

E-ISSN : 2746 - 2536

# Jurnal Listrik, Instrumentasi, dan Elektronika Terapan

Volume 4 No. 1 - April 2023

**Departemen Teknik Elektro dan Informatika**  
Sekolah Vokasi Universitas Gadjah Mada  
Jl. Yacaranda, Sekip Unit IV, Yogyakarta. 55281

## EDITORIAL BOARD

### Editor-in-Chief

Muhammad Arrofiq, Universitas Gadjah Mada

### Associate Editor

Candra Febri Nugraha, Universitas Gadjah Mada

### Editor

Budi Sumanto, Universitas Gadjah Mada

Esa Prakasa, Badan Riset dan Inovasi Nasional

Fahmizal, Universitas Gadjah Mada

Fakih Irsyadi, Universitas Gadjah Mada

Imroatul Hudati, Universitas Gadjah Mada

Suhono, Universitas Gadjah Mada

### Copyeditor

Jans Hendry, Universitas Gadjah Mada

Ahmad Adhiim Muthahhari, Universitas Gadjah Mada

### Assistant Editor

Andi Fariel, Universitas Gadjah Mada

Rismanda Kusumadewi, Universitas Gadjah Mada

Published by:

- 1. Identifikasi Sistem Motor DC dan Penerapan Kendali PID, LQR, dan Servo Tipe 1 Berbasis Arduino-MATLAB** 1 - 9  
Muhammad Rakha Firdaus, Tegar Arif Berbudi, Salima Nurrahma, Galang Izzaulhaq, Imroatul Hudati
- 2. Linierisasi Model Sistem Ball and Beam dan Metode Kendali LQG Berbasis MATLAB-Simulink** 10 - 16  
Hanifah Suwardi, Anisa Septyaning Choir, Salima Nurrahma, Imroatul Hudati
- 3. Implementasi Kontrol PID pada Kopel Motor DC dengan Menggunakan Filter Kalman** 17 - 22  
Salima Nurrahma, Tegar Arif Berbudi, Muhammad Rakha Firdaus, Galang Izzaulhaq, Imroatul Hudati
- 4. Kontrol Motor DC dengan Kendali Linear Quadratic Regulator dan Filter Kalman Menggunakan GUI MATLAB** 23 - 31  
Imroatul Hudati, Nurul Izzah Wijayakusuma, Aditya Yusuf Ismail, Alfonzo Aruga Paripurna Barus, Hendi Budianto
- 5. Design of Water Quality Monitoring System for Koi Fish Farming Using NodeMCU ESP32 and Blynk Application Based on Internet of Things** 32 - 37  
Dendi Rosandi, Junaidi Junaidi, Donni Kis Apriyanto, Arif Surtono

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

Muhammad Rakha Firdaus<sup>1\*</sup>, Tegar Arif Berbudi<sup>1</sup>, Salima Nurrahma<sup>1</sup>, Galang Izzaulhaq<sup>1</sup>, Imroatul Hudati<sup>1</sup>

<sup>1</sup>Departemen 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).

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

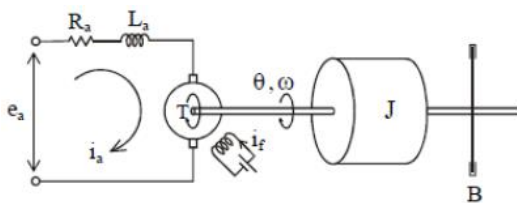
Setelah itu transformasi Laplace dari (1) ditunjukkan pada (2).

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

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

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).

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

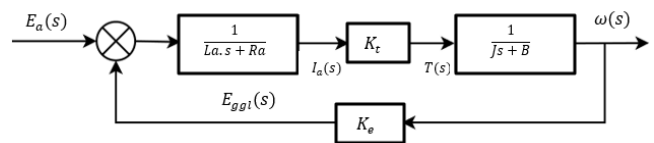
Dengan  $K_e$  adalah konstanta tegangan gaya gerak listrik balik. Transformasi Laplace menjadi (11).

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

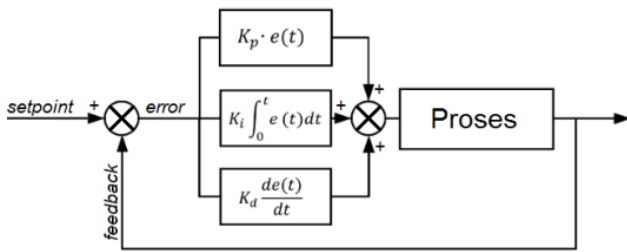
Dari hasil analisis perhitungan sebelumnya, maka diperoleh blok diagram dari pemodelan sistem motor DC seperti pada Gambar 2. Maka fungsi alih dari pemodelan motor DC didapatkan, seperti pada (12).

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

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



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  $\tau_i$  merupakan waktu integral dan  $\tau_d$  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).

$$J = \int (x^T Qx + u^T Ru) dt \quad (18)$$

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).

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

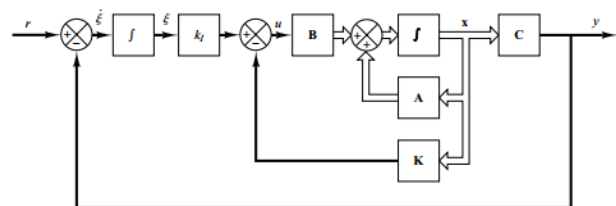
E. Desain Sistem Servo Tipe 1

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
- $\xi$  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).

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

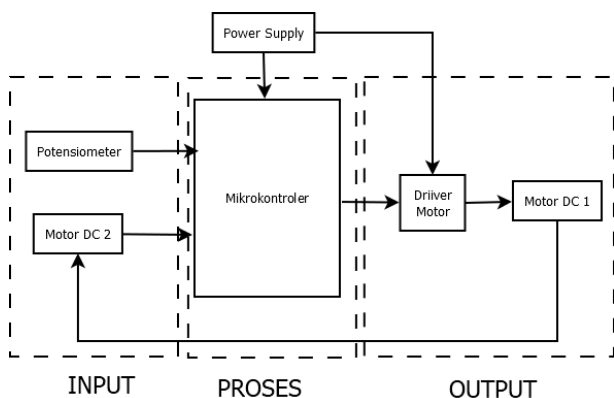
### III. METODOLOGI

#### A. Perancangan Perangkat Keras

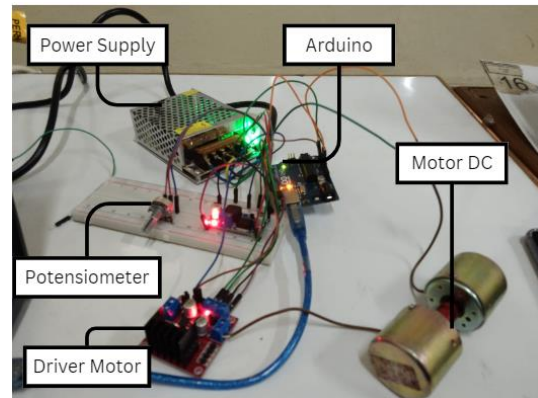
Proses perancangan dilakukan dengan membuat *wiring diagram* dari kopling motor DC dengan bantuan dari *software fritzing*. Gambar 5 merupakan *wiring diagram* dari sistem motor kopling dengan Arduino Uno sebagai pengontrol sistem. Arduino akan mendapatkan suplai tegangan sebesar 12 volt yang akan disalurkan 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



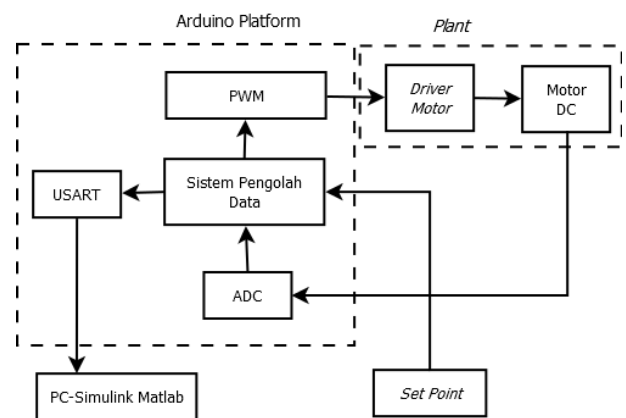
Gambar 6. *Hardware* identifikasi sistem motor DC

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.

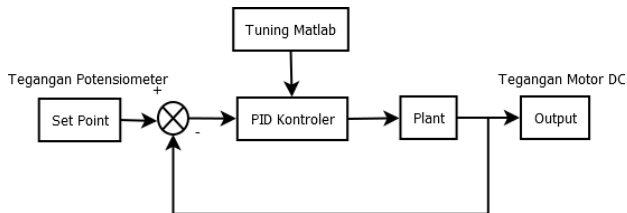
Nilai parameter PID akan di-*tuning* menggunakan fitur *tuning* PID pada MATLAB. Blok *Plant* merupakan hasil identifikasi sistem yang telah dimodelkan dalam bentuk fungsi alih. Data keluaran yang merupakan tegangan keluaran dari motor DC akan menjadi nilai *feedback* dan akan dibandingkan dengan nilai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

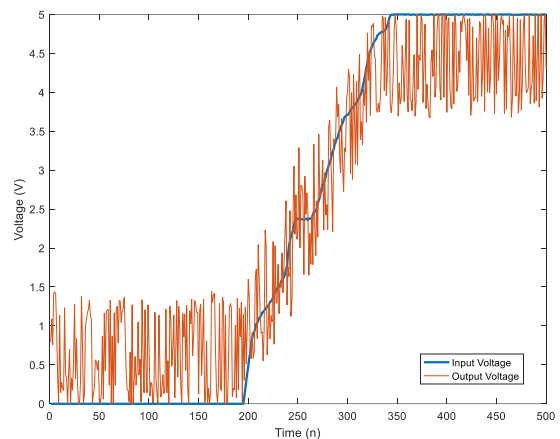
**IV. HASIL DAN PEMBAHASAN**

**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.

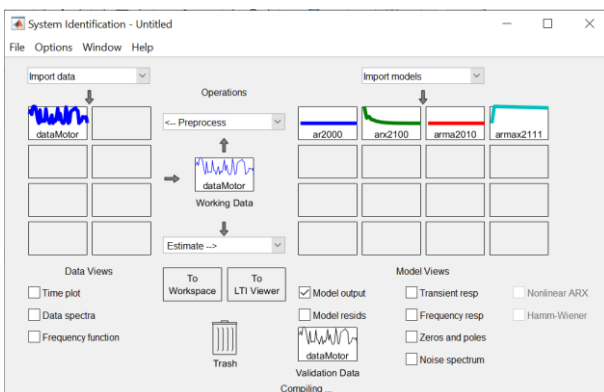
Pada Gambar 9 dapat dilihat bahwa pada saat nilai potensiometer ditingkatkan, terlihat terjadi kenaikan dari 0 V sampai dengan 5 V. Semakin besar tegangan pada potensiometer, semakin tinggi nilai PWM. Hal ini menyebabkan perubahan kecepatan putaran motor DC. Semakin besar kecepatan motor DC penggerak 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**

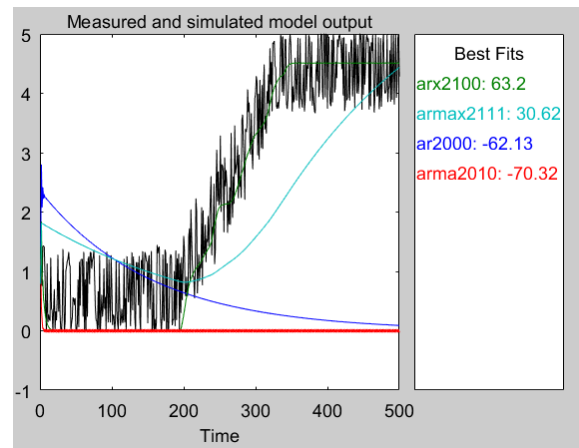
Setelah didapatkan data masukan dan keluaran dari hasil proses akuisisi data, selanjutnya adalah melakukan identifikasi sistem. Proses identifikasi sistem dilakukan dengan *System Identification Toolbox* dari MATLAB. Dengan bantuan MATLAB, didapatkan pemodelan sistem dalam bentuk AR, ARX, ARMA, dan ARMAX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

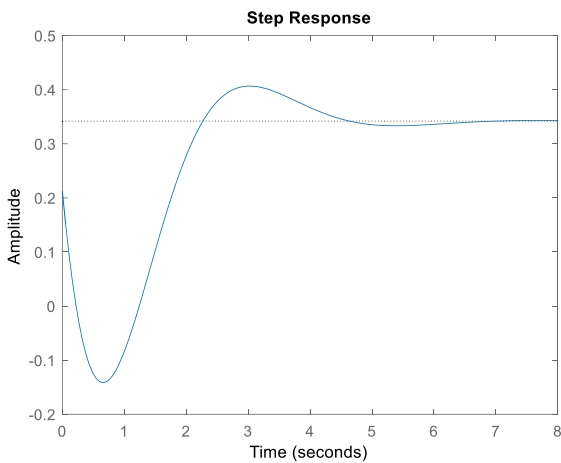
**C. Pengontrol PID, LQR, dan Servo Tipe 1**

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

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

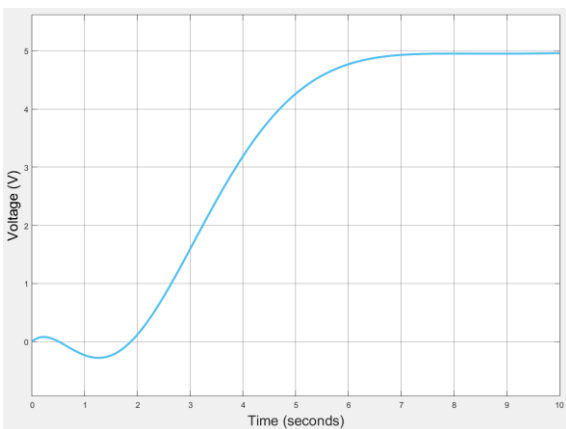


Gambar 12. Respons sistem sebelum ditambahkan kendali

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.

