

Identifikasi Sistem Motor DC dan Penerapan Kendali PID, LQR, dan *Servo* Tipe 1 Berbasis Arduino-MATLAB

Muhammad Rakha Firdaus^{1*}, Tegar Arif Berbudi¹, Salima Nurrahma¹, Galang Izzaulhaq¹, Imroatul Hudati¹

¹Departemen Teknik Elektro dan Informatika, Universitas Gadjah Mada; tegar.a.b@mail.ugm.ac.id, salima.nurrahma@mail.ugm.ac.id, galangizzaulhaq@mail.ugm.ac.id, imroatul.hudati@ugm.ac.id

*Korespondensi: rakhafdr30@mail.ugm.ac.id

Abstract – *There has been a lot of research on the application of control to DC motors. However, there are often obstacles regarding the selection of controllers that are not appropriate, resulting in the control of DC motors becoming unstable. So to overcome this, a system identification process is needed on the DC motor. This paper will explain the process of identifying the DC motor coupling system using the System Identification Toolbox in MATLAB. After obtaining the modeling, several types of control will be applied, namely PID, LQR, and type-1 servo to be compared to determine effective control for the DC motor system. From the results of the system identification process, it is obtained that the system model closest to the reference model is the ARX model with a best-fit value of 63.2%. Furthermore, this ARX model will be used as a mathematical model of the system to which PID, LQR, and type-1 servo controls will be added. From the experimental results, it is found that the best type of control in this system is to use type-1 servo control which produces the smallest MSE value of 4.9897.*

Keywords – *identification system, LQR, PID, type-1 servo*

Intisari – Penelitian mengenai penerapan kendali pada motor DC telah banyak dilakukan. Namun sering terjadi kendala mengenai pemilihan pengontrol yang tidak tepat sehingga mengakibatkan pengendalian motor DC menjadi tidak stabil. Sehingga untuk mengatasi hal ini, diperlukan suatu proses identifikasi sistem pada motor DC. Pada makalah ini menjelaskan proses identifikasi sistem kopling motor DC dengan menggunakan *System Identification Toolbox* pada MATLAB. Setelah mendapatkan pemodelan tersebut maka akan diterapkan beberapa macam kendali yaitu PID, LQR, dan *servo* tipe 1 untuk dibandingkan untuk menentukan kendali yang efektif untuk sistem motor DC tersebut. Dari hasil proses identifikasi sistem diperoleh bahwa model sistem yang paling mendekati dengan model referensi adalah model ARX dengan nilai *best-fit* sebesar 63,2%. Selanjutnya model ARX ini akan digunakan sebagai model matematis sistem yang akan ditambahkan kendali PID, LQR, dan *servo* tipe 1. Dari hasil percobaan didapatkan bahwa jenis kendali paling baik pada sistem ini adalah dengan menggunakan kendali *servo* tipe 1 yang menghasilkan nilai MSE paling kecil yaitu sebesar 4,9897.

Kata kunci – *identifikasi sistem, LQR, PID, servo tipe 1*

I. PENDAHULUAN

Motor arus searah atau biasa disebut motor DC (*direct current*) merupakan salah satu jenis penggerak yang sering digunakan dalam teknologi kontrol baik di sektor industri maupun di sektor rumah tangga [1]. Prinsip kerja dari motor DC ini adalah arah medan magnet stator akan selalu berada pada posisi yang berlawanan arah dengan arah medan magnet rotor yang mana apabila medan magnet yang berlawanan maka akan saling tarik menarik dan magnet yang searah akan saling tolak-menolak dan ketika daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu [2].

Penelitian mengenai penerapan kendali pada motor DC telah banyak dilakukan. Namun, sering terjadi kendala mengenai pemilihan pengontrol yang tidak tepat sehingga mengakibatkan pengendalian motor DC menjadi tidak stabil. Sehingga untuk mengatasi hal ini, diperlukan suatu proses identifikasi sistem pada motor DC untuk mengetahui dan mempelajari sifat dari sistem motor DC tersebut [3]. Dengan itu, dapat ditentukan tindakan yang tepat agar sistem berperilaku sesuai yang diinginkan.

Untuk memperoleh pemodelan matematis hasil identifikasi sistem dari sebuah sistem terdapat dua cara, yaitu

dengan pendekatan analisis dan eksperimen [4]. Pemodelan sistem dari motor DC dapat diperoleh melalui eksperimen dengan cara mengamati data masukan dan data keluaran dari motor DC [5]. Hubungan antara masukan dan keluaran akan dievaluasi dengan sistem identifikasi *tool* yang terdapat pada MATLAB.

Penelitian ini menggunakan dua motor DC yang digabungkan menggunakan kopling sebagai sebuah sistem. Sistem akan diidentifikasi menggunakan *System Identification Toolbox* pada MATLAB yang terintegrasi dengan Arduino sehingga masukan dan keluaran sistem tersebut dapat diamati secara *real time*. Setelah mendapatkan pemodelan tersebut, kemudian diterapkan beberapa macam kendali yaitu PID, LQR, dan *servo* tipe 1 untuk menentukan kendali yang paling efektif. Untuk menentukan kendali yang efektif dapat digunakan parameter seperti nilai *rise time*, *settling ime*, *overshoot*, dan *error steady state*. Dengan menentukan dan mengoptimalkan parameter tersebut, sistem kendali dapat bekerja dengan lebih efektif dan menghasilkan performa yang optimal.

II. DASAR TEORI

A. Identifikasi Sistem

Identifikasi sistem merupakan suatu cara yang digunakan untuk memperoleh model dari suatu sistem yang kemudian akan dibandingkan dengan hasil masukan dan keluaran yang didapatkan dari hasil percobaan [6]. Untuk melakukan proses identifikasi sistem dapat dilakukan dengan melakukan langkah-langkah sebagai berikut yaitu [4]:

- 1) Pada sistem yang akan diidentifikasi harus tersedia data masukan dan keluarannya
- 2) Pemilihan struktur model
- 3) Estimasi parameter-parameter model
- 4) Validasi model yang telah diidentifikasi

B. Model Matematis Motor DC

Motor DC merupakan peralatan yang menerapkan prinsip elektromagnetik dasar dengan mengubah energi listrik menjadi energi mekanik [7]. Pemodelan matematis dari motor DC didapatkan dari pendekatan rangkaian ekuivalen dengan menggunakan hukum-hukum rangkaian listrik serta prinsip hubungan elektro mekanis. Sebuah motor DC terdiri dari rangkaian elektronis dan rangkaian mekanis yang dapat dilihat pada Gambar 1.

