

## IMPLEMENTASI DAN ANALISIS KENDALI KECEPATAN MOTOR BLDC 1 KW TANPA BEBAN MENGUNAKAN ALGORITMA PID

---

**Fatkhur Rohman<sup>1</sup>, Muhammad Arif Nur Huda<sup>2</sup>**

<sup>1,2</sup>Teknik Otomotif-Elektronik Politeknik Negeri Malang

<sup>1</sup>fatkhur\_rohman@polinema.ac.id

<sup>2</sup>arifnurhuda1997@gmail.com

### Abstrak

Saat ini motor *Brushless Direct Current* (BLDC) banyak diaplikasikan di berbagai bidang dikarenakan memiliki keunggulan dibandingkan motor *Direct Current* (DC). Namun, untuk menghasilkan respon kecepatan yang diharapkan, motor BLDC tersebut membutuhkan suatu pengendalian. Salah satu pengendali kecepatan adalah menggunakan algoritma Proporsional, Integral, dan Derivatif (PID). Tujuan penelitian ini adalah untuk merancang kendali kecepatan PID pada motor BLDC 1 kW berbasis arduino uno menggunakan *Matlab Simulink*, dan untuk mengetahui pengaruh variasi nilai parameter  $K_p$ ,  $K_i$ ,  $K_d$  terhadap respon kecepatan motor BLDC 1 kW. Selain itu, untuk menghasilkan respon kecepatan motor dengan *rise time* yang cepat, *overshoot* yang kecil dan *error steady state* yang rendah. Metode Pengambilan data dilakukan dengan cara memvariasikan nilai parameter PID menggunakan blok diagram *Matlab Simulink* yang dieksekusi oleh Arduino Uno. Hasil penelitian menunjukkan bahwa variasi nilai parameter PID mempengaruhi respon kecepatan motor BLDC 1 kW. Dengan cara *trial & error* didapatkan nilai parameter PID terbaik dengan nilai  $K_p=1,5$ ;  $K_i=10,5$ ; dan  $K_d=0,04$ .

**Kata-kata kunci:** Algoritma PID, *error steady state*, motor BLDC, *overshoot*, *rise time*.

### **Abstract**

Currently Brushless Direct Current (BLDC) motor is widely applied in various fields because it has advantages over Direct Current (DC) motor. However, to produce the expected speed response, the BLDC motor requires a control. One of the speed controllers is using Proportional, Integral and Derivative (PID) algorithm. The purpose of the study was to design the PID speed control on the BLDC motor based on arduino uno in Matlab Simulink, and to know the effect of variations in the parameter values  $K_p$ ,  $K_i$ ,  $K_d$  to the speed response of BLDC 1 kW motor. In addition, to produce motor speed response with a fast rise time, a small overshoot and a low steady state error. The method of data retrieval is done by varying the value of the PID parameter using the block diagram Matlab Simulink executed by Arduino Uno. The results showed that variations in the value of the PID parameter affected the speed response of the BLDC 1 kW motor. By trial & error, the best PID parameter values are obtained with the value  $K_p = 1.5$ ;  $K_i = 10.5$ ; and  $K_d = 0.04$ .

**Keywords:** BLDC motor, error steady state, overshoot, PID algorithm, rise time.

## **1. PENDAHULUAN**

Seiring dengan berkembangnya teknologi maka dirancanglah suatu motor arus searah tanpa sikat atau motor *Brushless Direct Current* (BLDC). Saat ini, motor BLDC sudah banyak diaplikasikan untuk berbagai bidang dikarenakan memiliki keunggulan dibandingkan motor *Direct Current*. Adapun beberapa kelebihan seperti tidak memerlukan perawatan dikarenakan tidak memiliki sikat, tidak menghasilkan suara bising, dan memiliki berat yang baik untuk rasio ukuran daya [1]. Namun, untuk menghasilkan respon kecepatan yang diharapkan, motor BLDC tersebut membutuhkan suatu pengendalian. Karena tanpa pengaturan kecepatan motor maka respon kecepatan yang didapatkan masih kurang baik [2].