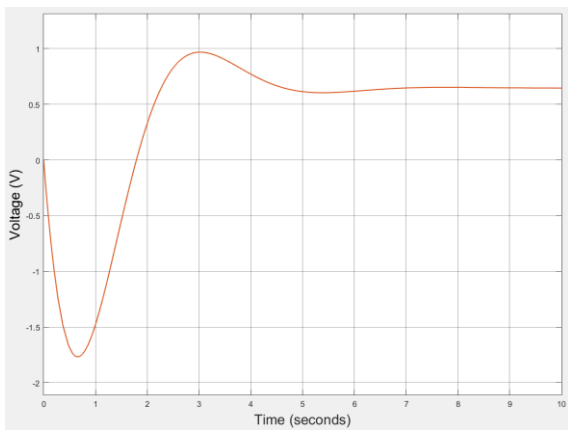
Besaran dari matriks J akan menentukan nilai K dan  $K_i$  yang akan dihasilkan. Semakin besar nilai matriks *gain* K, maka akan semakin cepat respons sistem, tetapi *overshoot* dan *steady-state error* semakin besar. Sebaliknya, semakin kecil nilai matriks *gain* K, semakin lambat respons sistem, tetapi *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	$\begin{bmatrix} 0,4 & 0,2 \\ 0,2 & 1,3 \end{bmatrix}$	$\begin{bmatrix} 0,004 & 0,002 \end{bmatrix}$



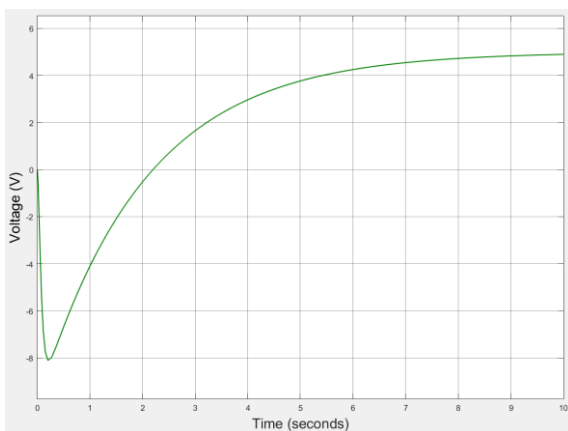
Gambar 14. Respons sistem dengan kendali LQR

Tabel 4. Penguatan K dan Ki

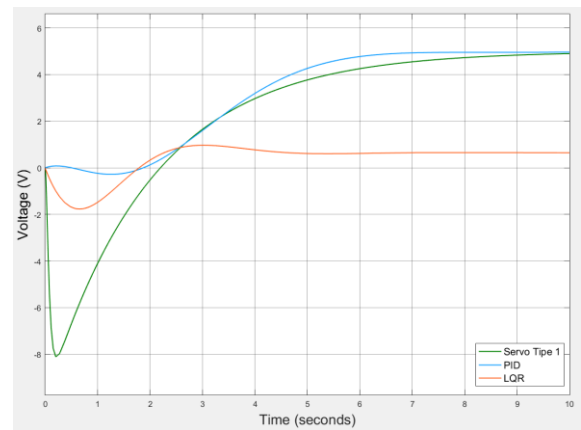
Ahat Bhat J	Khat	
	K=[Khat(1) Khat(2)]	Ki=-Khat(3)
$Ahat = \begin{bmatrix} -1,69 & -2,48 & 0 \\ 1 & 0 & 0 \\ 1,21 & -0,3 & 0 \end{bmatrix}$	[0,04 1,75]	[-0,93]
$Bhat = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$		
$J = [-0,5 - 20 - 30]$		

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. Muhardian 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.

# Linierisasi Model Sistem *Ball and Beam* dan Metode Kendali LQG Berbasis MATLAB-Simulink

Hanifah Suwardi<sup>1</sup>, Anisa' Septyaning Choir<sup>1</sup>, Salima Nurrahma<sup>1</sup>, Imroatul Hudati<sup>1\*</sup>

<sup>1</sup>Departemen Teknik Elektro dan Informatika, Universitas Gadjah Mada; hanifah.suwardi@mail.ugm.ac.id, salima.nurrahma@mail.ugm.ac.id, anisaseptyaningcho@mail.ugm.ac.id,

\*Korespondensi: imroatul.hudati@mail.ugm.ac.id

**Abstract** – *There are some unstable control systems, it can be an important problem in control, like systems in aerospace and flight. This can be studied simply in the ball and beam system. The ball and beam system uses an LQG (Linear Quadratic Gaussian) controller to determine the effect of stability and optimal system to reach the set point and the results of the slope angle on the beam at the set point. There is a combination of LQR with the element matrix parameters Q and R which can be searched using a trial and error technique with the condition that the Q matrix is a positive semidefinite while the R matrix is a positive definite matrix. As well as the use of gain Kalman filter as an estimator. Two experiments are given on the Q and R matrix elements which will be used in three distance/position displacement tests. The design of LQG on ball and beam systems is carried out in the MATLAB-Simulink software. Then the results of the angle of inclination of the beam (beam) obtained 14° to make the ball reach a set point of 10 cm, 17° for a set point of 20 cm, and 28° for a set point of 30 cm.*

**Keywords** – *ball and beam, angle, LQG, LQR, linearization*

**Intisari** – Sistem kontrol yang tidak stabil menjadi sebuah masalah yang penting dalam pengontrolan, seperti sistem dalam *aerospace* dan penerbangan. Hal ini dapat dipelajari secara sederhana pada sistem *ball and beam*. Pada sistem *ball and beam* digunakan pengontrol LQG (*linear quadratic gaussian*) untuk mengetahui pengaruh stabilitas maupun optimalnya sistem mencapai *set point* serta hasil kemiringan sudut pada balok (*beam*) pada titik *set point*. Terdapat kombinasi LQR dengan parameter matriks elemen Q dan R yang dapat dicari menggunakan teknik *trial and error* dengan syarat matriks Q ialah *semidefinite positive* sedangkan matriks R ialah matrik definit positif Serta penggunaan *gain filter* Kalman sebagai *estimator*. Dilakukan dua eksperimen pada elemen matriks Q dan R yang akan digunakan pada tiga pengujian perpindahan jarak/posisi. Rancangan LQG pada sistem *ball and beam* dilakukan di *software* MATLAB-Simulink. Diperoleh hasil sudut kemiringan *beam* sebesar 14° untuk membuat *ball* mencapai di posisi *set point* 10 cm, 17° untuk *set point* 20 cm, dan 28° untuk *set point* 30 cm.

**Kata kunci** – *ball and beam, sudut, LQG, LQR, linierisasi*

## I. PENDAHULUAN

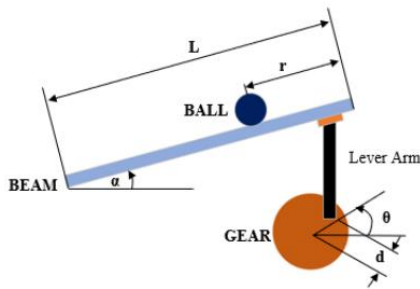
Terdapat beberapa jenis sistem kontrol, salah satunya adalah dalam aspek keseimbangan yang bertujuan untuk menjaga keseimbangan suatu objek terhadap objek lainnya. Sistem kontrol yang tidak stabil menjadi permasalahan dalam pengontrolan, baik sistem di *aerospace* dan penerbangan. Sistem *ball and beam* atau "*balancing a ball on beam*". bertujuan untuk mengontrol posisi bola untuk mencapai titik referensi atau *set point* dan menghilangkan gangguan dari eksternal maupun internal.

*Beam and ball system* merupakan alat yang umum digunakan untuk mempelajari sistem kendali dan memiliki berbagai metode kendali yang dapat diterapkan [1]. *Ball and beam system* merupakan sebuah sistem yang terdiri dari sebuah bola (*ball*) dan sebuah bidang segi empat atau balok (*beam*) dengan sensor diletakkan disalah satu sisi dari balok untuk mendeteksi posisi untuk dapat menyesuaikan di *set point* [2]. Sinyal pengontrolan dapat diturunkan dengan memberi *feedback* informasi dari posisi bola. Sinyal tegangan akan masuk ke motor *servo*, kemudian torsi yang dihasilkan oleh motor akan menggerakkan *beam* untuk berputar ke posisi atau sudut yang diinginkan [3]. Dengan demikian, bola dapat ditempatkan pada posisi yang diinginkan.

Pada sistem *ball and beam* digunakan pengontrol LQG (*linear quadratic gaussian*). Beberapa penelitian telah

menggunakan LQG di berbagai sistem seperti sistem suspensi, pendulum terbalik, dan sistem *ball and beam*. Penelitian sebelumnya yang membahas kontrol optimal pada sistem *ball and beam* menunjukkan bahwa metode kendali LQG memberikan respons yang baik karena dapat menjaga *measurement noise* dan dapat menstabilkan sistem dengan baik [4]. Terdapat pula yang menyatakan bahwa pengendali LQG mampu menjaga *robustness*, mengurangi *overshoot*, memperbaiki *error steady state*, dan mengembalikan nilai keluaran menuju *set point* setelah sistem mendapat gangguan. Hal ini terjadi karena pengendali tersebut memiliki filter Kalman yang berperan mengestimasi gangguan proses dan gangguan pengukuran [5].

Pada Gambar 1 merupakan diagram skematis dari sistem *ball and beam* yang mana sistem ini terdiri dari sebuah bidang segi empat (*beam*) dan sebuah bola (*ball*). Sistem *ball and beam* mengedepankan sistem yang stabil dan seimbang. Artikel ini membahas kontrol optimal LQG pada sistem *ball and beam* sebagai *tracking* keluaran sudut yang dihasilkan berdasarkan titik awal posisi *ball* pada *beam*. Pokok bahasan artikel ini mencakup model matematis sistem *ball and beam*, linierisasi, desain LQG pada sistem dengan menggunakan MATLAB-Simulink, dan hasil keluaran *step* serta analisis respons tiap percobaan posisi awal bola.



Gambar 1. Diagram skematis sistem ball and beam

II. METODOLOGI PENELITIAN

Perancangan kendali LQG memiliki beberapa tahapan yang meliputi:

- 1) Melakukan pemodelan matematis sistem
- 2) Linierisasi model matematis
- 3) Membangun *feedback regulator* (K)  
Mencari *gain regulator* (K), sebelumnya sudah ditentukan besarnya matriks pembobot dan R serta mencari nilai P yang merupakan solusi dari persamaan aljabar Riccati. Matriks Q dan R dicari dengan teknik *trial and error* sampai didapatkan kinerja *gain regulator* yang sesuai keinginan.
- 4) Merancang *estimator* atau filter Kalman  
Sesuai dengan tahapan perancangan telah diperoleh matriks A, B dan C pada persamaan *state space*. Selanjutnya, matriks Q dan R sebagai berikut:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; R = [10]$$

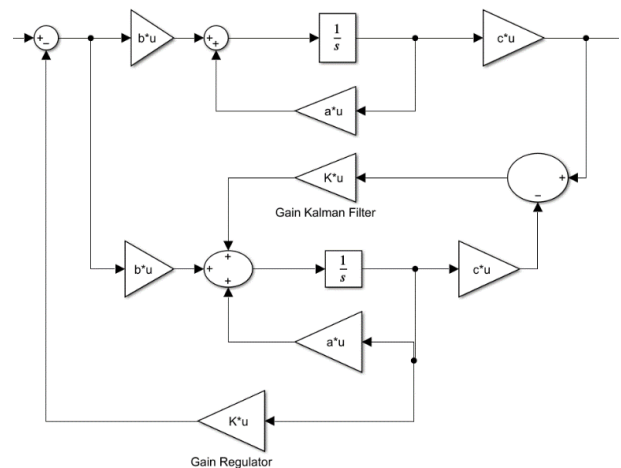
Diperoleh *gain regulator*:

$$K = [2,1605 \quad 0,8003 \quad 2,0386 \quad 0,4213]$$

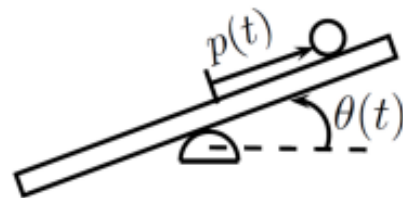
Serta telah diperoleh *gain filter* Kalman:

$$K_f = \begin{bmatrix} 10,9639 \\ 55,1032 \\ 30,0676 \\ 113,4881 \end{bmatrix}$$

Kemudian, rancangan blok diagram dilakukan untuk membuktikan hasil keluaran sistem berupa *step response*. Gambar 2 menunjukkan diagram blok LQG pada sistem *ball and beam*. Semua hasil matriks diperoleh menggunakan MATLAB. Untuk mencari gain regulator maupun gain Kalman dapat menggunakan *syntax*.



Gambar 2. Sistem LQG ball and beam



Gambar 3. Sistem ball and beam

III. MODEL MATEMATIS SISTEM BALL AND BEAM

A. Ball and Beam System Model

Terdapat beberapa metode untuk mencari persamaan dan model matematis sistem *ball and beam*. Teknik Lagrangian digunakan untuk mendapatkan persamaan dinamis pada sistem *ball and beam*. Metode Jacobian Linearization, digunakan untuk mendapatkan pendekatan yang linear terdapat persamaan dinamis di sekitar titik stabil. *State space* digunakan untuk memperoleh bentuk persamaan akhir pada sistem *ball and beam* seperti pada Gambar 3.

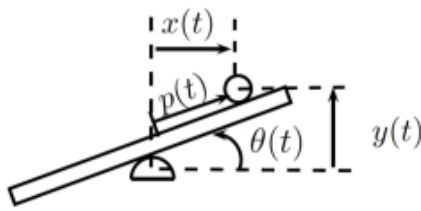
Bola menggelinding pada balok tanpa tergelincir dalam pengaruh gaya gravitasi. Balok berada pada posisi miring akibat torsi eksternal untuk mengontrol posisi bola pada balok. Koordinat umum yang menggambarkan sistem dinyatakan pada (1) dengan  $p(t)$  adalah posisi bola pada balok dan  $\theta(t)$  adalah sudut balok.

$$q(t) = \begin{bmatrix} p(t) \\ \theta(t) \end{bmatrix} \tag{1}$$

Lagrangian dari suatu sistem adalah kuantitas yang didefinisikan pada (2).

$$L = K - U \tag{2}$$

Dengan  $K$  adalah energi kinetik dan  $U$  adalah energi potensial dari sistem. Untuk mendapatkan nilai  $K$  dan  $U$ , perlu definisikan koordinat Cartesius  $x(t)$  dan  $y(t)$  seperti yang ditunjukkan pada Gambar 4.



Gambar 4. Koordinat Cartesius dan koordinat umum

Energi kinetik pada *beam* dihitung menggunakan (3).

$$K_1 = \frac{1}{2} J \dot{\theta}^2 \quad (3)$$

Dengan  $J$  merupakan momen inersia. Kemudian energi kinetik pada bola dihitung menggunakan (4).

$$K_2 = \frac{1}{2} J_b \dot{\theta}_b + \frac{1}{2} m v_b^2 \quad (4)$$

Dengan  $\dot{\theta}_b$  kecepatan sudut bola dan  $v_b$  adalah kecepatan linier bola. Nilai  $\dot{\theta}_b$  dapat diekspresikan menjadi bentuk koordinat umum sebagai (5).

$$\dot{\theta}_b = \frac{p}{r} \quad (5)$$

Dengan  $r$  merupakan jari-jari bola. Dapat diekspresikan  $v_b$  ke bentuk koordinat umum pada (6).

$$v_b^2 = \dot{x}^2 + \dot{y}^2 \quad (6)$$

$$x = p \cos \theta \quad (7)$$

$$\dot{x} = \dot{p} \cos \theta - p \dot{\theta} \sin \theta \quad (8)$$

$$\dot{x}^2 = \dot{p}^2 \cos^2 \theta + p^2 \dot{\theta}^2 \sin^2 \theta \quad (9)$$

$$y = p \sin \theta \quad (10)$$

$$\dot{y} = \dot{p} \sin \theta + p \dot{\theta} \cos \theta \quad (11)$$

$$\dot{y}^2 = \dot{p}^2 \sin^2 \theta + 2p\dot{p}\dot{\theta} \cos \theta \sin \theta + p^2 \dot{\theta}^2 \cos^2 \theta \quad (12)$$

Substitusi (9) dan (12) ke (6), diperoleh (13).

$$v_b^2 = \dot{p}^2 + p^2 \dot{\theta}^2 \quad (13)$$

Substitusi (5) dan (13) ke dalam (4), dapat diperoleh energi kinetik pada bola dalam bentuk koordinat umum (14).

$$K_2 = \frac{1}{2} \left( \frac{J_b}{r^2} + m \right) \dot{p}^2 + \frac{1}{2} m p^2 \dot{\theta}^2 \quad (14)$$

Energi potensial sistem, yaitu (15).

$$U = m g p \sin \theta \quad (15)$$

Substitusi (3) dan (14) ke (2), diperoleh bentuk Lagrangian untuk sistem ini yang ditunjukkan pada (16).

$$L = \frac{1}{2} \left( \frac{J_b}{r^2} + m \right) \dot{p}^2 + \frac{1}{2} (m p^2 + J) \dot{\theta}^2 - m g p \sin \theta \quad (16)$$

Persamaan pertama Lagrange,

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{p}} \right) - \frac{\partial L}{\partial p} = 0 \quad (17)$$

$$\frac{\partial L}{\partial p} = \left( \frac{J_b}{r^2} + m \right) \dot{p} \quad (18)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{p}} \right) = \left( \frac{J_b}{r^2} + m \right) \ddot{p} \quad (19)$$

$$\frac{\partial L}{\partial p} = m p \dot{\theta}^2 - m g \sin \theta \quad (20)$$

Substitusi (18), (19) dan (20) ke (17), diperoleh persamaan pertama untuk gerak pada sistem *ball and beam* (21).

$$\left( \frac{J_b}{r^2} + m \right) \ddot{p} + m g \sin \theta - m p \dot{\theta}^2 \quad (21)$$

Persamaan kedua *Lagrange* ditunjukkan pada (22).

