

Kontrol *Proporsional Integral Derivatif* (PID) pada Kecepatan Sudut Motor DC dengan Pemodelan Identifikasi Sistem dan *Tuning*

ALFIAN MA'ARIF*, RYAN ISTIARNO, SUNARDI

Program Studi Teknik Elektro, Universitas Ahmad Dahlan, Indonesia
Email: alfianmaarif@ee.uad.ac.id

Received 13 November 2020 | *Revised* 26 November 2020 | *Accepted* 25 Desember 2020

ABSTRAK

Penelitian ini mengusulkan tentang sistem kontrol kecepatan sudut Motor Direct Current (DC) menggunakan kontrol Proporsional Integral Derivatif (PID). Pemodelan motor DC menggunakan model identifikasi sistem agar model sistem dapat mendekati sistem sesungguhnya. Data identifikasi sistem adalah nilai masukan tegangan dan nilai keluaran kecepatan sudut. Representasi model adalah model fungsi alih. Nilai kontrol PID didapatkan dengan fitur Tuning PID dengan Matlab. Perangkat penelitian adalah Arduino, sensor encoder, driver motor dan Motor DC. Pada pengujian, kendali motor DC dengan PID mampu untuk mendapatkan respon yang baik dengan nilai respon terbaik, rise time 9,4286 detik, settling time 18,5 detik dan overshoot 2 persen. Nilai variasi PWM untuk memperoleh model dan respon sistem motor DC yang bagus yaitu nilai variasi PWM 5, nilai variasi PWM 10 dan nilai variasi PWM 50, 150, 255. Dengan menggunakan metode ini, proses tuning kontrol PID dapat lebih efektif dan efisien.

Kata kunci: Motor DC, Identifikasi Sistem, Kontrol PID, Tuning Matlab, Kecepatan

ABSTRACT

This study proposes a direct current (DC) motor angular speed control system using Proportional Integral Derivative (PID) control. DC motor modeling uses a system identification model so that the system model can approach the real system. The system identification data is the input voltage value and the angular velocity output value. Model representation is a transfer function model. PID control values are obtained with the PID Tuning feature with Matlab. The research devices are Arduino, encoder sensor, motor driver and DC motor. In testing, the DC motor control with PID was able to get a good response with the best response value, rise time of 9.4286 seconds, settling time of 18.5 seconds and overshoot 2 percent. The PWM variation values to obtain a good DC motor system model and response are the PWM variation value 5, the PWM variation value 10 and the PWM variation value 50, 150, 255. By using this method, the PID control tuning process can be more effective and efficient.

Keywords: DC Motor, System Identification, PID Control, Matlab Tuning, Speed

1. PENDAHULUAN

Motor DC (*Direct Current*) adalah peralatan elektromagnetik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik (**Masrukhan, dkk, 2016**). Prinsip kerja dari motor DC adalah arah medan magnet *rotor* selalu berusaha berada pada posisi yang berlawanan arah dengan arah medan magnet *stator* yang akan saling bertolak – menolak dan daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu (**Waluyo, dkk, 2013**).

Pada umumnya Kecepatan putar motor DC mengalami penurunan akibat dari pembebanan sehingga mengakibatkan putarannya menjadi lambat dan kecepatannya tidak konstan. Pengendalian kecepatan motor DC bisa dilakukan dengan mengubah besaran nilai tegangan masukannya (**Purbowaskito & Hsu, 2017**). Misalnya jika beban bertambah yang mengakibatkan berkurangnya kecepatan putar motor DC, maka untuk menaikkan kecepatannya kembali bisa dilakukan dengan menaikkan nilai tegangan input pada motor DC tersebut. Oleh karena itu, diperlukan pengendali (kontroler) agar kecepatan motor DC dapat stabil dengan beban yang berubah.

Kebutuhan akan sistem kontrol yang lebih efektif dan efisien di era modern ini semakin meningkat, mengingat jumlah *plant* yang akan dikontrol semakin banyak dan memiliki struktur yang semakin kompleks (**Khadari, dkk, 2019**). Salah satu jenis sistem kontrol yang banyak digunakan saat ini adalah sistem kontrol *Proportional Integral Derivative* (PID) (**Wiriawan & Irawan, 2016**). Sistem kontrol ini memiliki struktur sederhana dan mudah dalam pengaplikasiannya (**Maarif, dkk, 2020**). Sistem kontrol PID ini memiliki parameter-parameter kontrol, yaitu parameter Proporsional (KP), parameter integral (KI), dan parameter derivatif (KD) (**Febriawan & Aji, 2020**). Ketiga parameter tersebut diturunkan dari perhitungan matematis pada metode PID konvensional.

Dalam sistem kontrol PID adanya beberapa macam aksi kontrol, di antaranya yaitu aksi kontrol Proporsional, aksi kontrol integral dan aksi kontrol derivatif (**Djalal & Rahmat, 2020**). Masing-masing aksi kontrol ini mempunyai keunggulan tertentu, seperti aksi kontrol proporsional mempunyai keunggulan *rise time* yang cepat dan mengurangi *error* keadaan stabil, aksi kontrol integral mempunyai keunggulan untuk memperkecil *error* keadaan stabil dan aksi kontrol derivatif mempunyai keunggulan untuk meredam *overshoot* (**Iswanto, dkk, 2020**).

Terdapat beberapa jenis pendekatan pemodelan motor DC seperti pendekatan menggunakan hukum fisika dan identifikasi sistem (**Tang, dkk, 2017**). Pemodelan motor DC menggunakan pemodelan identifikasi sistem merupakan pemodelan yang cukup baik dengan menggunakan pendekatan input dan *output* sistem. Hal tersebut disebabkan karena pemodelan dengan identifikasi sistem akan menghasilkan model yang mendekati model sistem yang sesungguhnya (**Fahmizal, dkk, 2018**).

Proses penentuan nilai pengendali PID biasanya dilakukan dengan proses *trial and error* atau proses coba-coba (**Anggraini, dkk, 2020**). Proses ini tidak efektif karena belum tentu memberikan hasil yang baik dan menghabiskan waktu dan tenaga (**Djalal & Rahmat, 2017**). Oleh karena itu diperlukan metode untuk menentukan nilai parameter pengendali dengan cara yang efektif (**Achanta & Pamula, 2018**). Metode yang dapat digunakan untuk menentukan nilai parameter kontrol PID adalah metode tuning PID pada Matlab (**Hammoodi, dkk, 2020**). Metode ini memiliki kelebihan menghemat waktu dan dapat memberikan nilai parameter PID sesuai dengan respon sistem yang diinginkan (*rise time* kecil, *settling time* kecil, *overshoot* kecil dan tanpa *undershoot*). Oleh karena itu, penelitian ini bertujuan untuk mendapatkan nilai

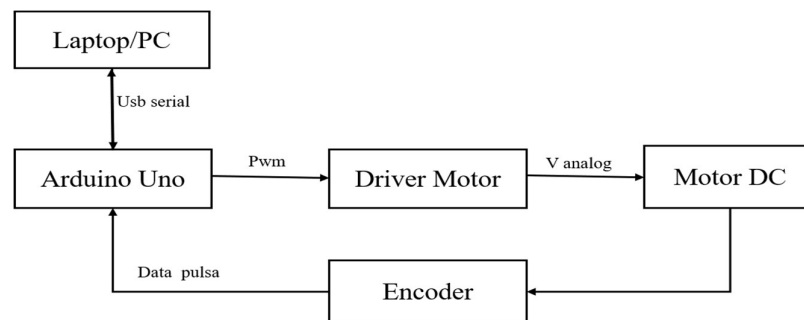
parameter kontrol PID yang terbaik dan efektif dengan menggunakan pemodelan identifikasi sistem dan proses *tuning* PID dengan *Matlab*.

