

# Tuning Parameter Pengendali PID dengan Metode Algoritma Genetik pada Motor DC

## Tuning of PID Controller Parameters with Genetic Algorithm Method on DC Motor

Eka Widya Suseno<sup>1\*</sup>, Alfian Ma'arif<sup>2\*</sup>, Riky Dwi Puriyanto<sup>3</sup>

<sup>1,2,3</sup>Teknik Elektro Universitas Ahmad Dahlan

<sup>1,2,3</sup>Jalan Ring Road Selatan, Tamanan, Banguntapan, Bantul DIY 55166

eka1600022028@webmail.uad.ac.id<sup>1\*</sup>, alfian.maarif@te.uad.ac.id<sup>2</sup>, rikydp@ee.uad.ac.id<sup>3</sup>

**Abstrak** –Saat ini, pengendali Proportional Integral Derivative (PID) digunakan secara umum untuk mendapatkan solusi optimum. Solusi dikatakan optimum apabila output di kehidupan nyata sesuai dengan output yang telah ditentukan. Oleh karena itu, pengendali adalah suatu hal yang dibutuhkan. Tantangan dalam menggunakan pengendali adalah tuning parameter untuk mencari konstanta parameter PID seperti Proporsional Gain (KP), Waktu Integral (KI) dan Waktu Derivatif (KD). Untuk memaksimalkan kinerja motor DC, pengaturan pengendali PID yang tepat merupakan hal yang sangat penting. Desain pengendali PID sebagai pengendali motor DC sudah sering dilakukan. Penggunaan pengendali PID membutuhkan pengaturan parameter yang tepat untuk mendapatkan kinerja yang optimal pada motor. Metode yang umum dalam menentukan parameter pengendali PID adalah trial and error. Namun hasil yang didapat tidak membuat pengendali PID optimal dan justru akan merusak sistem. Oleh karena itu, penelitian ini menggunakan salah satu metode penalaan parameter PID dengan menggunakan metode cerdas berbasis Genetic Algorithm (Algoritma Genetik) untuk mengoptimasi dan menentukan parameter yang tepat dari PID. Algoritma genetik adalah salah satu algoritma yang menggunakan genetika sebagai model algoritmanya. Algoritma genetik terinspirasi dari meniru proses seleksi alam, yaitu proses yang menyebabkan evolusi biologis. Konsep inilah yang diadaptasi dan diterapkan dengan baik untuk menala parameter PID. Penggunaan metode algoritma genetik dapat memberikan hasil yang lebih baik pada setiap iterasinya. Hasil penelitian menunjukkan bahwa overshoot yang dihasilkan karena adanya respon kecepatan setelah penambahan PID adalah kurang dari 10%.

**Kata Kunci:** Pengendali PID, algoritma genetik, motor DC.

**Abstract** –Currently, Proportional Integral Derivative (PID) controllers are generally used to obtain the optimum solution. The solution is said to be optimum if the output in real life matches the output determined. Therefore, the controller is needed. The challenge in using the controller is tuning parameters to find constants of PID parameters such as Proportional Gain (KP), Integral Time (KI) and Derivative Time (KD). In order to maximize the performance of a DC motor, proper PID controller settings are crucial. The design of PID controllers as DC motor controllers has often been done. The use of a PID controller requires setting the right parameters to get optimal performance on the motor. The common method for determining PID controller parameters is trial and error. However, the results obtained do not make the PID controller optimal and will actually damage the system. Therefore, this study uses one of the PID parameter tuning methods by using an intelligent method based on Genetic Algorithm to optimize and determine the appropriate parameters of PID. Genetic algorithm is an algorithm that uses genetics as a model algorithm. Genetic algorithms are inspired by imitating the process of natural selection, the process that causes biological evolution. This concept is well adapted and applied for tuning PID parameters. The

use of genetic algorithm methods can give better results in each iteration. The results showed that the resulting overshoot due to the speed response after the addition of PID was less than 10%.

**Keywords:** PID controllers, Genetic Algorithm, DC motor.

## 1. Pendahuluan

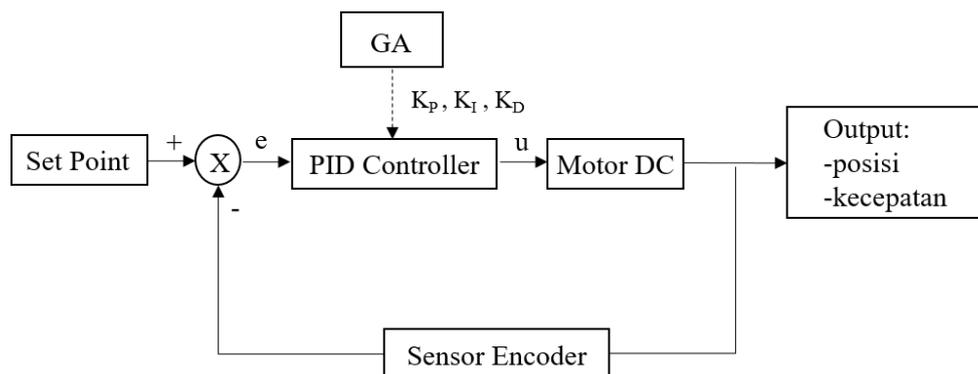
Pengendali *Proportional Integral Derivative* (PID) merupakan pengendali konvensional yang masih banyak digunakan dalam dunia industri dibandingkan jenis kendali yang lebih modern. Pengendali PID memiliki 3 parameter, yaitu Konstanta Proporsional (KP), Konstanta Integral (KI), dan Konstanta Derivatif (KD). Tujuannya adalah untuk mendapatkan respon sistem yang optimal sesuai dengan spesifikasi perancangan yang diinginkan [1]. Teknik penalaan PID ada beberapa metode, diantaranya metode Ziegler-Nichols [2], metode Fuzzy [2], dan metode Astrom-relay. Kebanyakan penalaan metode ini membutuhkan pengetahuan yang luas tentang sistem kontrol sehingga perlunya alternatif untuk otomatisasi penalaan dan praktisi yang tidak memerlukan pengetahuan sistem kontrol secara mendalam. Penggunaan pengendali PID membutuhkan pengaturan parameter yang tepat untuk mendapatkan kinerja yang optimal pada motor. Metode yang umum dalam menentukan parameter pengendali PID adalah *trial and error* [3]. Namun hasil yang didapat tidak membuat pengendali PID optimal dan justru akan merusak sistem.