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = \tau \quad (22)$$

Dengan  $\tau$  adalah torsi eksternal yang diaplikasikan ke *beam*. Diperoleh persamaan menggunakan pendekatan yang sama.

$$\frac{\partial L}{\partial \theta} = (m p^2 + J) \dot{\theta} \quad (23)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) = 2 m p \dot{p} \dot{\theta} + (m p^2 + J) \ddot{\theta} \quad (24)$$

$$\frac{\partial L}{\partial \theta} = -m g p \cos \theta \quad (25)$$

Substitusi (24) dan (25) ke (22), diperoleh persamaan kedua untuk gerak pada sistem *ball and beam*.

$$(m p^2 + J) \ddot{\theta} + 2 m p \dot{p} \dot{\theta} + m g p \cos \theta = \tau \quad (26)$$

Persamaan (21) dan (26) merupakan persamaan gerak untuk sistem *ball and beam*.

### B. Linear State Variable Representation

Persamaan gerak yang telah diperoleh untuk sistem *ball and beam* dapat ditulis dalam representasi *state variable*. Pertama mendefinisikan *state vector* seperti pada (27).

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix} = \begin{bmatrix} p(t) \\ \dot{p}(t) \\ \theta(t) \\ \dot{\theta}(t) \end{bmatrix} \quad (27)$$

*State vector* terdiri dari variabel yang diperlukan untuk menentukan respons sistem yang diberikan *input* dan kondisi saat ini [6]. Kemudian persamaan gerak dapat dituliskan dengan *variable state vector* pada (16).

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{m(x_1 x_4^2 - g \sin x_3)}{\frac{J_b}{r^2} + m} \\ x_4 \\ -\frac{2m x_1 x_2 x_4 - m g x_1 \cos x_3 + \tau}{(m x_1^2 + J)} \end{bmatrix} = f(x, \tau) \quad (28)$$

Titik operasi dapat didefinisikan dengan posisi bola  $p_0$  dan kecepatan, sudut serta kecepatan sudut *beam* bernilai nol.

$$x_0 = \begin{bmatrix} p_0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (29)$$

Dapat didefinisikan juga *input* operasi yang diperlukan untuk mempertahankan titik operasi ini, diperoleh torsi dari (28) ke nol kemudian mengevaluasi pada titik operasi ini.

$$u_\tau = mgp_0 \quad (30)$$

Persamaan torsi yang diperlukan untuk mempertahankan bola tetap pada posisi  $p_0$ . Persamaan Jacobian dari sisi kanan berhubungan dengan hasil *state vector*  $x$ .

$$\frac{\partial f}{\partial x}(x, \tau) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mx_4^2}{r^2+m} & 0 & \frac{-mg\cos x_3}{r^2+m} & \frac{2mx_1x_4}{r^2+m} \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_4}{\partial x_1} & -\frac{2mx_1x_4}{mx_1^2+J} & \frac{mgx_1\sin x_3}{mx_1^2+J} & -\frac{2mx_1x_2}{mx_1^2+J} \end{bmatrix} \quad (31)$$

Dengan,

$$\frac{f_4}{x_1} = \frac{(-2mx_2x_4 - mg\cos x_3)(mx_1^2 + J) - (-2mx_1x_2x_4 - mgx_1\cos x_3 + \tau)}{(mx_1^2 + J)^2} \quad (32)$$

Menghitung titik operasi pada representasi *state variable* matriks A, B dan C

$$A = \frac{\partial f}{\partial x}(x_0, \tau_0) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-mg}{r^2+m} & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{mg}{mp_0+J} & 0 & 0 & 0 \end{bmatrix} \quad (33)$$

$$B = \frac{\partial f}{\partial \tau}(x_0, \tau_0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (34)$$

$$C = [1 \ 0 \ 0 \ 0] \quad (35)$$

Dengan menggunakan matriks tersebut, maka dapat diperoleh bentuk representasi *state space* sebagai berikut [7].

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (36)$$

$$y(t) = Cx(t) \quad (37)$$

Substitusi (33) dan (34) dengan parameter sistem *ball and beam* pada Tabel 1. Sehingga dapat dihasilkan *state space* matriks sebagai berikut. Bentuk representasi *state space* ini, sistem dapat dianalisis dan pengendali dapat dirancang.

Tabel 1. Parameter sistem *ball and beam*

Parameter	Jumlah
(m) Massa bola	0,11 kg
(L) panjang balok	0,3 m
(r) Jari-jari bola	0,015 m
(g) Percepatan gravitasi	9,8 m/s <sup>2</sup>
(J) Momen Inersia balok	19 × 10 <sup>-3</sup> kg.m <sup>2</sup>
(J <sub>b</sub> ) Momen Inersia bola	9,99 × 10 <sup>-6</sup> kg.m <sup>2</sup>
(p <sub>0</sub> ) Posisi bola	m
(α) sudut servo	°
(θ) sudut balok	°

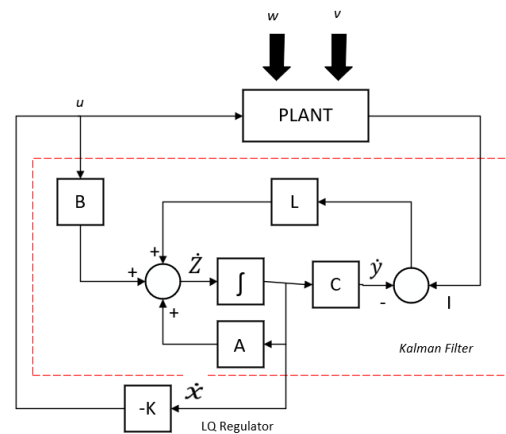
$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & -7 & 0 \\ 0 & 0 & 0 & 1 \\ -56,8 & 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 52,6 \end{bmatrix} u(t) \quad (38)$$

$$y(t) = [1 \ 0 \ 0 \ 0] \quad (39)$$

#### IV. DESAIN KENDALI LQG SISTEM BALL AND BEAM

Kendali LQG sebagai teknik kendali modern yang mengombinasikan kendali LQR dengan filter Kalman sebagai *estimator variable* keadaannya sehingga dapat diaplikasikan dalam bentuk *state-space* yang digunakan untuk mendesain dinamik *optimal regulator* [8].

Filter Kalman dapat mengeliminasi gangguan yang terdapat pada LQG, yaitu *gaussian white noise* [9]. *Gaussian white noise* merupakan sinyal acak yang memiliki intensitas yang sama dengan frekuensi yang berbeda yang berpengaruh terhadap respons sistem [10]. Gambar 5 menunjukkan diagram blok LQG.



Gambar 5. Diagram blok LQG



Dengan model *state space* LQG:

$$\dot{x}(t) = Ax(t) + Bu(t) + v(t) \quad (40)$$

$$y(t) = Cx(t) + w(t) \quad (41)$$

Dengan,

$u$	sinyal kendali <i>input</i>
$y$	<i>output</i> proses
$w$	Gangguan proses
$v$	<i>white noise</i>
$A$	matriks <i>plant</i>
$B$	matriks kendali <i>input</i>
$C$	matriks pengukuran.
$K$	<i>gain feedback regulator</i>
$K_f$	<i>gain Kalman</i>

Dengan penerapan *optimal feedback* yang dinyatakan dalam bentuk blok diagram dapat dinyatakan bahwa untuk mencari sinyal kontrol optimal  $u$  diperlukan *gain*  $K$  dan *gain*  $K_f$  yang optimal [11]. Agar sistem tetap stabil, diperlukan penguatan pengendali dan *estimator* yang stabil pula.

Nilai  $K$  dan  $K_f$  dapat diperoleh secara terpisah. Penentuan nilai  $K$  dilakukan dengan adanya sistem yang sudah ditentukan yaitu dengan metode LQR. Nilai penguat regulator (*gain*  $K$ ) ditunjukkan pada (42).

$$K = R_C^{-1} B^T S \quad (42)$$

Untuk mencari nilai  $S$  dapat menggunakan persamaan Riccati (43).

$$A^T S + SA - SBR_C^{-1} B^T S + Q_C = 0 \quad (43)$$

Dengan  $Q_C$  adalah matriks *positive-definite* (atau *positive-semidefinite*) Hermitian atau matriks simetris riil dan  $R_C$  adalah matriks *positive-definite* Hermitian atau matriks simetris riil [12]. Hubungan  $Q_C$  untuk menentukan matriks keadaan dan  $R$  untuk menentukan matriks kendali.

Sedangkan penentuan nilai  $K_f$  optimal dilakukan dengan sistem bersifat yang belum ditentukan kejadiannya (belum ditentukan) yaitu indeks performansi variansi kesalahan minimum atau dapat dituliskan [13]:

$$J(t_0) = \frac{1}{2} S(T) x^2(T) + \frac{1}{2} \int_{t_0}^T (Qx^2 + Ru^2) \quad (44)$$

Filter Kalman merupakan *estimator* optimal yang berfungsi mengestimasi keadaan dan menyaring *noise* [6]. Proses *estimator* dengan menggunakan bentuk *feedback control filter* mengestimasi keadaan pada suatu waktu dan kemudian memperoleh *feedback* dalam bentuk *noise* [14]. Dengan demikian, persamaan untuk filter Kalman terbagi menjadi dua kelompok, yaitu:

- *Time update eq.*  
Bertanggung jawab untuk memproyeksikan ke depan (dalam waktu).
- *Measurement update eq.*

Bertanggung jawab atas *feedback* yaitu dengan memasukkan pengukuran yang baru ke dalam estimasi apriori untuk mendapatkan estimasi posteriori yang lebih banyak [14].

*Estimator* bekerja dengan menekan nilai *error covariant* sekecil mungkin [13]. Maka, indeks performansinya atau *cost function*-nya dapat ditulis:

$$J = E\{[\hat{x} - x]^T [\hat{x} - x]\} \quad (45)$$

Dengan  $\hat{x}$  adalah nilai estimasi dari variabel  $x$  dalam fungsi waktu. Estimasi *state variable*  $\hat{x}$  diperoleh dari sistem dinamik filter Kalman sebagai berikut:

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f(y - \hat{y}) \quad (46)$$

Sedangkan, untuk menentukan penguatan filter Kalman pada *estimator* digunakan (47).

$$K_f = PC^T R_f^{-1} \quad (47)$$

Matriks  $P$  diperoleh dari persamaan Riccati (48) berikut.

$$0 = PA^T + AP - PC^T R_f^{-1} CP + GQ_f G^T \quad (48)$$

Dengan asumsi matriks  $A$  dan  $B$  terkendali dan  $C$  teramat  $Q_f \geq 0, R_f > 0$ , maka Kalman filter akan stabil. Matriks  $Q_f$  dan  $R_f$  adalah *noise covariance*, identifikasi *noise covariance* adalah tugas yang memungkinkan untuk implementasi *tuning* pada *state estimator* dan kineja pengendalian. Dengan *noise* proses  $w(t) \sim (0, Q_f)$  dan pengukuran  $n(t) \sim (0, V^2 R_f)$  adalah *white noise*.

## V. ANALISIS DAN PEMBAHASAN

Sistem *ball and beam* memiliki masukan posisi awal serta diberikan *set point* atau posisi akhir yang ditentukan. Keluaran sistem yang dihasilkan adalah sudut sesuai posisi *reference point*. Namun, perlu digarisbawahi bahwa penelitian ini melakukan percobaan secara eksperimen menggunakan MATLAB-Simulink tanpa purwarupa (*model-based*) untuk memastikan perhitungan dengan metode LQG mampu mengendalikan dengan stabil pada sistem *ball and beam*.

Diberikan beberapa percobaan berdasarkan *initial position point*. Bola diletakan secara acak sepanjang balok tersebut, dengan beberapa *set point* yang ditetapkan serta *trial dan error* dalam pencarian matriks  $Q$  dan  $R$ . Kemudian, *step response* pada masing-masing percobaan dianalisis.

### A. Pengujian 1: Set Point 10 cm dan Initial Point 0 cm

Pengujian sistem *ball and beam* dilakukan dengan memberikan *set point* tetap bernilai 10 cm. Hasil respons sistem ditunjukkan pada Gambar 6. Berdasarkan Gambar 6, sistem menghasilkan respons dengan *overshoot* sebesar 19,9%, *rise time* selama 0,48 s, *settling time* selama 3,7 s, *time delay* selama 0,62 s, dan *peak time* selama 1,22 s. *Steady state error* dihitung dengan (49).

$$Ess = Setpoint - X_{ss} = 10 - 9,148 = 0,852 \quad (49)$$

Steady state error merupakan perbedaan jarak antara keluaran sistem ( $X_{ss}$ ) dan posisi *set point* (posisi yang diinginkan).

Nilai keluaran sistem ( $X_{ss}$ ) dapat diidentifikasi melalui hasil respons sistem serta dapat diidentifikasi hasilnya setelah parameter *settling time* terukur. Pada pengujian diperoleh  $X_{ss}$  sebesar 9,148, dapat dilihat Gambar 6. Garis merah ditujukan posisi *set point* dan garis hitam ditujukan untuk hasil response sistem. Secara umum, sistem yang stabil akan memiliki *steady state error* sama dengan nol.

Perpindahan jarak yang diberikan dari *initial point* 0 cm hingga *set point* 10 cm menghasilkan sudut balok yang dipengaruhi oleh sudut motor *servo* pula. Diperoleh sudut sekitar  $14^\circ$  untuk mencapai *set point* 10 cm.

**B. Pengujian 2: Set Point 20 cm dan Initial Point 12 cm**

Pengujian sistem *ball and beam* dilakukan dengan memberikan *set point* tetap bernilai 20 cm. Hasil respon sistem ditunjukkan pada Gambar 7. Berdasarkan Gambar 7, sistem menghasilkan respons dengan *overshoot* sebesar 21,341%, *rise time* selama 0,486 s, *peak time* selama 1,19 s, *settling time* selama 4,2 s, *delay time* selama 0,6 s. *Steady state error* dihitung dengan (50) dengan hasil  $X_{ss}$  sebesar 18,34.

$$Ess = Setpoint - X_{ss} = 20 - 18,34 = 1,66 \quad (50)$$

Perpindahan jarak pada pengujian ini menghasilkan sudut maksimal sekitar  $17^\circ$  karena respons sinyal pada Gambar 7 menunjukkan sifat *underdamp case*. Akibatnya, untuk mencapai keadaan stabil, sudut akan terus berkurang hingga sesuai dengan *set point*.

