalah motor dc. Pada sistem yang dibangun ini *serial encoder* akan menghasilkan posisi sudut pada putaran motor dc, *control pin driver* akan mengkoneksikan dengan PID mikrokontroler, pwm akan mengolah putaran atau modulasi dari lebar pulsa yang dihasilkan, dan button adalah tombol untuk memulai putaran motor.

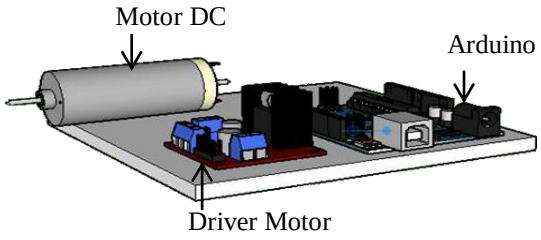
Tabel 1. Pin I/O Plan motor DC.

No.	Arduino	L298N	Motor DC	Encoder	Tombol
1	Pin 13	-	-	-	Kaki 1
2	Pin 8	IN1	-	-	-
3	Pin 9	IN2	-	-	-
4	Pin 10	ENA	-	-	-
5	Pin 2	-	-	Out	-
6	5 Volt	Vcc	-	-	-
7	3,3 Volt	-	-	3,3 Volt	Kaki 2
8	GND	GND	-	GND	-
9	-	M1	M1	-	-
10	-	M2	M2	-	-

2.3. Perancangan Perangkat Keras

Perancangan perangkat keras dilakukan dengan membuat gambaran desain tiga dimesi untuk memberikan gambaran keluaran sebuah plan yang akan diuji. Gambar 3 adalah sebuah plan desain tiga dimensi yang akan mengimplementasikan hasil dari metode Diferensial Evolusi.

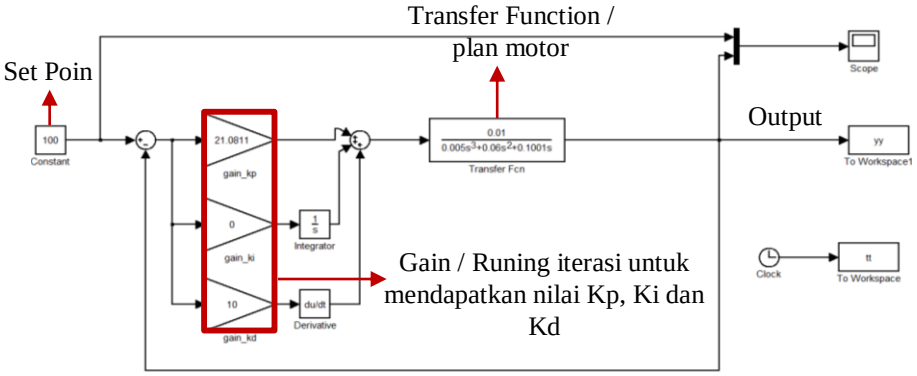
ISSN (e): 2540-9123
ISSN (p): 2502-1982



Gambar 3. Desain sistem rangkaian 3D.

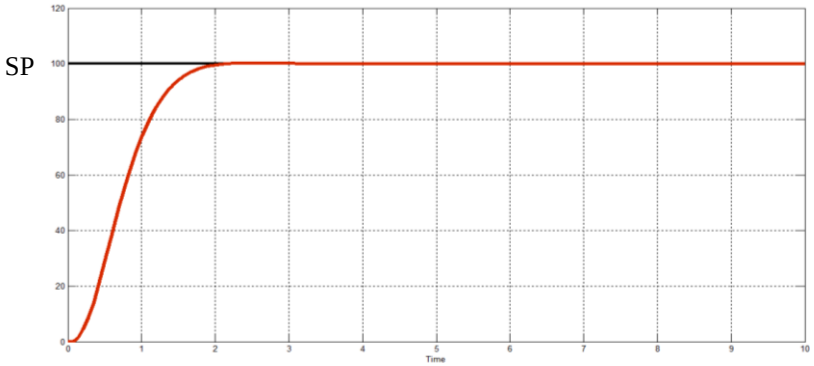
2.4. Pengujian Sistem Metode DE

Pengujian metode DE yang digunakan pada penelitiana ini diuji dengan simulasi menggunakan Matlab. Simulasi dilakukan dengan mencari nilai terbaik *gain tuning Kp, Ki* dan *Kd* yang kemudian ditampilkan pada plan *simulink* Matlab.



Gambar 4. Simulink program DE.

Berdasarkan hasil simulasi diperoleh respon sistem yang ditunjukkan pada Gambar 5. Pada Gambar 5 dapat diobservasi apakah nilainya melebihi nilai set point yang sudah ditentukan sebelumnya dan dapat diketahui apakah terjadi *overshoot* atau tidak. Hasil simulasi yang tidak terjadi overshoot dapat digunakan pada tuning PID Motor DC.



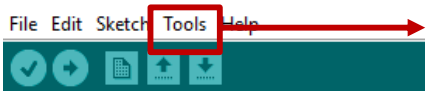
Gambar 5. Respon sistem.

2.5. Pengujian Sistem dengan Plan *Hardware*

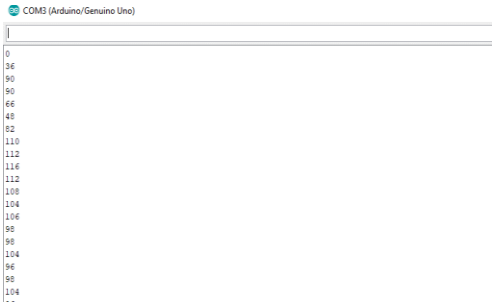
Pengujian dengan metode DE diawali dengan membuat *tuning* parameter secara otomatis, di mana nilai parameter dalam DE sangat berpengaruh terhadap performa DE sendiri. Pada pengujian ini dilakukan sebanyak 100 kali, dan mengambil nilai keluaran dari PID motor dc yang berupa kecepatan. Program *float Kp, Ki* dan *Kd* dimasukkan pada pencarian sebelumnya dengan metode Diferensial Evolusi dan dilakukan mode *tools* untuk menampilkan putaran motor dc yang

ISSN (e): 2540-9123
ISSN (p): 2502-1982

dihasilkan. Pada Gambar 6 menunjukkan data putaran motor DC dengan 100 kali pengambilan data. Selanjutnya, data tersebut dianalisis untuk melihat keadaan sinyal yang dihasilkan.



