

# SISTEM KENDALI KECEPATAN MOTOR DC MENGUNAKAN HYBRID PID-PSO

**Karan Gunawan<sup>1</sup>, Sugiarto<sup>2</sup>, Joko Prasajo<sup>3</sup>**

Program Studi Teknik Elektro, Institut Teknologi Nasional Yogyakarta

Email : [Karangunawan34@gmail.com](mailto:Karangunawan34@gmail.com)

## **Abstrak**

Motor DC merupakan mesin yang sering dipakai di industri. industri pada pengaplikasiannya motor dc berperan untuk mengendalikan putaran pengerolan baja lembaran dan akan menghasilkan baja lembaran tipis. Untuk mengendalikan kecepatan putar motor dc memerlukan kontrol yang tepat salah satu pengendalian yang populer di perindustrian yaitu dengan kendali PID karena PID terbukti dapat memberikan performa kontrol yang baik dan mempunyai algoritma sederhana yang mudah dipahami. Akan tetapi PID mempunyai kekurangan yaitu dengan adanya *overshoot*. Untuk mengatasinya dibutuhkan supervisi pada sistem kendali PID untuk mengurangi nilai *overshoot* yang terjadi dengan kecerdasan buatan seperti PSO yang dianggap tepat untuk permasalahan tersebut karena mempunyai kelebihan kecepatan dan sederhana. Dengan menggunakan PSO kita dapat mencari parameter PID yang memberikan kinerja yang baik serta respon yang cepat dalam mengatur prosesnya.

Kata Kunci : Motor Dc, Proportional Integral Derivative, Particle Swarm Optimization, Overshoot, Matlab.

## **Abstract**

*DC motor is a machine that is often used in industry. industry in its application, the dc motor plays a role in controlling the rotation of sheet steel rolling and will produce thin sheet steel. To control the rotational speed of the DC motor requires proper control, one of the popular controls in the industry is PID control because PID is proven to provide good control performance and has a simple algorithm that is easy to understand. However, PID has a drawback, namely the presence of overshoot. To overcome this, supervision of the PID control system is needed to reduce the overshoot value that occurs with artificial intelligence such as PSO which is considered appropriate for these problems because it has the advantages of speed and simplicity. By using PSO we can find PID parameters that provide good performance and fast response in regulating the process.*

*Keywords: Dc motor; Proportional Integral Derivative, Particle Swarm Optimization, Overshoot, Matlab.*

## 1. PENDAHULUAN

Motor DC merupakan mesin yang sering dipakai di industri. Salah satu yang menggunakan mesin tersebut yaitu di perindustrian pabrik baja, pada pengaplikasiannya motor dc berperan untuk mengendalikan putaran pada *cold rolling mill* yang merupakan proses pengerolan baja lembaran dan akan menghasilkan baja lembaran tipis. Akan tetapi kecepatan putar *motor dc* akan mengalami penurunan dan tidak konstan dikarenakan adanya pembebanan pada *cold rolling mill* sehingga akan menghasilkan lembaran baja yang kurang baik, untuk mengatasi kecepatan putar yang mengalami penurunan maka motor dc menggunakan sistem pengendalian agar putaran yang dihasilkan tetap optimal.

Salah satu pengendalian yang populer di Perindustrian dan sudah digunakan untuk berbagai sistem kendali khususnya untuk pengendalian kecepatan putaran yaitu dengan kendali PID. Sudah sejak lama kendali PID terbukti dapat memberikan performa kontrol yang baik dan mempunyai algoritma sederhana yang mudah dipahami.

Dibalik kemudahan dalam menggunakan kendalinya, PID mempunyai kekurangan yaitu dengan adanya *overshoot*. Untuk mengatasi masalah ini, dibutuhkan supervisi pada sistem kendali PID untuk mengurangi nilai *overshoot* yang terjadi dengan kecerdasan buatan seperti PSO yang dianggap tepat untuk permasalahan tersebut karena mempunyai kelebihan kecepatan dan sederhana. Dengan menggunakan PSO kita dapat mencari parameter PID yang memberikan kinerja yang baik serta respon yang cepat dalam mengatur prosesnya.