Pemodelan dari motor DC dapat dilakukan melalui dua tahapan yaitu dengan pemodelan elektris dan pemodelan mekanis. Masukan dari motor DC adalah berupa tegangan arus searah dan keluarannya adalah kecepatan sudut. Dari pemodelan sistem elektris pada Gambar 1 maka didapatkan persamaan diferensial seperti pada (1).

$$e_a(t) - e_{ggl}(t) = L_a \cdot \frac{di_a(t)}{dt} + R_a \cdot i_a(t) \quad (1)$$

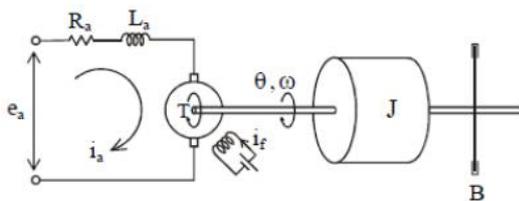
Setelah itu transformasi Laplace dari (1) ditunjukkan pada (2).

$$E_a(s) - E_{ggl}(s) = (L_a \cdot s + R_a)I(s) \quad (2)$$

$$I(s) = \frac{1}{L_a \cdot s + R_a} (E_a(s) - E_{ggl}(s))$$

Torsi pada motor berbanding lurus dengan hasil kali dari arus kumparan jangkar dan medan magnetik yang dihasilkan oleh penguat medan, yang berbanding lurus dengan arus medan. Relasi ini ditunjukkan pada (3).

$$B = \frac{\mu_f \cdot N_f \cdot i_f(t)}{2\pi \ell_f} = K_B \cdot i_f(t) \quad (3)$$



Gambar 1. Pemodelan sistem motor DC [4]

Dengan K_B adalah konstanta medan magnetik, sehingga menjadi (4).

$$T = K_B \cdot i_f(t) i_a(t) \ell_a r_a n_a \quad (4)$$

$$T = K_{TM} \cdot i_f(t) i_a(t)$$

Dengan K_{TM} adalah konstanta torsi motor. Karena arus medan i_f konstanta maka menjadi (5).

$$T = K_{TM} \cdot i_a(t) \quad (5)$$

Transformasi Laplace (6).

$$T(s) = K_{TM} \cdot i_a(s) \quad (6)$$

Torsi yang dihasilkan motor bekerja terhadap inersia dan gesekan viskositas, sehingga menjadi (7).

$$T = J \frac{d\omega(t)}{dt} + B\omega(t) \quad (7)$$

Transformasi Laplace menjadi (8).

$$T(s) = Js\omega(s) + B\omega(s) \quad (8)$$

$$\theta(s) = \frac{1}{Js + B} T(s)$$

Besarnya tegangan gaya gerak listrik berbanding lurus dengan hasil kali dari arus medan dan kecepatan sudut motor (9).

$$e_{ggl}(t) = K \cdot i_f(t) \cdot \omega(t) \quad (9)$$

Dengan K dan i_f adalah konstanta sehingga menjadi (10).

$$e_{ggl}(t) = K_e(t) \cdot \omega(t) \quad (10)$$

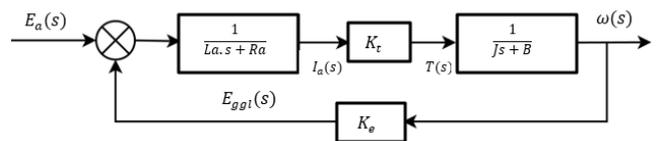
Dengan K_e adalah konstanta tegangan gaya gerak listrik balik. Transformasi Laplace menjadi (11).

$$E_{ggl}(s) = K_e \cdot \omega(s) \quad (11)$$

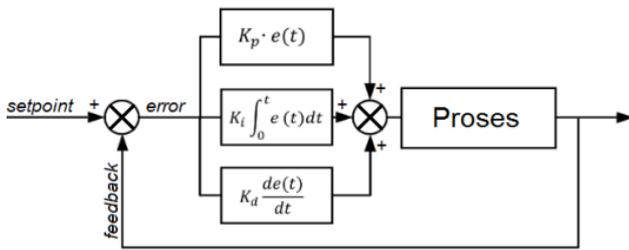
Dari hasil analisis perhitungan sebelumnya, maka diperoleh blok diagram dari pemodelan sistem motor DC seperti pada Gambar 2. Maka fungsi alih dari pemodelan motor DC didapatkan, seperti pada (12).

$$\frac{C(s)}{R(s)} = \frac{Kt}{(Las + Ra)(Js + B)} \cdot \frac{KtKe}{1 + (Las + Ra)(Js + B)} \quad (12)$$

$$\frac{C(s)}{R(s)} = \frac{Kt}{JLas^2 + (BLa + JRa)s + (RaB + KtKe)}$$



Gambar 2. Blok diagram sistem motor DC



Gambar 3. Blok diagram pengontrol PID

C. Pengendali PID

Proportional-integral-derivative (PID) merupakan kontrol mekanisme umpan balik yang sering digunakan di dunia industri. Kendali PID digunakan dalam *system loop* tertutup yang memiliki umpan balik dari keluaran sistem untuk mencapai respons yang diinginkan. Kontrol PID terdiri dari proporsional, integral dan derivatif dan saling memiliki pengaruh terhadap yang lain [8].

Gambar 3 menunjukkan bahwa jika ketiga parameter tersebut dikombinasikan secara optimal akan menghasilkan respons keluaran yang baik dan memberikan informasi yang relevan dan berguna secara jelas. Kendali PID memiliki masukan berupa sinyal kesalahan $e(t)$ dan keluaran pengontrol merupakan sinyal kontrol $u(t)$. Hubungan antara masukan dan keluaran pengontrol PID tertulis pada (13) [9].

$$u(t) = K_p \left(e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + \tau_d \frac{de(t)}{dt} \right) \quad (13)$$

Hasil Laplace dari (13) adalah:

$$U(s) = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) E(s) \quad (14)$$

Pada (14), K_p merupakan penguatan proporsional dan τ_i merupakan waktu integral dan τ_d adalah waktu diferensial. Ketiga parameter tersebut dapat ditentukan nilainya. Sehingga fungsi alih pengontrol PID ditunjukkan oleh (15).