Menampilkan Serial Monitor Data



Gambar 6. Menampilkan putaran motor DC.

2.6. Algoritma Evolusi Diferensial

Differential Evolution diperkenalkan oleh Storn dan Price pada 1995. Menurut Storn dan Price, DE memiliki 3 buah parameter kontrol yaitu sebagai berikut.

(1) Ukuran populasi.
(2) F yaitu faktor skala berfungsi untuk mengatur *difference* vektor juga untuk menghindari stagnasi selama proses pencarian.
(3) CR yaitu *Crossover* dalam nilai (0,1) yang berfungsi sebagai pengendali laju dari perhitungan.

Keuntungan utama metode DE dapat dirangkum dalam tiga poin: tidak memerlukan parameter yang banyak, tidak mudah terjebak dalam solusi lokal, dan cepat untuk berkumpul [10]. Algoritma DE Standar membutuhkan setidaknya 4 langkah sebagai berikut.

1. Inisialisasi generasi populasi yang ditunjukkan pada Persamaan 1.

$$P_{ij} = rand_{p1j}(0,1)\left(x_{ij}^{u} + x_{ij}^{L}\right) + x_{ij}^{L} \quad (1)$$

di mana i adalah jumlah individu (populasi), j adalah jumlah variabel, $rand_{p1j}(0,1)$ adalah nilai acak dari 0 hingga 1, x_{ij}^{u} adalah batas atas, x_{ij}^{L} adalah batas bawah.

2. Tahap mutasi dengan memilih tiga individu acak dalam populasi,

$$h_{ij}(t+1) = x_{p1j}(t) + F\left(x_{p2j}(t) + x_{p3j}(t)\right), \quad (2)$$

sedangkan fungsi tanpa solusi lokal adalah,

$$h_{ij}(t+1) = x_{bj}(t) + F\left(x_{p2j}(t) + x_{p3j}(t)\right), \quad (3)$$

di mana, x_{bj} adalah individu-individu terbaik. F adalah faktor mutasi yang dipilih antara 0 - 2. Secara umum, dapat diatur dari 0,3 ke 0,6 agar proses konvergensi lebih cepat dan meminimalisir agar prosesnya tidak terjebak dalam solusi lokal;

3. Tahap c*rossover* untuk membuat vektor baru,

ISSN (e): 2540-9123
ISSN (p): 2502-1982

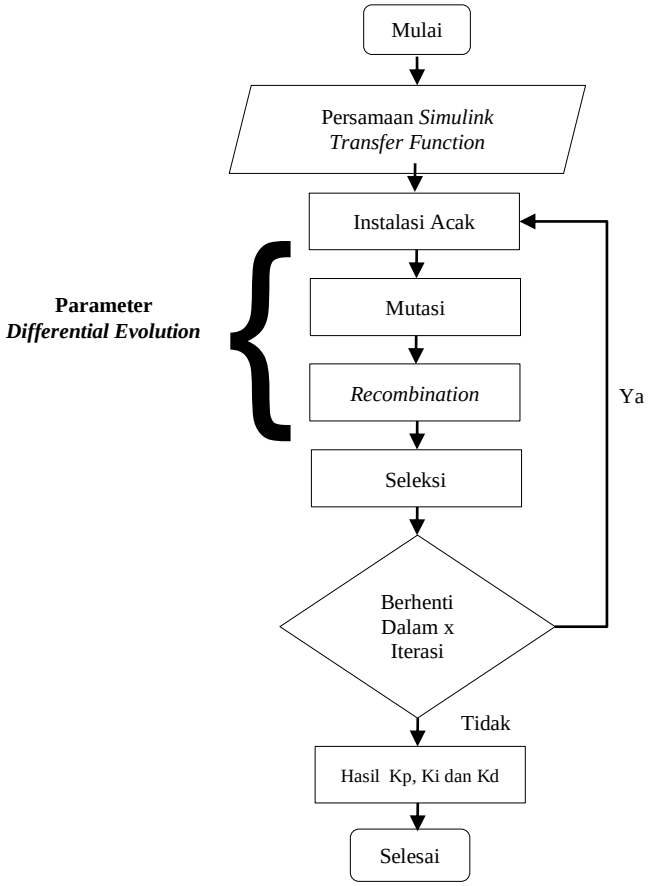
$$v_i(t+1) = \begin{cases} h_{ij}(t+1), & rand_{ij} \leq CR \\ P_{ij}(t), & rand_{ij} > CR \end{cases} \quad (4)$$

Dengan CR adalah faktor *crossover* dalam kisaran [0,1]. Secara umum, dapat diatur dari 0,6 hingga 0,9.

4. Mengevaluasi vektor dengan membandingkannya dengan orang tuanya. Jika lebih baik daripada orangtuanya, maka lakukanlah penggantian.

$$P_i(t+1) = \begin{cases} v_i(t+1), & f\big(v_{ij}(t+1)\big), \dots, v_{in}(t+1) < f(x_{x1}(t), \dots, x_{in}(t)) \\ x_{ij}, & f\big(v_{ij}(t+1)\big), \dots, v_{in}(t+1) \geq f(x_{x1}(t), \dots, x_{in}(t)) \end{cases} \quad (5)$$

Pada Gambar 4 ditunjukkan *flowchart* iterasi yang menggunakan metode DE dengan ketiga parameternya yaitu *G* sebagai instalasi acak, *F* sebagai mutasi data dan *Cr* atau recombination kali silang data.