2. METODE

Penelitian ini menggunakan metode eksperimen untuk merancang model motor DC menggunakan pemodelan identifikasi sistem dan nilai kendali PID di dapatkan dari proses identifikasi sistem menggunakan *software* *Matlab*. Penelitian ini memiliki beberapa tahap perancangan sistem, baik pada perancangan perangkat keras (*hardware*) maupun perancangan perangkat lunak (*software*).

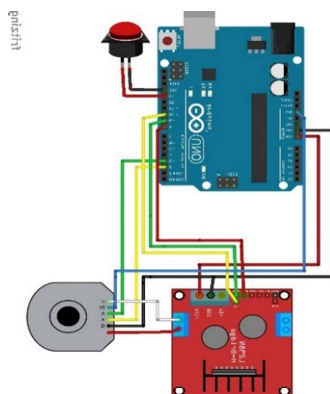
2.1 Desain Sistem

Perancangan sistem ini terdiri dari beberapa hal, di antaranya diagram blok sistem, diagram pengkabelan, diagram blok kontrol PID, dan diagram alir sistem PID. Perancangan sistem perangkat keras dalam bentuk diagram blok sistem diperlihatkan pada Gambar 1. Diagram pengkabelan dari sistem dapat dilihat pada Gambar 2 dan rancangan perangkat keras sistem ditunjukkan pada Gambar 3.

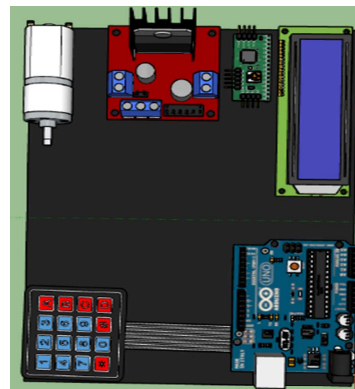


Gambar 1. Diagram blok sistem

Gambar 1 adalah diagram blok pemodelan motor DC yang menggunakan *Arduino Uno* sebagai pengendali sistem. *Arduino* mendapat suplai tegangan dari sumber tegangan yang sudah disiapkan sebesar 12 volt yang disalurkan pada *driver* motor untuk menggerakkan motor DC. Selanjutnya gerakan putar motor DC akan dicatat oleh sensor *encoder* yang akan diberikan pada *Arduino Uno* agar dapat dilihat hasilnya dalam *software* *Arduino IDE* pada menu serial monitor dan serial *plotter* pada Komputer.



Gambar 2. Wiring diagram motor DC

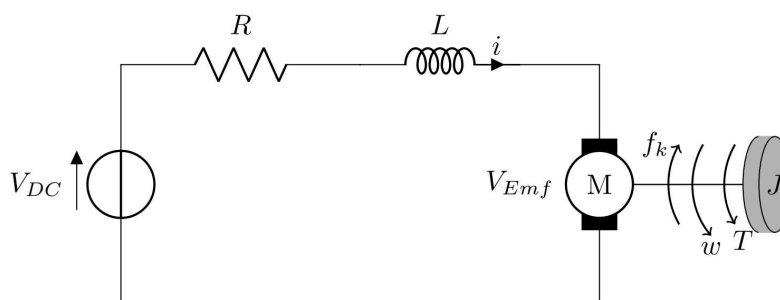


Gambar 3. Rancangan perangkat keras

Secara umum, perancangan perangkat keras dibuat untuk mengontrol kecepatan sudut motor DC, melihat respon sistem dan juga sebagai alat pengambilan data yang diperlukan untuk membuat model motor DC menggunakan pendekatan pemodelan identifikasi sistem. Data yang akan digunakan berupa tegangan dan kecepatan putar motor DC.

2.2 Pemodelan Motor DC

Motor DC merupakan suatu perangkat elektronis pertama yang digunakan dalam sejarah industri perkembangan sistem transmisi daya untuk mengubah besaran listrik menjadi besaran mekanik. Prinsip kerjanya apabila motor diberi daya maka arus akan mengalir ke dalam motor dan menghasilkan torsi putar yang sebanding dengan arusnya. Model motor DC yang digunakan adalah representasi fungsi alih (*transfer function*). Input sistem adalah tegangan yang diberikan pada motor DC yaitu V . Sedangkan *output* sistem berupa kecepatan sudut motor DC yaitu omega (ω). Sirkuit listrik dari *armature* motor DC dan besaran fisiknya adalah seperti Gambar 4.



Gambar 4. Sirkuit listrik dari motor DC

Pemodelan motor DC dapat dilakukan melalui 2 tahap yaitu dengan pemodelan elektrik dan pemodelan mekanik. Masukan motor DC adalah tegangan DC dan keluaran motor DC adalah kecepatan, serta torsi dari motor DC. Pemodelan elektrik motor DC dengan menggunakan analisis hukum kirchoff tentang tegangan diperoleh persamaan dalam domain frekuensi sebagai

$$I = \frac{V_{DC} - K_e \omega}{R + Ls} \quad (1)$$

dengan I adalah arus listrik, V_s adalah tegangan suplai, R adalah resistansi, L_s adalah induktansi, ω adalah omega dan K_e adalah konstanta back EMF. Serta pemodelan mekanik motor DC menggunakan pendekatan hukum Newton tentang rotasi diperoleh

$$\omega(Js + b) = K_t I \quad (2)$$

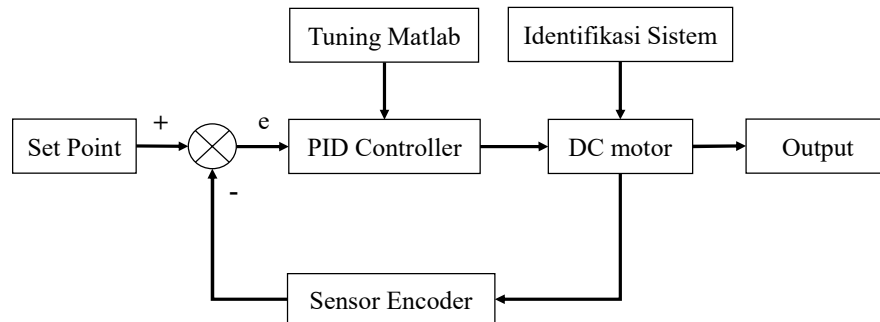
dengan J adalah nilai inersia, b adalah konstanta gaya gesek, K_t adalah konstanta torsi. Kedua persamaan mekanik dan elektrik disubstitusi untuk memperoleh model kecepatan sudut motor DC sebagai