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (15)$$

Tipe pengontrol proporsional K_p akan memberikan pengaruh pengurangan nilai *rise-time* tetapi tidak menghapus nilai *steady-state error*.

Kontrol integral K_i akan memberikan pengaruh menghilangkan *error steady state* tetapi menurunkan respons *transient*. Sedangkan pengontrol derivatif K_d akan memberikan pengaruh meningkatkan stabilitas sistem dan mengurangi nilai *overshoot*.

D. Linear Quadratic Regulator (LQR)

Linear quadratic regulator (LQR) merupakan salah satu kontrol optimal untuk menstabilkan dan mengontrol sistem dengan mengubah lokasi *pole* [8]. Metode ini menggunakan pendekatan *state equation* dan aljabar Ricatti yang digunakan

untuk mengoptimalkan proses sistem dengan bentuk linear [10]. Untuk melakukan kontrol, LQR mengambil *state* sistem dinamis dan masukan kontrol. Untuk mendesain sebuah kontrol LQR, sistem matriks A dan B harus bersifat *controllable* [11].

Apabila diketahui suatu sistem yang direpresentasikan dalam bentuk *state space* (16).

$$\dot{x} = Ax + Bu \quad (16)$$

State feedback dari kontrol u adalah $u = -Kx$ sehingga menjadi (17).

$$\dot{x} = (A - BK)x \quad (17)$$

Dengan K diperoleh dari minimalisasi indeks kinerja (18).

$$J = \int (x^T Q x + u^T R u) dt \quad (18)$$

Dengan Q merupakan matriks *semi-definite* positif dan R merupakan matriks *definite* positif. Kemudian, nilai K diperoleh dengan (19).

$$K = R^{-1} B^T P \quad (19)$$

Dengan menggunakan solusi persamaan Ricatti, maka nilai dari matriks P dapat dihitung dengan nilai matriks A, B, dan C dengan (20).

$$A^T P + PA - P B R^{-1} B^T P + Q = 0 \quad (20)$$

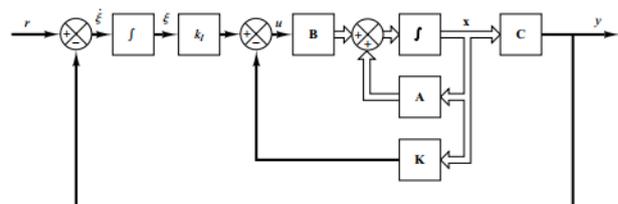
E. Desain Sistem Servo Tipe 1

Sistem *servo* tipe 1 merupakan salah satu teknik kontrol *tracking* yang dirancang agar sinyal keluaran dapat mengikuti sinyal referensi yang diberikan. Sistem ini menggunakan matriks *gain* kontrol K dan *integrator* K_i yang didapatkan dengan teknik *pole placement* maupun LQR [12]. Blok diagram sistem *servo* tipe 1 ditunjukkan oleh Gambar 4.

Diasumsikan bahwa sebuah *plant* mempunyai bentuk persamaan (21) sebagai berikut.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (21)$$

Jika *plant* tidak memiliki integrator, maka perlu ditambahkan integrator pada bagian *feed forward*, sehingga menjadi (22).



Gambar 4. Blok diagram sistem servo tipe 1 tanpa integrator

$$\begin{aligned} u &= -Kx + k_1\xi \\ \dot{\xi} &= r - y = r - Cx \end{aligned} \quad (22)$$

Dengan,

- x state vector plant
- u sinyal kontrol
- y sinyal output
- ξ output integrator
- r sinyal input referensi

Maka fungsi alih dari sistem menjadi (23).

$$G_p(s) = C(sI - A)^{-1}B \quad (23)$$

Sehingga state equation menjadi (24).

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\xi}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \xi(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \quad (24)$$

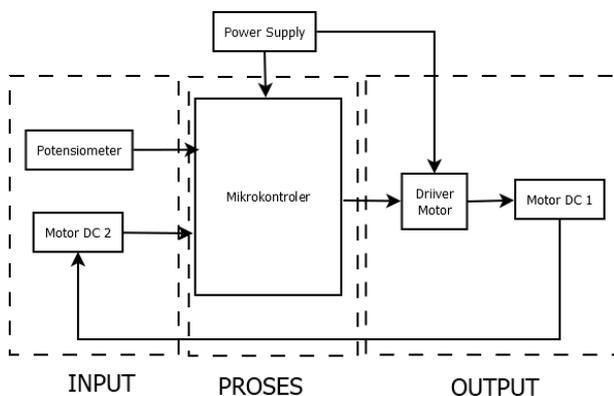
III. METODOLOGI

A. Perancangan Perangkat Keras

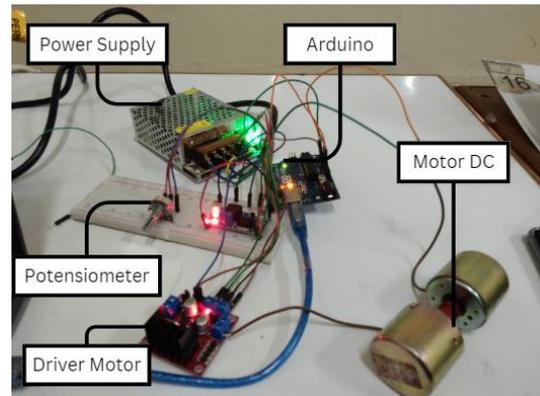
Proses perancangan dilakukan dengan membuat *wiring diagram* dari kopling motor DC dengan bantuan dari *software fritzing*. Gambar 5 merupakan *wiring diagram* dari sistem motor kopling dengan Arduino Uno sebagai pengontrol sistem. Arduino akan mendapatkan suplai tegangan sebesar 12 volt yang akan disalurkan ke *driver* motor untuk menggerakkan motor DC pertama. Besar PWM akan diatur menggunakan potensiometer. Pada saat motor DC pertama berputar, maka *shaft* motor DC kedua akan ikut berputar, sehingga akan menghasilkan tegangan yang akan dibaca oleh Arduino. Tegangan tersebut merupakan data keluaran yang akan digunakan untuk keperluan identifikasi sistem. Gambar 6 menunjukkan *hardware* yang telah dirancang.

B. Perancangan Sistem Akuisisi Data

Sistem akuisisi data diperlukan untuk pengambilan data dari sistem menuju komputer untuk melakukan identifikasi sistem. Diagram blok rancangan sistem akuisisi data dapat dilihat pada Gambar 7.