## 2. METODE PENELITIAN

**Motor DC** atau motor arus searah adalah mesin yang mengubah energi listrik DC menjadi energi mekanis, konstruksi motor DC sangat mirip dengan generator DC. Mesin yang bekerja baik sebagai generator baik pula bekerja sebagai motor. Suatu perbedaan di dalam konstruksinya sebaiknya diperhatikan antara motor dan generator. Karena motor sering kali dioperasikan di lokasi yang mungkin mudah mendapatkan kerusakan mekanis debu, lembab atau korosif, maka motor biasanya lebih tertutup rapat dibandingkan generator. Pada motor arus searah pengaturan putarannya mudah dan dapat diatur dalam daerah yang sangat lebar. Generator arus searah bekerja berdasarkan prinsip hukum *faraday*, sedangkan arah gaya yang menimbulkan kopel pada arus searah berdasarkan kaidah tangan kiri. Untuk mesin arus searah berlaku suatu persamaan umum :

$$E = vt - Ia \cdot Ra$$

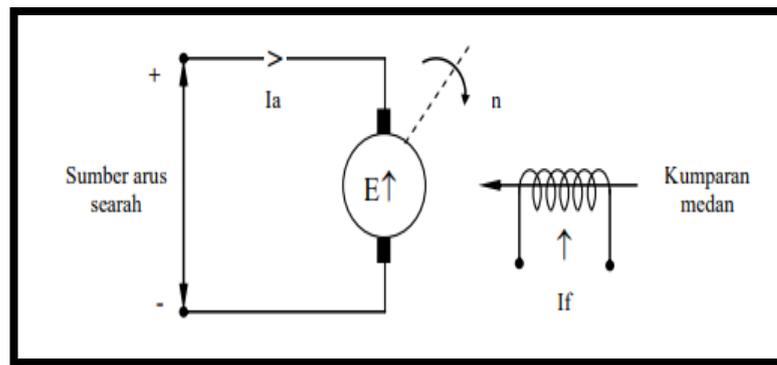
Atau

$$t = E + Ia \cdot Ra$$

dengan :

- E :Tegangan jangkar / ggl lawan (*Volt*)
- Vt :Tagangan sumber/ tegangan jala-jala (*volt*)
- Ia :Arus jangkar (*Ampere*)
- Ra :Tahanan jangkar (*Ohm*).

Dalam kehidupan kita sehari-hari motor DC dapat kita lihat pada motor *starter* mobil, pada *tape recorder*, pada mainan anak-anak dan sebagainya. Sedangkan pada pabrik-pabrik motor motor DC kita jumpai pada *elevator*, *konveyer* dan sebagainya. Untuk lebih jelasnya dapat dilihat pada gambar 2.1 berikut ini :



2. 1 Gambar arus searah

Pada motor a

= konstan.

**PID (*Proportional-Integral-Derivative*)** merupakan sistem kontrol untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Parameter kontrol PID terdiri atas kontrol *Proportional*, *Integral* dan *Derivative*. Dimana penggunaannya bisa hanya digunakan kontrol P atau PI atau PID, tergantung respon yang kita inginkan

Beberapa *terminologi* dasar untuk mempermudah memahami PID *Controller* :

- a. *Error* : Merupakan jumlah perangkat yang tidak melakukan sesuatu dengan benar.
- b. *Proportional* (P) : Istilah proporsional berbanding lurus dengan kesalahan saat ini, pengendali *proportional* juga bertindak sebagai penguat yang mampu mengubah *output* dari sistem secara *proportional* tanpa memberikan efek dinamik pada kinerja pengendali tersebut.
- c. *Integral* (I) : Istilah *Integral* bergantung pada kesalahan kumulatif yang dibuat selama jangka waktu (t), pengendali *integral* merupakan pengendali yang berfungsi untuk memperbaiki respon *steady state* dari sistem sehingga pengendali ini mampu memperkecil *error* sistem.

- d. *Derivative (D)* : Istilah turunan tergantung laju perubahan kesalahan. Pengendali ini merupakan suatu pengendali yang berfungsi untuk memperbaiki respon *transien* dari sistem.
- e. *Constant (factor)* : Setiap istilah P, I, D dimasukkan dalam kode dengan mengalikannya terhadap konstanta masing-masing.
  - 1. P-faktor ( $K_p$ ) : Nilai konstan yang digunakan untuk menambah atau mengurangi dampak *Proportional*.
  - 2. D-faktor ( $K_d$ ) : Nilai konstan yang digunakan untuk menambah atau mengurangi dampak dari *Derivatif*.
  - 3. I-faktor ( $K_i$ ) : Nilai konstan yang digunakan untuk menambah atau mengurangi dampak dari *Integral*.