Pada penelitian ini, Algoritma Genetik [4] [5] digunakan untuk menentukan parameter pengendali PID pada motor DC [6-8] sehingga pengendali dapat menghasilkan tanggapan sistem yang baik (*satisfactory response*), khususnya pada kasus-kasus *plant sample* yang memiliki orde tinggi. Algoritma Genetik merupakan suatu teknik optimasi berbasis evolusi alam, yaitu melalui proses mutasi, pindah silang dan seleksi [9]. Setelah mendapatkan model sistem, Algoritma Genetik [10] digunakan secara *off-line* dengan simulasi untuk menentukan parameter pengendali PID. Hasil yang diperoleh saat pengujian dan simulasi akan dirangkum dalam tabel pengaruh nilai parameter PID dengan metode Algoritma Genetik.

## 2. Metode Penelitian

### 2.1. Desain Sistem

Dalam desain sistem ini, digunakan algoritma genetik untuk menentukan parameter pengendali PID. Diagram blok pengendali PID dengan fungsi alih ditunjukkan pada Gambar 1.

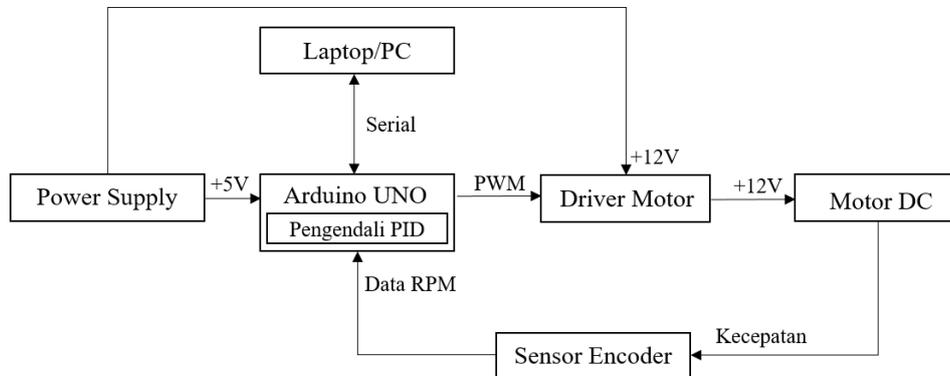


Gambar 1. Diagram blok *tuning* pengendali PID.

Gambar 1 menunjukkan bahwa *input* sistem berupa sebuah *tuning* parameter PID dengan menggunakan Algoritma Genetik (GA) sebagai metode pengendalinya untuk mencari konstanta parameter PID [11]. *Plant* yang digunakan ialah sebuah sistem kendali umpan balik yang keluarannya berupa sebuah kecepatan dari motor DC.

## 2.2. Diagram Blok *Hardware*

Selain adanya diagram blok *tuning* pengendali PID, pada penelitian ini dibuat juga diagram blok *hardware* serta *wiring hardware* untuk menunjukkan penjelasan perangkat keras yang digunakan. Diagram blok *hardware* ditunjukkan pada Gambar 2.

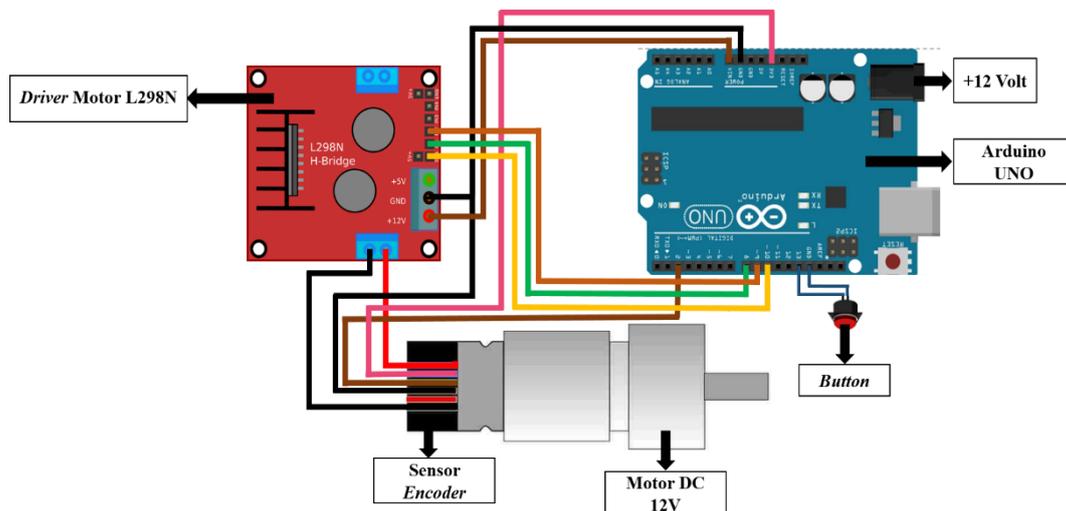


Gambar 2. Diagram blok *hardware*.

Pada Gambar 2 diperlihatkan diagram blok pemodelan motor DC yang menggunakan *board* Arduino Uno R3 [12] sebagai pengendali sistem. *Board* Arduino mendapat *supply* tegangan dari *power supply* yang sudah disiapkan sebesar 12 volt yang disalurkan pada *driver* motor untuk menggerakkan motor DC. Selanjutnya gerakan putar motor DC [8] akan dicatat oleh *encoder* yang akan diberikan pada Arduino Uno R3 agar dapat dilihat hasilnya pada *software* Arduino Ide pada menu serial *monitor* dan serial *plotter*.

## 2.3. Diagram Pengkabelan *Hardware*

*Wiring diagram* atau diagram pengkabelan dari sistem yang dibuat dapat dilihat pada Gambar 3.