Ada beberapa metode pengendalian yang dapat diterapkan pada motor BLDC, salah satunya adalah penggunaan pengendali *Proportional, Integral, dan Derivative* (PID). Pengendali PID merupakan teknik kendali yang sudah umum diterapkan di dunia rekayasa kendali. Sistem kendali PID adalah sistem pengontrolan

melalui kombinasi dari tiga macam kendali, yakni kendali *proportional*, kendali *integral*, dan kendali *derivative*.

Adapun tujuan perancangan kontroler PID adalah untuk mengurangi *steady state error*, mengurangi *overshoot*, meniadakan osilasi, serta memperkecil *rise time* dan *settling time* [3]. Sehingga kestabilan putaran dan kinerja pada motor BLDC akan lebih baik setelah adanya penerapan kontroler PID ini.

## 2. KAJIAN PUSTAKA

### 2.1 PENELITIAN SEBELUMNYA

Penelitian dengan judul Pengendali Kecepatan Motor DC pada Mobil Listrik, memperoleh parameter control PID yang stabil adalah pada  $K_p = 1.36$ ,  $K_i = 0.2$ , dan  $K_d = 1.33$ . Dimana dengan nilai tersebut sesuai untuk pengontrolan motor mulai 100 rpm – 1100 rpm. Pada perancangan PID tersebut menggunakan *plant* mobil listrik dan mencari nilai *plant* dengan cara *trial* dan *error*. [4]

Penelitian dengan judul Rancang Bangun Pengendalian Kecepatan Brushless DC Motor Tipe A2212/10T 1400KV Menggunakan Kontroller PID Berbasis *Labview*, memperoleh hasil bahwa respon kecepatan dari motor BLDC menggunakan kontroler PID mampu mencapai *setpoint* yang diinginkan dengan  $\tau = 1,089$  detik dan *error steady state* 4,3% (tanpa beban) dan  $\tau = 1,20$  detik dan *error steady state* 3.7% (dengan beban. Respon tersebut menunjukkan perbaikan dari respon system tanpa kontroler dengan  $\tau = 0,833$  detik dan *error steady state* 223.4 %.[5]

### 2.2 BRUSHLESS DIRECT CURRENT MOTOR (BLDC)

Motor *Brushless Direct Current* (BLDC) salah satu jenis motor yang saat ini banyak diaplikasikan secara luas diberbagai bidang. Motor BLDC digunakan di banyak industri seperti otomotif, penerbangan, medis, perangkat otomasi industri dan instrumentasi. Sesuai dengan namanya, motor BLDC tidak menggunakan *brush* atau sikat didalamnya untuk keperluan komutasi, namun komutasinya diatur secara elektronik. Motor BLDC ini merupakan pilihan yang tepat untuk aplikasi yang

membutuhkan keandalan tinggi, efisiensi tinggi, dan rasio *power-to-volume* tinggi [6]. Secara umum, motor BLDC memiliki karakteristik torsi yang lebih baik dengan *range* kecepatan yang tinggi dibandingkan dengan motor *Direct Current* (DC).



Gambar 1. Motor BLDC 1 kW

### 2.3 SISTEM PENGENDALIAN PID.

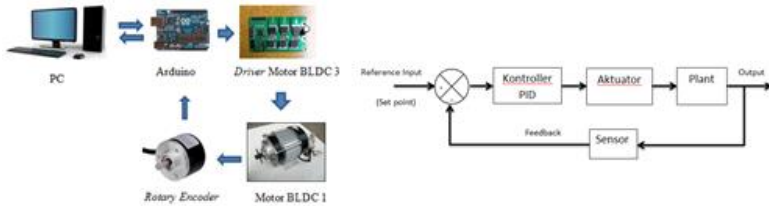
Pengendali PID merupakan teknik kendali yang sudah umum diterapkan di dunia rekayasa kendali. Sistem kendali PID adalah sistem pengontrolan melalui kombinasi dari tiga macam kendali, yakni kendali *proportional*, kendali *integral*, dan kendali *derivative* [7]. Kendali ini cukup mudah diterapkan dibandingkan dengan kendali yang lain dan banyak diaplikasikan pada mesin-mesin industri. Kontroler PID ini banyak diterapkan pada berbagai *plant* untuk mengatur kestabilan sistem misalnya kecepatan putaran motor, pengaturan intensitas cahaya lampu, dan kestabilan suhu.