$$\omega = \frac{K_t}{(Js + b)(Ls + R) + K_e K_t} V_s \quad (3)$$

2.3 Perancangan Perangkat Lunak Sistem PID

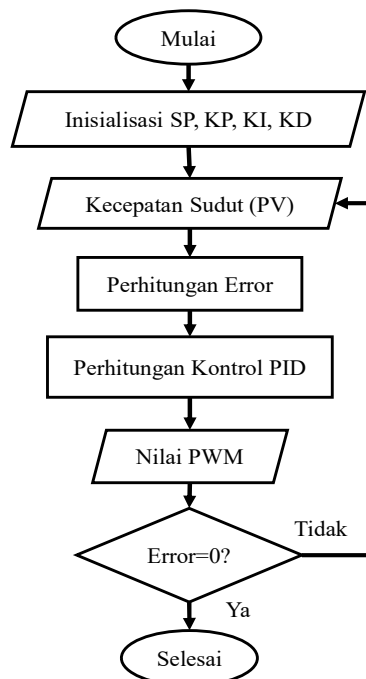
Parameter pengontrol Proporsional Integral derivatif (PID) selalu didasari atas tinjauan terhadap karakteristik sistem yang diatur (*plant*) (Apriaskar, dkk, 2019). Dengan demikian

bagaimanapun rumitnya suatu *plant*, perilaku *plant* tersebut harus diketahui terlebih dahulu sebelum pencarian parameter PID itu dilakukan. Perancangan perangkat lunak sistem PID meliputi perancangan blok diagram kontrol PID dan juga perancangan sistem identifikasi menggunakan menu *ident* pada perangkat lunak *Matlab*. Diagram blok kontrol PID pada motor DC menggunakan pendekatan identifikasi sistem dapat dilihat pada Gambar 5.



Gambar 5. Diagram Blok kontrol PID

Gambar 5 menjelaskan tentang diagram blok sistem kontrol PID. Blok PID akan dituning menggunakan fitur tuning PID pada Matlab. Blok *Plant* adalah sistem motor DC yang akan dimodelkan dengan proses identifikasi sistem menggunakan fitur Identifikasi *Matlab (ident)*. Keluaran sistem berupa kecepatan sudut (omega) motor DC yang dibaca menggunakan sensor *encoder*. Data kecepatan sudut menjadi nilai umpan balik dan dibandingkan dengan nilai *set point* (nilai referensi). Hasil perbandingan nilai tersebut dinamakan dengan nilai *error* yang menjadi masukan untuk pengendali (Ardiansyah, 2017). Pengendali akan mengeluarkan sinyal kendali sebagai masukan untuk sistem Motor DC. Diagram alir atau *flowchart* cara kerja program PID pada kendali motor DC dengan pengendali PID menggunakan pendekatan pemodelan identifikasi sistem terdapat pada Gambar 6.

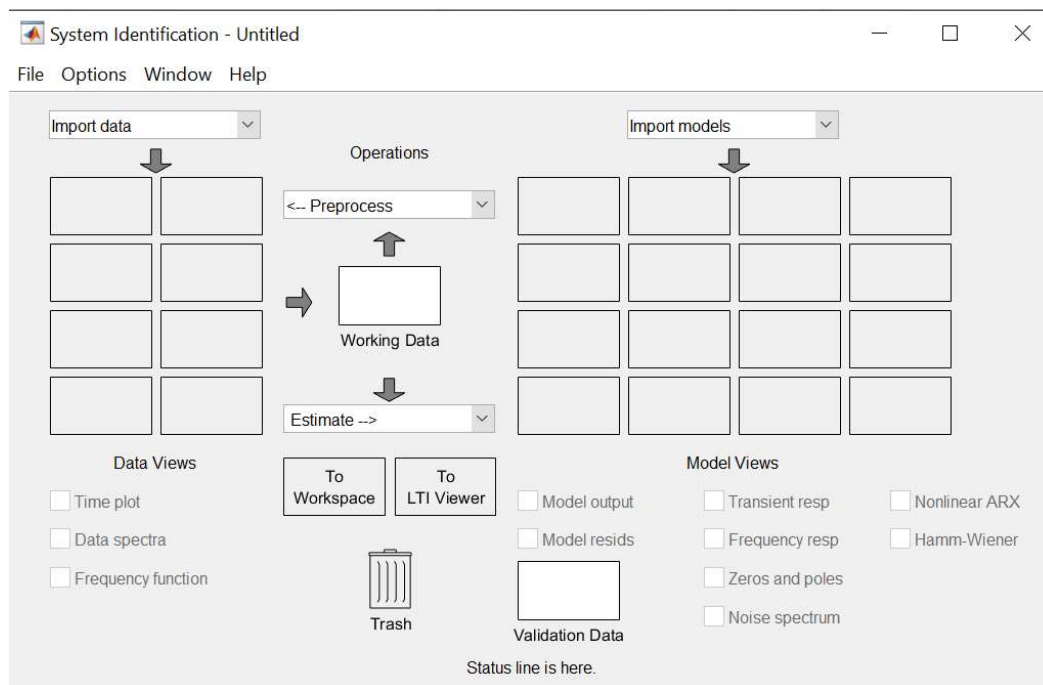


Gambar 6. Diagram alir sistem PID

Diagram alir sistem PID diperlihatkan pada Gambar 6. Bagian awal program adalah inisialisasi nilai set point (SP), kontrol proporsional (KP), kontrol integral (KI), dan kontrol (KD). Nilai kontrol PID diperoleh dari hasil tuning dengan Matlab. Proses selanjutnya yaitu membaca nilai kecepatan sudut (*Present Value* (PV)) dan membandingkan dengan nilai *set point* (SP) untuk memperoleh nilai *error*. Dengan menggunakan nilai *error*, pengendali akan menghitung nilai sinyal kendali dan mengirimkannya ke motor DC. Kecepatan motor DC akan dibaca oleh sensor *encoder* dan dibandingkan kembali dengan nilai *set point* untuk menghitung *error*. Kontrol PID akan menghitung nilai sinyal kendali dengan nilai *error* yang baru dan memberikan nilai sinyal kendali yang baru ke motor DC. Proses ini akan terus berjalan untuk menjaga kecepatan motor DC memiliki nilai *error* sebesar nol.