Gambar 5. Wiring diagram motor DC



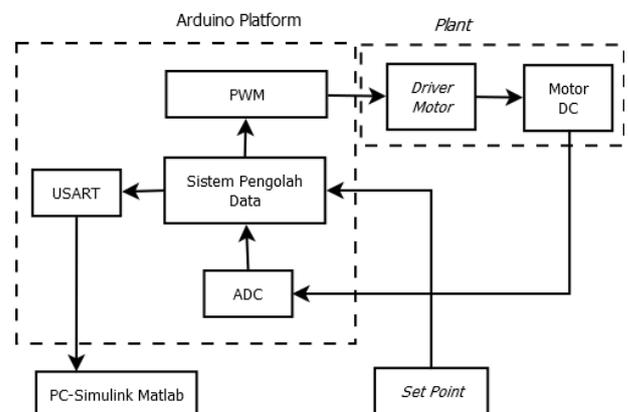
Gambar 6. Hardware identifikasi sistem motor DC

Akuisisi data menggunakan Arduino sebagai sistem untuk pengontrol dan pengolah data. Arduino akan mengontrol nilai PWM yang akan diberikan ke *driver* motor untuk mengatur kecepatan motor DC pertama, lalu akan membaca data serial yang dihasilkan oleh motor DC kedua berupa data tegangan. Untuk menyimpan data serial Arduino digunakan *software* MATLAB yang terhubung secara serial dengan Arduino sehingga data-data tersebut dapat diolah untuk keperluan identifikasi sistem.

C. Pemodelan Identifikasi Sistem MATLAB

Setelah mendapatkan data dari proses akuisisi data, langkah selanjutnya adalah melakukan identifikasi sistem menggunakan *System Identification Toolbox* di MATLAB, [13]. Dengan menggunakan fitur tersebut, maka dapat dilakukan simulasi dan analisis sistem dinamis dengan *graphical user interface* (GUI).

Untuk menggunakan *System Identification Toolbox*, yang harus dilakukan pertama kali adalah memasukkan data. Data yang dimaksud adalah data masukan berupa hasil tegangan oleh potensiometer dan data keluaran yang merupakan tegangan keluaran dari motor DC. Data tersebut telah diakuisisi sebelumnya menggunakan Arduino yang terhubung dengan MATLAB.



Gambar 7. Diagram blok rancangan sistem akuisisi data [4]

Langkah selanjutnya adalah melakukan proses *pre-proces* data. Pada proses ini dapat dipilih representasi model matematis yang diinginkan. Pada penelitian ini digunakan representasi model polinomial, yaitu model AR, ARX, ARMA, dan ARMAX.

Setelah mendapatkan model matematis, selanjutnya dilakukan pengujian kesesuaian respons dari model hasil estimasi dengan respons sistem referensi. Model yang menghasilkan nilai *best-fit* yang paling tinggi akan digunakan sebagai model matematis yang akan digunakan. Model kemudian ditambah dengan pengontrol untuk mengatur respons sistem.

D. Perancangan Kontroler PID

Sebelum menentukan nilai paramater pengontrol proporsional integral derivatif (PID), karakteristik sistem harus ditinjau terlebih dahulu [14]. Nilai-nilai parameter dari pengontrol PID dicari secara *trial and error*. Spesifikasi umum yang digunakan dalam perancangan pengontrol PID adalah *rise-time*, *settling-time*, *maximum overshoot*, dan *error steady state* terhadap masukan yang diberikan [15]. Perubahan pada setiap nilai parameter akan memengaruhi parameter yang lainnya sehingga digunakan MATLAB untuk melakukan *tuning* nilai PID secara otomatis. Diagram blok kontrol PID dapat dilihat pada Gambar 8.

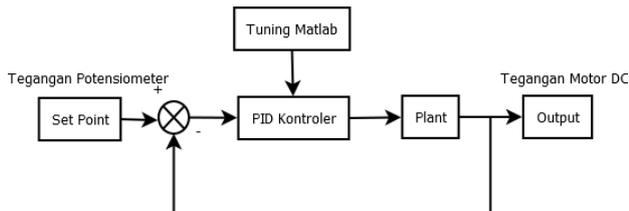
Nilai parameter PID akan di-*tuning* menggunakan fitur *tuning* PID pada MATLAB. Blok *Plant* merupakan hasil identifikasi sistem yang telah dimodelkan dalam bentuk fungsi alih. Data keluaran yang merupakan tegangan keluaran dari motor DC akan menjadi nilai *feedback* dan akan dibandingkan dengan nilai *set point*, yaitu tegangan keluaran dari potensiometer. Pengontrol PID akan menghitung nilai *error* antara *set point* yang dengan nilai keluaran sistem.

E. Perancangan LQR dan Sistem Servo Tipe 1

Sebelum menerapkan kendali LQR dan sistem *servo* tipe 1, terlebih dahulu diamati respons dari *open loop system* dalam bentuk fungsi alih yang didapatkan dari hasil identifikasi sistem. Hal ini dilakukan untuk mengetahui respons sistem sebelum diberikan kendali.

Untuk menerapkan kendali LQR dan *servo* tipe 1, fungsi alih diubah menjadi bentuk *state space* (21) sebagai berikut.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{21}$$



Gambar 8. Diagram blok kontrol PID

Selanjutnya adalah analisis *controlability* dari sistem, dengan seluruh *behaviour state* (x) harus dapat dikendalikan oleh sinyal kendali (u). *Controllability* sistem dapat diuji dengan matriks *controlability*. Sebuah sistem akan dikatakan *completely controllable* jika matriks *controlability* (C_M) mempunyai *rank* yang sama dengan orde sistem, atau *full rank*. Dengan kata lain, jika determinan matriks tidak sama dengan nol, maka sistem tersebut *completely controllable*.

$$\begin{aligned} C_M &= [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B] \\ A^T P + PA - PBR^{-1}B^T P + Q &= 0 \end{aligned} \tag{25}$$

Apabila sistem *fully controllable*, selanjutnya adalah menentukan nilai matriks Q dan R dan dengan menggunakan persamaan Ricatti pada (25). Jika nilai dari matriks P didapatkan, maka nilai K pada *feedback* akan didapatkan.

Selanjutnya dalam merancang sistem *servo* tipe 1 yaitu merepresentasikan sistem dalam bentuk *state space*. Lalu dengan matriks *gain* kontrol K dan *integrator* K_i yang didapatkan menggunakan teknik LQR sebelumnya.