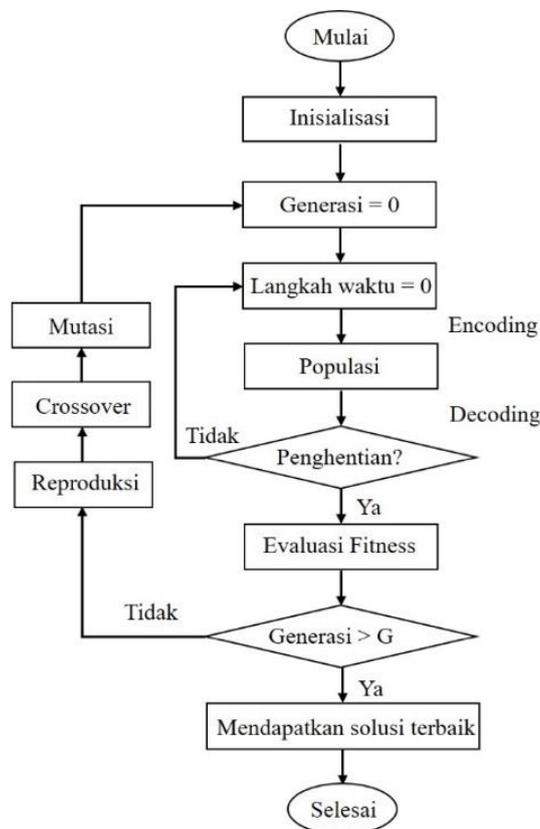


Gambar 3. *Wiring hardware*.

Perancangan perangkat keras dibuat untuk mengontrol motor DC, dan *driver* motor [13] untuk melihat respon sistem dan juga sebagai alat pengambilan data yang diperlukan untuk memodelkan motor DC menggunakan pendekatan pemodelan identifikasi sistem, data yang akan digunakan berupa tegangan dan kecepatan putar motor DC.

## 2.4. Algoritma Genetik

Prosedur Algoritma Genetik [2] digambarkan dalam bentuk *flowchart* yang ditunjukkan pada Gambar 4.



Gambar 4. *Flowchart* program algoritma Genetik.

Berdasarkan Gambar 4, standar utama Algoritma Genetik mencakup empat operator sebagai berikut:

- Seleksi GA adalah operasi seleksi yang akan memilih solusi induk.
- Selama fase reproduksi GA, individu dipilih dari populasi dan direkombinasi, menghasilkan keturunan yang pada gilirannya akan terdiri dari generasi berikutnya.
- Crossover* mengambil dua induk dan memotong *string* kromosom induk di beberapa posisi yang dipilih secara acak, untuk menghasilkan dua segmen “kepala” dan dua segmen “ekor”.
- Mutasi diterapkan pada setiap anak secara individual, setelah *crossover*.

## 3. Hasil dan Pembahasan

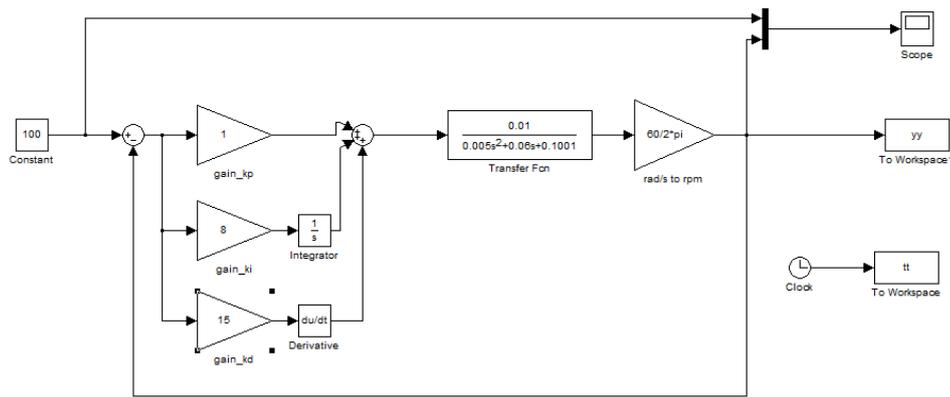
### 3.1. Pemilihan Model *Transfer Function* Motor DC

Model TF Motor DC tersebut diambil dari jurnal [1], yaitu

$$TF \text{ MDC} = \frac{0.01}{0.005s^2 + 0.06s + 0.1001} \quad (1)$$

### 3.2. Pengujian menggunakan Metode *Trial and Error*

Pengujian dengan metode *trial and error* atau metode coba-coba pada nilai KP, KI dan KD dilakukan untuk mengetahui perbandingan parameter PID yang didapatkan. Pengujian dengan metode tersebut menggunakan *plant* motor DC pada Matlab seperti ditunjukkan pada Gambar 5.



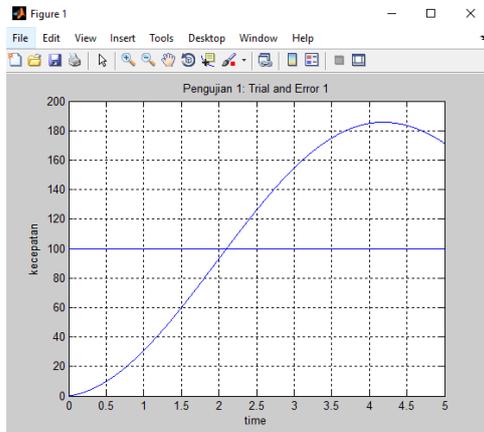
Gambar 5. Plant motor DC pada matlab.

Hasil dari pengujian menggunakan metode *trial and error* ini ditunjukkan pada Tabel 1.

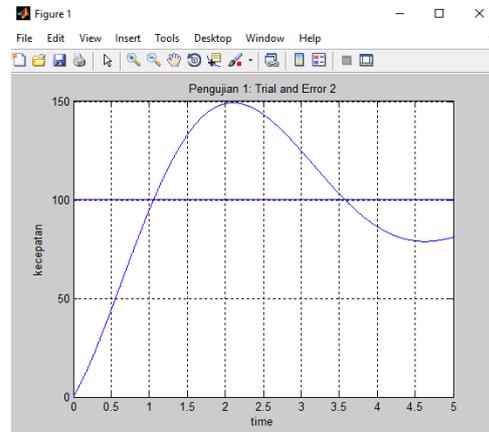
Tabel 1. Pengujian 1 menggunakan metode *trial and error*

No.	Parameter Pengendali PID			Risetime	Settling Time	Overshoot
	KP	KI	KD			
1	1	8	15	1,4440	NaN	85,6825
2	10	25	15	0,8129	NaN	49,2620

Pengujian dilakukan sebanyak lima kali dengan hasil berupa grafik pada *plant* motor DC seperti yang ditunjukkan pada Gambar 6 dan Gambar 7.



Gambar 6. Grafik data nomor 1 pengujian *trial and error*.



Gambar 7. Grafik data nomor 2 pengujian *trial and error*.