Gambar 4. *Flowchart* pencarian iterasi menggunakan metode DE.

3. Hasil dan Pembahasan

Pada bagian pembahasan ini, dilakukan analisis berdasarkan hasil pengujian jumlah spesies, pengujian parameter kontrol DE, dan pengujian manual tuning kontroler. Hasil Sistem kontrol DE sendiri digunakan untuk mengevaluasi kinerja kontroler yang diusulkan.

ISSN (e): 2540-9123
ISSN (p): 2502-1982

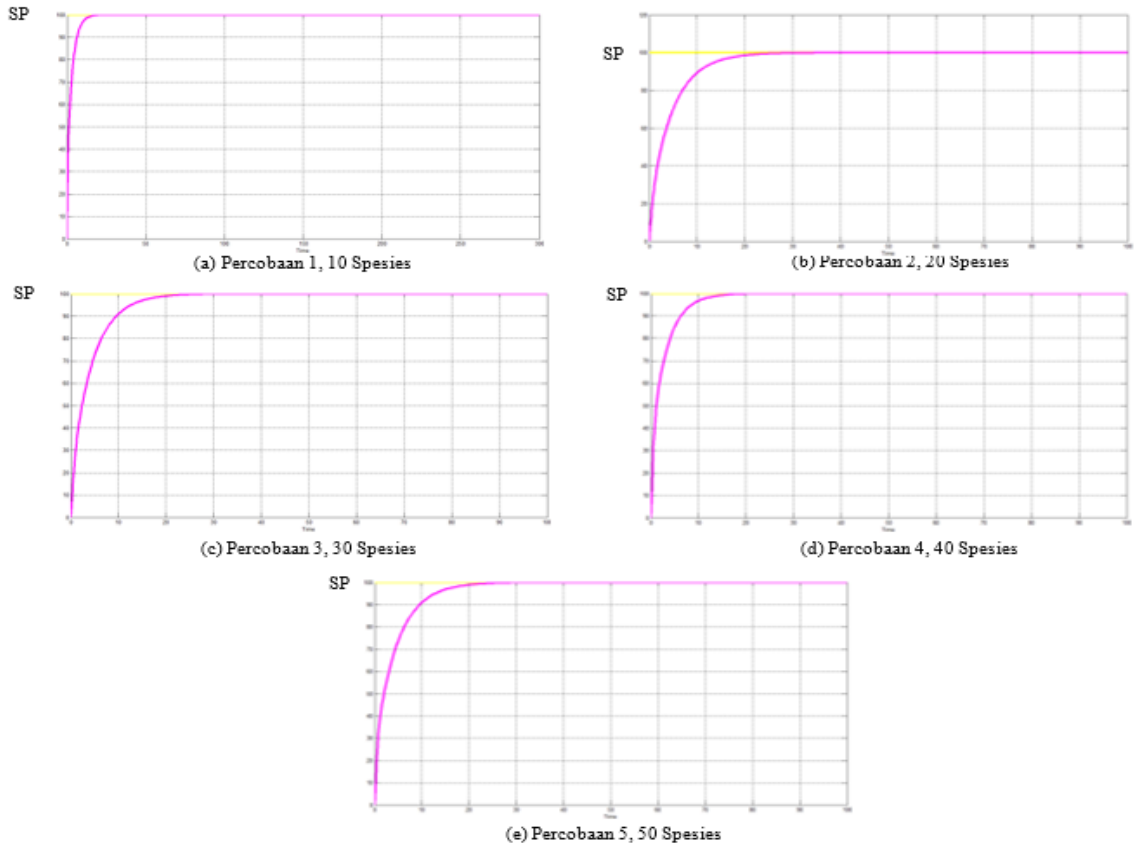
3.1. Pengujian Parameter DE 1

Pada pengujian parameter DE 1 ini, dilakuka beberapa variasi nilai spesies/iterasi dengan mengubah nilai setiap pengujiannya. Pada setiap perubahan nilai yang dilakukan terhadap jumlah nilai variasi tersebut menentukan durasi waktu pencarian untuk mendapatkan hasil parameter *Kp*, *Ki* dan *Kd*. Semakin besar nilai maka pencarian membutuhkan waktu yang lebih lama dalam pencariannya untuk mendapatkan nilai yang paling terbaik.

Tabel 2. Pengujian jumlah spesies.

Pengujian Jumlah Spesies	Kp	Ki	Kd	Respons			Waktu Naik
				tr	*ts*	*ov*	
10	4,9071	4,046	0,20536	6,1887	11,5813	0	300
20	3,3344	2,388	2,5352	9,9961	18,1484	0,0014	78,3060
30	3,0314	2,5913	0,57013	9,2372	16,7424	0	100
40	5,591	4,1561	0,55475	6,1357	11,6174	0	100
50	4,122	2,714	0,42374	9,9274	17,1245	0	100

Gambar 5 menunjukkan sinyal hasil pengujian pada jumlah spesies/iterasi. Percobaan dilakukan dengan mengubah nilai percobaan dengan ukuran yang berbeda-beda yaitu 10, 20, 30, 40, dan 50. Dari hasil pengujian ini didapatkan nilai *Kp, Ki* dan *Kd* yang bervariasi. Sebagai contoh, dapat dilihat pada percobaan 2 dengan ukuran 20 didapatkan nilai melebihi batas nilai Set Poin dengan nilai *overeshoot* 0,0014. Hasil sinyal uji menunjukkan bahwa algoritma memberikan kinerja yang cukup baik dan cukup mudah diterapkan dalam kinerja PID.