**C. Pengujian 3: Set Point 30 cm dan Initial Point 27 cm**

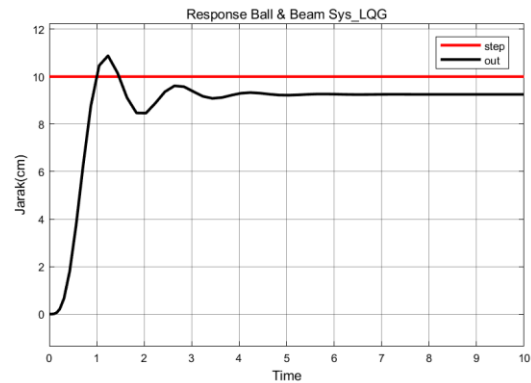
Pengujian sistem *ball and beam* dilakukan dengan memberikan *set point* tetap bernilai 30 cm. Hasil respons sistem ditunjukkan pada Gambar 8. Berdasarkan Gambar 8, sistem menghasilkan respons dengan *overshoot* sebesar 24,37%, dan *rise time* selama 0,496 s, *settling time* selama 6,7 s, *time delay* selama 0,68 s, *time peak* selama 1,2 s. *Steady state error* dihitung dengan (51).

$$EEss = Setpoint - X_{ss} = 30 - 27,43 = 2,58 \quad (51)$$

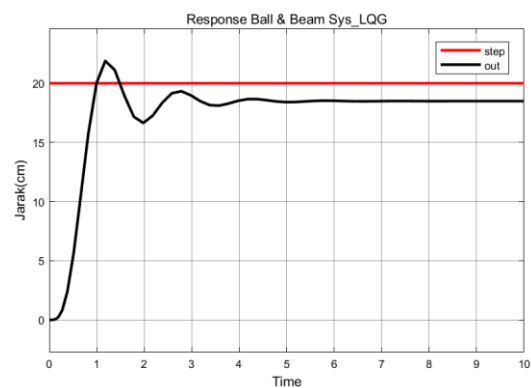
Diperoleh perpindahan jarak pada pengujian sebesar  $28^\circ$ . Dari tiga percobaan tersebut dapat diringkas kembali informasi *step response*, sebagai berikut.

Tabel 2. Hasil respons step dengan eksperimen matriks Q dan R pertama pada ketiga percobaan

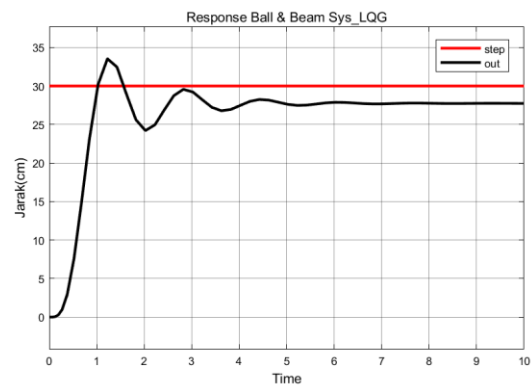
	Matriks		Karakteristik Respons Transient				
	Q	R	Mp (%)	RT (s)	ST (s)	Amp	Ess
1	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	[200]	19,9	0,5	3,7	9,15	0,85
2			21,3	0,5	4,2	18,34	1,66
3			24,3	0,51	6,7	27,43	2,58



Gambar 6. Respons sistem *ball and beam* dengan *set point* 10 cm dan *initial point* 0 cm



Gambar 7. Respons sistem *ball and beam* dengan *set point* 20 cm dan *initial point* 12 cm



Gambar 8. Respons sistem *ball and beam* dengan *set point* 30 cm dan *initial point* 27 cm

Tabel 3. Hasil respons step dengan eksperimen matriks Q dan R kedua pada ketiga percobaan

	Matriks		Karakteristik Respons Transient				
	Q	R	Mp (%)	RT (s)	ST (s)	Amp	Ess
1			13	0,6	3,1	9,21	0,79
2	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$	[500]	13	0,6	3	18,4	1,6
3			16	0,56	3,1	27,64	2,36

Berdasarkan Tabel 2, matriks Q dan R didapatkan secara eksperimen, serta diberikan tiga percobaan yang masing-masing memiliki *setpoint* dan *initial point* yang berbeda. *Range setpoint* yang digunakan adalah 10 cm s.d. 30 cm sedangkan *initial point* diberikan secara acak. Begitu pun dengan Tabel 3 [15].

Bila dibandingkan hasilnya yang lebih baik ada pada Tabel 3. Nilai *overshoot* mengalami penurunan dibandingkan Tabel 2. *Amplitude* masing-masing percobaan di Tabel 3 mengalami peningkatan hampir menyentuh *set point*. Matriks R bila ditambahkan nilainya akan menyebabkan *overshoot* menurun dan mengurangi osilasi. Kemudian, matriks Q bila diubah nilainya, dengan syarat *semidefinite-positive*, maka menghasilkan respons yang osilasinya berlebihan, tetapi dapat mendekati *steady state* [16].

## VI. SIMPULAN

Pada penelitian ini dapat disimpulkan bahwa terdapat desain kendali *linear quadratic* lainnya selain LQR yaitu LQG. Metode pengendali ini ialah kombinasi metode LQR serta filter Kalman sebagai *estimator* yang berfungsi untuk mengestimasi keadaan dan menyaring gangguan. Percobaan ini dilakukan untuk mengetahui apakah metode kendali ini mampu mengendalikan sistem *ball and beam* sehingga dapat mencapai *set point*. Hasilnya, semua percobaan stabil, tetapi tidak mampu mencapai *set point*. Namun, bila dilakukan eksperimen *trial and error* terhadap nilai Q dan R kembali, maka respons sistem dimungkinkan akan stabil dan optimal mencapai *set point*.

## PENELITIAN LANJUTAN

Saran pengembangan penelitian dalam sistem *ball and beam* adalah menggunakan *real plant* agar dapat mengetahui sistem apakah stabil maupun optimal bila menggunakan metode kendali LQG.

## DAFTAR PUSTAKA

- [1] N. O. Dewi and P. W. Rusimamto, "Perancangan Sistem Pengendalian Ball and Beam Menggunakan Perangkat Lunak LabVIEW Berbasis Fuzzy Logic Controller," *Jurnal Teknik Elektro*, vol. 8, no. 2, 2019.
- [2] D. Martinez and F. Ruiz, "Nonlinear model predictive control for a Ball&Beam," *2012 IEEE 4th Colombian Workshop on Circuits and Systems, CWCAS 2012 - Conference Proceedings*, no. 2, 2012, doi: 10.1109/CWCAS.2012.6404073.
- [3] T. Trapp and W. Selby, "Ball and Beam Balance Final Report," 2009.
- [4] R. Soni and Sathans, "Optimal control of a ball and beam system through LQR and LQG," *Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC 2018*, no. Icisc, pp. 179–184, 2018, doi: 10.1109/ICISC.2018.8399060.
- [5] J. Dofela, "DESAIN KENDALI OPTIMAL METODE LINEAR QUADRATIC GAUSSIAN (LQG) UNTUK PENGENDALIAN SISTEM SUSPENSİ KENDARAAN SEPEREMPAT." Universitas Islam Negeri Sultan Syarif Kasim Riau, 2019.
- [6] A. Srivastava and B. Pratap, "Nonlinear robust observers for ball and beam system: A comparative analysis," *2015 2nd International Conference on Recent Advances in Engineering and Computational Sciences, RA ECS 2015*, no. December, pp. 22–27, 2016, doi: 10.1109/RA ECS.2015.7453281.
- [7] "Ball & Beam : System Modelling," *Control Tutorials for MATLAB & SIMULINK*.
- [8] T. TAUFİKURRAHMAN, "DESAIN KONTROLER LINEAR QUADRATIC GAUSSIAN (LQG) PADA SISTEM PENDULUM TERBALIK." University of Muhammadiyah Malang, 2017.
- [9] F. L. Q. Regulator, "Cerita singkat tentang Linear Quadratic Regulator".
- [10] F. L. Lewis, *Applied Optimal Control & Estimation (Digital Design & Implementation)*. Prentice Hall, 1992.
- [11] K. Ryu and Y. Oh, "Balance control of ball-beam system using redundant manipulator," *2011 IEEE International Conference on Mechatronics, ICM 2011 - Proceedings*, pp. 403–408, 2011, doi: 10.1109/ICMECH.2011.5971319.
- [12] T. TAUFİKURRAHMAN, "DESAIN KONTROLER LINEAR QUADRATIC GAUSSIAN (LQG) PADA SISTEM PENDULUM TERBALIK." University of Muhammadiyah Malang, 2017.
- [13] J. Dofela, "DESAIN KENDALI OPTIMAL METODE LINEAR QUADRATIC GAUSSIAN (LQG) UNTUK PENGENDALIAN SISTEM SUSPENSİ KENDARAAN SEPEREMPAT." Universitas Islam Negeri Sultan Syarif Kasim Riau, 2019.
- [14] G. Welch and G. Bishop, "An introduction to the Kalman filter," 1995.
- [15] D. Kinoshita and K. Yoshida, "Stabilizing Control for a ball and beam system considering the restricted beam angle and ball speed," *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2019*, pp. 170–176, 2019, doi: 10.23919/SICE.2019.8859847.
- [16] K. T. Lee, S. Y. Oh, and H. L. Choi, "Bounded control of a ball and beam system in the absence of feedback," *International Conference on Control, Automation and Systems*, vol. 2017-October, no. Iccas, pp. 1167–1169, 2017, doi: 10.23919/IC CAS.2017.8204389.

# Implementasi Kontrol PID pada Kopel Motor DC dengan Menggunakan Filter Kalman

Salima Nurrahma<sup>1,\*</sup>, Tegar Arif Berbudi<sup>1</sup>, Muhammad Rakha Firdaus<sup>1</sup>, Galang Izzaulhaq<sup>1</sup>, Imroatul Hudati<sup>1</sup>

<sup>1</sup>Departemen Teknik Elektro dan Informatika, Universitas Gadjah Mada; tegar.a.b@mail.ugm.ac.id, rakhafdr30@mail.ugm.ac.id, galangizzaulhaq@mail.ugm.ac.id, imroatul.hudati@ugm.ac.id

\*Korespondensi: salima.nurrahma@mail.ugm.ac.id;

**Abstract** – This paper describes a DC motor coupling system that uses several control methods including linear quadratic regulator (LQR), linear quadratic tracking (LQT), PID and uses Kalman filter. The methods used aim to obtain stable results from the DC motor coupling system used. To get the appropriate results, several stages were carried out such as hardware design where this process was carried out by pairing all the components used including the power supply, DC motor driver, Arduino UNO, potentiometer, and the two coupled DC motors. This system uses an input voltage of 12 V and will be processed by the motor driver to drive the DC motor. Based on the results that have been obtained, it can be observed that when the system uses the Kalman filter and uses the PI control the system response results have a small steady state error (ess). This is obtained because when the Kalman filter is added, it can reduce noise in the DC motor voltage readings and produces smoother output, and is less affected by high noise. After conducting experiments with the Kalman filter, the PID control was gradually added to its usage. It was found that when the P control was used, it provided the best response that reached the set point value with an RMSE of 0.1071 and an ess of +0.0874.

**Keywords** – DC motor coupling, PID, Kalman filter, LQR, LQT

**Intisari** – Artikel ini menjelaskan mengenai sistem kopel motor DC yang menggunakan beberapa metode kendali meliputi *linear quadratic regulator* (LQR), *linear quadratic tracking* (LQT), PID serta menggunakan *Kalman filter*. Metode - metode yang digunakan bertujuan untuk mendapatkan hasil yang stabil dari sistem kopel motor DC yang digunakan. Untuk mendapatkan hasil yang sesuai maka dilakukan beberapa tahapan seperti perancangan perangkat keras di mana proses ini dilakukan dengan memasang seluruh komponen yang digunakan meliputi *power supply*, *driver motor DC*, *Arduino UNO*, *potensio* dan kedua motor DC yang telah dikopel. Sistem ini menggunakan *input* tegangan sebesar 12 V dan kemudian diolah oleh *driver motor* untuk menggerakkan motor DC. Berdasarkan hasil yang telah didapat, dapat diamati bahwa saat sistem menggunakan Kalman filter dan menggunakan kendali PI hasil respons sistem memiliki *error steady state* (ess) yang kecil. Hal ini diperoleh karena pada saat ditambahkan, hal tersebut dapat mengurangi *noise* dalam pembacaan nilai tegangan motor DC dan menghasilkan keluaran yang lebih halus dan bebas dari *noise* tinggi. Setelah dilakukan percobaan dengan filter Kalman, selanjutnya ditambahkan kendali PID secara bertahap dalam penggunaannya. Didapatkan pada saat kendali P digunakan memberikan respons paling baik yang mencapai nilai *set point* dengan RMSE sebesar 0,1071 dengan ess sebesar +0,0874.

**Kata kunci** – kopel motor DC, PID, filter Kalman, LQR, LQT

## I. PENDAHULUAN

Kopel motor DC terdiri dari dua motor DC yang digabungkan menjadi satu di mana satu motor DC berfungsi sebagai penggerak dan yang lain sebagai sensor untuk pembacaan kecepatan dan tegangan *output* yang dihasilkan oleh sistem. Pada sistem kontrol tegangan keluaran motor DC membutuhkan perangkat keras yang terdiri dari catu daya DC, laptop, *DC motor driver*, *potensio* dan kopel untuk menggabungkan kedua motor.

Sistem kopel motor DC memiliki tegangan 12 V dengan *input* yang diatur oleh *potensio* dan menggunakan kendali untuk mendapatkan hasil *output* yang sesuai. Pembacaan hasil *output* yang telah dikontrol oleh beberapa metode akan dibaca oleh Arduino sehingga dapat dibandingkan metode mana yang sesuai dan benar.

Motor DC memiliki kekurangan dalam hal pengaturan keluaran kecepatan atau tegangan dari motor DC. Maka dari itu untuk menekan tingginya *overshoot* perlu dilakukan pengaturan sehingga kecepatan dan tegangan keluaran motor DC sesuai dengan nilai *set point* yang diinginkan dengan *rise*

*time* yang tinggi dan *overshoot* yang kecil. Pengaturan yang diperlukan berupa kendali *linear quadratic regulator* (LQR), *linear quadratic tracking* (LQT), *proportional-integral-derivative* (PID) dan filter Kalman. Motor DC bekerja secara non linier terutama pada saat diberi variasi beban. Terdapat beberapa penelitian pengendalian motor DC yang telah dilakukan, seperti pada penelitian [1]. Pengendalian kecepatan motor DC dilakukan dengan menggunakan metode kendali PID. Pengendali konvensional seperti PID umum digunakan karena memiliki performa yang baik dalam mengendalikan sistem linier. Kelemahan pada metode tersebut adalah sulit disesuaikan dengan perubahan kondisi sistem yang cepat atau tugas kendali yang kompleks. Penelitian hanya dilakukan dengan menggunakan simulasi sehingga tidak memperhatikan kondisi sesungguhnya dari kecepatan motor DC. Pada penelitian [2], dilakukan pengendalian kecepatan motor DC dengan menggunakan kendali adaptif. Pada penelitian ini, hanya dilakukan pada simulasi dengan menganalisis model motor DC. Kelemahan pada metode ini adalah pengujian dilakukan dengan menggunakan analisis motor DC sehingga membuat model referensi yang diberikan tidak sesuai dengan kondisi

sesungguhnya dari motor DC. Penelitian [3] tentang kendali motor DC, dengan menggunakan metode *fuzzy logic*, di mana data yang digunakan diambil dari PCI 6221. Kelemahan dari metode ini adalah penentuan *membership function* yang cukup rumit. Peneliti harus cukup berpengalaman agar dapat menentukan *membership function* yang sesuai dengan sistem yang akan dikendalikan.