2.4 Pemodelan Identifikasi Sistem *Matlab*

Perangkat lunak *Matlab* dalam penelitian ini digunakan untuk melakukan proses identifikasi sistem dan juga untuk menguji hasil identifikasi sistem yang telah didapatkan. Proses identifikasi sistem sendiri menggunakan salah satu aplikasi *software* Matlab yang bernama menu *ident* (Ramdani, 2015). Perancangan perangkat lunak *ident* digunakan untuk mengolah atau mengidentifikasi sampel data yang didapatkan dari pengujian motor DC berupa tegangan dan kecepatan putar motor DC. Untuk tampilan *interface* menu *Ident* pada *software* Matlab ditunjukkan pada Gambar 7. Bagian *import data* berfungsi untuk memasukkan data model ke blok *data views*, blok *preprocess* berfungsi untuk menyaring data dan blok *estimate* berfungsi untuk mengestimasi (mengidentifikasi model). Model hasil identifikasi akan ditampilkan pada bagian blok model views. Selanjutnya respon model dapat dilihat pada bagian *Model Output*.



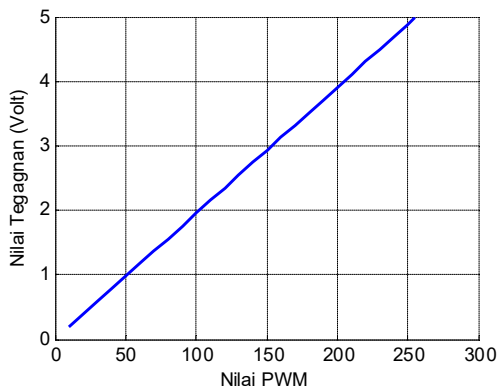
Gambar 7. Interface menu *Ident* software Matlab

Sistem *ident* pada *Matlab* memungkinkan pengguna untuk menyesuaikan model, respon dan tampilan jendela, termasuk memasukkan dan menentukan jenis model (fungsi alih atau *state space*). Untuk memperoleh model dari suatu sistem dapat dilakukan dengan cara meng-*export* data dari aplikasi seperti *simulink* atau aplikasi lain seperti *controlsystemtoolbox* ke ruang kerja (*workspace*) *Matlab*.

3. HASIL DAN PEMBAHASAN

3.1 Data Keluaran Tegangan Arduino

Pengujian data keluaran tegangan Arduino dilakukan dengan memberikan nilai masukan PWM pada motor DC. Pengujian ini bertujuan melihat nilai tegangan yang diberikan kepada motor DC berdasarkan nilai PWM yang dimasukkan pada Arduino. Setelah nilai PWM diberikan pada motor DC langkah selanjutnya adalah melihat nilai keluaran menggunakan alat bantu ukur tegangan atau multimeter pada PIN 10 Arduino. Nilai pembacaan tegangan pada Arduino dapat dilihat pada Gambar 8.



Gambar 8. Data Keluaran Tegangan PWM Arduino

Data keluaran tegangan Arduino tersebut dapat disimpulkan bahwa semakin tinggi nilai PWM pada motor DC semakin besar pula tegangan yang diberikan oleh Arduino. Jadi, dapat dikatakan bahwa nilai tegangan motor DC yang diberikan Arduino adalah sebesar 4,99 volt dengan nilai masukan PWM maksimal 255 PWM.

3.2 Data Kecepatan Putar Motor DC

Pengujian kecepatan putar motor DC dilakukan dengan memberikan nilai memasukkan PWM pada motor DC yang digunakan untuk melihat kecepatan putar atau RPM yang dihasilkan oleh motor DC tersebut. Nilai RPM yang di hasilkan oleh motor dc akan terlihat dalam serial monitor dalam program Arduino yang digunakan.

3.2.1 Nilai variasi PWM 5

Pengambilan data kecepatan putar motor DC dengan nilai variasi PWM 5 didapatkan hasil percobaan sebanyak 510 data RPM. Hasil pembacaan kecepatan putar motor DC (RPM) dengan nilai variasi PWM 5 dapat dilihat dalam Gambar 9. Setiap data PWM diambil 10 buah data dan dirata-rata.

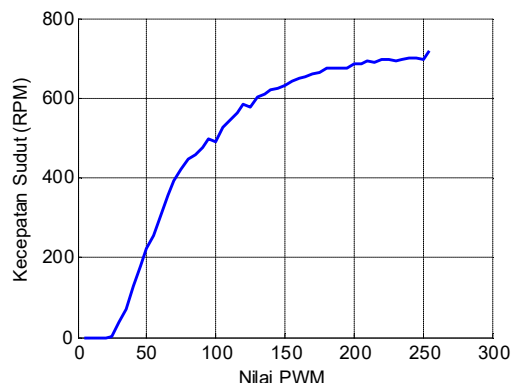
3.2.2 Nilai variasi PWM 10

Pengambilan data kecepatan putar motor DC dengan nilai variasi PWM 10 didapatkan hasil percobaan sebanyak 260 data RPM. Hasil pembacaan kecepatan putar motor DC (RPM) dengan nilai variasi PWM 10 dapat dilihat dalam Gambar 10. Setiap data PWM diambil 10 buah data dan dihitung nilai rata-ratanya.

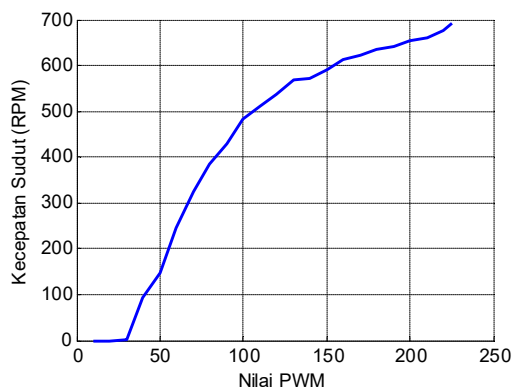
3.2.3 Nilai variasi PWM 50, 150, dan 255

Pengambilan data kecepatan putar motor dc dengan nilai variasi PWM 50, 150, dan 255 didapatkan hasil percobaan sebanyak 500 data RPM. Hasil pembacaan kecepatan putar motor DC (RPM) dengan nilai variasi PWM 50, 150, dan 255 dapat dilihat dalam Tabel 1. Nilai PWM

50 diambil 100 data, Nilai PWM 150 diambil 300 data dan nilai PWM 255 diambil 100 data, kemudian masing-masing data tersebut dirata-rata.