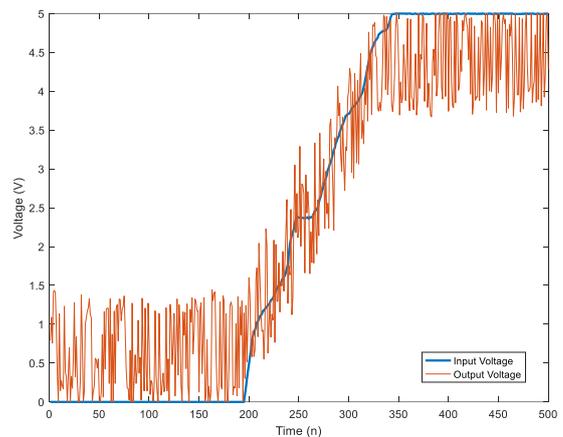
IV. HASIL DAN PEMBAHASAN

A. Akuisisi Data dari Arduino ke MATLAB

Akuisisi data dilakukan dengan menghubungkan *microcontroller* ke MATLAB, sehingga data masukan dari potensiometer dan data keluaran dari motor DC akan dibaca dan disimpan oleh MATLAB untuk keperluan identifikasi sistem. Gambar 9 menunjukkan hasil pembacaan data masukan dan keluaran sistem.

Terdapat dua data yang disajikan, yaitu pembacaan tegangan pada potensiometer (biru) dan pembacaan tegangan keluaran motor DC (oranye). Dari kedua data tersebut akan dihitung rata-rata nilai *error* dengan persamaan RMSE (26).

$$RMSE = \sqrt{\sum \frac{(v1 - v2)^2}{n}} \tag{26}$$



Gambar 9. Data masukan dan keluaran Arduino

Variabel $v1$ merupakan data masukan dari potensiometer dan $v2$ merupakan data keluaran dari motor DC. Sementara itu, n merupakan jumlah data, yaitu 500. Berdasarkan hasil perhitungan, diperoleh nilai *error* sebesar 0,7258.

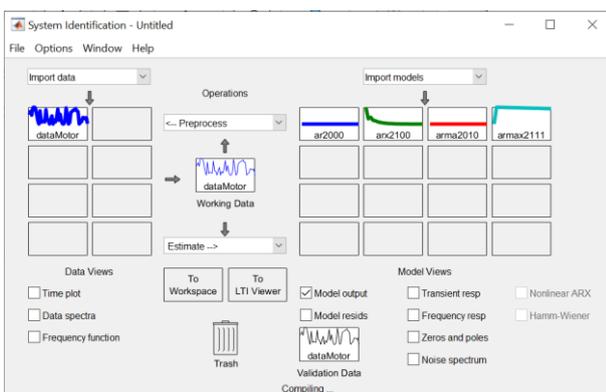
Pada Gambar 9 dapat dilihat bahwa pada saat nilai potensiometer ditingkatkan, terlihat terjadi kenaikan dari 0 V sampai dengan 5 V. Semakin besar tegangan pada potensiometer, semakin tinggi nilai PWM. Hal ini menyebabkan perubahan kecepatan putaran motor DC. Semakin besar kecepatan motor DC penggerak maka akan menyebabkan semakin besarnya putaran dan keluaran tegangan motor DC pembaca. Sehingga Gambar 9 menunjukkan grafik dari keluaran akan mengikuti sinyal referensi dari tegangan masukan. Namun, pada data keluaran terdapat banyak *noise* sehingga nilai bacaan tegangan keluaran menjadi tidak stabil pada nilai yang sudah ditetapkan.

B. Identifikasi Sistem

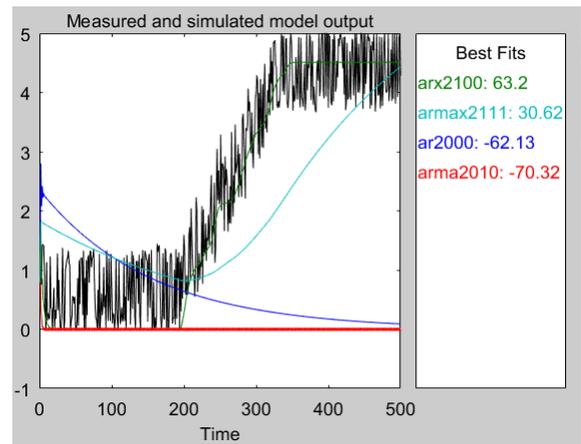
Setelah didapatkan data masukan dan keluaran dari hasil proses akuisisi data, selanjutnya adalah melakukan identifikasi sistem. Proses identifikasi sistem dilakukan dengan *System Identification Toolbox* dari MATLAB. Dengan bantuan MATLAB, didapatkan pemodelan sistem dalam bentuk AR, ARX, ARMA, dan ARMAX dengan melakukan proses estimasi. Didapatkan hasil identifikasi sistem pada *interface* seperti pada Gambar 10.

Dari proses estimasi akan didapatkan bentuk diskrit dari setiap pemodelan sehingga dapat diketahui bagaimana bentuk pemodelan sistem dalam bentuk AR, ARX, ARMA, dan ARMAX. Kemudian, akan dianalisis bagaimana kesesuaian respons dari setiap pemodelan untuk menentukan pemodelan yang menghasilkan respons yang hampir sama dengan sinyal referensinya. Hasil uji kesesuaian respons dapat dilihat pada Gambar 11.

Grafik berwarna hitam merupakan respons sistem atau disebut sebagai sinyal referensi. Selain itu juga dapat dilihat hasil grafik dari setiap pemodelan.



Gambar 10. Proses pemodelan sistem dengan *System Identification Toolbox*



Gambar 11. Grafik uji kesesuaian respons

Dari keempat grafik pemodelan tersebut dapat dilihat bahwa respons yang paling mendekati sinyal referensi adalah pemodelan dalam bentuk ARX. Selain itu, dapat dilihat bahwa *System Identification Toolbox* juga menunjukkan nilai *best-fit* yang dihasilkan dari setiap pemodelan. Nilai *best-fit* pemodelan ARX adalah yang paling tinggi jika dibandingkan pemodelan yang lainnya, yaitu 63,2%. Sehingga dapat disimpulkan bahwa pemodelan sistem dalam bentuk polinomial yang akan menghasilkan respons terbaik adalah pemodelan ARX. Untuk itu, pada pemodelan sistem ini akan ditambahkan pengontrol untuk mengatur respons sistem yang diinginkan.