Gambar 5. Sinyal hasil pengujian jumlah spesies.

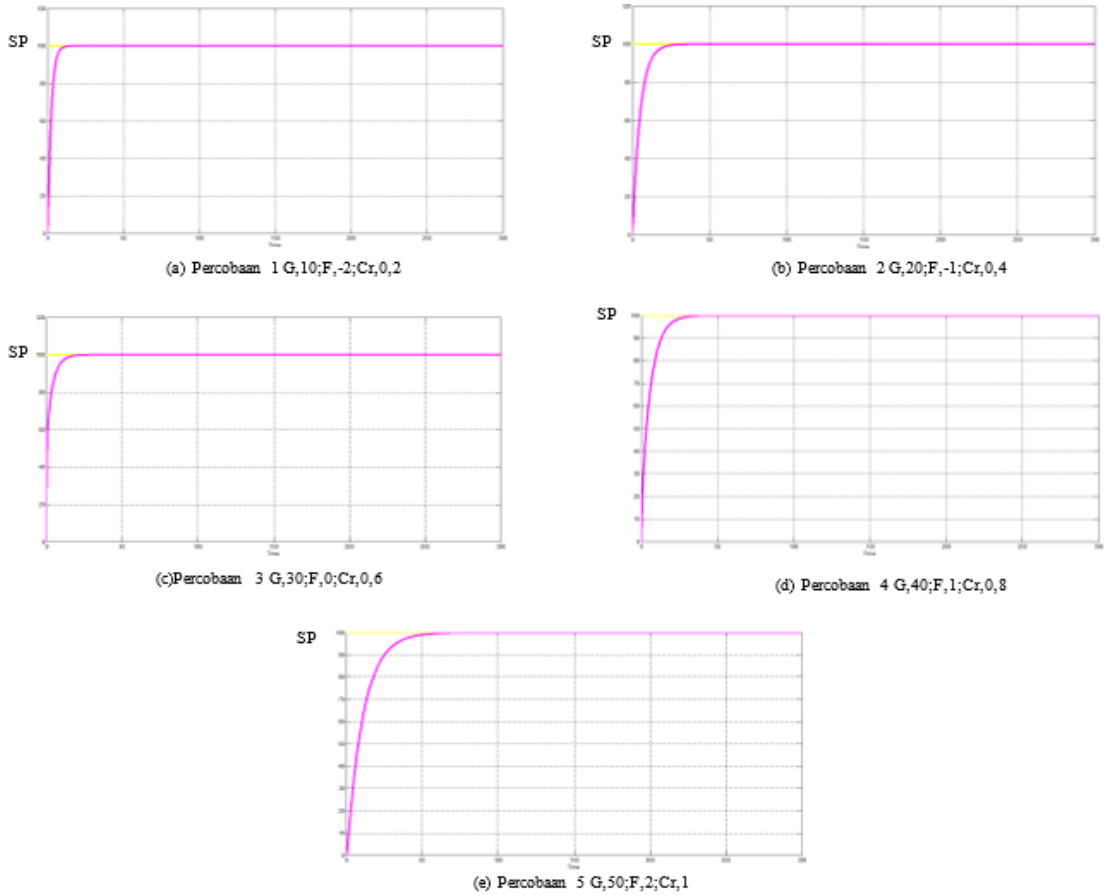
3.2. Pengujian Parameter DE 2

Pada pengujian 2 ini dilakukan variasi pada parameter kontrol DE, dengan mengubah nilai setiap pengujiannya. Pada setiap perubahan nilai yang dilakukan jumlah nilai variasi menentukan nilai Kp, Ki, Kd dan nilai respons pada sinyal hasil.

Tabel 3. Pengujian parameter kontrol DE.

Pengujian Parameter Kontrol Diferensial Evolusi			Jumlah Spesies	*Kp*	*Ki*	*Kd*	Respons			Waktu Naik
G	*F*	*Cr*					*tr*	*ts*	*ov*	
10	-2	0,2	10	2,081	4,0894	2,2664	4,8113	8,3788	0,0011	27,3899
20	1	0,4	10	1,6314	2,0075	3,7542	10,9361	18,9734	0,0018	278,9377
30	0	0,6	10	12,3388	4,9414	1,6967	6,0039	12,7824	0,0102	297,3831
40	1	0,8	10	2,2799	1,8574	0,41406	12,7733	22,8959	0	300
50	2	1	10	0	0,8828	0	23,5587	42,5400	0	300

Pada Gambar 6. diperoleh sinyal hasil pada parameter kontrol DE. Percobaan dilakukan dengan mengubah nilai *G*, *F*, dan *Cr*. Hasil *Kp*, *Ki* dan *Kd* ditunjukkan pada Tabel 3. Berdasarkan hasil percobaan dengan mengubah nilai parameter DE didapatkan hasil yang cukup baik. Hasil sinyal uji menunjukkan bahwa algoritma masih memberikan nilai terbaik pada hasil pencariannya. Mengingat nilai pengujian yang di rubah adalah parameter DE sendiri, masih memberikan nilai yang cukup baik. Hasil yang ditampilkan adalah nilai yang paling terbaik pada proses pencariannya.



Gambar 6. Hasil sinyal pengujian parameter kontrol DE.