### 3. METODOLOGI

Jenis penelitian yang dilakukan ini adalah penelitian eksperimen dan terapan. Penelitian eksperimen dilakukan dengan cara mencoba secara *trial and error* terhadap variabel penelitian. Pengambilan data pada alat ini akan dilakukan tanpa memberi pembebanan pada motor BLDC. Kemudian kecepatan motor BLDC akan divariasikan (400 rpm, 800 rpm, 1200 rpm). Selanjutnya setiap kecepatan tersebut (rpm) akan diberi sekurang kurangnya 5 variasi nilai  $K_p$ ,  $K_i$ , dan  $K_d$  dan melalui variasi parameter PID tersebut akan dilihat grafik hubungan antara kecepatan dan waktu pada *Matlab Simulink software* melalui perangkat komputer.

### 3.1 BLOK DIAGRAM PENELITIAN

Untuk tujuan pengambilan data Gambar 2 menunjukkan perencanaan blok diagram untuk penelitian ini.



Gambar 2. Diagram Blok Sistem

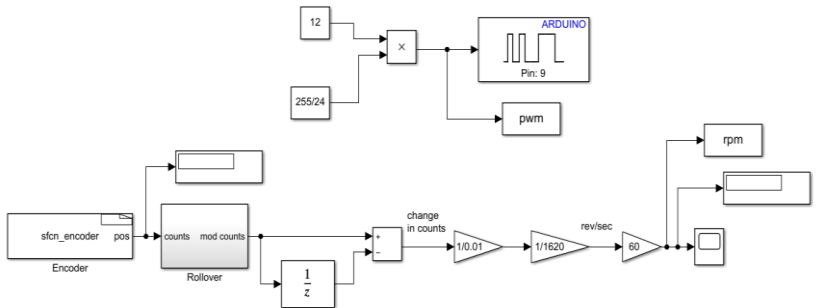
Blok Diagram diatas merupakan rangkaian sistem *close loop* (loop tertutup) pengendali kecepatan motor BLDC 1 kW menggunakan Algoritma PID. Sistem tersebut memiliki beberapa komponen utama diantaranya adalah Arduino Uno sebagai kontroller PID, *Driver Motor* sebagai aktuator (penggerak motor BLDC), Motor BLDC sebagai *Plant*, *Rotary encoder* sebagai pendeteksi kecepatan yang juga sebagai *feedback* (umpan balik) ke sistem, dan terakhir adalah PC atau laptop untuk memonitoring respon kecepatan motor secara *real time* melalui *Matlab Simulink Software* yang digunakan sebagai data penelitian.

### 3.2 RANCANG BANGUN SOFTWARE

Pada penelitian ini menggunakan papan arduino uno sebagai *hardware* kontroler PID. Umumnya arduino diprogram dengan menulis kode C/C++ di arduino IDE, namun pada penelitian ini arduino uno sebagai kontroler PID diprogram menggunakan *Matlab Simulink*.

#### 3.2.1 PERMODELAN SISTEM *OPEN LOOP*

Pembuatan model *simulink open loop* bertujuan untuk mendapatkan fungsi alih (*transfer function*) motor BLDC yang akan digunakan sebagai simulasi pada *Software Matlab Simulink*.



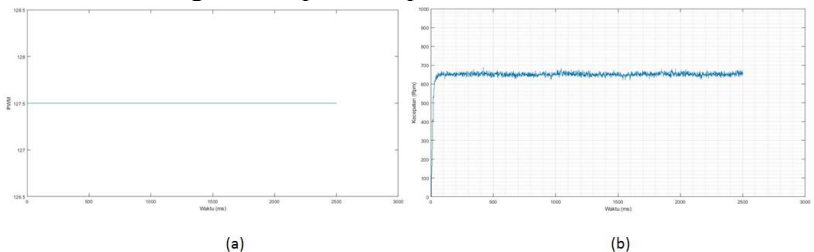
Gambar 3. Model Simulink Sistem *Open Loop*

Pembuatan model *simulink* ini adalah menggabungkan program pengontrol PWM dengan program pembacaan kecepatan. Blok ini juga berfungsi untuk mengidentifikasi sistem untuk mendapatkan nilai *transfer function* (fungsi alih) dari motor BLDC 1 kW. Program dari pengontrol PWM terdiri dari 2 blok *constan*, 1 blok *product*, blok PWM, dan blok *to workspace*. Blok *constan* bernilai 12 dan 255/24, 12 berarti nilai tegangan yang kita inginkan, 255/24 artinya nilai PWM maksimal disbanding nilai tegangan maksimal.