***Particle Swarm Optimization***, yang disingkat sebagai PSO didasarkan pada perilaku sebuah kawanan serangga, seperti semut, rayap. Algoritma PSO sendiri meniru perilaku sosial organisme ini. Perilaku sosial terdiri dari tindakan individu dan pengaruh dari individu-individu lain dalam suatu kelompok. Dalam PSO, kawanan diasumsikan mempunyai ukuran tertentu atau tetap dengan setiap partikel posisi awalnya terletak di suatu lokasi yang acak dalam ruang multidimensi. Sebagai contoh, misalnya perilaku burung-burung dalam kawanan burung. Meskipun setiap burung memiliki keterbatasan dalam hal kecerdasan, biasanya seekor burung akan mengikuti kebiasaan (*rule*) seperti berikut:

- a. Posisi seekor burung tidak berada terlalu dekat dengan burung yang lain.
- b. Burung tersebut akan mengarahkan terbangnya ke arah rata-rata ke seluruh kawanan burung.
- c. Akan memposisikan diri dengan rata-rata posisi burung yang lain dengan menjaga sehingga jarak antar burung dalam kawanan itu tidak terlalu jauh.

Dengan demikian perilaku kawanan burung akan didasarkan pada kombinasi dari 3 faktor simpel berikut:

- a. *Kohesi* : terbang bersama
- b. *Separasi* : jarak jangan terlalu dekat
- c. *Penyesuaian* : mengikuti arah Bersama

Jadi PSO sendiri dikembangkan berdasarkan pada model berikut:

- a. Ketika seekor burung mendekati target atau makanan (atau bisa minimum atau maksimum suatu fungsi tujuan) secara cepat 26 mengirim informasi kepada burung-burung yang lain dalam kawanan tertentu.
- b. Burung yang lain akan mengikuti arah menuju ke makanan tetapi tidak secara langsung.
- c. Ada komponen yang tergantung pada pikiran setiap burung, yaitu memorinya tentang apa yang sudah dilewati pada waktu sebelumnya.

Kemudian terdapat dua faktor yang memberikan karakter terhadap status partikel pada ruang pencarian yaitu posisi partikel dan kecepatan partikel. Algoritma PSO *orisinil* dijelaskan dalam persamaan berikut:

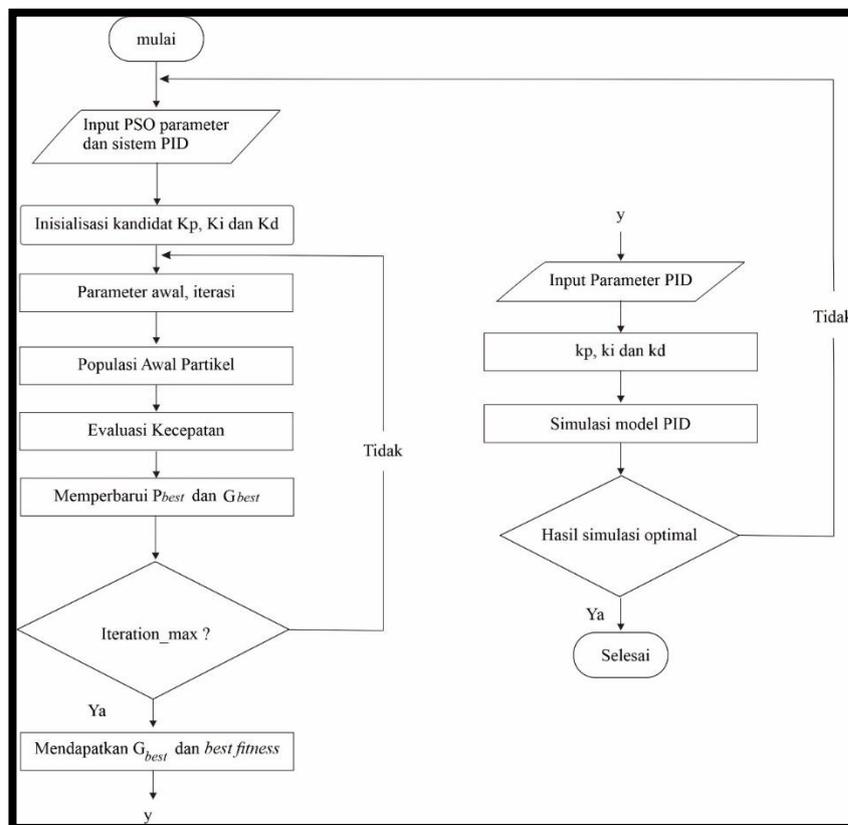
$$Xi(t) = xi1, xi2(t), \dots xin(t)$$

$$vi(t) = vi1(t), vi2(t), \dots Vin(t)$$

dengan,

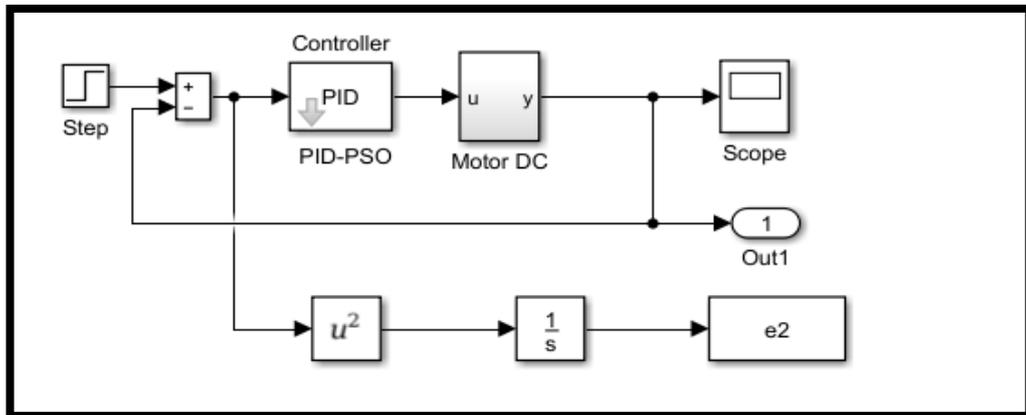
- X : posisi partikel
- V : kecepatan partikel
- i : indeks partikel
- t : iterasi ke-t
- N : ukuran dimensi ruang.

Alur penelitian ini dapat dilihat dengan detail pada diagram alir tahapan pengoptimalan dalam penggunaan simulator penalaan parameter kendali PID dengan kecerdasan buatan berbasis PSO yang terlihat pada gambar ini.

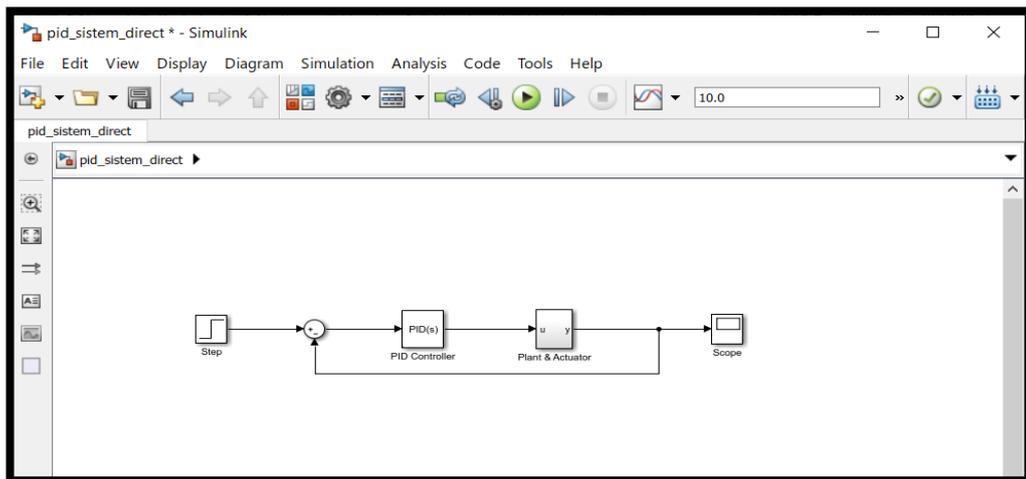


2. 2 alur diagram penelitian

Untuk membandingkan nilai respon mana yang terbaik dengan nilai parameter yang telah ditentukan pada penggunaan sistem kendali untuk motor DC. Pada gambar 4.1 terlihat model *Simulink* untuk melakukan penalaan parameter kendali dengan sistem PID-PSO dan gambar 4.2 yaitu gambar *Simulink* kendali PID.