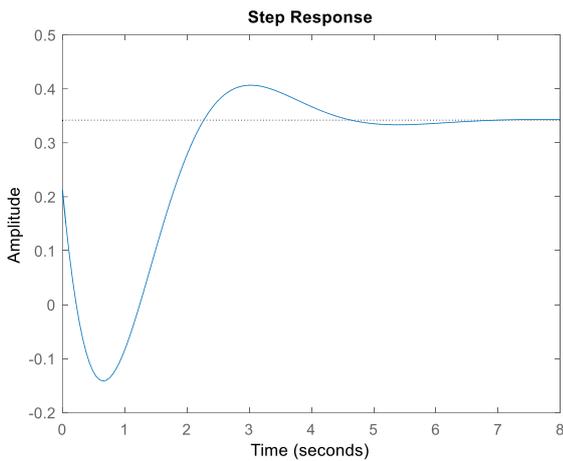
C. Pengontrol PID, LQR, dan Servo Tipe 1

Sebelum melakukan pengontrolan, dilakukan proses konversi bentuk diskrit dari pemodelan ARX ke bentuk *state space*. Hal ini dikarenakan proses pengendalian hanya bekerja pada sistem kontinu. Proses dilakukan dengan dengan mengubah model diskrit yang didapat pada proses pemodelan sistem menjadi bentuk *state space* menggunakan perhitungan MATLAB. Tabel 1 menunjukkan hasil konversi diskrit ke kontinu.

Setelah bentuk *state space* sistem didapatkan, maka proses selanjutnya adalah mengamati respons sistem sebelum ditambahkan kendali. Hal ini bertujuan untuk mengetahui mekanisme pengontrolan yang tepat untuk memperbaiki respons sistem.

Tabel 1. Hasil konversi diskrit ke kontinu

Model ARX Diskrit	
$A(z) = 1 - 0,3068 z^{-1} - 0,3124 z^{-2}$	
$B(z) = 0,3438$	
Model State Space	
$A = \begin{bmatrix} -1,699 & -2,484 \\ 1 & 0 \end{bmatrix}$	$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
$C = [-1,21 \quad 0,3219]$	$D = 0,2123$

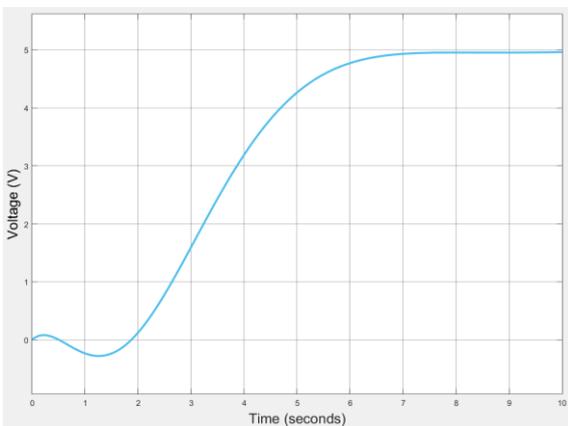


Gambar 12. Respons sistem sebelum ditambahkan kendali

Gambar 12 menunjukkan bahwa respons sistem sebelum diberi kendali memiliki *undershoot* yang cukup besar, memiliki *overshoot* sebesar 18,9%, dan belum mencapai nilai *set point* yang diberikan. Untuk itu, diperlukan sistem kendali untuk memperbaiki respons sistem.

1) *Propotional, Integral, Derivative (PID)*: Proses selanjutnya yaitu merancang pengontrolan PID dengan menentukan nilai parameter yang tepat agar respons sistem sesuai yang diinginkan. Untuk membantu penentuan nilai parameter-parameter tersebut digunakan fitur *tuning* PID pada MATLAB. Fitur ini dapat menentukan nilai parameter PID secara otomatis dengan mengubah respons sistem sesuai yang diinginkan. Dari proses *tuning* PID, didapatkan nilai yang terbaik, yaitu nilai konstanta proporsional sebesar 0, integral senilai 0,77, dan derivatif sebesar 0. Gambar 13 menunjukkan respons sistem dengan kontrol PID.

Setelah penambahan kendali PID pada sistem, respons sistem akan mencapai kestabilannya pada nilai *set point*-nya, yaitu 5 V dan sistem juga memiliki *undershoot* yang kecil dan tidak memiliki *overshoot*. Namun, sistem ini masih memerlukan waktu yang cukup lama untuk mencapai nilai *steady state*-nya.



Gambar 13. Respons sistem dengan kontrol PID

2) *Linear Quadratic Regulator (LQR)*

Selanjutnya akan diamati bagaimana pengaruh dari penambahan kontrol LQR pada sistem. Sebelum menerapkan kontrol LQR, *controllability* dari sistem harus diperiksa terlebih dahulu. Dengan MATLAB, didapatkan bahwa sistem bersifat *fully controllable* karena matriks *controllability* sama dengan *rank* sistem. Hasil perhitungan menggunakan MATLAB disajikan pada Tabel 2.

Proses selanjutnya adalah mencari elemen matriks K dengan LQR. Nilai K dicari menggunakan perintah berikut pada MATLAB “[S, e, k, y]=care(A, B, Q, R)”. Pada penelitian ini nilai matriks Q dan matriks R ditentukan secara *trial and error* untuk mencari hasil respons sistem yang terbaik. Pada Tabel 3 disajikan hasil terbaik dengan variasi nilai matriks Q dan R.

Diperoleh respons sistem yang terbaik dengan nilai matriks $Q = [1 \ 0; 0 \ 1]$ dan matriks $R = 10$. Namun, kendali LQR ini tidak mampu membuat sistem mencapai nilai *set point* yang telah ditentukan. Kendali LQR hanya dapat mengurangi nilai *overshoot* sistem menjadi 13%. Grafik respons dapat dilihat pada Gambar 14.

3) *Sistem Servo Tipe 1*: Pengendalian sistem yang terakhir adalah menggunakan *servo* tipe 1. Untuk menggunakan kontrol jenis ini, diperlukan nilai *gain* penguatan K dan penguatan K_i . Kedua nilai gain dapat dicari menggunakan perintah “ $K_{hat} = acker(A_{hat}, B_{hat}, J)$ ”. Nilai J merupakan elemen matriks 3x1 yang ditentukan secara *trial and error* untuk menghasilkan nilai penguatan K dan penguatan K_i terbaik.