Adapun perhitungan secara matematis adalah sebagai berikut:

$$\text{nilai PWM} = 12 \times \frac{255}{24} = 127,5 \quad (1)$$

Nilai 127,5 tersebut merupakan nilai PWM sebagai input dari sistem. Blok PWM sebagai blok keluaran diatur pada pin 9. Berikut Adalah grafik respon kecepatan sistem lub terbuka

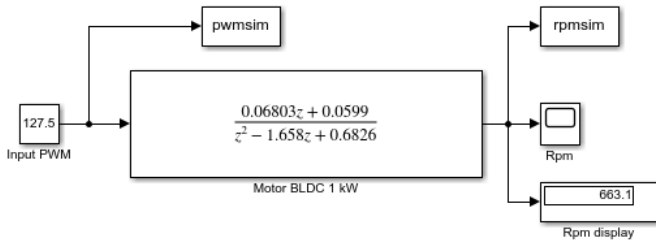


Gambar 4. Pengujian Karakteristik (a) Input PWM (b) Ouput kecepatan

Langkah selanjutnya adalah memasukkan data input dan output berupa grafik ke dalam aplikasi *System Identification* di Matlab. Setelah memasukkannya ke dalam *system identification* maka akan didapatkan nilai *transfer function* sebagai berikut:

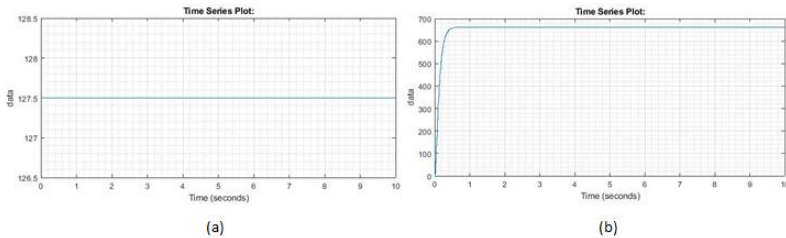
$$tf = \frac{0,06803z+0,0599}{z^2 - 1,658z+0,6826} \quad (2)$$

Untuk memastikan bahwa nilai *transfer function* (fungsi alih) benar dan akurat maka perlu divalidasi terlebih dahulu. Berikut adalah model *simulink* validasi nilai *transfer function*



Gambar 5. Diagram Blok *Transfer Function*

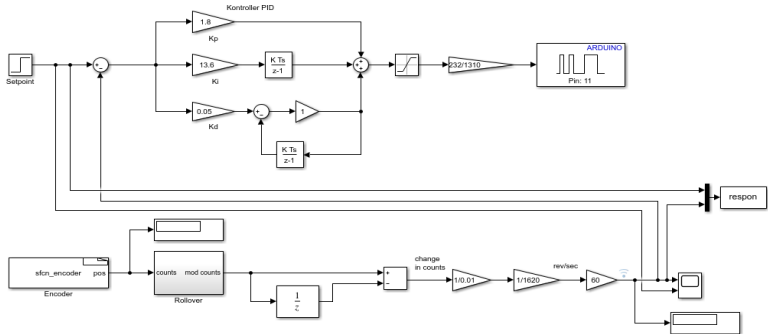
Berikut adalah grafik hasil simulasi pengujian nilai *transfer function* sistem lup terbuka.