2. 3 Gambar Simulink kendali PID-PSO



2. 4 Gambar Simulink kendali PID

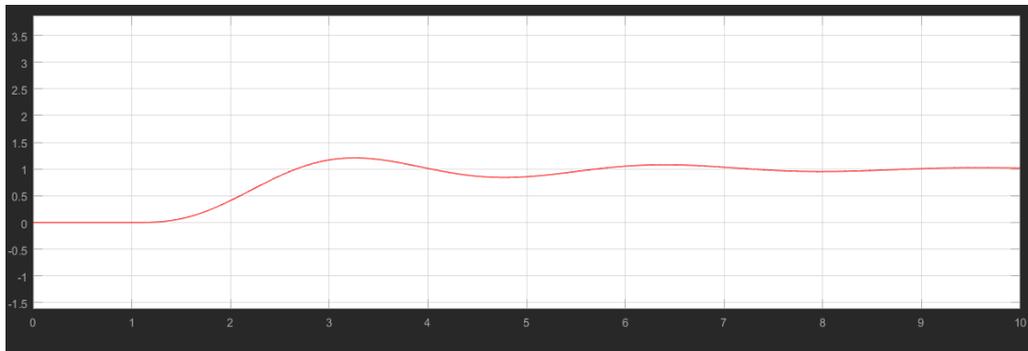
### 3. HASIL DAN PEMBAHASAN

Pada tahap ini pengujian kendali menggunakan PID-PSO sebagai sistem simulink dari kendali motor dc sebagaimana dilihat pada gambar 4.1 yang Dimana merupakan kendali dengan *hybrid* PID-PSO dan pada gambar 4.2 merupakan sistem kendali dengan PID . Adapun hasil uji respon dan simulasi terdapat pada berikut:

- a. Pada tahap ini dilakukan dengan 3 kali pengujian dengan masing-masing input

No	Kp	Kd
1.	1	1
2.	1	2
3.	2	3

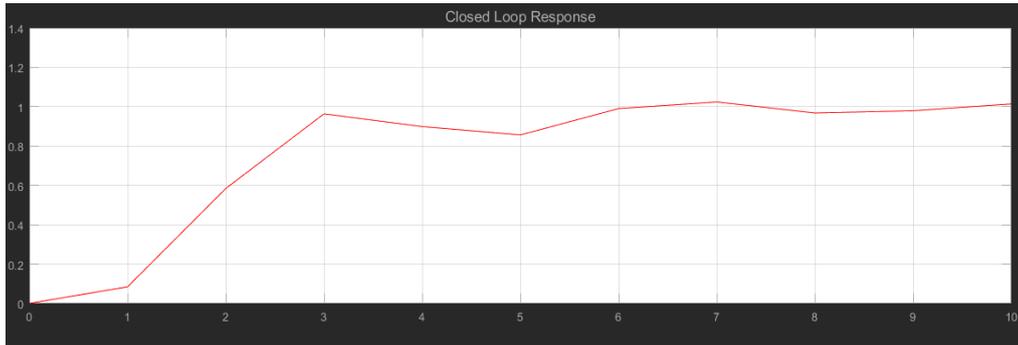
Sehingga menghasilkan nilai *presentase overshout* yang terlihat pada gambar 3.1 dan 3.3 yang menampilkan hasil dalam bentuk grafik dan pada gambar 3.2 dan 3.4 yaitu yang menghasilkan data dari gambar grafik tersebut.



3. 1 Grafik tanggapan sistem menggunakan PID

▼ Overshoots / Undershoots	
+ Preshoot	0.595 %
+ Overshoot	18.452 %
+ Undershoot	17.663 %
+ Settling Time	--
- Preshoot	--
- Overshoot	--
- Undershoot	--
- Settling Time	--

3. 2 Data respon Grafik kendali PID



3. 3 Grafik tanggapan sistem menggunakan PID-PSO

▼ Overshoots	
Average slope of all falling edges	
+ Preshoot	0.515 %
+ Overshoot	2.577 %
+ Undershoot	2.710 %
+ Settling Time	--
- Preshoot	--
- Overshoot	--
- Undershoot	--
- Settling Time	--

3. 4 Data Grafik kendali sistem PID-PSO

Pengujian respon dilakukan dengan percobaan 3 kali, hasilnya setiap parameter yang menggunakan kendali PID-PSO memiliki keunggulan nilai *presentase overshoot* yang lebih baik di setiap percobaannya dibandingkan dengan sistem kendali yang hanya menggunakan *controller* PID.