Pada penelitian ini, dilakukan kopel motor DC menggunakan kendali PID untuk mengatur mengendalikan kecepatan motor DC dengan masukan tegangan yang dikendalikan dengan potensiometer. Selain itu, penelitian kali ini juga menggunakan filter Kalman untuk mendapatkan hasil respons yang sesuai.

## II. DASAR TEORI

### A. Discrete PID Controller

Kontrol PID merupakan kontrol dengan mekanisme umpan balik yang sering digunakan di dunia industri. PID berfungsi untuk mengatur putaran motor yaitu kontrol proporsional yang memiliki kelebihan *rise time* yang cepat, kontrol integral untuk memperkecil nilai *error* dan kontrol derivatif untuk meredam *overshoot* [4].

Kendali PID digunakan dalam sistem *loop* tertutup yang memiliki umpan balik dari *output* sistem untuk mencapai respon yang diinginkan. Kontrol PID terdiri dari proporsional, integral dan derivatif dan saling memiliki pengaruh terhadap yang lain [4].

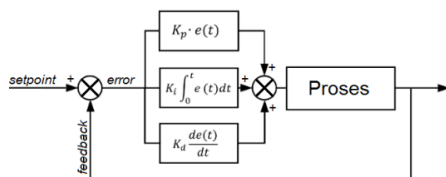
Pada Gambar 1 ditunjukkan bahwa jika ketiga parameter tersebut dikombinasikan secara optimal akan menghasilkan respons keluaran yang baik. PID memiliki masukan berupa sinyal kesalahan  $e(t)$  dan keluaran pengendali merupakan sinyal kontrol  $u(t)$ . Hubungan antara masukan dan keluaran pengendali PID pada (1).

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

Hasil Laplace pada (1) menjadi (2).

$$U(s) = K_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right) E(s) \quad (2)$$

Pada (2),  $K_p$  merupakan penguatan proporsional dan  $\tau_i$  merupakan waktu integral dan  $\tau_d$  adalah waktu diferensial. Ketiga parameter tersebut dapat ditentukan nilainya. Sehingga fungsi alih pengendali PID ditunjukkan oleh (3).



Gambar 1. Blok diagram pengendali PID

Tabel 1. Pengaruh tiap parameter PID

Respons CL	Rise Time	Overshoot	Settling Time	Error Steady State
Kp	Naik	Meningkat	Perubahan Kecil	Menurun
Ki	Menurun	Meningkat	Meningkat	Hilang
Kd	Perubahan Kecil	Menurun	Menurun	Perubahan Kecil

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (3)$$

Tipe pengendali proporsional  $K_p$  akan memberikan pengaruh pengurangan nilai *rise time* akan tetapi tidak menghapus nilai *error steady state* [5]. Kontrol integral akan memberikan pengaruh menghilangkan *error steady state* akan tetapi berakibat menurunkan respons *transient*. Sedangkan pengendali derivatif Kd akan memberikan pengaruh meningkatkan stabilitas sistem dan mengurangi nilai *overshoot*. Efek dari setiap pengendali dalam sistem *closed loop* hal ini dapat dilihat pada Tabel 1.

Pada pengendali PID memiliki tiga parameter yang mempengaruhi hasil dari kontrol yang dirancang. Berdasarkan Tabel 1 didapatkan hasil pengaruh beberapa parameter seperti parameter Kp ditambahkan nilainya atau diperbesar maka akan berpengaruh meningkatkan nilai *rise time* dan *overshoot* pada hasil kontrol. Sebaliknya pada nilai *settling time* dan nilai *error steady state* akan menurun. Jika menambahkan nilai parameter Ki maka nilai *overshoot* dan *settling time* akan meningkat, sedangkan nilai *rise time* akan menurun dan nilai *error steady state* akan hilang. Terakhir jika menambahkan nilai Kd jika ditambahkan akan memberikan pengaruh pada nilai *overshoot* dan *settling time* akan menurun, sedangkan nilai *rise time* dan *error steady state* akan berubah sedikit.

### B. Filter Kalman

Filter Kalman merupakan metode yang digunakan untuk melakukan estimasi suatu nilai [6]. Kalman Filter ini digunakan untuk meminimalisasi *error* estimasi *mean squared* pada linier stokastik menggunakan *noisy linear sensors*. Metode ini sering disebut dengan *linear quadratic estimator (LQE)* karena meminimalisasi fungsi kuadrat dari eror estimasi pada sebuah sistem dinamis linear dengan *white measurement* dan *disturbance noise*.

Filter Kalman lebih dipergunakan bagi permasalahan umum estimasi pada suatu keadaan dari proses kontrol diskrit yang dibangun dari penurunan persamaan stokastik linier [7]. Filter Kalman membahas masalah umum untuk memperkirakan keadaan proses waktu diskrit terkontrol yang diatur oleh persamaan linier stokastik (4).

$$x_k = Cx_{k-1} + Du_k + w_k \quad (4)$$

Pada waktu ke- $k$  dengan  $x_{k-1}$  menyatakan vector keadaan dan  $u_k$  sebagai vektor dengan model pengukuran  $z \in R$  dengan,

$$z_k = H x_k + v_k \tag{5}$$

Variabel acak  $w_k$  dan  $v_k$  menyatakan proses dan pengukuran *noise*. Masing-masing diasumsikan independen dan memiliki probabilitas distribusi normal. Matriks *covariant noise* dapat berubah pada setiap kali pengukuran, namun dalam hal ini *noise* diasumsikan konstan. C, D dan H masing-masing merupakan matriks yang elemennya adalah koefisien dari komponen masing-masing vektor.

Filter Kalman mengestimasi suatu proses dengan menggunakan suatu bentuk kontrol *feedback* berulang. Persamaan pada filter Kalman dibagi dalam dua tahap persamaan pembaruan waktu (*time update*) dan persamaan pembaruan pengukuran (*measurement update*) [8].

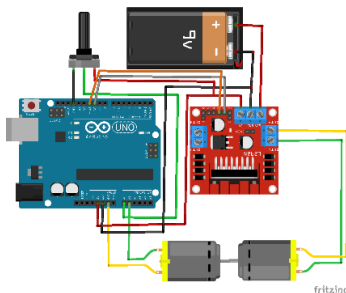
Persamaan pembaharuan waktu yang ada akan digunakan untuk memproyeksikan keadaan saat ini dan estimasi *covariant error*-nya untuk mendapatkan estimasi *priori* untuk langkah selanjutnya.

### III. METODOLOGI

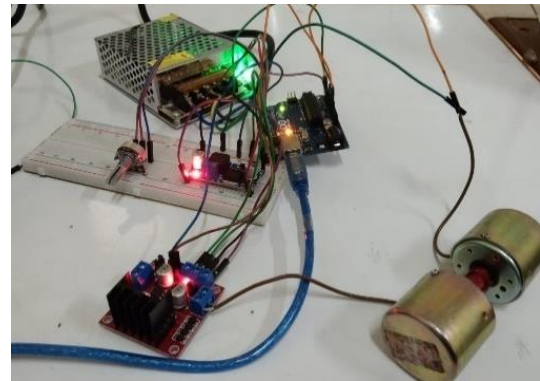
#### A. Perancangan Perangkat Keras

Perancangan perangkat keras dilakukan dengan membuat *wiring diagram* kopel motor DC menggunakan bantuan dari *software* Fritzing. Gambar 2 menunjukkan *wiring diagram* dari sistem kopel motor yang menggunakan Arduino UNO sebagai pengontrol sistem. Gambar 3 menunjukkan rancangan perangkat keras dari kopel motor DC.

Arduino dan *driver motor* akan mendapatkan suplai tegangan dari sumber tegangan sebesar 12 V. *Driver motor* digunakan untuk menggerakkan motor DC pertama, di mana besar PWM untuk *driver* motor tersebut akan diatur menggunakan potensiometer. Pada saat motor DC pertama berputar, maka *shaft* motor DC kedua akan ikut berputar dikarenakan kedua motor sudah dalam keadaan terkopel sehingga pada saat motor DC kedua bergerak, akan dihasilkan tegangan pada kedua terminalnya dan tegangan tersebut akan dibaca oleh Arduino. Data tegangan motor DC kedua tersebut akan berfungsi sebagai data *output* yang akan digunakan untuk keperluan identifikasi sistem.



Gambar 2. *Wiring diagram* motor DC



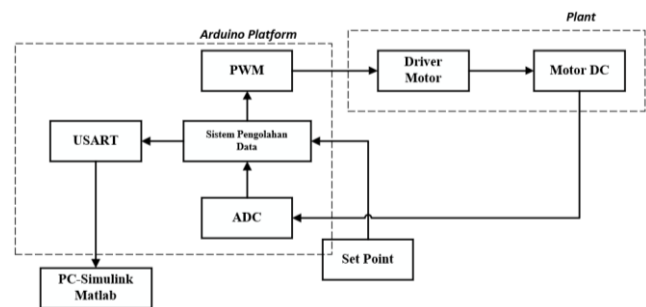
Gambar 3. Perangkat keras kopel motor DC

#### B. Perancangan Sistem Akuisisi Data

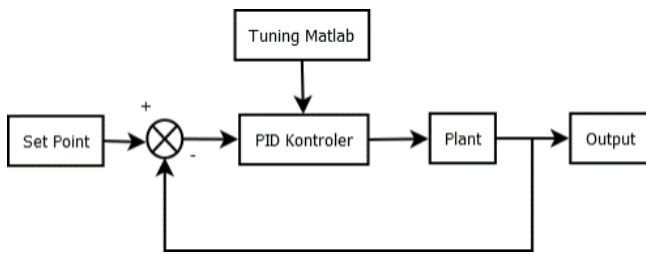
Dalam penelitian ini dibutuhkan suatu sistem akuisisi data yang diperlukan untuk melakukan pengambilan data dari sistem menuju ke komputer. Hal ini diperlukan karena untuk melakukan identifikasi sistem diperlukan data masukan dan keluaran dari sistem tersebut, sehingga diperlukan rancangan sistem akuisisi data yang mana diagram blok rancangan sistem akuisisi data dapat dilihat pada Gambar 4.

Dari rancangan blok diagram di atas dapat diketahui bahwa akuisisi data pada penelitian ini menggunakan Arduino sebagai sistem untuk pengontrol dan pengolah data pada sistem. Arduino akan membaca tegangan pada potensiometer sebagai nilai *set point*. Nilai *set point* tersebut digunakan untuk mengatur besarnya nilai PWM yang akan diberikan ke *driver* motor untuk mengatur kecepatan putaran motor DC pertama.

Dengan konfigurasi kopel antara kedua motor, Arduino akan membaca tegangan yang dihasilkan oleh motor DC kedua secara serial. Untuk menyimpan data serial tersebut, digunakan *software* MATLAB yang telah terhubung secara serial dengan Arduino sehingga hasil data yang telah didapatkan oleh Arduino akan dicatat oleh MATLAB sehingga MATLAB akan menyimpan data, baik data masukan potensiometer atau data keluaran tegangan motor kedua sehingga data-data tersebut dapat diolah selanjutnya untuk keperluan identifikasi sistem.



Gambar 4. Diagram blok rancangan sistem akuisisi data [9]



Gambar 5. Diagram blok kontrol PID

### C. Perancangan Pengendali PID

Sebelum menentukan nilai parameter pengontrol PID, sebuah sistem terlebih dahulu harus ditinjau bagaimana karakteristik dari sistem tersebut [10]. Nilai-nilai parameter dari pengendali PID akan dicari secara *trial and error*. Perubahan pada setiap nilai parameter akan memengaruhi parameter yang lainnya sehingga pada penelitian ini digunakan bantuan dari MATLAB untuk melakukan *tuning* nilai [11] PID secara otomatis. Gambar 5 menunjukkan rancangan blok diagram kontrol PID.

Dari blok diagram di atas dapat diketahui bahwa nilai parameter PID akan diatur menggunakan fitur pengaturan PID pada MATLAB. Blok *Plant* merupakan hasil identifikasi sistem yang telah dimodelkan dalam bentuk transfer fungsi. Data *output* akan menjadi nilai *feedback* dan akan dibandingkan dengan nilai *set point*. Pengendali PID akan menghitung nilai *error* dari beda antara nilai *set point* yang telah ditetapkan dengan nilai *output* pada sistem.

## IV. HASIL DAN PEMBAHASAN

### A. Filter Kalman

Sebelum diberikan filter Kalman, tegangan keluaran yang diukur pada motor DC kedua memiliki banyak *noise*. Adanya *noise* ini dapat mengganggu proses umpan balik dalam perhitungan sinyal kontrol. Gambar 6 menunjukkan grafik tegangan keluaran dari motor DC kedua selama 1000 kali *sampling*.

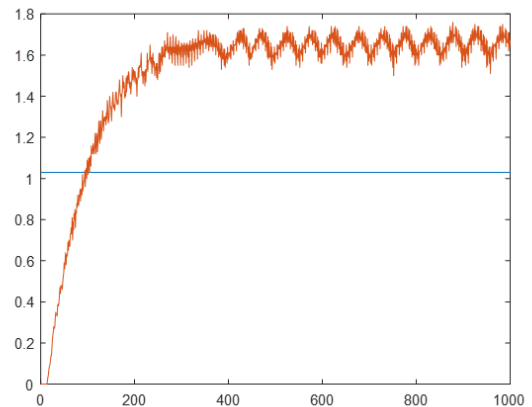
Filter Kalman dapat mengurangi *noise* pada pembacaan nilai tegangan motor DC [8]. Filter Kalman mengurangi *noise* dengan melakukan estimasi nilai (*state*) selanjutnya berdasarkan nilai-nilai sebelumnya. Filter Kalman memprediksi *state* dari nilai sebelumnya kemudian memperbarui *state* berdasarkan prediksi serta nilai saat ini. Setelah tegangan keluaran diberi Filter Kalman, hasil keluarannya lebih halus dan bebas dari *noise* tinggi. Grafik tegangan keluaran dari motor DC kedua setelah diberi Filter Kalman ditunjukkan pada Gambar 7.

### B. Pengendali PID

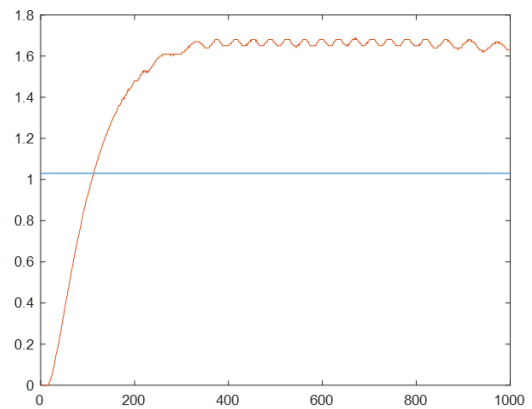
Sebelum diberikan kendali, tegangan keluaran yang diukur dari motor DC kedua memiliki *rise time* yang cukup lama dan memiliki *error steady state* yang sangat besar seperti pada Gambar 6 dan Gambar 7. Untuk mempercepat *rise time* dan juga mengurangi nilai *error steady state* tersebut,

dibutuhkan suatu kendali. Kendali yang diimplementasikan pada sistem ini adalah kendali *proportional* (P), kendali *proportional-integral* (PI), dan kendali PID.