Gambar 6. Simulasi Karakteristik (a) input (pwm) dan (b) output (kecepatan)

Terlihat bahwa pembacaan kecepatan pada simulasi dan eksperimen (menggunakan sensor *Rotary encoder*) menunjukkan hasil yang relatif sama, yaitu sekitar  $\pm 650 \text{ rpm}$  dan dapat disimpulkan bahwa nilai *transfer function* sudah bisa mewakili *plant* berupa motor BLDC 1 kW

### 3.2.2 PERMODELAN SISTEM KONTROL PID



Gambar 7. Model *Simulink* Kontrol PID

Perancangan Kontrol PID dilakukan dengan menggunakan *Software Matlab Simulink*. Model *Simulink* terdiri dari 2 bagian utama yaitu rangkaian blok diagram algoritma PID dan rangkaian blok diagram pembacaan kecepatan motor (rpm). Program pembacaan kecepatan motor sama dengan program yang dibuat sebelumnya, akan tetapi program tersebut fungsinya juga sebagai sinyal *feedback* (masukan sebagai) ke sistem PID.

### 3.2.3 FIRMWARE OVER THE AIR (FOTA) UPDATE

Sebagai sarana untuk menanam program pada perangkat, penelitian ini menggunakan metode FOTA yaitu menanam firmware arduino pada perangkat controller secara wireless memanfaatkan modul IOT ESP8266. Berikut ini adalah hasil pengujian rentang durasi proses update FOTA pada beberapa nilai baudrate yang didukung oleh protocol AVRISP 2 bootloader.

Tabel 1 Kecepatan Tranfer Data menggunakan FOTA

No	Ukuran hex file (kb)	Baudrate (bps)	Durasi (ms)
1	1	9600	210
2	2	9600	326
3	4	9600	443
4	8	9600	962



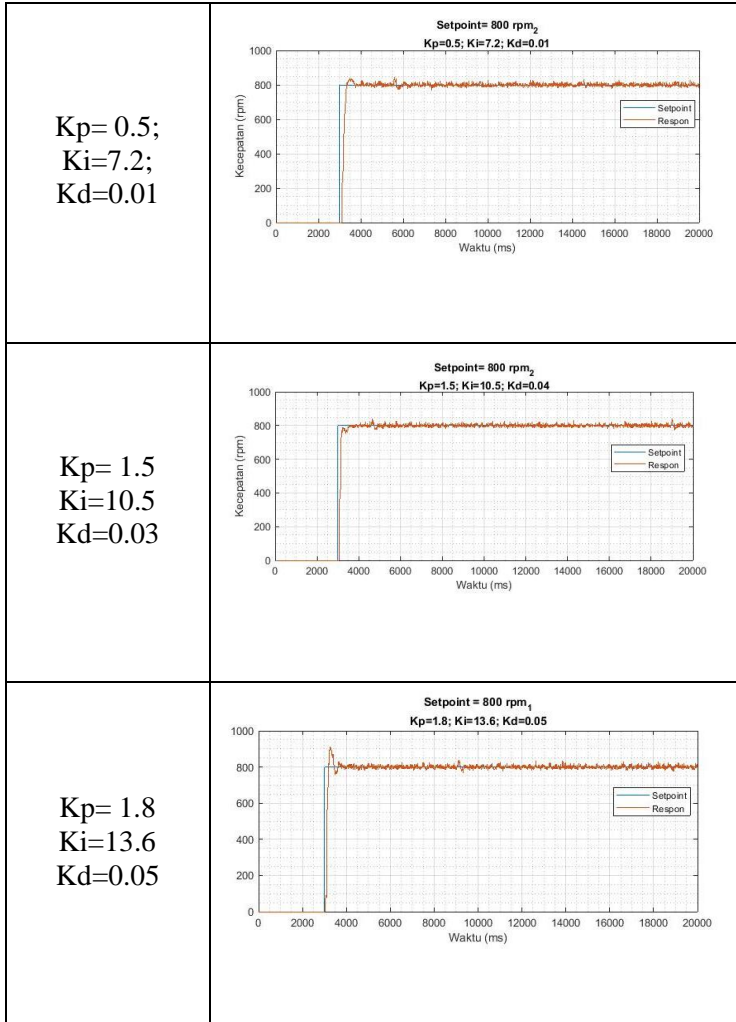
5	16	9600	1638
6	1	115200	43
7	2	115200	65
8	4	115200	74
9	8	115200	150
10	16	115200	502

#### 4. HASIL DAN PEMBAHASAN

Berikut adalah hasil grafik respon kecepatan dengan beberapa variasi nilai parameter PID pada salah satu sampel rpm yakni dengan *setpoint* 800 rpm. Variasi nilai parameter PID sebelumnya telah disimulasikan terlebih dahulu dengan *transfer function* (fungsi alih).