Gambar 9. Data Rata-Rata Kecepatan Putar Motor DC dengan nilai variasi PWM 5



Gambar 10. Data Rata-Rata Kecepatan Putar Motor DC dengan nilai variasi PWM 10

Tabel 1. Data Kecepatan Putar Motor DC dengan nilai variasi PWM 50, 150, dan 255

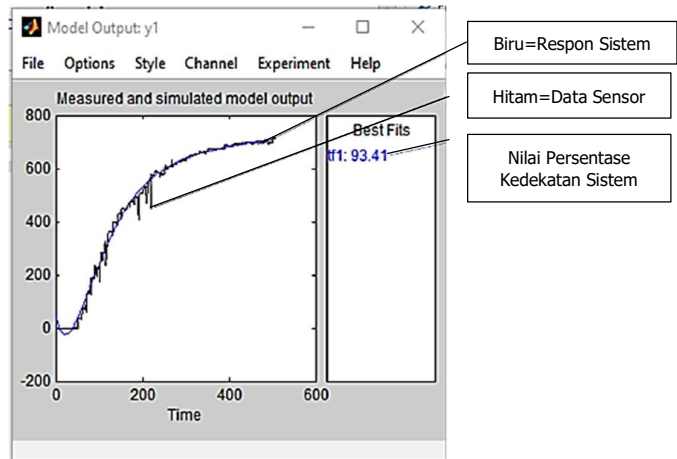
No	PWM	Jumlah Percobaan	Rata-rata RPM
1	50	100x	224,72
2	150	300x	610,7067
3	255	100x	706,95

3.3 Respon Sistem

Respon sistem adalah perubahan perilaku *output* terhadap perubahan sinyal input. Respon sistem berupa kurva yang akan menjadi dasar untuk menganalisis karakteristik sistem selain menggunakan persamaan/model matematika. Bentuk kurva respon sistem dapat dilihat setelah mendapatkan sinyal input. Untuk melihat respon sistem dari data kecepatan putar motor DC/ RPM motor DC dapat menggunakan menu *ident* pada *Matlab*.

3.3.1 Respon Sistem Nilai Variasi PWM 5

Respon sistem dari nilai variasi PWM 5 yang telah didapatkan yang kemudian di lakukan proses identifikasi menggunakan menu *ident* yang terdapat pada program *Matlab*. Hasil proses dapat dilihat pada Gambar 11. Respon sistem kecepatan putar motor DC dengan variasi nilai jarak 5 terdapat 2 hasil kurva (hitam dan biru).

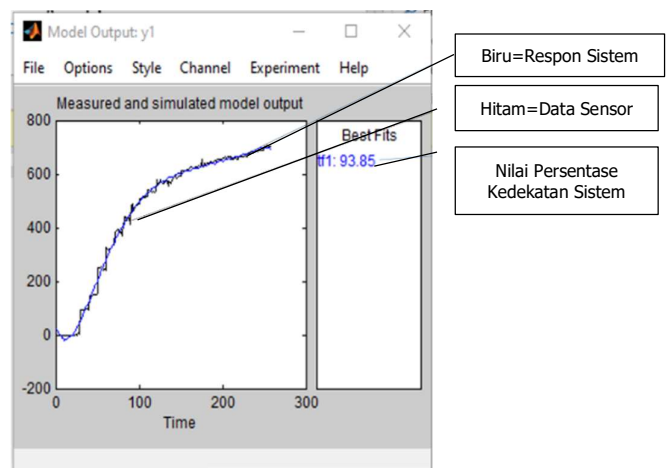


Gambar 11. Respon sistem nilai variasi PWM 5

Kurva berwarna hitam adalah sebagai kurva masukan dari data kecepatan putar motor DC dengan variasi nilai jarak 3 dan kurva yang berwarna biru adalah hasil respon sistem yang didapatkan dari proses identifikasi yang memiliki kedekatan dengan data sesungguhnya sebesar 93,41%.

3.3.2 Respon Sistem Nilai Variasi PWM 10

Data respon sistem dari nilai variasi PWM 10 kemudian dilakukan proses identifikasi sistem menggunakan menu *ident* yang terdapat pada program Matlab. Hasil grafik proses identifikasi sistem dapat dilihat pada Gambar 12. Respon sistem kecepatan putar motor DC dengan variasi nilai selisih PWM 10 angka mendapatkan hasil 2 buah kurva (hitam dan biru). Kurva berwarna hitam adalah kurva masukan dari data kecepatan putar motor DC dengan variasi nilai jarak 10 dan kurva yang berwarna biru adalah hasil respon sistem yang didapatkan dari proses identifikasi sistem. Antara kurva data respon sistem dan kurva data identifikasi sistem memiliki tingkat persentase kemiripan sebesar 93,85% dibandingkan dengan data asli.

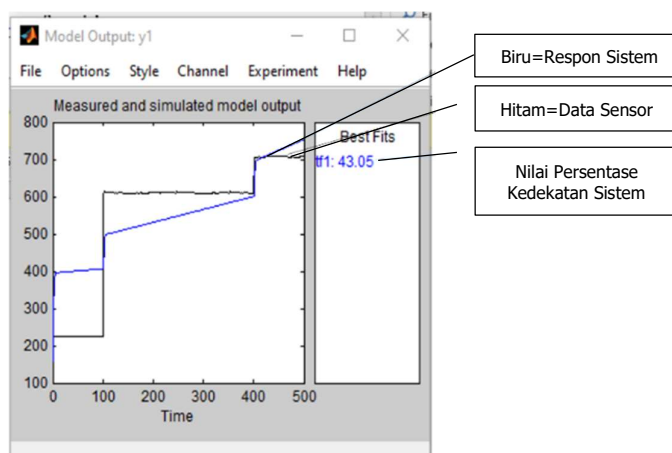


Gambar 12. Respon sistem nilai variasi PWM 10

3.3.3 Respon Sistem Nilai Variasi PWM 50, 150, 255

Respon sistem dari nilai variasi PWM 50, 150 dan 255 yang telah didapatkan yang kemudian dilakukan proses identifikasi menggunakan menu *ident* yang terdapat pada program Matlab dapat dilihat pada Gambar 13. Respon sistem kecepatan putar motor DC dengan nilai variasi

PWM 50, 150 dan 255 terdapat 2 hasil kurva (hitam dan biru) yang merupakan tampilan respon sistem yang didapatkan dari proses identifikasi. Kurva berwarna hitam adalah sebagai kurva masukan atau input dari data kecepatan putar motor DC dan kurva yang berwarna biru adalah hasil respon sistem yang didapatkan dari proses identifikasi yang memiliki presentase kemiripan dengan data sistem sebesar 43,05%.



Gambar 13. Respon sistem nilai variasi PWM 50, 150 dan 255

3.4 Pemodelan Matematis Motor DC

Pemodelan matematis motor DC didapatkan dari pemrosesan nilai estimasi dari hasil data pemodelan yang dilakukan. Pemrosesan estimasi pada menu *ident* dilakukan dengan masuk pada menu estimasi dari *mydata* yang telah diperoleh yang selanjutnya mengatur jumlah *pole* dan *zero* yang akan digunakan untuk mengatur pemrosesan estimasi. Setelah pengaturan *pole* selesai selanjutnya tinggal melakukan eksekusi pada hasil menu estimasi yang telah dilakukan. Nilai *pole* yang digunakan adalah 2 dan nilai *zero* 1 dikarenakan kontrol PID hanya dapat mengendalikan maksimal sistem order 2.