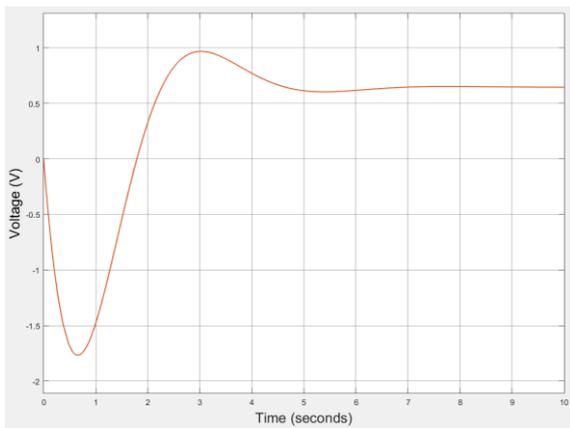
Besaran dari matriks J akan menentukan nilai K dan K_i yang akan dihasilkan. Semakin besar nilai matriks *gain* K, maka akan semakin cepat respons sistem, tetapi *overshoot* dan *steady-state error* semakin besar. Sebaliknya, semakin kecil nilai matriks *gain* K, semakin lambat respons sistem, tetapi *overshoot* dan *steady-state error* semakin kecil. Sedangkan nilai *gain* K_i digunakan untuk meminimalkan nilai *steady-state error* dari sistem.

Tabel 2. Matriks controllability dan rank sistem

Matriks Controllability	Rank Sistem
$Cm = \begin{bmatrix} 1 & -1.699 \\ 0 & 1 \end{bmatrix}$	2

Tabel 3. Matriks S dan K pada variasi nilai LQR

No.	Q	R	S	K
1	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	1	$\begin{bmatrix} 0,3 & 0,1 \\ 0,1 & 1,3 \end{bmatrix}$	$\begin{bmatrix} 0,3 & 0,1 \end{bmatrix}$
2	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	10	$\begin{bmatrix} 0,4 & 0,2 \\ 0,2 & 1,3 \end{bmatrix}$	$\begin{bmatrix} 0,04 & 0,02 \end{bmatrix}$
3	$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$	1	$\begin{bmatrix} 2,2 & 1,5 \\ 1,5 & 11,8 \end{bmatrix}$	$\begin{bmatrix} 2,2 & 1,5 \end{bmatrix}$
4	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	100	$\begin{bmatrix} 0,4 & 0,2 \\ 0,2 & 1,3 \end{bmatrix}$	$\begin{bmatrix} 0,004 & 0,002 \end{bmatrix}$



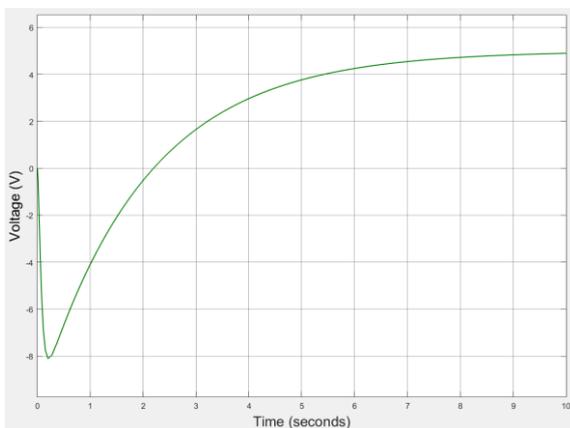
Gambar 14. Respons sistem dengan kendali LQR

Tabel 4. Penguatan K dan Ki

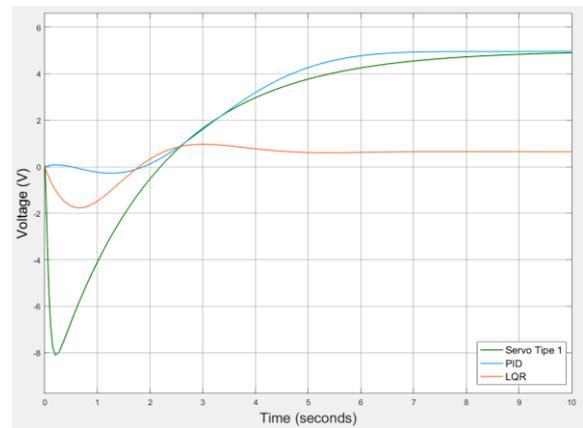
Ahat Bhat J	Khat	
	K=[Khat(1) Khat(2)]	Ki=-Khat(3)
$Ahat = \begin{bmatrix} -1,69 & -2,48 & 0 \\ 1 & 0 & 0 \\ 1,21 & -0,3 & 0 \end{bmatrix}$	[0,04 1,75]	[-0,93]
$Bhat = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$		
$J = [-0,5 - 20 - 30]$		

Dari hasil percobaan didapatkan bahwa respons terbaik sesuai dengan performansi yang diinginkan, yaitu nilai *rise-time* yang kecil serta nilai *steady-state error* yang kecil, didapatkan ketika nilai $J = [-0,5 - 20 - 30]$. Nilai penguatan K dan Ki disajikan pada Tabel 4.

Setelah mendapatkan nilai penguatan K dan penguatan Ki, selanjutnya akan diamati bagaimana respons sistem setelah ditambahkan penguatan K dan Ki. Respons sistem pada Gambar 15 menunjukkan bahwa sistem dapat mencapai nilai kestabilannya pada nilai *set point* 5 V dan sistem tidak memiliki *overshoot*. Namun, sistem pada awalnya mengalami *undershoot* terlebih dahulu dan sistem akan mulai stabil setelah tujuh detik.