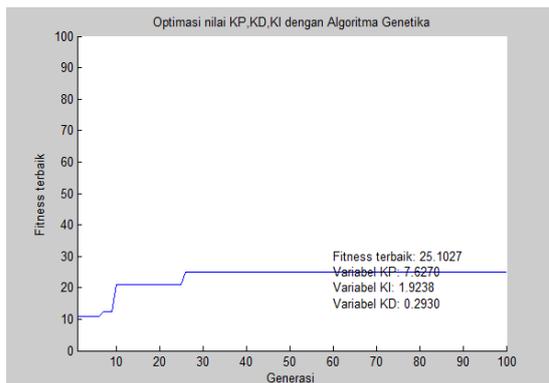
Berdasarkan pengujian yang dilakukan, diperoleh nilai *overshoot* yang sangat besar yaitu 85,6825% untuk Gambar 6 dan 49,2620% untuk Gambar 7. Nilai *settling time* yang didapat bernilai NaN. NaN adalah singkatan dari *Not a Number* (bukan angka). NaN ialah nilai tipe data numerik yang mewakili nilai yang tidak ditentukan atau tidak terwakili, terutama dalam aritmatika titik-mengambang. Dalam pengujian menggunakan metode *trial and error* ini menunjukkan bahwa semua percobaan yang dilakukan bernilai NaN dan terjadi pada data *settling time*. Pengujian menggunakan metode *trial and error* tidak efektif dalam mencari parameter PID yang diinginkan.

### 3.3. Pengujian menggunakan Metode Algoritma Genetik (GA)

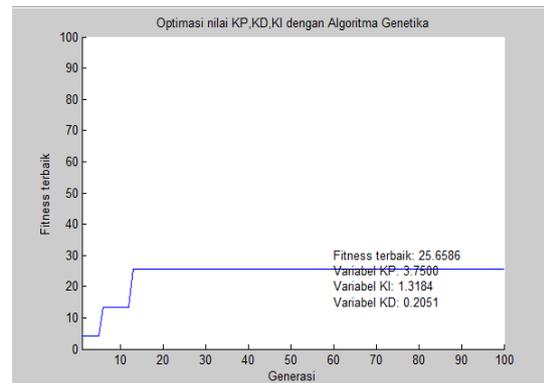
Pengujian menggunakan metode Algoritma Genetik dilakukan dengan tiga macam pengujian yang dipengaruhi oleh fungsi Algoritma Genetik itu sendiri. Fungsi Algoritma Genetik di antaranya adalah proses mutasi, proses pindah silang dan jumlah generasi variasi. Proses pengujian dijalankan dengan Matlab.

#### 3.3.1. Pengujian menggunakan GA dengan Pengaruh Pmutasi

Pengaruh Pmutasi dilakukan dengan proses memasukkan nilai Pmutasi 0,8 dan 1. Selanjutnya memasukkan nilai Psilang 0,6 pada semua data yang dicobakan dan menggunakan jumlah generasi 100. Hasil simulasi untuk *transfer function* motor DC pada persamaan pemilihan model ditampilkan pada Gambar 8 dan Gambar 9.

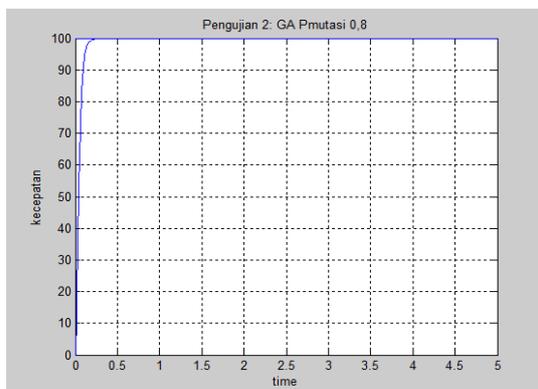


Gambar 8. Hasil simulasi Pmutasi 0,8.

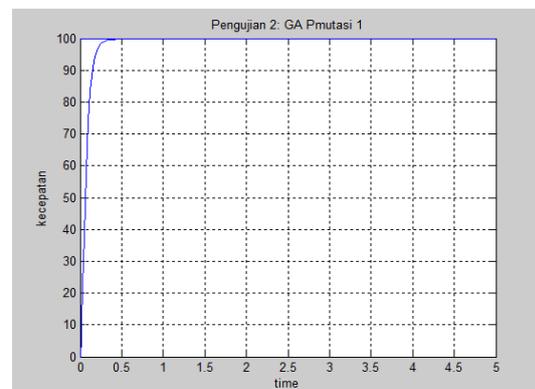


Gambar 9. Hasil simulasi Pmutasi 1.

Pengujian selanjutnya ialah memasukkan nilai parameter PID (KP, KI, KD) yang didapatkan dari hasil simulasi Pmutasi pada *plant* motor DC Matlab. Hasil yang ditampilkan dari pengujian tersebut berupa grafik yang ditunjukkan pada Gambar 10 dan Gambar 11.



Gambar 10. Grafik hasil simulasi Pmutasi 0,8.



Gambar 11. Grafik hasil simulasi Pmutasi 1.

Dari hasil simulasi Pmutasi 0,8 dan Pmutasi 1, diperoleh nilai *overshoot* yang sama, yaitu sebesar 0%. Hasil yang ditunjukkan pada grafik di atas bisa dianggap lebih baik dari grafik dengan metode *trial and error*. Akan tetapi, nilai *overshoot* tersebut belum tentu mutlak saat pengujian menggunakan alat. Hasil dari pengujian menggunakan metode GA dengan pengaruh Pmutasi ini ditunjukkan pada Tabel 2.

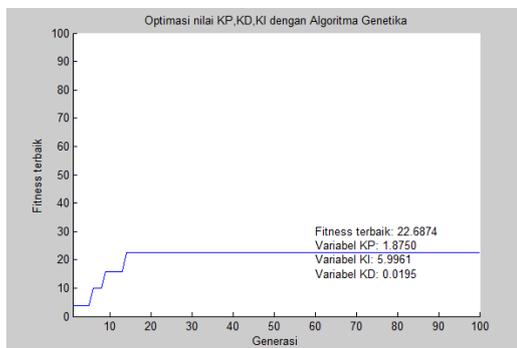
Tabel 2. Pengujian 2 menggunakan Metode GA dengan Pengaruh Pmutasi.