Hasil dari pemrosesan proses estimasi dapat dilihat pada kolom *import models* yang telah terisi. Hasil menu estimasi berupa pemodelan matematis dari motor DC dapat dilihat dengan melakukan double klik pada kolom *import model* dengan nama *tf1*. Pemodelan matematis diperoleh pada proses estimasi oleh menu *ident* program Matlab didapatkan berupa nilai fungsi alih pemodelan motor DC. Nilai fungsi alih yang didapatkan pada proses identifikasi berupa pemodelan matematis dapat dilihat pada Persamaan (4), (5), dan (6). Persamaan (4) sampai Persamaan (6) secara berurutan adalah model dari respon variasi PWM 5, PWM 10 dan PWM 50, 150 dan 255.

$$\frac{0,396s + 0,0001456}{s^2 + 0,03599 + 0,0003185} \quad (4)$$

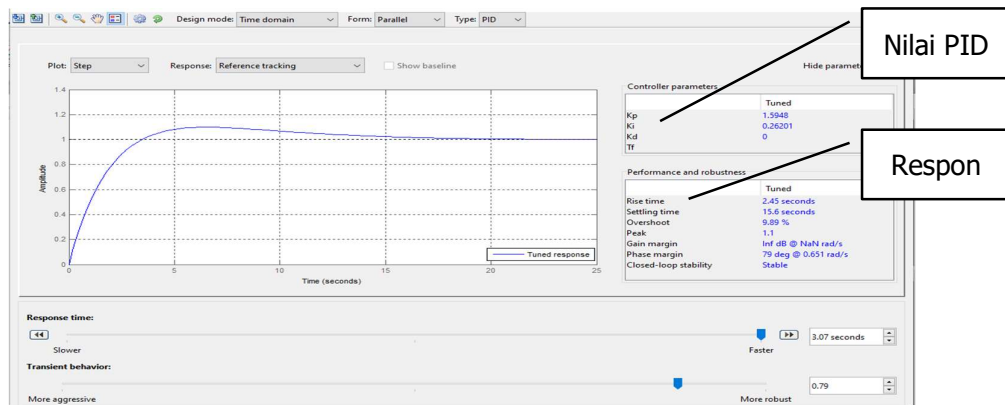
$$\frac{0,5186s + 0,001131}{s^2 + 0,04933 + 0,001086} \quad (5)$$

$$\frac{0,863s + 0,00222}{s^2 + 0,09585 + 0,00000018} \quad (6)$$

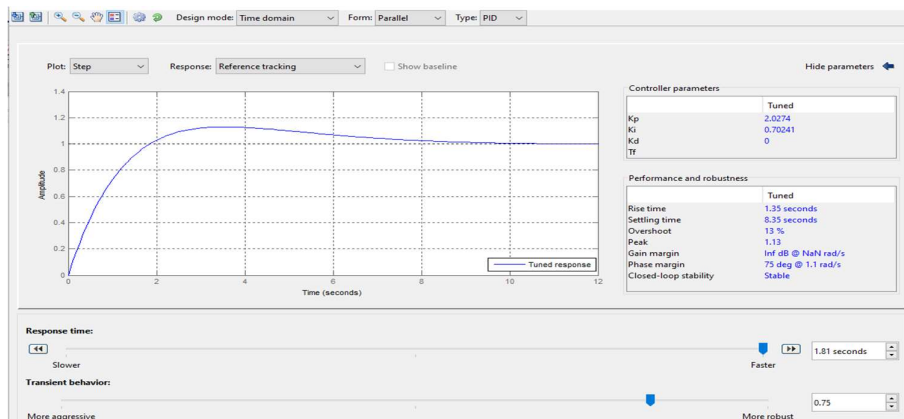
Dari nilai persamaan matematis yang didapatkan dari Persamaan (4), (5), dan (6), kemudian dapat dicari nilai kontrol PID yang akan digunakan untuk mengontrol motor DC.

3.5 Nilai Kontrol PID

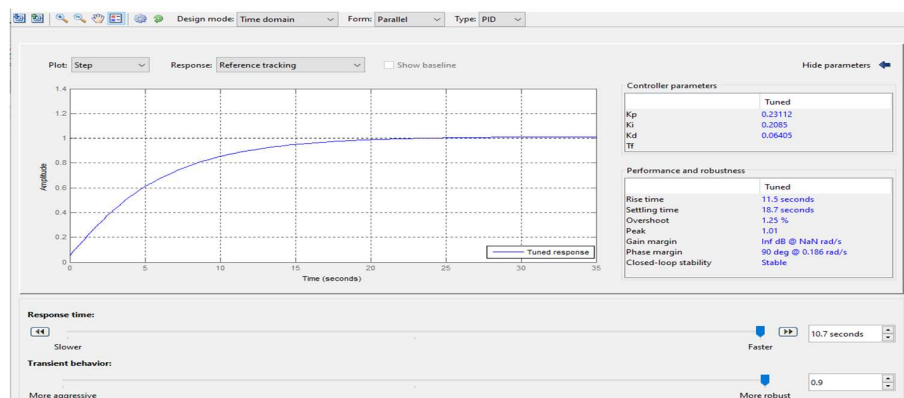
Pemrosesan nilai kontrol PID dilakukan dengan masuk pada menu *PID Tuning* pada *Matlab*. Hasil pemrosesan menu *PID Tuning* dari model matematis motor DC dari nilai variasi PWM 5, PWM 10 dan PWM 50, 150, 255, dapat dilihat pada Gambar 14, Gambar 15 dan Gambar 16. Pengaturan nilai dilakukan dengan menggeser bagian *response time* dan *transient behavior*. Selanjutnya tinggal mengamati grafik respon dan mencari respon yg paling baik. Untuk melihat hasil menu *PID tuning* berupa nilai kontrol KP, KI, dan KD dapat dilihat pada menu bagian kanan.



Gambar 14. Nilai kontrol PID variasi PWM 5



Gambar 15. Nilai kontrol PID variasi PWM 10



Gambar 16. Nilai kontrol PID variasi PWM 50, 150, 255

Nilai kontrol PID dari variasi PWM 3 berupa nilai kontrol yang akan digunakan untuk pengontrolan motor DC adalah $K_P=1,5948$; $K_I=0,26201$; $K_D=0$. Sementara, nilai kontrol PID dari variasi PWM 10, nilai yang didapatkan adalah $K_P=2,0274$; $K_I=0,70241$; $K_D=0$. Lalu, nilai kontrol PID dari variasi PWM 50, 150 dan 255 yang didapatkan adalah $K_P=0,23112$; $K_I=0,2085$; $K_D=0,06405$. Dari nilai kontrol tersebut akan digunakan untuk mengontrol motor DC.

3.6 Hasil Uji Alat

Hasil uji coba alat dilakukan dengan memasukkan nilai pengendali PID yang telah didapatkan pada percobaan sebelumnya. Pengujian alat ini bertujuan untuk melihat respon dari motor DC terhadap nilai kontrol menggunakan program Arduino IDE.