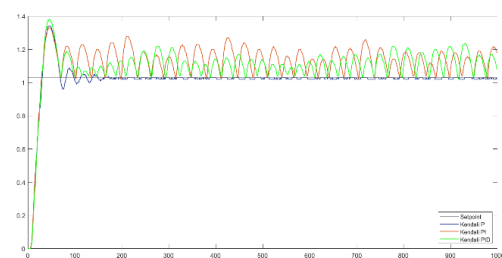
1) *Kendali P*: Nilai konstanta proporsional ( $K_p$ ) yang digunakan pada kendali ini adalah 5. Kendali P memberikan respon keluaran yang mencapai set point dengan cepat. Nilai *rise time*-nya adalah 27 waktu *sampling*. Keluaran mengalami *overshoot* sebesar 30,09% dan *settling time*-nya sekitar 90 waktu *sampling*. Namun, masih terdapat *error steady state* sebesar -0,0042 V dan RMSE sebesar 0,0087.



Gambar 6. Grafik tegangan keluaran sebelum adanya filter Kalman



Gambar 7. Grafik tegangan keluaran setelah adanya filter Kalman



Gambar 8. Grafik respons kendali P, PI, dan PID dengan filter Kalman

Tabel 2. Perbandingan nilai error kendali P, PI, dan PID

Kendali	$e_{ss}$ (V)	RMSE
P	-0,0042	0,0087
PI	+0,1121	0,1299
PID	+0,0874	0,1071

2) *Kendali PI*: Kendali PI ini menggunakan nilai konstanta proporsional ( $K_p$ ) sebesar 5 dan konstanta integral ( $K_i$ ) sebesar 1. Kendali PI pada motor DC tidak bisa menghasilkan keluaran yang stabil sesuai dengan set point namun memiliki nilai *rise time* yang cepat yakni 25 waktu sampling. Terdapat *overshoot* sebesar 29,13%, di mana keluarannya mengalami *ripple* secara terus-menerus dengan nilai rata-rata yang relatif besar. Nilai rata-rata *error steady state*-nya paling besar dari kendali lain yaitu +0,1121 V dengan nilai RMSE sebesar 0,1299.

3) *Kendali PI*: Nilai konstanta proporsional ( $K_p$ ), integral ( $K_i$ ), dan derivatif ( $K_d$ ) yang digunakan berturut-turut adalah 5, 15, dan 10. Kendali PID pada motor DC juga tidak dapat memberikan respon keluaran yang stabil, tetapi *rise time*-nya tetap cepat sebesar 26 waktu sampling. Keluaran mengalami *overshoot* yang paling tinggi dari kendali sebelumnya yakni sebesar 33,98%. Keluarannya tetap mengalami *ripple* tinggi di atas nilai *set point*. *Ripple* ini mengakibatkan besarnya nilai rata-rata eror yakni +0,0874 V dengan nilai RMSE sebesar 0,1071.

Gambar 8 menunjukkan grafik tegangan keluaran motor DC dengan kendali P, PI, dan PID yang telah disaring dengan filter Kalman. Dari grafik tersebut, terlihat bahwa kendali P memiliki respons yang paling baik karena dapat mencapai *set point* tanpa adanya *ripple* yang signifikan.

Tabel 2 menunjukkan perbandingan respons *transient* dari kendali P, PI, dan PID dengan Filter Kalman. Dari nilai *error steady state* dan RMSE masing-masing kendali, diperoleh bahwa kendali P merupakan metode kendali terbaik untuk sistem ini yang mampu menghasilkan respons yang mencapai nilai *set point*. Berbeda dengan kendali P, kendali PI dan PID tidak berhasil membuat respons sistem sesuai dengan nilai *set point*. Hal ini dapat dilihat dari RMSE kendali PI dan PID yang nilainya sekitar 0,1 lebih besar dari RMSE kendali P.

Kendali P merupakan satu-satunya kendali yang mampu menghasilkan respons yang mencapai nilai *set point* tanpa adanya osilasi terus-menerus. Akibat ini juga, kendali P memiliki nilai *error steady state* yang kecil jika dibandingkan dengan kendali PI maupun PID, yakni 3% lebih kecil. Dengan menggunakan kendali proporsional, respons sistem dapat dengan stabil mencapai *set point* yang artinya kecepatan motor kopel dapat dikontrol melalui masukan potensiometer.

### V. SIMPULAN

Pada percobaan sistem kopel motor DC yang telah dilakukan menggunakan beberapa metode kendali meliputi LQR, LQT, PID serta menggunakan filter Kalman. Metode-

metode yang digunakan ini bertujuan untuk mendapatkan hasil yang stabil dari sistem kopel motor DC yang digunakan. Berikut merupakan paparan hasil percobaan yang didapatkan:

1. Proses penggunaan filter Kalman berpengaruh dalam mengurangi *noise* pada pembacaan nilai tegangan motor DC. Filter Kalman memprediksi *state* dari nilai sebelumnya kemudian memperbarui *state* berdasarkan prediksi serta nilai *state* saat ini. Setelah tegangan keluaran diberi filter Kalman, hasil keluarannya lebih stabil dan *noise* yang diakibatkan oleh *noise* proses ataupun *noise* sensor berkurang.
2. Memasuki perancangan pengendali PID, nilai-nilai parameter yang digunakan dari pengendali PID ini kami dapatkan secara *trial and error*. Perubahan pada setiap nilai parameter akan memengaruhi parameter yang lainnya sehingga pada penelitian ini digunakan bantuan dari MATLAB untuk melakukan penyetelan nilai PID secara otomatis.
3. Sebelum menerapkan kendali LQR, terlebih dahulu dilakukan pengamatan terhadap respons dari *open loop* sistem dalam bentuk fungsi alih yang didapatkan dari hasil identifikasi sistem. Hal ini dilakukan untuk mengetahui respons sistem sebelum diberikan kendali.
4. Selanjutnya dilakukan pengendalian respons sistem menggunakan kendali LQR dan LQT. Didapatkan hasil dari kedua kendali tersebut yang tidak jauh berbeda di mana keluaran yang dihasilkan tetap memiliki *delay* dan memiliki batas atas maksimum 2,1 V.
5. Berdasarkan hasil yang telah didapat dan respons sistem setelah diberikan kendali, dapat diamati bahwa saat sistem menggunakan filter Kalman dan menggunakan kendali P, hasil respons sistem memiliki *error steady state* paling kecil, sebesar -0,0042 V, 3% lebih kecil dari *error steady state* kendali PI dan PID.

### DAFTAR PUSTAKA

- [1] N. Mishra, S. Bajpai, and E. Narayan, "Speed Control of DC Motor using PID Controller FED H-Bridge," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 10, pp. 274–282, 2020, doi: 10.35940/ijitee.f3967.0891020.
- [2] A. Khanna and T. Gaur, "Model Predictive Control Of Dc Motor Model In Matlab," *Int. J. Sci. Eng. Res.*, vol. 8, no. 4, pp. 82–85, 2017, [Online]. Available: <http://www.ijser.org>.
- [3] A. K. Rajagiri, S. R. Mn, S. S. Nawaz, and T. Suresh Kumar, "Speed control of DC motor using fuzzy logic controller by PCI 6221 with MATLAB," *E3S Web Conf.*, vol. 87, no. 201 9, pp. 1–6, 2019, doi: 10.1051/e3sconf/20198701004.
- [4] A. Ma'Arif, Iswanto, N. M. Raharja, P. A. Rosyady, A. R. C. Baswara, and A. A. Nuryono, "Control of DC Motor Using Proportional Integral Derivative (PID): Arduino Hardware Implementation," *Proceeding - 2020 2nd Int. Conf. Ind. Electr. Electron. ICIEE 2020*, pp. 74–78, 2020, doi: 10.1109/ICIEE49813.2020.9277258.
- [5] V. Yadav and V. K. Tayal, "Tuner," pp. 442–445, 2018.
- [6] A. N. A. Syarifuddin, D. A. Merdekawati, and E. Apriliani, "Perbandingan Metode Kalman Filter, Extended Kalman Filter, dan Ensambl Kalman Filter pada Model penyebaran virus HIV/AIDS," *Limits J. Math. Its Appl.*, vol. 15, no. 1, p. 17, 2018, doi: 10.12962/limits.v15i1.3344.
- [7] P. S. Madhukar and L. B. Prasad, "State Estimation using Extended Kalman Filter and Unscented Kalman Filter," *Proc. - 2020 Int.*



- [8] Y. Xu, K. Xu, J. Wan, Z. Xiong, and Y. Li, "Research on Particle Filter Tracking Method Based on Kalman Filter," *Proc. 2018 2nd IEEE Adv. Inf. Manag. Commun. Electron. Autom. Control Conf. IMCEC 2018*, no. Imcec, pp. 1564–1568, 2018, doi: 10.1109/IMCEC.2018.8469578.
- [9] 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.
- [10] 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.
- [11] S. Balamurugan and A. Umarani, "Study of Discrete PID Controller for DC Motor Speed Control Using MATLAB," *2020 Int. Conf. Comput. Inf. Technol. ICCIT 2020*, no. 1, pp. 358–363, 2020, doi: 10.1109/ICCIT-144147971.2020.9213780.
- [12] C. Xiangxian, Z. Yue, and H. Hai, "Train Speed Control Algorithm Based on PID Controller and Single-Neuron PID Controller," in 2010 Second WRI Global Congress on Intelligent Systems, 2010, pp. 107–110. doi: 10.1109/GCIS.2010.41.
- [13] C. Zhao and Z. Hua, "Design of Motor Speed Control System Based on STM32 Microcontroller," in 2022 International Conference on Computation, Big-Data and Engineering (ICCBE), 2022, pp. 274–276. doi: 10.1109/ICCBE56101.2022.9888225.
- [14] M. E. A. Mohamed, G. Yanling, B. A. rahman Osman, and A. A. Mohamed, "Design of Speed Control System of DC motor Based on PID tuning with fuzzy define weighting point," in 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), 2019, pp. 1–6. doi: 10.1109/ICCCEEE46830.2019.9071374.
- [15] S. Sahoo, B. Subudhi, and G. Panda, "Optimal speed control of DC motor using linear quadratic regulator and model predictive control," in 2015 International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE), 2015, pp. 1–5. doi: 10.1109/EPETSG.2015.7510130.

# Kontrol Motor DC dengan Kendali *Linear Quadratic Regulator* dan Filter Kalman Menggunakan GUI MATLAB

Imroatul Hudati<sup>1,\*</sup>, Nurul Izzah Wijayakusuma<sup>1</sup>, Aditya Yusuf Ismail<sup>1</sup>, Alfonzo Aruga Paripurna Barus<sup>1</sup>,  
Hendi Budiando<sup>1</sup>

<sup>1</sup>Departemen Teknik Elektro dan Informatika, Universitas Gadjah Mada; nurulizzah@mail.ugm.ac.id,  
adityayusuf00@mail.ugm.ac.id, adityayusuf00@mail.ugm.ac.id, hendi.b@mail.ugm.ac.id

\*Korespondensi: imroatul.hudati@ugm.ac.id

**Abstract** – DC motor is one of the electronic devices that is used in the robotic, industry and sometime can be found in daily life. DC motors usage is quite a lot, where the user is expected to be able monitor the speed or the position of the DC motor and be able to make adjustments. Noise is a part that cannot be ignored in controlling the motor DC speed. Numerous studies have been conducted. Previous research monitored the system response without considering the noise. The aims of this study are to implement the linear quadratic regulator (LQR) control and provide noise in the form of random number which will be filtered using a Kalman filter to reduce the noise and achieve a more stable system response. The provision of control and filter methods will be monitored by the user through an easy-to-understand display. The LQR control method will be given to DC motors in order to reduce the error between the output and input. LQR control use six variations of the Q index weighting will be given with an R value of 1. The most optimal Q result is [0.1 0 0; 0 0.1 0; 0 0 0.1] with a rise time is 2.20 seconds and a peak overshoot is 0.09 and the system response results are displayed. in the MATLAB GUI with the addition results after being given a Kalman filter.

**Keywords** – DC motor, LQR, Kalman Filter, MATLAB GUI

**Intisari** – Penggunaan Motor DC menjadi salah satu perangkat elektronik yang digunakan pada bidang robotika, industri bahkan banyak ditemui dalam kehidupan sehari – hari. Penggunaan motor DC yang cukup banyak, di mana user diharapkan dapat memantau kecepatan atau posisi motor DC dan dapat melakukan pemantauan terhadap noise yang terjadi pada sistem. Terdapat banyak penelitian yang telah dilakukan. Penelitian sebelumnya hanya memantau respons sistem tanpa memperhatikan noise yang dihasilkan. Penelitian ini bertujuan untuk menerapkan kendali *linear quadratic regulator* (LQR) dan memberikan noise berupa random number yang kemudian diberi filter Kalman sehingga noise tersebut dapat dikurangi dan respon sistem menjadi lebih stabil. Pemberian metode kendali dan filter diharapkan dapat dipantau oleh user melalui sebuah tampilan yang mudah dipahami. Metode kendali LQR akan diberikan kepada motor DC untuk mengurangi besarnya kesalahan antara output terhadap input. Pada kendali LQR akan diberikan enam variasi pembobotan indeks Q dengan R bernilai 1 didapatkan hasil Q yang paling optimal ialah [0,1 0 0; 0,1 0 0; 0 0 0,1] dengan rise time sebesar 2,20 detik dan peak overshoot sebesar 0,09 dan hasil respons sistem ditampilkan pada GUI MATLAB dengan adanya penambahan hasil sistem setelah diberikan filter Kalman.

**Kata kunci** – kecepatan motor DC, LQR, filter Kalman, GUI MATLAB

## I. PENDAHULUAN

Motor DC merupakan salah satu bagian penting sebagai aktuator dalam sebuah sistem seperti sebagai penggerak lengan robot, konveyor pada industri, penggerak alternator [1] dan lain sebagainya. Motor DC menjadi perangkat yang cukup penting yang menyebabkan terdapat berbagai penelitian untuk mengendalikan posisi dan kecepatan motor DC. Kecepatan dan posisi motor DC tersebut harus dapat diamati oleh pengguna agar pengendalian dapat lebih mudah untuk dilakukan. Pada penggunaan motor DC, sangat erat kaitannya dengan adanya noise. Hal inilah yang menyebabkan sistem dibuat agar dapat lebih tahan terhadap noise yang dihasilkan oleh sistem itu sendiri.

Kontrol optimal menjadi salah satu pendekatan kontrol modern yang diterapkan pada motor DC untuk mengatur kecepatan dengan mengintegrasikan optimisasi ke desain pengontrol yang tidak optimal [2]. Pengendalian kecepatan motor DC terdapat masukan dan keluaran yang dapat dikontrol dengan cara membuat pemodelan sistem, salah satunya ialah model *state space* atau ruang keadaan. Hal ini dapat diterapkan pada salah satu metode kontrol yang digunakan dalam sistem kontrol optimal yaitu *linear quadratic regulator* (LQR) [3], [4].