No.	Pmutasi	Psilang	Gen	<i>Fitness</i> Terbaik	Gain KP	Gain KI	Gain KD	<i>Risetime</i>	<i>Settling Time</i>	<i>Overshoot</i>
1	0,8	0,6	100	25,1027	7,6270	1,9238	0,2930	0,0813	0,1500	0
2	1	0,6	100	25,6586	3,7500	1,3184	0,2051	0,1288	0,2428	0

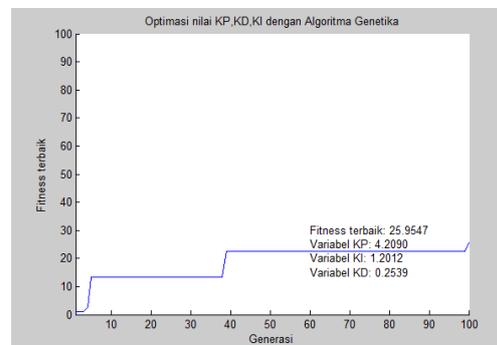
Dari hasil pengujian 2 ini diperoleh nilai *overshoot* 0 pada seluruh data. *Overshoot* yang didapatkan sudah optimal dengan nilai parameter PID yang sudah memenuhi syarat, yaitu  $KP \geq KI \geq KD$ .

### 3.3.2 Pengujian menggunakan GA dengan Pengaruh Psilang

Pengaruh Psilang dilakukan dengan proses memasukkan nilai Psilang 0,2; 0,4. Selanjutnya memasukkan nilai Pmutasi 0,6 pada semua data yang dicobakan dan menggunakan jumlah generasi 100. Hasil simulasi untuk *transfer function* motor DC pada persamaan pemilihan model ditampilkan pada Gambar 12 dan Gambar 13.

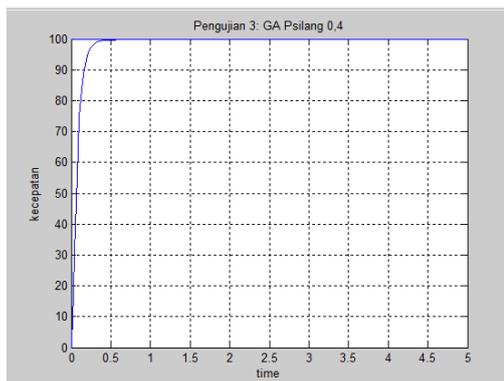


Gambar 12. Hasil simulasi Psilang 0,2.

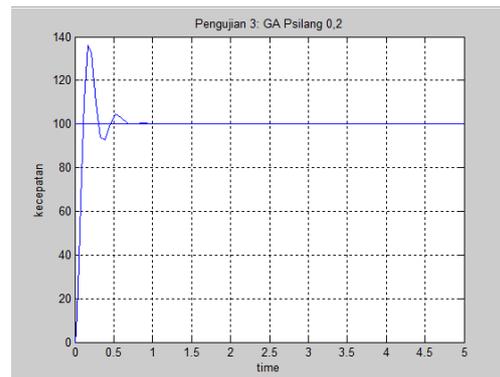


Gambar 13. Hasil simulasi Psilang 0,4.

Pengujian selanjutnya ialah memasukkan nilai parameter PID (KP, KI, KD) yang didapatkan dari hasil simulasi Psilang pada *plant* motor DC Matlab. Hasil yang ditampilkan dari pengujian tersebut berupa grafik yang ditunjukkan pada Gambar 14 dan Gambar 15.



Gambar 14. Grafik hasil simulasi Psilang 0,2.



Gambar 15. Grafik hasil simulasi Psilang 0,4.

Hasil yang diperoleh dari simulasi Psilang 0,2 memiliki nilai *overshoot* yang besar, yaitu 36,1315%. Sedangkan simulasi Psilang 0,24 menghasilkan nilai *overshoot* sebesar 0%. Hasil yang ditunjukkan pada grafik di atas bisa dianggap lebih baik dari grafik dengan metode *trial and error*, tetapi belum lebih baik dari grafik metode GA Pmutasi yang menampilkan grafik tanpa *overshoot* lebih banyak. Hasil dari pengujian menggunakan metode GA dengan pengaruh Psilang ini ditunjukkan pada Tabel 3.

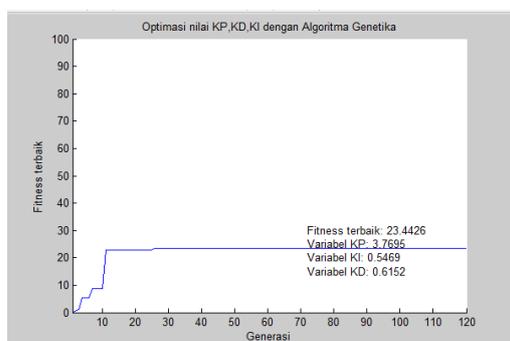
Tabel 3. Pengujian 3 menggunakan Metode GA dengan Pengaruh Psilang.

No.	Pmutasi	Psilang	Gen	Fitness Terbaik	Gain KP	Gain KI	Gain KD	Risetime	Settling Time	Overshoot
1	0,6	0,2	100	22,6874	1,8750	5,9961	0,0195	0,0746	0,6191	36,1315
2	0,6	0,4	100	25,9547	4,2090	1,2012	0,2539	0,1444	0,2737	0

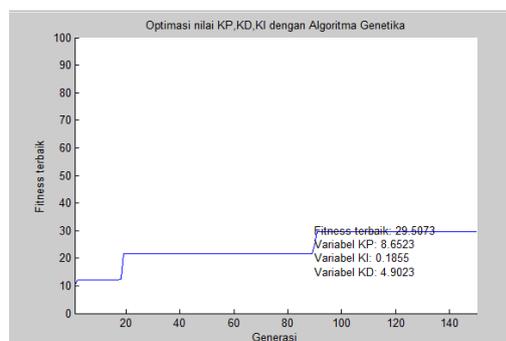
Dari hasil pengujian 3 ini diperoleh nilai *overshoot* 0 pada data nomor 2. *Overshoot* yang didapatkan sudah optimal dengan nilai parameter PID yang sudah memenuhi syarat, yaitu  $KP \geq KI \geq KD$ .