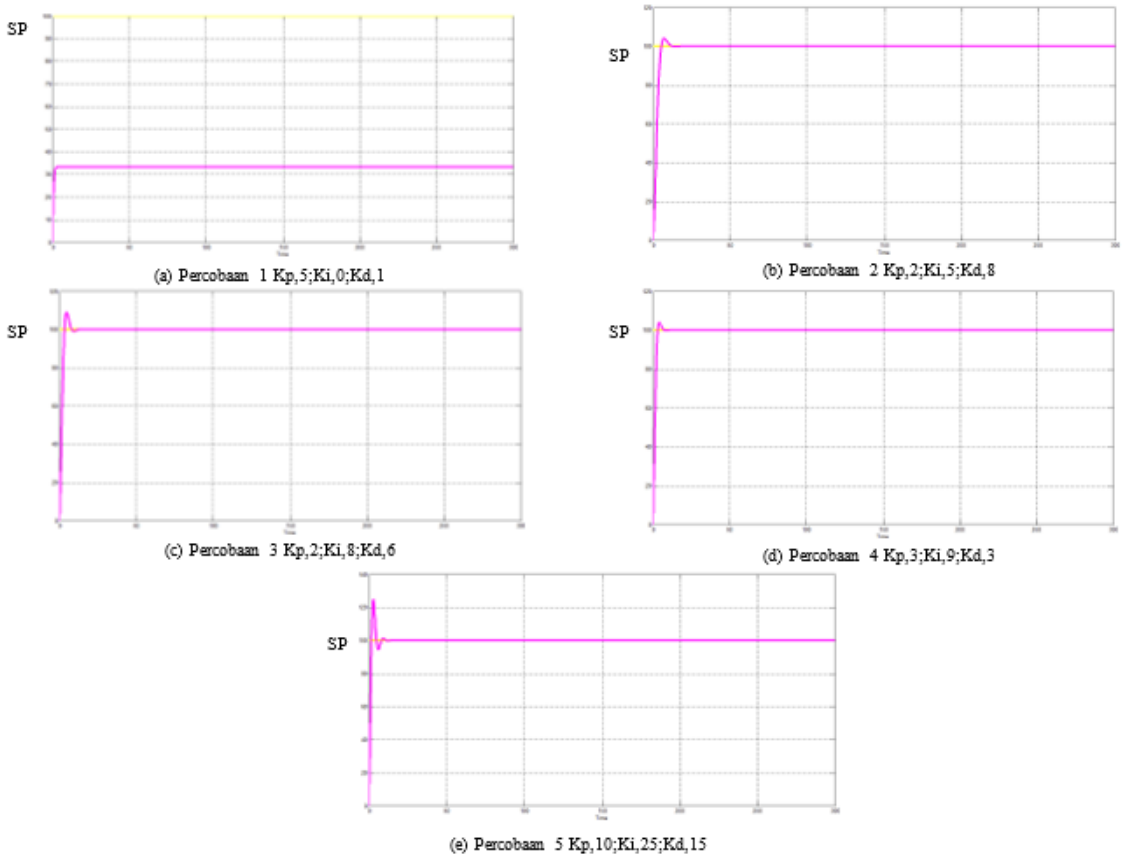
3.3. Pengujian *Trial* dan *Error* Tuning PID

Pada pengujian ini, dilakukan variasi terhadap nilai gain/*tuning* PID DE, dengan mengubah nilai setiap pengujiannya. Pada setiap perubahan nilai Kp, Ki, Kd dimasukkan secara manual atau dengan cara coba-coba, dengan mengubah *gain*/*tuning* untuk mendapatkan nilai parameter yang baik.

Tabel 4. Pengujian manual tuning PID.

Pengujian Parameter Kontrol Diferensial Evolusi			Jumlah Spesies	*Kp*	*Ki*	*Kd*	Respons			Waktu Naik
G	*F*	*Cr*					*tr*	*ts*	*ov*	
10	0,5	0.8	10	5	0	1	0	0	0	8,8661
10	0,5	0,8	10	2	5	8	3,4943	9,4728	4,0816	7,0411
10	0,5	0,8	10	2	8	6	2,2061	7,0003	9,1331	4,5003
10	0,5	0,8	10	3	9	3	1,9773	5,4363	4,0553	4,0566
10	0,5	0,8	10	10	25	15	1,1937	7,2840	24,9459	2,7176

Gambar 7. Menunjukkan sinyal hasil pada manual *tuning* DE. Percobaan dilakukan dengan mengubah nilai percobaan 1. *Kp* = 5, *Ki* = 0, *Kd* = 1. Percobaan 2. *Kp* = 2, *Ki* = 5, *Kd* = 8, Percobaan 3. *Kp* = 2, *Ki* = 8, *Kd* = 6, Percobaan 4. *Kp* = 3, *Ki* = 9, *Kd* = 3, dan Percobaan 5 . *Kp* = 10, *Ki* = 25, *Kd* = 15. Hasil Kp, Ki dan Kd ditunjukkan pada Tabel 4. Hasil sinyal uji menunjukkan nilai respons sistem terhadap *tuning* kurang baik, pencarian *tuning* manual juga bisa dilakukan dengan mengikuti kaidah pencarian Kp, Ki dan Kd.



Gambar 7. Hasil sinyal pengujian manual tuning PID.

3.4. Pengujian Trial dan Error Tuning dengan Hardware

Hasil pengujian dengan metode DE akan diimplementasikan ke dalam plan *hardware* sebagai *setting tuning Kp*, *Ki* dan *Kd*, di mana hasil terbaik DE akan digunakan sebagai nilai acuan untuk mendapatkan respons sistem pada PID.