Penelitian pada pengendalian motor DC sudah banyak dilakukan seperti pada penelitian [5], yang melakukan pengendalian terhadap motor DC dengan menggunakan kendali PID. Hasil kendali akan ditampilkan pada tampilan *front panel* LabVIEW. Kendali dan tampilan pada penelitian ini telah berjalan dengan baik, tetapi sistem belum memberikan filter pada noise. Seperti yang dapat dilihat pada hasil penelitian, hasil noise yang diberikan cukup besar. Pada penelitian [6], dilakukan kendali kecepatan motor DC dengan menggunakan metode PID yang ditala dengan menggunakan metode *Ant Colony*. Penalaan yang dilakukan masih kurang tepat sehingga hasil respons menunjukkan redaman yang cukup besar. Pada penelitian [7], dilakukan pengendalian motor DC dengan menggunakan metode MRAC PID. Metode MRAC PID merupakan penggabungan dua metode, di mana PID akan membantu MRAC memperkecil kesalahan yang dihasilkan oleh output sistem. Pada penelitian ini dilakukan uji coba ketahanan sistem terhadap gangguan dengan menganalisis diagram Bode. Pada diagram tersebut akan ditampilkan respons frekuensi sehingga dapat diamati besar nilai *gain* yang dapat ditoleransi oleh sistem. Penelitian ini masih terpaku pada hasil simulasi dan belum mempertimbangkan beberapa keadaan lain seperti adanya noise yang kemungkinan besar dapat terjadi.

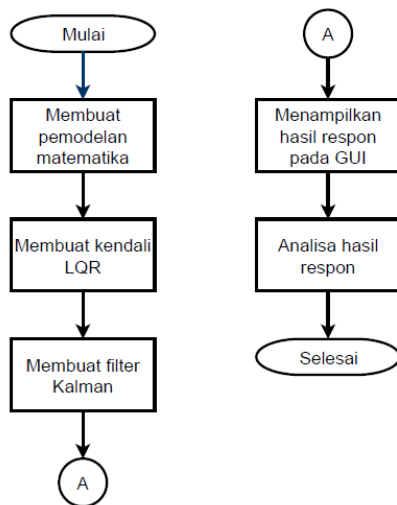
Pada penelitian ini sistem yang dipilih ialah sistem pemodelan motor DC dengan menggunakan kendali LQR. Pengontrol LQR memiliki dua buah parameter yang harus ditentukan untuk menghasilkan aksi kontrol yang diharapkan, yaitu matriks bobot Q dan R [8]. Namun, terdapat kemungkinan jika pada hasil LQR terdapat *noise* sehingga akan mempengaruhi hasil respons sistem. *Noise* yang diberikan pada penelitian ini merupakan *random number* yang ditambahkan pada hasil kendali LQR dan filter Kalman digunakan sebagai filter yang berfungsi untuk mereduksi hasil respons sistem yang mengandung *noise*. Pada penelitian ini dilakukan *tuning* pembobotan indeks Q melalui eksperimen dan hasil dari setiap variasi akan dibandingkan sehingga dapat mengetahui bobot Q yang paling optimal yaitu yang memiliki *rise time* cepat dan *peak overshoot* kecil. Pada penelitian ini juga dikembangkan GUI dengan menggunakan MATLAB untuk mengamati hasil respons sistem yang dihasilkan.

II. METODOLOGI

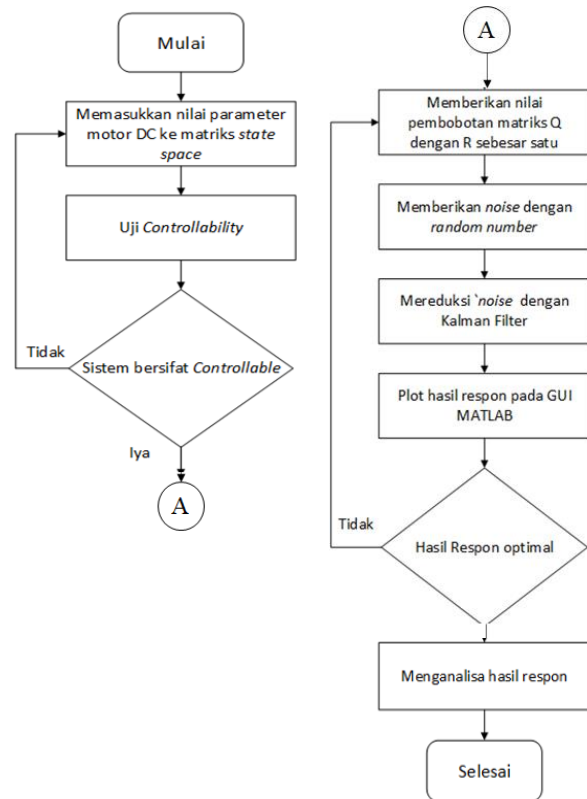
Pada penelitian ini, terdapat beberapa langkah yang dilakukan agar didapatkan hasil yang diharapkan. Gambar 1 menunjukkan rancangan dari penelitian ini. Pada tahap awal, dilakukan perancangan model matematika dari motor DC yang selanjutnya akan diberikan kendali LQR terhadap sistem tersebut. Hasil pengukuran diberikan *noise* berupa *white noise*, yang selanjutnya hasil pengukuran akan masuk menuju *filter Kalman*. Hasil respons akan diamati pada GUI MATLAB.

A. Flowchart

Perancangan sistem menggunakan *software* MATLAB baik dalam perhitungan model matematis, pengujian *controllability*, pemberian *noise* dan filter Kalman, pengambilan data dan perhitungan *mean square error* (MSE). Pada pengujian sistem, dilakukan sebanyak enam variasi dan pada setiap variasi diambil 30 data untuk perhitungan MSE yang akan berpengaruh dalam penentuan indeks performansi yang paling optimal untuk sistem. Gambar 2 menunjukkan bagan alir program pada MATLAB.



Gambar 1. Blok diagram rancangan penelitian



Gambar 2. Bagan alir program utama

B. Perancangan Model Matematis Motor DC

Sesuai dengan representasi motor DC, berdasarkan hukum II Newton akan didapatkan (1).

$$J\ddot{\theta} = T - \omega - d \tag{1}$$

$$T = K_{Ti} \tag{2}$$

$$d = b\dot{\theta} \tag{3}$$

Dengan,

- $J$  inersia
- $\theta$  sudut putaran dari *shaft*
- $T$  torsi
- $\omega$  gangguan torsi
- $d$  redaman torsi
- $K_{Ti}$  Konstanta Torsi

Dengan menyubstitusikan (2) dan (3) pada (1), akan didapatkan (4).

$$J\ddot{\theta} + b\dot{\theta} = K_{Ti} - \omega \tag{4}$$

Arus jangkar memiliki persamaan tersendiri seperti yang ditunjukkan pada (5).  $e_m$  merupakan gaya gerak listrik balik yang diberikan oleh (6). Dengan menyubstitusikan (6) dengan (5), didapatkan (7).

$$Li + Ri = v - e_m \tag{5}$$

$$e_m = K_e\dot{\theta} \tag{6}$$

$$Li + Ri = v - k_e\dot{\theta} \tag{7}$$

Persamaan (4) dan (7) untuk mengatur dinamika motor DC, untuk lebih jelasnya ditunjukkan oleh (8) dan (9). Dari persamaan tersebut diketahui bahwa  $v$  merupakan tegangan yang digunakan untuk mengendalikan motor.

$$J\ddot{\theta} + b\dot{\theta} = K_{Ti} - \omega \tag{8}$$

$$Li + Ri = v - K_e\dot{\theta} \tag{9}$$

C. Perancangan Kendali LQR

Linear quadratic regulator (LQR) adalah suatu metode kendali modern karena menerapkan sistem *multiple input multiple output* (MIMO), di mana sistem yang akan dikendalikan dibuat menjadi bentuk *state space* sehingga pengendalian dapat dilakukan dari beberapa masukan dan menghasilkan beberapa keluaran [9]. LQR disebut linier karena model dan bentuk kontrolnya berupa linier, sedangkan disebut kuadratik karena *cost function* pada kendali ini berbentuk kuadratik. Fungsi tersebut harus dikesalkan, serta referensi yang diberikan bukanlah berupa fungsi waktu sehingga disebut regulator [10], [11].

Gambar 3 menampilkan blok diagram dari LQR, dengan persamaan ruang keadaan (*state space*) seperti pada (10) dan (11). dan persamaan indeks performansinya diberikan oleh (12). Di mana  $Q$  adalah matriks definit positif (atau semi definit positif).

$R$  adalah matriks definit positif, dan  $u$  tidak dibatasi. Tujuan utama dari sistem kendali optimal adalah untuk meminimalkan indeks performansinya sehingga sistem menjadi stabil.

$$\dot{x}(t) = Ax(t) + u(t) \tag{10}$$

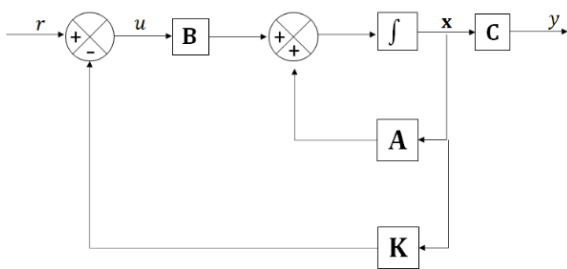
$$y(t) = Cx(t) + Du(t) \tag{11}$$

$$J = \int_0^\infty (x^t Q x + u^T R u) dt \tag{12}$$

Berdasarkan sistem yang ditentukan seperti (10) nilai matriks  $K$  ditentukan dari persamaan vektor kendali optimal seperti (13). Sehingga persamaan indeks performansi ( $J$ ) diminimalkan. Nilai matriks  $P$  didapatkan dengan cara menyelesaikan persamaan Aljabar Riccati tereduksi, seperti (14).

$$u(t) = -(R^{-1}B^T P)x(t) = -Kx(t) \tag{13}$$

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{14}$$



Gambar 3. Contoh kendali LQR [8]

Setelah itu, disubstitusikan nilai matriks  $P$  ke (15). Hasil nilai matriks  $K$  yang didapat merupakan nilai matriks optimalnya.

$$K = R^{-1}B^T P \tag{15}$$

Pada kendali LQR, model sistem diubah ke *state space* dengan bentuk umum persamaan (10) dan (11). Dan membuat variabel *state* seperti berikut.

- Variabel *state*

$$\begin{aligned} x_1 &= \theta \\ x_2 &= \dot{\theta} \\ x_3 &= i \\ y &= x_1 \\ u &= V \end{aligned}$$

- Persamaan *state*

$$\dot{x}_1 = x_2 \tag{16}$$

$$\dot{x}_2 = \frac{K_t}{J} x_3 - \frac{b}{J} x_2 \tag{17}$$

$$\dot{x}_3 = \frac{R}{L} x_3 - \frac{K_e}{L} x_2 + \frac{1}{L} u \tag{18}$$

- *State space*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{K_t}{J} \\ 0 & -\frac{K_e}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u \tag{19}$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{20}$$

$$D = 0 \tag{21}$$

Indeks performansi pada kendali LQR dipilih sesuai dengan bagian yang akan dioptimalkan. Pada penelitian ini yang akan dioptimalkan ialah bobot indeks  $Q$  dengan  $R$  sebesar satu.

D. Perancangan Filter Kalman

Secara umum, Persamaan untuk metode filter Kalman dituliskan dalam bentuk (22)-(26). [12]. dengan  $\hat{x}$  adalah estimasi *state*,  $F$  adalah matriks transisi,  $u$  adalah variabel kendali,  $B$  adalah matriks kendali,  $P$  matriks variasi *state*,  $Q$  adalah matriks variasi proses,  $y$  adalah variabel pengukuran,  $H$  adalah matriks perhitungan,  $K$  adalah penguatan Kalman,  $R$  adalah matriks variasi pengukuran.

- Tahap prediksi

$$\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1} + B_t u_t \tag{22}$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t \tag{23}$$

- Tahap *update* :

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(Y_t - H_t) \hat{x}_{t|t-1} \quad (24)$$

$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1} \quad (25)$$

$$P_{t|t} = (I - K_t H_t) P_{t|t-1} \quad (26)$$

### E. Perancangan GUI MATLAB

Perancangan GUI MATLAB terdiri dari perancangan tampilan GUI MATLAB dan perancangan program. Penjelasan terkait tiap bagian akan dijelaskan sebagai berikut.

1) *Tampilan GUI MATLAB*: Perancangan *graphical user interface* (GUI) pada MATLAB seperti yang ditunjukkan Gambar 4. Tampilan GUI terdiri dari panel *state space* matriks A B C D, parameter Matriks Q dan R, hasil nilai K, tombol plot, dan plot hasil respons sistem. Pada plot hasil respons sistem terdiri dari tiga plot grafik yaitu:

1. Grafik respons awal yang merupakan respons dari *state space* yang diberi kendali LQR.
2. Grafik nilai random sebagai *noise* yang diberikan ke hasil grafik LQR.
3. Grafik hasil filter Kalman yang merupakan grafik LQR setelah mendapatkan *noise* dan diberi *filter Kalman* untuk mereduksi *noise* tersebut.

2) *Program GUI MATLAB*: Untuk dapat menampilkan hasil grafik plot, maka tampilan GUI MATLAB perlu diprogram yang terdapat pada fitur *editor* di mana terdapat panel matriks yang digunakan untuk mengisi matriks *state space* A B C D juga matriks Q dan R, program tersebut dimasukkan pada fungsi *push button object*. Nilai-nilai yang diberikan pada matriks akan diproses menggunakan persamaan LQR seperti yang ditunjukkan sebagai berikut.

```
K = lqr(A,B,Q1,R1);
Set(handles.K1, 'string', K);
sys = ss(A-B*K,B,C,D);
```

Persamaan yang diberikan merupakan rumus yang sudah terdapat pada MATLAB, sehingga penulisan hanya disesuaikan dengan nama variabel yang diberikan [13]. Setelah mendapatkan nilai K dan hasil plot sistem LQR maka diberikan *random number*.

- *Time duration/LQR*

```
t = 1 : 100;
y = step(t,sys);
y1 = y.';
ywn = wgn(101,1,0)
yn = y1 + ywn.'
yn = y1 + 1*rand(size(t));
axes(handles.axes1);
plot(t,y);
```

- *Noise*

```
Axis(handles.noise);
Plot(t, yn);
```



Gambar 4. GUI pada MATLAB

*Random number* yang diberikan pada sistem merupakan representasi dari sebuah *noise* sehingga sistem perlu diberikan sebuah filter. Pada penelitian ini filter yang digunakan ialah filter Kalman, seperti yang ditunjukkan oleh (22)-(26).

Rumus filter Kalman yang dibuat pada MATLAB, untuk memberikan filter pada pemodelan sistem motor DC dengan menggunakan LQR dan data akan ditampilkan dalam bentuk grafik. Pada tampilan GUI menunjukkan grafik respons sistem hasil dari *plotting* perintah MATLAB tersebut.