Tabel 2 Grafik Respon Kecepatan *Setpoint* 800 rpm

Parameter PID	Grafik
<p><math>K_p=0.8</math>  <math>K_i=6</math>  <math>K_d=0.02</math></p>	<p>Setpoint = 800 rpm<sub>2</sub>  <math>K_p=0.8; K_i=6; K_d=0.02</math></p>
<p><math>K_p=1.6</math>  <math>K_i=17.2</math>  <math>K_d=0.04</math></p>	<p>Setpoint = 800 rpm<sub>1</sub>  <math>K_p=1.6; K_i=17.2; K_d=0.04</math></p>



Setelah mengambil beberapa data penelitian berupa data grafik respon kecepatan motor BLDC 1 kW maka setelah itu adalah mengubahnya kedalam data numerik agar untuk memudahkan data analisis. Data numerik tersebut meliputi nilai *rise time* dalam satuan detik atau sekon, *Overshoot* yang disajikan dalam presentase, dan *Steady State Error* yang juga dalam satuan persentase.

Tabel 3 Nilai Rata-Rata Rise Time

<b>Rise time (detik)</b>			
<b>Parameter PID</b>	<b>400 rpm</b>	<b>800 rpm</b>	<b>1200 rpm</b>
Kp=0.8; Ki=6.0; Kd=0.02	0.620	0.660	0.853
Kp=1.6; Ki=17.2; Kd=0.04	0.143	0.193	0.443
Kp=0.5; Ki= 7.2; Kd=0.01	0.363	0.347	0.460
Kp=1.5; Ki=10.5; Kd=0.04	0.190	0.247	0.423
Kp=1.8; Ki=13.6; Kd=0.05	0.153	0.183	0.410

Pada tabel 3 diketahui bahwa respon kecepatan motor berupa nilai *rise time* tercepat pada *setpoint* 400 rpm yaitu sebesar 0.143 detik yaitu saat menggunakan parameter PID dengan Kp=1.6; Ki=17.2; Kd=0.04. Nilai *rise time* tercepat untuk *setpoint* 800 rpm dan 1200 rpm, berturut-turut adalah sebesar 0,183 detik dan 0,410 detik yaitu saat menggunakan nilai parameter PID Kp=1.8; Ki=13.6; Kd=0.05.

Tabel 4 Nilai Rata-rata Overshoot

<b>Overshoot (%)</b>			
<b>Parameter PID</b>	<b>400 rpm</b>	<b>800 rpm</b>	<b>1200 rpm</b>
Kp=0.8; Ki=6.0; Kd=0.02	0.000	0.000	0.000
Kp=1.6; Ki=17.2; Kd=0.04	26.833	22.417	3.859
Kp=0.5; Ki= 7.2; Kd=0.01	8.583	6.958	3.667
Kp=1.5; Ki=10.5; Kd=0.04	0.000	0.000	3.720
Kp=1.8; Ki=13.6; Kd=0.05	13.250	14.708	3.972

Pada tabel 4 diketahui nilai *overshoot* paling tinggi saat *setpoint* 400 rpm dan 800 rpm berturut-turut adalah sebesar 26,883% dan 22,417% yaitu pada saat menggunakan nilai parameter PID Kp=1.6; Ki=17.2; Kd=0.04, sedangkan untuk *setpoint* 1200 rpm nilai *overshoot* paling tinggi sebesar 3,972% yaitu saat menggunakan nilai parameter PID Kp=1.8; Ki=13.6; Kd=0.05.