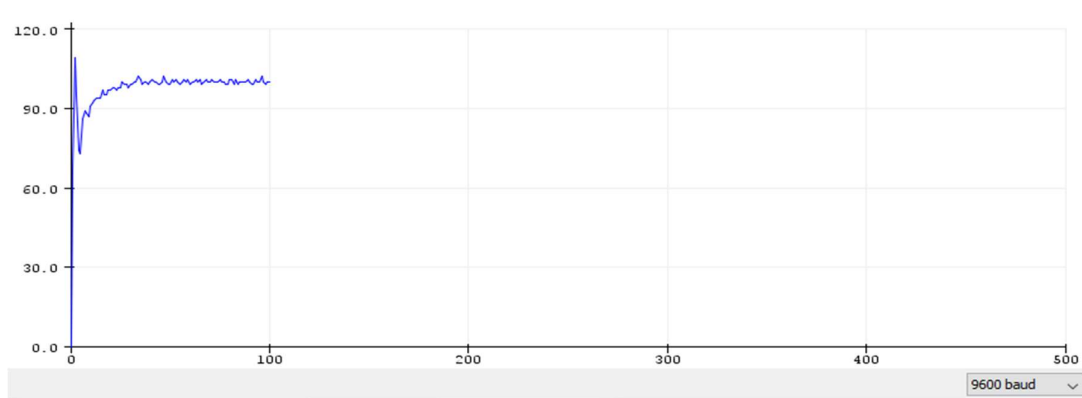
3.6.1 Hasil Uji Coba Alat Variasi PWM 5

Hasil uji coba alat dengan nilai variasi PWM 5 dilakukan dengan memasukkan nilai kontrol PID yang telah didapatkan yaitu berupa nilai kontrol $K_P=1,5948$; $K_I=0,26201$; dan $K_D=0$ yang dimasukkan pada program Arduino IDE pada Gambar 17.

```
float kp=1.5948, ki=0.26201, kd=0;
int sp=100, error=0, last_error=0, sum_error=0;
{
rpm = (float)(encoderValue * 60 / ENCODEROUTPUT);
Serial.println(rpm);
encoderValue = 0;
error = sp - rpm;
sum_error = sum_error + error;
motorSpeed = (kp * error + ki*sum_error + kd * (error-last_error));
analogWrite(PWM, motorSpeed);
last_error = error + sum_error;
}
```

Gambar 17. Program Pengendali Kecepatan Motor DC

Respon terhadap sistem motor dc ditampilkan pada *serial plotter* program *Arduino IDE* berupa grafik respon motor DC. Grafik tersebut menunjukkan seberapa tingkat pengendali PID yang digunakan untuk mencapai nilai referensi dari pengendali tersebut. Tampilan grafik respon motor DC pada serial *Plotter Arduino IDE* yang ditunjukkan pada Gambar 18.



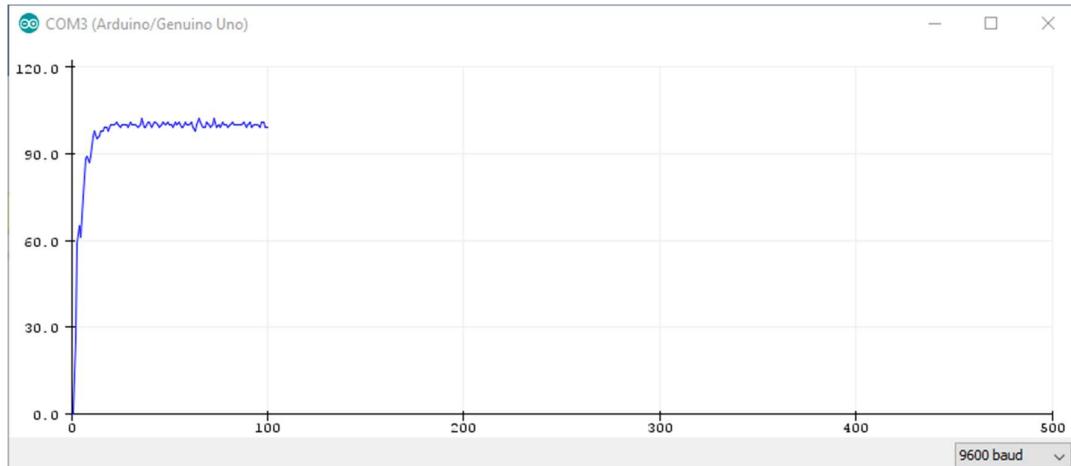
Gambar 18. Grafik respon motor DC variasi PWM 5

Dari grafik nilai variasi PWM 5 dengan pengaturan nilai kontrol $K_P=1,5948$; $K_I=0,26201$; $K_D=0$ tersebut mencapai nilai stabil atau konstan pada nilai set poin 100. Nilai *rise time* adalah 1,3646

detik; *settling time* 23,5 detik; dan *overshoot* 11 persen. Pada pengujian ini respon sistem tidak terlalu baik karena terdapat overshoot dan undershoot.

3.6.2 Hasil Uji coba Alat Variasi PWM 10

Hasil uji coba alat dengan nilai variasi PWM 10 dilakukan dengan memasukkan nilai kontrol PID yang telah didapatkan yang berupa nilai kontrol $K_P=2,0274$; $K_I=0,70241$; $K_D=0$. Tampilan grafik respon motor DC pada *serial plotter* Arduino IDE yang ditunjukkan pada Gambar 19.

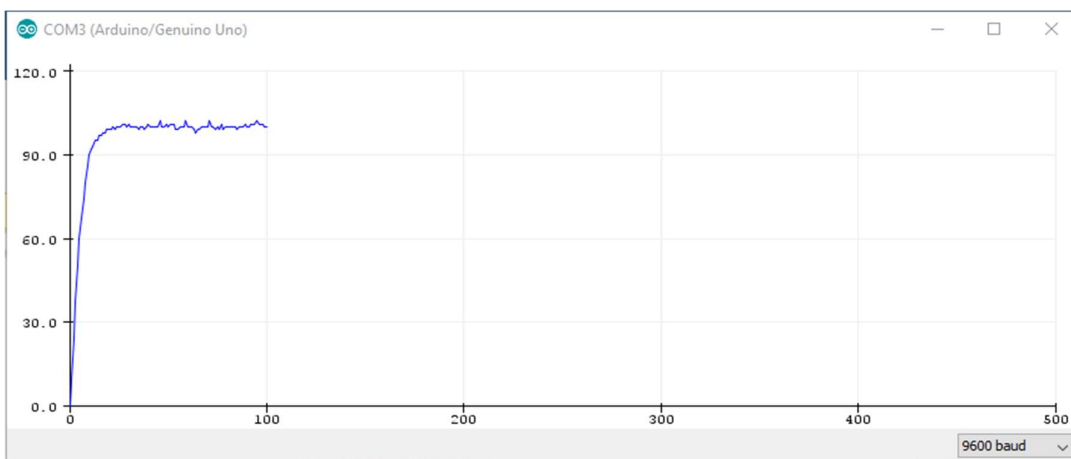


Gambar 19. Grafik respon motor DC variasi PWM 10

Dari grafik nilai variasi PWM 10 dengan pengaturan nilai kontrol $K_P=2,0274$; $K_I=0,70241$; $K_D=0$ tersebut mencapai nilai stabil atau konstan pada nilai set poin 100. Nilai *rise time* adalah 8,3654 detik, *settling time* 93,5 detik dan *overshoot* 3 persen. Pada pengujian ini, respon sistem sudah sudah baik tanp ada overshoot yang terlalu besar dan tanpa overshoot. Namun grafik kenaikan respon sistem masih agak kasar.