### 3.3.3 Pengujian menggunakan GA dengan Pengaruh Jumlah Generasi Variasi

Pengaruh jumlah generasi sangat menentukan parameter PID, jumlah generasi ini tidak mutlak harus lebih besar ataupun lebih kecil karena ditentukan dengan pergerakan dari nilai *fitness* terbaik. Jumlah generasi yang digunakan pada simulasi ini terdiri dari lima data, yaitu jumlah generasi 120 dan jumlah generasi 150. Selanjutnya memasukkan nilai Pmutasi dan Psilang 0,6 pada semua data yang dicobakan. Hasil simulasi untuk *transfer function* motor DC pada persamaan pemilihan model ditampilkan pada Gambar 16 dan Gambar 17.

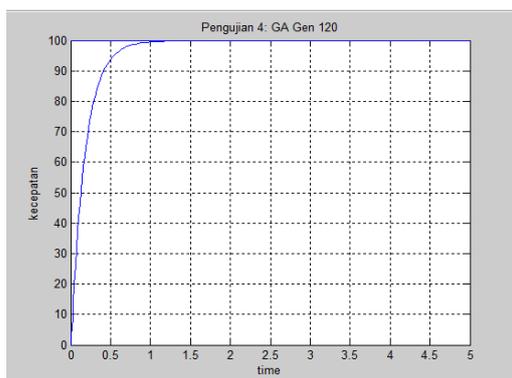


Gambar 16. Hasil simulasi jumlah generasi 120.

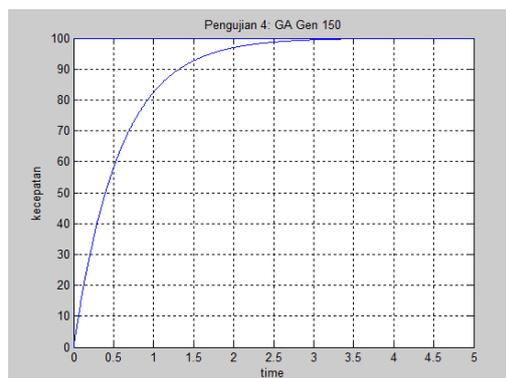


Gambar 17. Hasil simulasi jumlah generasi 150.

Pengujian selanjutnya ialah memasukkan nilai parameter PID (KP, KI, KD) yang didapatkan dari hasil simulasi jumlah generasi variasi pada *plant* motor DC Matlab. Hasil yang ditampilkan dari pengujian tersebut berupa grafik yang ditunjukkan pada Gambar 18 dan Gambar 19.



Gambar 18. Grafik hasil simulasi jumlah generasi 120.



Gambar 19. Grafik hasil simulasi jumlah generasi 150.

Hasil yang diperoleh dari simulasi jumlah generasi 120 dan jumlah generasi 150 memiliki nilai *overshoot* yang sama, yaitu 0%. Hasil yang ditunjukkan pada grafik di atas bisa dianggap lebih baik dari grafik dengan metode *trial and error* dan grafik dengan metode GA Pmutasi karena

menampilkan *overshoot* yang lebih banyak. Akan tetapi, nilai *overshoot* tersebut belum tentu mutlak saat pengujian menggunakan alat. Hasil dari pengujian menggunakan metode GA dengan pengaruh jumlah generasi variasi ini ditunjukkan pada Tabel 4.

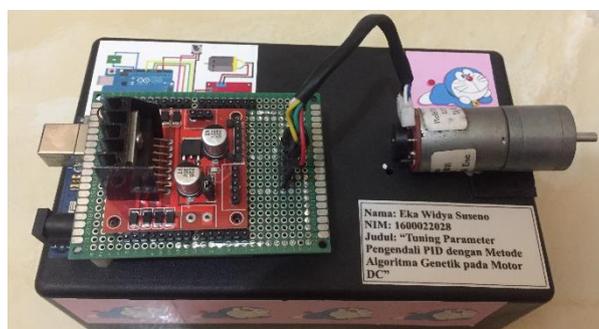
Tabel 3. Pengujian 4 menggunakan metode GA dengan pengaruh jumlah generasi variasi.

No.	Pmutasi	Psilang	Gen	<i>Fitness</i> Terbaik	Gain KP	Gain KI	Gain KD	<i>Risetime</i>	<i>Settling</i> <i>Time</i>	<i>Overshoot</i>
1	0,6	0,6	120	23,4426	3,7695	0,5469	0,6152	0,3889	0,7122	0
2	0,6	0,6	150	29,5037	8,6532	0,1855	4,9025	1,2614	2,2480	0

Dari hasil pengujian 4 ini diperoleh nilai *overshoot* 0 pada semua data. *Overshoot* yang didapatkan sudah optimal dengan nilai parameter PID yang sudah memenuhi syarat, yaitu  $KP \geq KI \geq KD$ .

### 3.4. Pengujian *Hardware*

*Hardware* yang digunakan pada pengujian tersebut ditunjukkan pada Gambar 20.



Gambar 20. *Hardware* pengendali PID dengan motor DC.

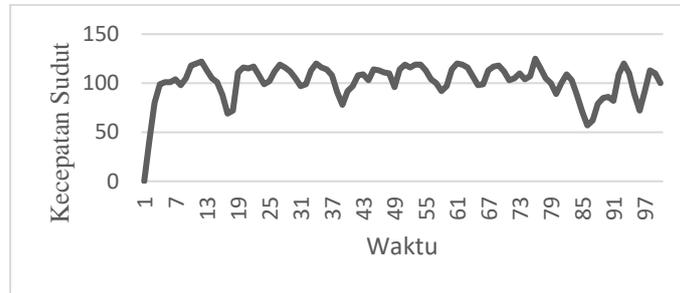
Pengujian *hardware* dilakukan setelah proses pengujian 1 sampai pengujian 4 mendapatkan hasil berupa 6 data nilai parameter PID (KP, KI, KD) menggunakan metode *trial and error* dan GA. Kemudian data tersebut diuji pada alat yang telah dirancang untuk mendapatkan hasil optimal yang sesuai dengan tujuan peneliti. Proses pengujian *hardware* dijalankan dengan aplikasi Arduino dan untuk menampilkan informasi respon sistem menggunakan Matlab. Hasil dari pengujian *hardware* ini ditunjukkan pada Tabel 5.

Tabel 5. Perbandingan parameter PID pada pengujian 1 sampai 4 menggunakan *hardware*.