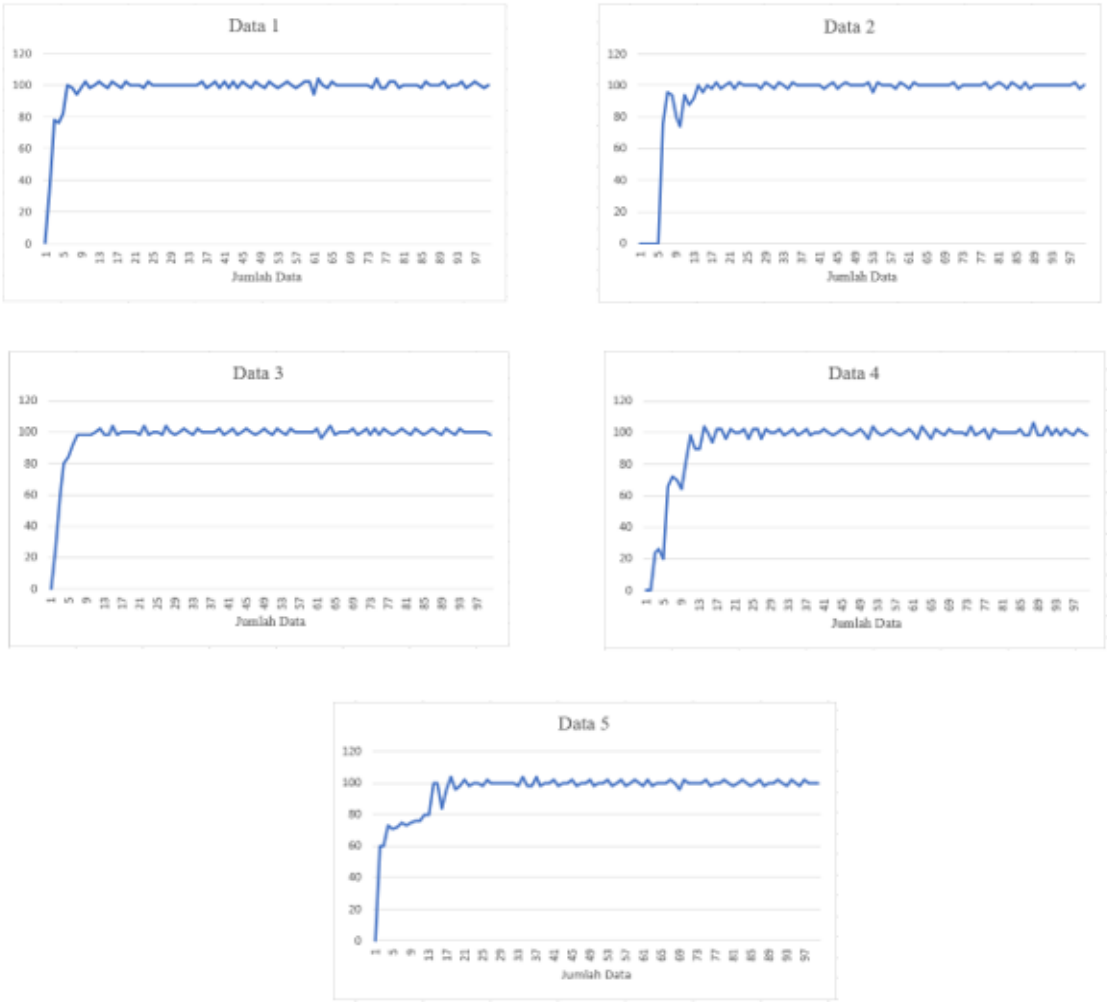
ISSN (e): 2540-9123
ISSN (p): 2502-1982

3.5. Pengujian Tuning Hasil Parameter DE 1

Pengujian *tuning* dilakukan dengan data pada Tabel 5. Data tabel diambil dari hasil terbaik Pengujian DE 1.

Tabel 5. Pengujian *tuning* hasil parameter DE 1.

NO.	*Kp*	*Ki*	*Kd*	Respons			Waktu Naik
				Tr	*ts*	*ov*	
1.	4,9071	4,046	0,20536	4,1813	75,3333	4,0000	62
2.	3,3344	2,388	2,5352	1,5684	53,3333	2,0000	18
3.	3,0314	2,5913	0,57013	4,3654	64,3333	4,0000	15
4.	5,591	4,1561	0,55475	8,0833	91,0000	6,0000	88
5.	4,122	2,714	0,42374	12,3333	69,3333	4,0000	18



Gambar 8. Sinyal respons pengujian *tuning* hasil parameter DE 1.

Hasil sinyal respons pengujian *tuning* DE 1 menunjukkan sinyal respons yang cukup baik pada plan. Hal ini ditunjukkan pada Gambar 8 dengan hasil respons sistem yang stabil dengan mencapai nilai set poin pada putaran motor dc.