#### 4. KESIMPULAN DAN SARAN

**Kesimpulan** yang dapat diambil dari hasil uji simulasi dan perancangan kendali motor dc dengan menggunakan kendali PID dan PSO yaitu :

1. Penambahan *Supervisi* PSO pada kendali PID menghasilkan *Overshoot* yang kecil. Pada percobaan pertama *overshoot* pada sistem menggunakan PID murni 18,452% dan setelah ditambahkan PSO *overshoot* jadi 2,577%, percobaan kedua *overshoot* menggunakan PID murni 74.561% dan 4.737% setelah ditambahkan PSO, pada percobaan ketiga nilai *overshoot* dengan PID 25.633% dan berkurang ketika ditambahkan PSO 5.851%.
2. KP pengurangan *overshoot* pada kendali PID dihasilkan dengan masing-masing percobaan berbeda, Dimana pada percobaan pertama  $K_p : 1$ ,  $K_d : 1$ , Pada

percobaan kedua nilai  $K_p$  : 1,  $K_d$  : 2, dan pada percobaan ketiga  $K_p$  : 2 dan  $K_d$  : 3.

**Saran** dari penelitian ini telah dirancang dan dilakukan dengan sesuai penelitian, akan tetapi perancangan sistem masih perlu dikembangkan lebih baik lagi yaitu dengan cara pembuatan simulasi dengan menggabungkan ke dalam GUI agar dapat melakukan proses hanya sekali dan langsung mendapatkan hasil yang sederhana.

#### 4. DAFTAR ISI

- ADHIM, A. (2016). *Pid Auto Tuning Menggunakan Pso Pada Sistem Fotovoltaik Penjejak Matahari Dua-Sumbu*.
- Aggrawal, A., Mishra, A. K., & Zeeshan, A. (2014). Speed Control of DC Motor Using Particle Swarm Optimization Technique by PSO Tuned PID and FOPID. *International Journal of Engineering Trends and Technology*, 16(2), 72–79. <https://doi.org/10.14445/22315381/ijett-v16p216>
- Ali, M., Umami, I., & Sopian, H. (2016). Particle Swarm Optimization (PSO) Sebagai Tuning PID Kendali Untuk Kecepatan Motor DC. *Intake : Jurnal Penelitian Ilmu Teknik Dan Terapan*, 7(1), 10–20. <http://ejournal.undar.ac.id/index.php/intake/article/view/382>
- Elektro, S. T., Teknik, F., & Surabaya, U. N. (2018). *RANCANG BANGUN SELF BALANCING TWINROTOR MENGGUNAKAN KENDALI PID TUNING PARTICLE SWARM OPTIMIZATION Nofrialdi*.
- Laksono, A. N. J., Argo, B. D., Hendrawan, Y., & Al Riza, D. F. (2014). Pemodelan Dan Optimasi Sistem Kontrol Pada Multiple Effect Evaporator Dengan Menggunakan Particle Swarm Optimization. *Jurnal Keteknikan Pertanian Tropis Dan Biosistem*, 3(1), 106–111. <https://jkptb.ub.ac.id/index.php/jkptb/article/view/259>
- MA'ARIF, A., ISTIARNO, R., & SUNARDI, S. (2021). Kontrol Proporsional Integral Derivatif (PID) pada Kecepatan Sudut Motor DC dengan Pemodelan Identifikasi Sistem dan Tuning. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 9(2), 374. <https://doi.org/10.26760/elkomika.v9i2.374>
- Myrtellari, A., Marango, P., & Gjonaj, M. (2015). *Optimal Control of DC Motors Using PSO Algorithm for Tuning PID Controller*. 15–22. <https://doi.org/10.33107/ubt-ic.2015.40>
- Ruswandi Djalal, M., Teknik Mesin, J., Studi Teknik Energi, P., Negeri Ujung Pandang, P., & Perintis Kemerdekaan km, J. (2017). Optimisasi Kontrol Pid Untuk Motor Dc Magnet Permanen Menggunakan Particle Swarm Optimization. *Technology Acceptance Model*, 8, 117–122.
- Suhartono, S. S. (2012). Optimasi Parameter Kendali PID Berbasis Algoritma Particle Swarm Optimization untuk Pengendalian Kecepatan Motor Induksi Tiga Fase. *Jurnal Teknik ITS*. <http://ejournal.its.ac.id/index.php/teknik/article/view/256>