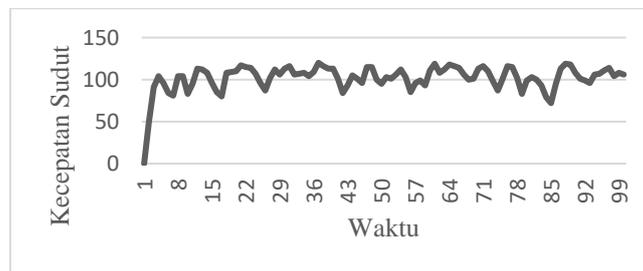
No.	Metode	Gain KP	Gain KI	Gain KD	<i>Risetime</i>	<i>Settling</i> <i>Time</i>	<i>Overshoot</i>
1	<i>Trial and Error</i> 1	1	8	15	2,2882	99,8000	25
2	<i>Trial and Error</i> 2	10	25	15	1,7756	NaN	20
3	Algoritma Genetik Pengaruh Pmutasi 0,8	7,6270	1,9238	0,2930	1,3754	42,2500	12
4	Algoritma Genetik Pengaruh Pmutasi 1	3,7500	1,3184	0,2051	2,7872	13,5000	2
5	Algoritma Genetik Pengaruh Psilang 0,2	1,8750	5,9961	0,0195	2,2917	94,2500	10
6	Algoritma Genetik Pengaruh Psilang 0,4	4,2090	1,2012	0,2539	2,6462	18,0000	2
7	Algoritma Genetik Pengaruh Gen 120	3,7695	0,5469	0,6152	1,3200	24,3333	7
8	Algoritma Genetik Pengaruh Gen 150	8,6532	0,1855	4,9025	3,4228	NaN	15

Pengujian *hardware* dilakukan sebanyak 8 kali sesuai dengan jumlah data dari pengujian sebelumnya. Urutan dari pengujian tersebut sudah sesuai dengan pengujian 1 sampai 4. Hasil dari

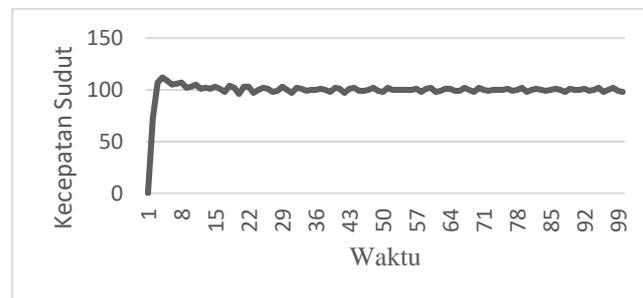
pengujian *hardware* berupa grafik kecepatan motor DC yang kemudian akan dibandingkan untuk mendapatkan nilai *overshoot* terkecil. Grafik tersebut ditunjukkan pada Gambar 21 sampai Gambar 28.



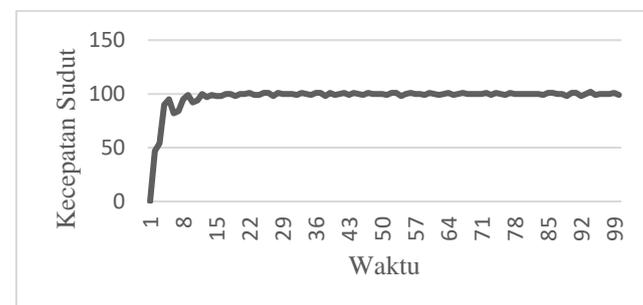
Gambar 21. Grafik metode *trial and error* 1 pada pengujian *hardware*.



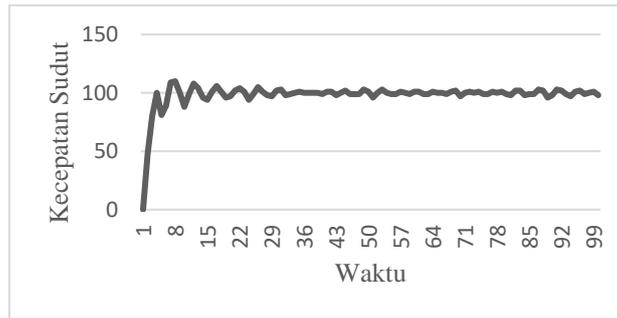
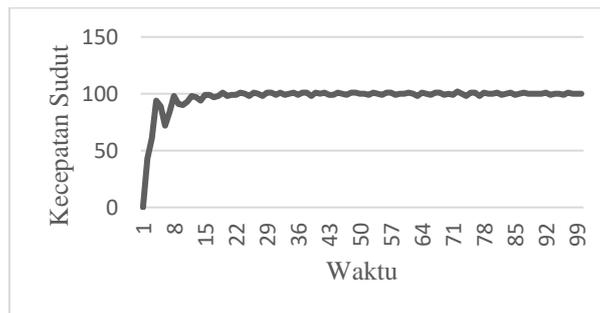
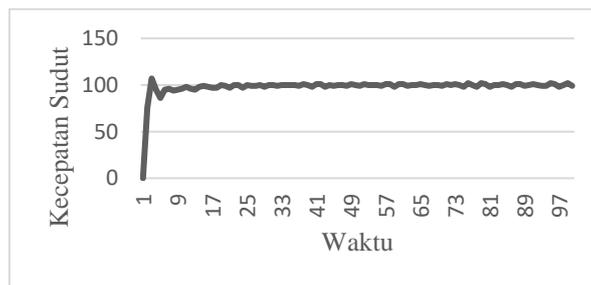
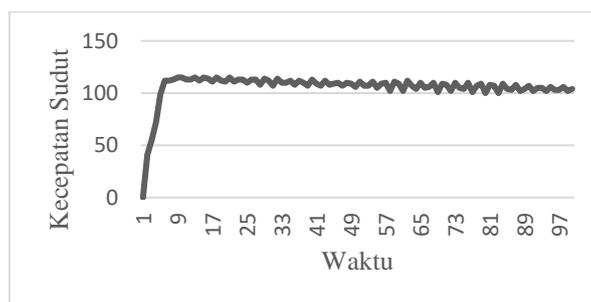
Gambar 22. Grafik metode *trial and error* 2 pada pengujian *hardware*.



Gambar 23. Grafik metode GA Pmutasi 0,8 pada pengujian *hardware*.



Gambar 24. Grafik metode GA Pmutasi 1 pada pengujian *hardware*.

Gambar 25. Grafik metode GA Psilang 0,2 pada pengujian *hardware*.Gambar 26. Grafik metode GA Psilang 0,4 pada pengujian *hardware*.Gambar 27. Grafik metode GA jumlah gen 120 pada pengujian *hardware*.Gambar 28. Grafik metode GA jumlah gen 150 pada pengujian *hardware*.