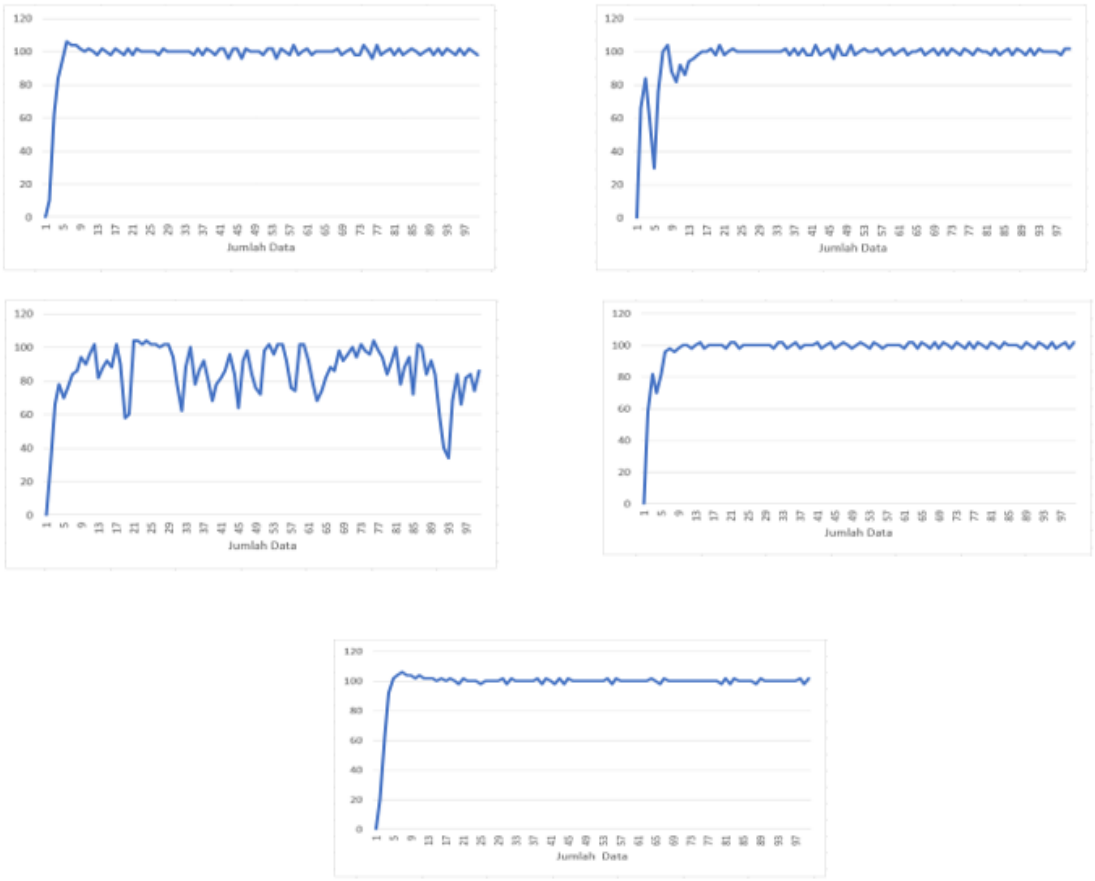
3.6. Pengujian Tuning Hasil Parameter DE 2

Pengujian *tuning* dilakukan dengan data pada Tabel 6. Data tabel diambil dari hasil terbaik Pengujian DE 2. Hasil sinyal respons pengujian *tuning* DE 2 menunjukkan hasil yang cukup baik dengan parameter tertentu, mengingat parameter DE dilakukan ubah variasi pada pencariannya.

ISSN (e): 2540-9123
ISSN (p): 2502-1982

Tabel 6. Pengujian *tuning* hasil parameter DE 2.

No.	*Kp*	*Ki*	*Kd*	Respons			Waktu Naik
				tr	*ts*	*ov*	
1.	2,081	4,0894	2,2664	2,5000	77,3333	6,0000	6
2.	1,6314	2,0075	3,7542	5,4318	50,3333	4,0000	8
3.	12,3388	4,9414	1,6967	7,1429	0	4,0000	21
4.	2,2799	1,8574	0,41406	4,3990	9	2,0000	14
5.	0	0,88286	0	2,4333	12	6,0000	7



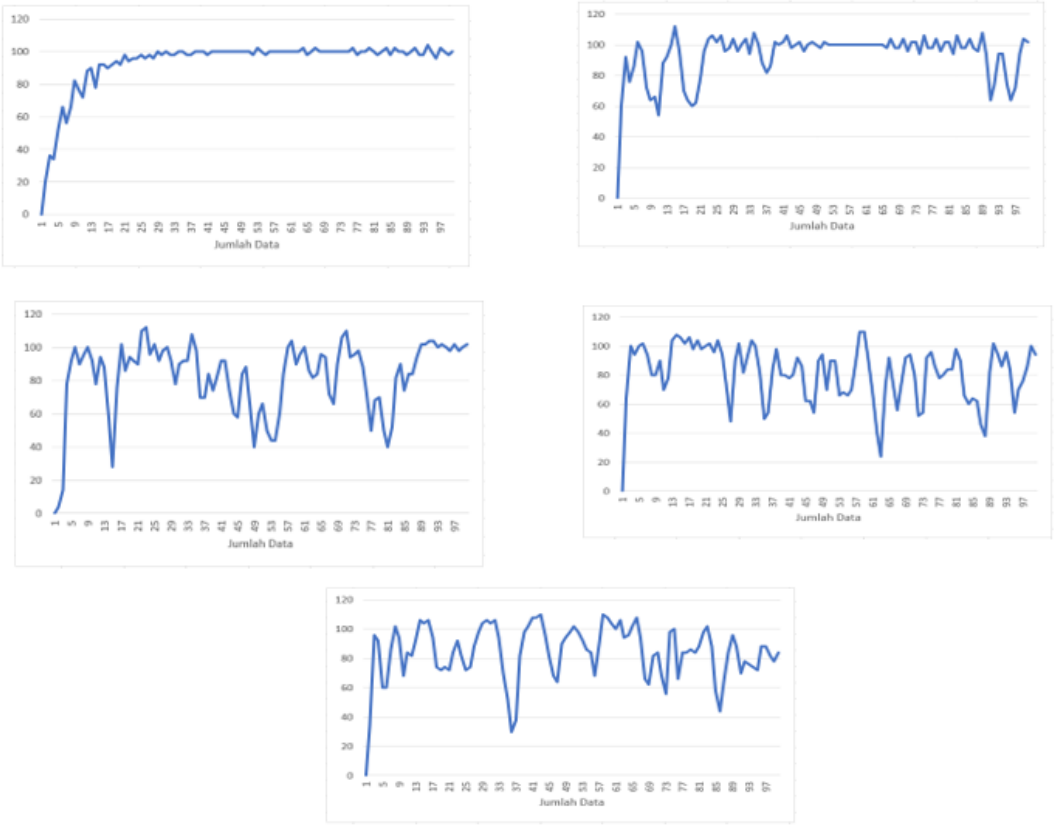
Gambar 9. Sinyal respons pengujian *tuning* hasil parameter DE 2.