3.6.3 Hasil Uji coba Alat Variasi PWM 50, 150 dan 255

Hasil uji coba alat dengan nilai variasi PWM 50, 150 dan 255 dilakukan dengan memasukkan nilai kontrol PID yang telah didapatkan yang berupa nilai kontrol $K_P=0,23112$; $K_I=0,2085$; dan $K_D=0,06405$. Tampilan grafik respon motor DC pada *serial plotter* Arduino IDE yang ditunjukkan pada Gambar 20.



Gambar 20. Grafik respon motor DC variasi PWM 50, 150 dan 255

Dari grafik nilai variasi PWM 50, 150 dan 255 dengan pengaturan nilai kontrol $K_P=0,23112$; $K_I=0,2085$; dan $K_D=0,06405$ tersebut mencapai nilai stabil atau konstan pada nilai set poin 100. Nilai *rise time* adalah 9,4286 detik; *settling time* 18,5 detik; dan *overshoot* 2 persen. Dari penelitian ini didapatkan hasil nilai kendali PID yang paling baik adalah dinilai kontrol variasi pwm 50, 150, dan 255 karena mempunyai respon sistem yang baik berupa nilai *rise time* yang cepat, *settling time* yang rendah, *overshoot* yang kecil, tanpa *undershoot* dan kenaikan grafik juga halus.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, kendali motor DC dengan pengendali PID menggunakan pendekatan pemodelan identifikasi sistem telah mampu untuk memberikan nilai kendali PID yang sesuai untuk spesifikasi motor yang diinginkan. Penelitian ini menggunakan beberapa parameter nilai variasi PWM dan mendapatkan 3 parameter PWM yang paling baik dalam memperoleh nilai respon sistem motor DC yaitu nilai variasi PWM 5, nilai variasi PWM 10 dan nilai variasi PWM gabungan 50, 150, 255. Dari ketiga variasi PWM tersebut didapatkan respon sistem yang berupa nilai *rise time*, *settling time*, *overshoot*, serta nilai *error* yang sangat kecil. Nilai variasi PWM 50, 150, dan 255 memiliki nilai respon sistem terbaik karena ciri-ciri pengendali PID yang bagus adalah pengendali PID yang memiliki nilai *rise time* yang cepat, mempunyai nilai *settling time* yang rendah serta memiliki nilai *overshoot* yang kecil.

DAFTAR RUJUKAN

- Achanta, R. K., & Pamula, V. K. (2018). DC motor speed control using PID controller tuned by jaya optimization algorithm. *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering*, (pp. 983–987).
- Anggraini, S. F., Ma'arif, A., & Puriyanto, R. D. (2020). Pengendali PID pada Motor DC dan Tuning Menggunakan Metode Differential Evolution (DE). *TELKA - Telekomunikasi Elektronika Komputasi Dan Kontrol*, 6(2), 147–159.
- Apriaskar, E., Azis Salim, N., & Prastiyanto, D. (2019). Performance Evaluation of Balancing Bicopter using P, PI, and PID Controller. *Jurnal Teknik Elektro*, 11(2), 44–49.
- Ardiansyah, R. A. (2017). Perancangan dan Pengujian Sistem Pengendali Sudut untuk Motor DC Brushless Menggunakan Kendali Algoritma P-D. *Jurnal Rekayasa Elektrika*, 13(2), 82–86.
- Djalal, M. R., & Rahmat. (2017). Optimisasi Kontrol PID Untuk Motor DC Magnet Permanen Menggunakan Particle Swarm Optimization. *Jurnal TAM (Technology Acceptance Model)*, 8(2), 117–122.
- Djalal, M. R., & Rahmat. (2020). Penalaan Optimal Kendali Motor DC Berbasis Ant Colony Optimization. *Jurnal Teknologi*, 12(1), 49–56.
- Fahmizal, F., Fathuddin, F., & Susanto, R. (2018). Identifikasi Sistem Motor DC dan Kendali Linear Quadratic Regulator Berbasis Arduino-Simulink Matlab. *Majalah Ilmiah Teknologi*

- Elektro*, 17(2), 299–306.
- Febriawan, A., & Aji, W. S. (2020). Rotating Control on Robots Indonesian Abu Robot Contest with PID and IMUBNO055 Controls. *Buletin Ilmiah Sarjana Teknik Elektro*, 2(1), 14–23.
- Hammoodi, S. J., Flayyih, K. S., & Hamad, A. R. (2020). Design and implementation speed control system of DC motor based on PID control and matlab simulink. *International Journal of Power Electronics and Drive Systems*, 11(1), 127–134.
- Iswanto, Ma'arif, A., Puriyanto, R. D., Raharja, N. M., & Rahmadhia, S. N. (2020). Arduino Embedded Control System of DC Motor Using Proportional Integral Derivative. *International Journal of Control and Automation*, 13(4), 658–667.
- Khadari, I., Subiyanto, & Prastiyanto, D. (2019). Simulasi Kontroler PID Tuning Menggunakan Logika Fuzzy dan Algoritma Genetika Sebagai Pengendali Kecepatan Motor DC. *Setrum*, 8(2), 186–196.
- Maarif, A., Puriyanto, R. D., & Hasan, F. R. T. (2020). Robot Keseimbangan dengan Kendali PID dan Kalman Filter. *IT Journal Research and Development (ITJRD)*, 4(2), 117–127.
- Masrukhan, M. N., Mulyo, M. P., Ajiatmo, D., & Ali, M. (2016). Optimasi Kecepatan Motor DC Menggunakan PID dengan Tuning Ant Colony Optimization (ACO) Controller. *Prosiding SENTIA-2016*, (pp. 49–52).
- Purbowaskito, W., & Hsu, C.-H. (2017). Sistem Kendali PID untuk Pengendalian Kecepatan Motor Penggerak Unmanned Ground Vehicle untuk Aplikasi Industri Pertanian. *Jurnal Infotel*, 9(4), 376–381.
- Ramdani, E. (2015). Parameter Identifikasi Transfer Fungsi Menggunakan MATLAB. *Setrum*, 4(1), 30–36.
- Tang, W. J., Liu, Z. T., & Wang, Q. (2017). DC motor speed control based on system identification and PID auto tuning. *Chinese Control Conference*, (pp. 6420–6423).
- Waluyo, Fitriansyah, A., & Syahril. (2013). Analisis Penalaan Kontrol PID pada Simulasi Kendali Kecepatan Putaran Motor DC Berbeban menggunakan Metode Heuristik. *ELKOMIKA*, 1(2), 79–92.
- Wiriawan, A. R. J., & Irawan, A. (2016). Pengaturan Kecepatan Motor DC dengan Kontrol Proporsional Integral Derifatif (PID) Berbasis LabView. *TELEKONTRAN*, 4(2), 13–24.