Dari hasil pengujian *hardware* menggunakan kecepatan motor DC dengan parameter PID sebagai *input*, dapat dilihat ada empat data yang sudah menampilkan *overshoot* minimum 10%, yaitu Gambar 24 dengan nilai *P*mutasi 1, Gambar 25 dengan nilai *Psilang* 0,2, Gambar 26 dengan nilai *Psilang* 0,4 dan Gambar 27 dengan nilai generasi 120. Tetapi data terbaik tersebut tidak dapat dilihat dari nilai *overshoot* saja. Peneliti harus membandingkan nilai parameter PID yang dihasilkan apakah sudah memenuhi syarat ( $K_P \leq K_I \leq K_D$ ) dan kecepatan yang didapat sudah stabil (*steady state*) sampai data ke-100, serta tidak ada kenaikan lonjakan yang signifikan

ataupun *undershoot* yang dinilai kurang baik. Temuan-temuan yang ditemukan pada keseluruhan percobaan ini dapat dilihat dari nilai parameter PID yang didapat serta grafik yang ditampilkan. Ada dua data pengujian terbaik dari segi nilai *overshoot* maupun nilai parameter PID yang menjadi *input*, yaitu pada Gambar 24, parameter PID nya adalah  $K_P = 3,7500$ ;  $K_I = 1,3184$  dan  $K_D = 0,2051$ . Untuk nilai *overshoot* = 2, *undershoot* = 0 dan untuk waktu tunak (*settling time*) = 13,5 serta waktu naik (*risetime*) = 2,7872. Kemudian Gambar 26, parameter PID nya adalah  $K_P = 4,2090$ ;  $K_I = 1,2012$  dan  $K_D = 0,2539$ . Untuk nilai *overshoot* = 2, *undershoot* = 0 dan untuk waktu tunak (*settling time*) = 18 serta waktu naik (*risetime*) = 2,6462.

#### 4. Kesimpulan

Setelah dilakukan penelitian dan mengolah data pada penelitian ini, algoritma Genetik untuk penalaan parameter pengendali PID ini sudah menemukan solusi optimal dengan *overshoot* di bawah 10%, dengan jumlah generasi 100 pada pengaruh Pmutasi 0,4 dan Pengaruh Psilang 0,8 serta pengaruh jumlah generasi variasi 120 dan 150. Metode algoritma genetik terbukti memberikan hasil sistem yang memiliki waktu tunak lebih baik dan lonjakan maksimum yang lebih kecil dibandingkan metode *trial and error* yang digunakan dalam simulasi. Proses pengujian *hardware* dengan metode algoritma genetik pada motor DC dapat berjalan dengan baik, sehingga didapatkan data terbaik berupa nilai *overshoot* di bawah 10% dan parameter PID yang memenuhi syarat. Proses pengujian tersebut menghasilkan dua data terbaik dengan nilai *overshoot* = 2 dan nilai parameter PID terbaik pada Pmutasi 1 ialah  $K_P = 3,7500$ ;  $K_I = 1,3184$  dan  $K_D = 0,2051$  serta nilai parameter PID terbaik pada Psilang 0,4 ialah  $K_P = 4,2090$ ;  $K_I = 1,2012$  dan  $K_D = 0,2539$ .

#### Referensi

- [1] R. Arindya, "Penalaan Kendali PID Untuk Pengendali Proses," *J. Teknol. Elektro*, vol. 8, no. 2, pp. 109-116, 2017, doi: 10.22441/jte.v8i2.1604.
- [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," *Journal of Physics: Conference Series*, 2019, doi: 10.1088/1742-6596/1373/1/012039.
- [3] M. Ruswandi Djalal, D. Ajiatmo, A. Imran, and I. Robandi, "Desain Optimal Kontroler Pid Motor DC Menggunakan Cuckoo Search Algorithm," *Prosiding SENTIA*, vol. 7, no. 1, pp. 121-126, 2015.
- [4] A. Josi, "Implementasi Algoritma Genetika Pada Aplikasi Penjadwalan Perkuliahan Berbasis Web Dengan Mengadopsi Model Waterfall," *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, vol. 2, no. 2, pp. 77-83, 2017, doi: 10.30591/JPIT.V2I2.517.G554.
- [5] A. N. Toscani and R. Roestam, "Pengembangan Sistem Penjadwalan Kuliah Menggunakan Algoritma Genetik (Studi Kasus : Pascasarjana Universitas Jambi)," *Jurnal Manajemen Sistem Informasi*, vol. 2, no. 2, pp. 379-393, 2017.
- [6] S. Agustina and N. Nugroho, "Analisa Motor DC (Direct Current) Sebagai penggerak Mobil Listrik," *Jurnal Mikrotiga*, vol.2, no. 1, pp. 28-34, 2015.
- [7] F. Fahmizal, M. Arraofiq, R. Adrian, and A. Mayub, "Robot Inverted Pendulum Beroda Dua (IPBD) dengan Kendali Linear Quadratic Regulator (LQR)," *Jurnal ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 7, no. 2, pp. 224-237, 2019, doi: 10.26760/elkomika.v7i2.224.
- [8] S. A. Ananda; and E. T. Soewangsa;, "Studi Karakteristik Motor DC Penguat Luar Terhadap Posisi Sikat," *Jurnal Teknik Elektro*, vol. 3, no. 1, pp. 51-56, 2003.
- [9] M. I. Irawan, I. Mukhlash, and F. Fernando, "Optimization of PID controller using genetic algorithm for missile's automatic steering system," *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 19, pp. 5058-5069, 2017.
- [10] V. Giordano, A. V. Topalov, O. Kaynak, and B. Turchiano, "Sliding-mode approach for on-line neural identification of robotic manipulators," in *2004 5th Asian Control*

*Conference*, 2004.

- [11] Dwi Ana Ratna Wati, *Sistem Kendali Cerdas*. Yogyakarta: Graha Ilmu, 2011.
- [12] Arduino, “Arduino Uno Rev3,” 2018. Available: <http://store.arduino.cc/products/arduino-uno-rev3>.
- [13] Agus Faudin, “Tutorial Arduino mengakses driver motor L298N,” 2017. Available: <https://www.nyebarilmu.com/tutorial-arduino-mengakses-driver-motor-l298n/>