3.7. Pengujian Trial dan Error Tuning PID pada Hardware

Pengujian *tuning* dilakukan dengan data pada Tabel 7. Data tabel diambil dari hasil terbaik Pengujian hasil manual. Hasil pengujian *tuning* manual menunjukkan respons sistem yang tidak baik, ditunjukkan seperti pada Gambar 10 dimana nilai *tuning* harus dilakukan beberapa kali percobaan untuk mendapatkan nilai yang baik pada *setting tuning gain Kp*, *Ki* dan *Kd*.

Tabel 7. Pengujian *trial* dan *error tuning* PID pada *hardware*.

No.	Kp	Ki	Kd	Respons			Waktu Naik
				tr	*ts*	*ov*	
1.	5	0	1	11,5000	96,3333	4,0000	94
2.	2	5	8	1,7708	100	12,0000	15
3.	2	8	6	2,2571	92,5000	12,0000	23
4.	3	9	3	1,5660	50	10,0000	58
5.	10	25	15	1,6091	22,2222	10,0000	43



Gambar 10**.** Sinyal Respons trial *dan* error tuning *PID pada* hardware.

4. Kesimpulan

Setelah dilakukan penelitian dan pengujian dengan menggunakan metode ini, dapat disimpulkan bahwa metode ini terintegrasi dengan baik yang artinya tuning PID dengan menggunakan metode DE dapat memberikan pengoptimalan dan efektivitas terhadap PID mikrokontroler. Proses pencarian dilakukan dengan pendekatan metode DE yang membuktikan bahwa hasil parameter DE 1 dan DE 2 pada matlab memberikan nilai terbaik pada pencariannya dengan size=10 dari hasil DE 1 dengan nilai *Kp* = 4,9071; *Ki* = 4,9071; *Kd* = 0,20536 dan nilai rad = 0. DE 2 pada pengujian dilakukan dengan mengubah nilai parameter DE sendiri dengan size = 10 dari hasil De 2 dengan nilai *Kp* = 2,081; *Ki* = 4,0894; *Kd* = 2,2664 dan rad = 0,0011. Hasil parameter yang digunakan adalah nilai terbaik pada pencariannya yang digunakan pada implementasi motor dc.

Ucapan Terima Kasih

Pada kesempatan ini penulis mengucapkan terimakasih yang sebanyak-banyaknya kepada bapak pembimbing yang sudah mendukung dan melakukan riset dalam penelitian ini.

Referensi

[1]. A. Gosh, S. Das, R. Mallipeddi, A. K. Das, and S. S. Dash, "A Modified Differential Evolution with Distance-based Selection for Continuous Optimization in Presence of Noise," IEEE Access, vol. 5, pp. 26944–26964, 2017, doi: 10.1109/ACCESS.2017.2773825.

[2]. A. Ma'Arif, H. Nabila, Iswanto, and O. Wahyunggoro, "Application of Intelligent Search Algorithms in Proportional-Integral-Derivative Control of Direct-Current Motor System," J. Phys. Conf. Ser., vol. 1373, no. 1, 2019, doi: 10.1088/1742-6596/1373/1/012039.

[3]. B. Guan, Y. Zhao, and Y. Li, "DESeeker: Detecting Epistatic Interactions Using a Two-Stage Differential Evolution Algorithm," IEEE Access, vol. 7, pp. 69604–69613, 2019, doi: 10.1109/ACCESS.2019.2917132.

[4]. B. Yang, Z. Zhang, and Z. Sun, "Computing Nonlinear LTS Estimator Based on a Random Differential Evolution Strategy," Tsinghua Sci. Technol., vol. 13, no. 1, pp. 59–64, 2008, doi: 10.1016/S1007-0214(08)70010-5.

[5]. FAHMIZAL, F. et al. (2019) 'Robot Inverted Pendulum Beroda Dua (IPBD) dengan Kendali Linear Quadratic Regulator (LQR)', ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika. doi: 10.26760/elkomika.v7i2.224, 7(2), pp. 224-237.

[6]. L. U. Factor, "Congestion Management of Power System with Interline Power Flow Controller Using Disparity," vol. 1, no. 3, pp. 76–85, 2015M. G. Villarreal-Cervantes, A. Rodriguez-Molina, C. V. Garcia-Mendoza, O.

[7]. Penaloza-Mejia, and G. Sepulveda-Cervantes, "Multi-Objective On-Line Optimization Approach for the DC Motor Controller Tuning Using Differential Evolution," IEEE Access, vol. 5, pp. 20393–20407, 2017, doi: 10.1109/ACCESS.2017.2757959.

[8]. R. A. Sarker, S. M. Elsayed, and T. Ray, "Differential evolution with dynamic parameters selection for optimization problems," IEEE Trans. Evol. Comput., vol. 18, no. 5, pp. 689–707, 2014, doi: 10.1109/TEVC.2013.2281528.

[9]. W. Lianghong, W. Yaonan, Z. Shaowu, and T. Wen, "Design of PID controller with incomplete derivation based on differential evolution algorithm," J. Syst. Eng. Electron., vol. 19, no. 3, pp. 578–583, 2008, doi: 10.1016/s1004-4132(08)60123-1.

[10]. W. Shahzad, Y. Ahsan, and E. Ahmed, "Drug Design and Discovery using Differential Evolution," *J. Appl. Environ. Biol. Sci.*, vol. 6, no. 12, pp. 16–26.

[11]. Y. L. Li, Z. H. Zhan, Y. J. Gong, W. N. Chen, J. Zhang, and Y. Li, "Differential Evolution with an Evolution Path: A DEEP Evolutionary Algorithm," IEEE Trans. Cybern., vol. 45, no. 9, pp. 1798–1810, 2015, doi: 10.1109/TCYB.2014.2360752.