Gambar 15. Respons sistem dengan kendali servo tipe 1



Gambar 16. Perbandingan Hasil Penerapan Kendali

Tabel 5. Nilai MSE dari tiap jenis kendali

Jenis Kendali	MSE
PID	5,0893
LQR	5,7307
Servo tipe 1	4,9897

4) *Perbandingan Kendali PID, LQR, dan Servo Tipe 1:* Setelah melakukan percobaan penambahan kendali pada sistem dengan berbagai jenis kendali yaitu kendali PID, kendali LQR, dan kendali servo tipe 1, diketahui bahwa setiap kendali memiliki kelebihan dan kekurangan masing-masing dalam memperbaiki respons sistem. Gambar 16 menunjukkan perbandingan respons dari beberapa kendali yang telah diaplikasikan. Nilai *Mean Square Error* (MSE) digunakan untuk menentukan jenis kendali yang terbaik dari di antara ketiga jenis kendali tersebut. MSE merupakan rata-rata nilai *error* antara masukan dan keluaran sistem. Tabel 5 menunjukkan nilai MSE yang didapatkan dari setiap jenis kendali. Dari Tabel 5 diketahui bahwa kendali dengan nilai rata-rata *error* terkecil adalah kendali menggunakan kendali servo tipe 1, dengan MSE sebesar 4,8979. Sehingga dapat disimpulkan bahwa kendali yang terbaik untuk digunakan pada sistem adalah kendali servo tipe 1 karena pada kendali servo tipe 1, *error* yang didapatkan akan dikendalikan oleh dua *gain* sehingga akan memperkecil kesalahan.

V. SIMPULAN

Dengan melakukan identifikasi sistem, maka pemodelan matematis dari suatu sistem akan dapat diketahui. Proses identifikasi sistem dimulai dari proses akuisisi data dari sistem hingga mengolahnya menggunakan *System Identification Toolbox* untuk memperoleh pemodelan matematis dalam bentuk AR, ARX, ARMA, dan ARMAX. Dari hasil pemodelan sistem, didapatkan bahwa pemodelan dengan model ARX akan menghasilkan pemodelan paling mendekati keluaran sistem dan menghasilkan *best-fit* yang paling tinggi, yaitu sebesar 63,2%. Setelah melakukan percobaan untuk menerapkan beberapa kendali yaitu PID, LQR, dan servo tipe 1, diketahui didapatkan bahwa kendali

PID merupakan kendali *servo* tipe 1 terbaik dengan menghasilkan nilai MSE sebesar 4,9897.

DAFTAR PUSTAKA

- [1] N. R. Wibowo, "Rancang Bangun Sistem Kendali Kecepatan Motor Dc Sebagai Media Pembelajaran Praktikum Sistem Kendali Menggunakan Labview," *JST (Jurnal Sains Ter.)*, vol. 6, no. 2, 2020, doi: 10.32487/jst.v6i2.775.
- [2] Anthoinete P.Y.Waroh, "Analisa dan simulasi sistem pengendali motor dc," *J. Ilm. Sains*, vol. 14, no. 2, p. 80, 2014.
- [3] X. L. Wang, T. Y. Yin, J. T. Chen, J. X. Liang, and Y. Li, "Rapid Design of DC Motor Speed Control System Based on MATLAB," *Appl. Mech. Mater.*, vol. 743, pp. 168–171, Mar. 2015, doi: 10.4028/www.scientific.net/AMM.743.168.
- [4] F. Fahmizal, F. Fathuddin, and R. Susanto, "Identifikasi Sistem Motor DC dan Kendali Linear Quadratic Regulator Berbasis Arduino-Simulink Matlab," *Maj. Ilm. Teknol. Elektro*, vol. 17, no. 2, p. 399, 2018, doi: 10.24843/mite.2018.v17i02.p20.
- [5] W. J. Tang, Z. T. Liu, and Q. Wang, "DC motor speed control based on system identification and PID auto tuning," *Chinese Control Conf. CCC*, pp. 6420–6423, 2017, doi: 10.23919/ChiCC.2017.8028376.
- [6] F. Mangkusasmito, D. Y. Tadeus, and A. Subari, "Implementasi Identifikasi Sistem Metode Black Box Pada Motor Dc Menggunakan Correlation Analysis Dan Model Arx," *Gema Teknol.*, vol. 20, no. 4, pp. 134–139, 2020, doi: 10.14710/gt.v20i4.29381.
- [7] A. MA'ARIF, R. ISTIARNO, and S. SUNARDI, "Kontrol Proporsional Integral Derivatif (PID) pada Kecepatan Sudut Motor DC dengan Pemodelan Identifikasi Sistem dan Tuning," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 9, no. 2, p. 374, 2021, doi: 10.26760/elkomika.v9i2.374.
- [8] R. Muhandian and K. Krismadinata, "Kendali Kecepatan Motor DC Dengan Controller PID dan Antarmuka Visual Basic," *JTEV (Jurnal Tek. Elektro dan Vokasional)*, vol. 6, no. 1, p. 328, 2020, doi: 10.24036/jtev.v6i1.108034.
- [9] A. R. Maulana, "Pengaturan Kecepatan Motor Dc Pada Simulator Hybrid Electric Vehicel (HEV) Menggunakan Kontroler Pid Berdasarkan Algoritma Particle Swarm Optimization (PSO)," p. 115, 2015, [Online]. Available: <http://repository.its.ac.id/60216/>.
- [10] Q. P. Syahna *et al.*, "ANALISIS RESPON SISTEM KENDALI LQR (LINEAR QUADRATIC REGULATOR) PADA SIMULASI GIMBAL," 2019.
- [11] R. Soni and Sathans, "Optimal control of a ball and beam system through LQR and LQG," *Proc. 2nd Int. Conf. Inven. Syst. Control. ICISC 2018*, no. Icisc, pp. 179–184, 2018, doi: 10.1109/ICISC.2018.8399060.
- [12] D. Rahmawati and T. Agustinah, "Stabilisasi Sistem Pendulum Kereta Menggunakan Kontroler Linear Quadratic Gaussian," *J. Tek. ITS*, vol. 6, no. 2, 2017, doi: 10.12962/j23373539.v6i2.24153.
- [13] E. Ramdani, "Parameter Identifikasi Transfer Fungsi Menggunakan MATLAB," *Setrum Sist. Kendali-Tenaga-elektronika-telekomunikasi-komputer*, vol. 4, no. 1, p. 30, 2016, doi: 10.36055/setrum.v4i1.463.
- [14] E. Apriaskar, F. Fahmizal, N. A. Salim, and D. Prastiyanto, "Performance Evaluation of Balancing Bicopter using P, PI, and PID Controller," *J. Tek. Elektro*, vol. 11, no. 2, pp. 44–49, 2019, doi: 10.15294/jte.v11i2.23032.
- [15] R. F. Nugroho, "Sistem Pengendalian Motor Dc Menggunakan Pid Dengan Metode Ziegler – Nichols (Implementasi Palang Pintu Parkir)," *Foreign Aff.*, 2019.