Tabel 5 Nilai Rata-Rata *Error Steady State*

<i>Error steady state (%)</i>			
<b>Parameter</b>	<b>400 rpm</b>	<b>800 rpm</b>	<b>1200 rpm</b>
Kp=0.8; Ki=6.0; Kd=0.02	8.583	4.833	3.833
Kp=1.6; Ki=17.2; Kd=0.04	7.917	4.583	3.333
Kp=0.5; Ki= 7.2; Kd=0.01	7.250	4.083	3.667
Kp=1.5; Ki=10.5; Kd=0.04	7.167	3.875	3.222
Kp=1.8; Ki=13.6; Kd=0.05	7.417	4.542	3.889

Pada tabel 5 nilai *error steady state* terendah pada semua sampel rpm (*setpoint* 400 rpm, 800 rpm, dan 1200 rpm) yaitu berturut-turut sebesar 7,167%, 3,875%, dan 3,222%. Nilai *error steady state* terendah tersebut didapatkan pada nilai parameter PID Kp=1.5; Ki=10.5; Kd=0.04. Dengan hasil tersebut menunjukkan bahwa nilai parameter PID Kp=1.5; Ki=10.5; Kd=0.04 adalah nilai parameter yang menghasilkan kecepatan paling stabil.

## 5. KESIMPULAN

Dari hasil penelitian yang telah dilakukan, ada beberapa kesimpulan yang dapat diambil yang diantaranya adalah sebagai berikut:

1. Untuk merancang sistem kendali kecepatan motor BLDC dengan kontrol PID menggunakan arduino uno diperlukan perancangan *hardware* dan *software*. Perancangan *hardware* meliputi perancangan sensor kecepatan dan perancangan *output* PID berupa PWM. Sedangkan perancangan *Software* menggunakan *Matlab Simulink* meliputi pembuatan program PID dan program pembacaan kecepatan dari sensor *rotary encoder*.
2. Untuk mendapatkan hasil respon kecepatan terbaik, pada penelitian ini menggunakan metode *trial* dan *error* dimana dilakukan percobaan pada beberapa variasi nilai Kp, Ki, dan Kd. Respon terbaik didapatkan ketika menggunakan nilai parameter PID Kp=1,5; Ki=10,5; dan Kd=0,04; dimana respon kecepatan pada semua kondisi

kecepatan memiliki nilai *rise time* yang cepat, *overshoot* yang kecil, dan juga *error steady state* yang kecil.

## 6. DAFTAR PUSTAKA

- [1] Kristiyono, R., Wahyunggoro, O., Nugroho, P. 2015. Siste Kendali Kecepatan Motor BLDC Menggunakan Algoritma *Hybrid PID Fuzzy*. *University Research Colloquium*. 116-124.
- [2] Zoni, M., Hidayat., Hidayatullah. 2017. Perbaikan Performance Kendali Kecepatan Motor Brushless DC (BLDC) Dengan Pengendali Logika Fuzzy. *Seminar Nasional dan Gelar Produk*. 299-308.
- [3] Bista, Dinesh. 2016. *Understanding and Design of an Arduino-based PID Controller*. Tesis tidak diterbitkan. Virginia: *Physics*, Virginia Commonwealth University.
- [4] Firmanto. 2014. *Pengendali Kecepatan Motor DC Pada Mobil Listrik*. Tugas Akhir tidak diterbitkan. Batam: Program Studi Teknik Elektronika Politeknik Negeri Batam.
- [5] Pratama, F.Y., Endryansyah. 2018. Rancang Bangun Pengendalian Kecepatan *Brushless Dc* Motor Tipe A2212/10t 1400 Kv Menggunakan Kontroler Pid Berbasis Labview. *Jurusan Teknik Elektro*, (Online), 7 (3), 157-166, (<https://jurnalmahasiswa.unesa.ac.id>), diakses 20 Desember 2018.
- [6] Wicaksono A.S., Effendie, R., Iskandar, E. 2016. Perancangan dan Implementasi Sistem Pengaturan Kecepatan Motor Bldc Menggunakan Kontroler Pi Berbasis *Neural-Fuzzy* Hibrida Adaptif. *Jurnal Teknik ITS*, 5 (2), 68-74.
- [7] Rosalina., Qosim, I., Mujirudin, M. 2017. Analisis Pengaturan Kecepatan Motor DC Menggunakan Kontrol PID (*Proportional Integral Derivative*). *Seminar Nasional TEKNOKA*, 2, 89-94.