Sistem juga diberikan perhitungan MSE yang berfungsi sebagai perhitungan *error* dengan persamaan (27). Hasil dari MSE dapat digunakan sebagai acuan dalam melihat keakuratan pengukuran sistem. Nilai MSE yang baik ialah hasil perhitungan dengan nilai MSE kecil.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \check{y}_i)^2 \quad (27)$$

Dengan

$y_i$  nilai respons *output*

$\check{y}_i$  nilai *input*

$n$  banyak data

- Filter Kalman

```
xk=0;
Pk=0;
Q= 10;
R=100;

for kk=1:100
    zk=yn(kk);
    Pk=Pk+Q;
    Kk = Pk/(Pk+R);
    xk=xk+(Kk*(zk-xk));
    Pk=(1-Kk)*Pk;
    buff_xk(kk)=xk;
    buff_zk(kk)=zk;
end
```

- Plot filter Kalman

```
axes(handles.kalman);
plot(t,buff_xk);
```

### III. HASIL DAN PEMBAHASAN

Nilai parameter motor DC yang digunakan sebagai pemodelan sistem dengan kendali LQR ditunjukkan oleh Tabel 1. Nilai parameter yang digunakan tersebut didapatkan dari implementasi motor DC untuk robot medis [15]. Untuk menentukan *controllability* sistem maka dilakukan pengujian sistem seperti berikut.

- Matriks sistem

$$\begin{aligned} A &= [\theta \ 1 \ \theta; \ \theta \ -b/J \ Kt/J; \ \theta \ -Ke/L \ -R/L]; \\ B &= [\theta; \ \theta; \ 1/L]; \\ C &= [1 \ \theta \ \theta]; \\ D &= \theta \end{aligned}$$

- Matriks Q dan R

$$\begin{aligned} Q &= [250 \ \theta \ \theta; \ \theta \ \theta \ \theta; \ \theta \ \theta \ 250]; \\ R &= \theta.001; \end{aligned}$$

- Pengujian *controllability*

$$\text{Uncontrollability} = \text{length}(A) - \text{rank}(cm)$$

- Hasil

$$\text{Uncontrollability} = \theta$$

Pengujian *controllability* sistem pada penelitian ini melihat dari matriks *uncontrollable*-nya di mana pada *syntax* tersebut menunjukkan bahwa matriks *uncontrollable* berjumlah nol, artinya sistem bersifat *controllable*. Pada sistem kendali LQR matriks Q dan R merupakan matriks dengan sifat *real definit positif*. Kecepatan respons variabel *state* dalam menuju kestabilan berhubungan dengan matriks Q sedangkan pengontrolan atau seberapa besar kontrol *input* yang dihasilkan berhubungan dengan matriks R. Oleh karena itu, untuk mendapatkan hasil respons sistem terbaik maka dilakukan *tuning* matriks Q dan R dengan cara mengubah

nilainya sehingga akan diperoleh variasi *gain* K. Pemilihan matriks Q dan R dilakukan secara *trial and error* [13], sehingga dilakukan pengamatan dari enam variasi matriks Q dan R, selanjutnya dilakukan pengamatan perubahan respons sistem yang dihasilkan.

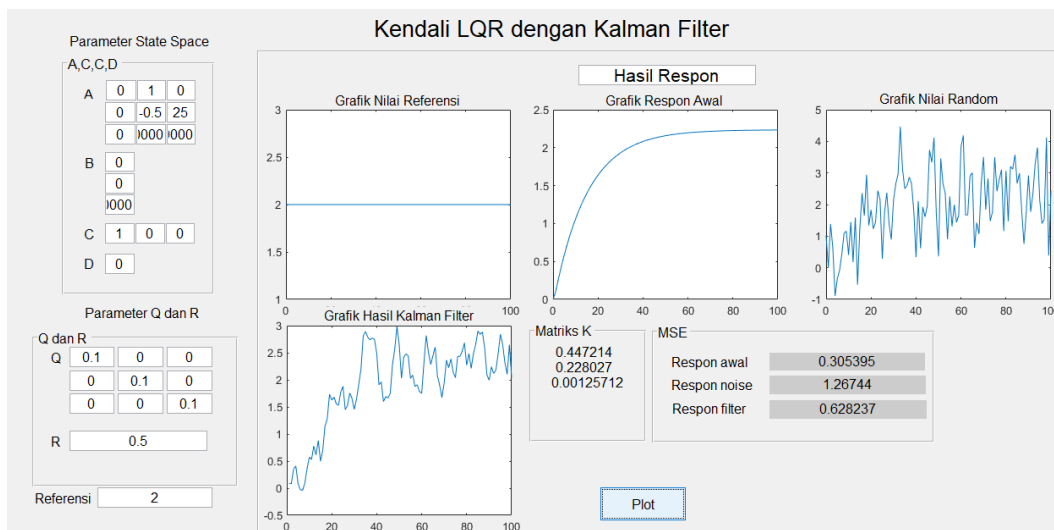
Penelitian ini menggunakan variasi parameter matriks Q yang dilakukan sebanyak enam variasi dengan nilai R sebesar 1. Hasil berupa respons sistem akan didapatkan dengan melakukan perhitungan MSE, maka akan diketahui parameter nilai Q dan R yang terbaik untuk sistem tersebut. Pengambilan data untuk tiap variasi dilakukan sebanyak 30 pengujian untuk mendapatkan perhitungan nilai *error* sistem pada saat diberi kendali LQR, ditambahkan *noise* dan saat pemberian filter Kalman sehingga dapat diketahui variasi dengan nilai MSE terkecil. Variasi pembobotan indeks Q pertama ditunjukkan oleh Gambar 5.

Matriks A B C D mengacu pada (15)-(20) yang merupakan hasil *state space* pemodelan motor DC. Nilai parameter Tabel 1 dimasukkan pada matriks *state space* sehingga menjadi:

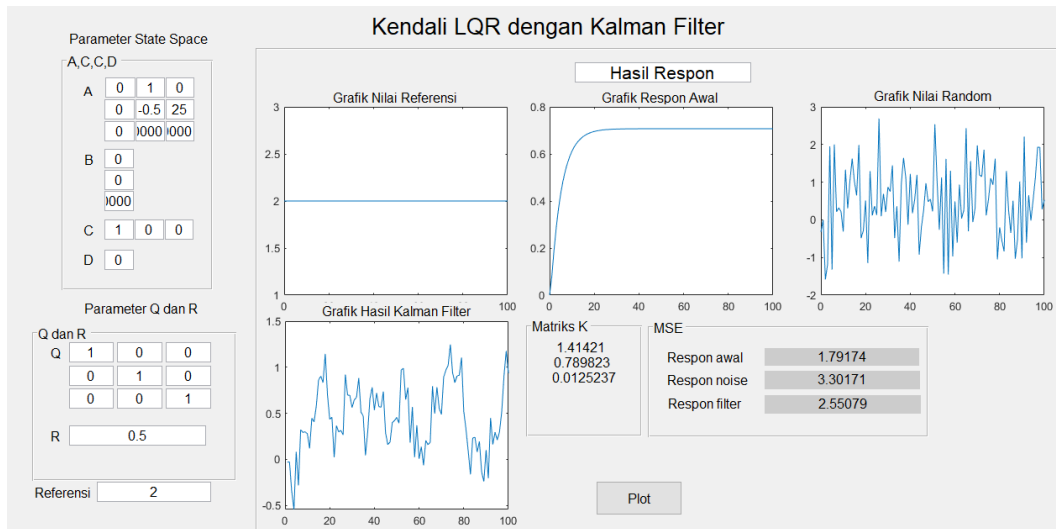
$$\begin{aligned} A &= [\theta \ 1 \ \theta; \ \theta \ -0.5 \ 25; \ \theta \ -5000 \ -80000] \\ B &= [\theta; \ \theta; \ 10000] \\ C &= [1 \ \theta \ \theta] \\ D &= \theta \end{aligned}$$

Tabel 1. Parameter motor DC

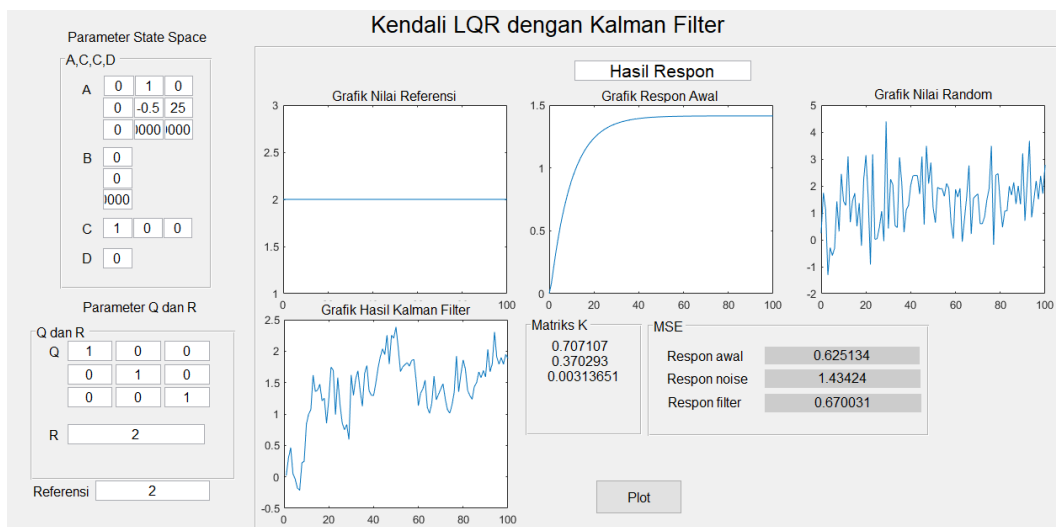
Parameter	Nilai
b	0,01 Nms
J	0,02 kgm <sup>2</sup>
Kt	0,5 Nm/Awp
Ke	0,5 V/rad/sec
L	0,0001 H
R	8 Ohm



Gambar 5. Matriks Q = 0.1 dan R = 0.5



Gambar 6. Matriks Q = 1 dan R = 0,5



Gambar 7. Matriks Q = 1 dan R = 2

Selanjutnya diberikan kendali LQR dengan indeks nilai  $Q = [0,1 \ 0 \ 0; 0 \ 0,1 \ 0; 0 \ 0 \ 0,1]$  dan  $R = 0,5$ . Nilai gain K ditunjukkan pada panel GUI dan juga di mana gain K tersebut akan mempengaruhi respons sistem. Pada Gambar 5 terdapat grafik respons awal sistem setelah diberi kendali LQR yang menunjukkan keadaan *steady state* sistem pada amplitudo 3,1 dengan *rise time* 5,15 s. Hasil respons LQR kemudian diberi *random number* sebagai *noise* seperti yang ditunjukkan pada grafik nilai acak. Berdasarkan hasil tersebut, respons sistem menjadi sangat tidak stabil sehingga perlu diberi filter Kalman yang ditunjukkan oleh grafik hasil filter Kalman dengan *noise* yang sudah tereduksi.

Variasi Q kedua dengan indeks  $Q = [1 \ 0 \ 0; 0 \ 1 \ 0; 0 \ 0 \ 1]$  dan  $R = 0,5$ . Sistem *steady state* pada amplitudo 1,41 dan *rise time* 3,072 s di mana juga diberikan *noise* berupa *random number* dan filter Kalman dengan respons seperti yang ditunjukkan oleh Gambar 6.

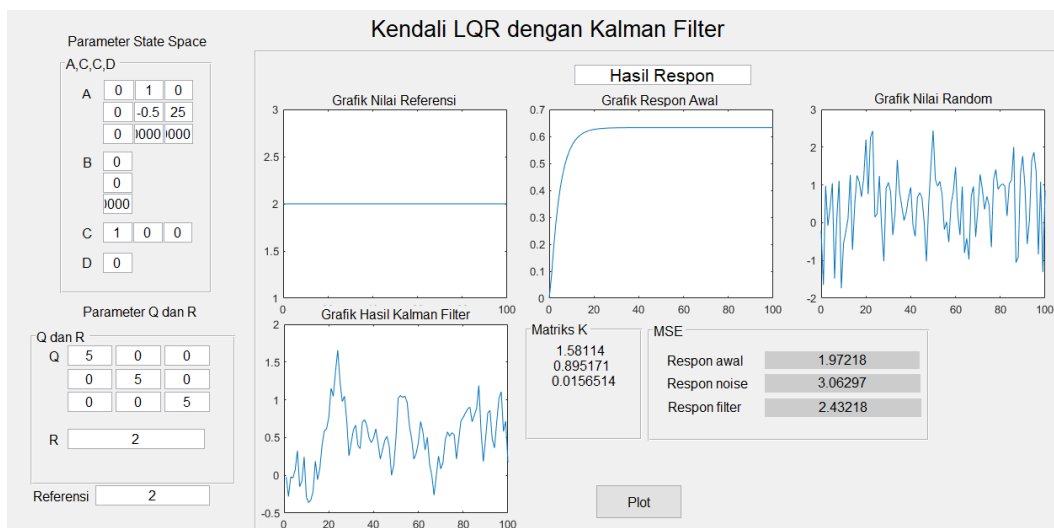
Variasi Q ketiga dengan indeks  $Q = [1 \ 0 \ 0; 0 \ 1 \ 0; 0 \ 0 \ 1]$  dan  $R = 2$ . Sistem *steady state* pada amplitudo 0,99 dan *rise*

*time* 2,683 di mana juga diberikan *noise* berupa *random number* dan diberi filter Kalman dengan respons seperti yang ditunjukkan oleh Gambar 7.

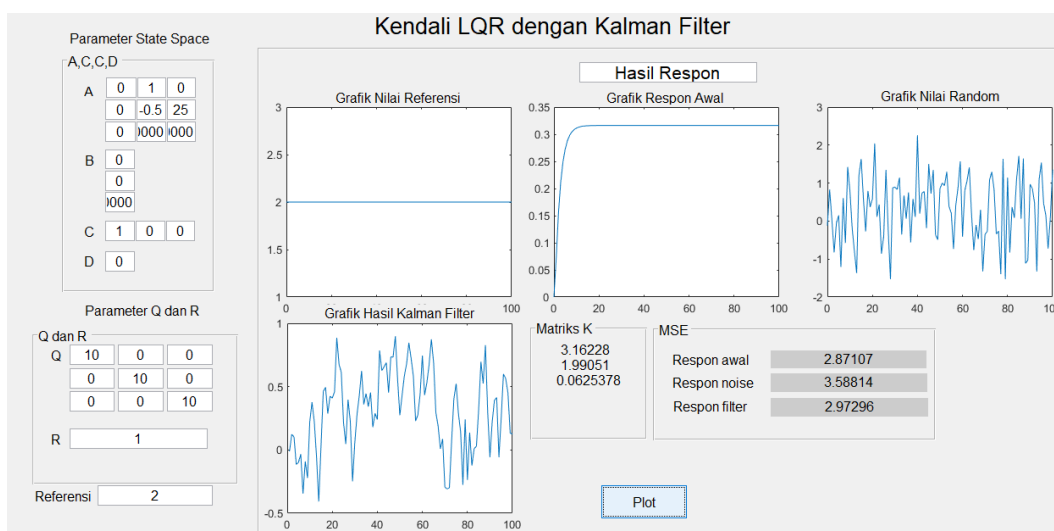
Variasi Q keempat dengan indeks  $Q = [5 \ 0 \ 0; 0 \ 5 \ 0; 0 \ 0 \ 5]$  dan  $R = 2$ . Sistem *steady state* pada amplitudo 0,446 dan *rise time* 2,683 s di mana juga diberikan *noise* berupa *random number* dan diberi filter Kalman dengan respons seperti yang ditunjukkan oleh Gambar 8.

Variasi Q kelima dengan indeks  $Q = [10 \ 0 \ 0; 0 \ 10 \ 0; 0 \ 0 \ 10]$  dan  $R = 1$ . Sistem *steady state* pada amplitudo 0,316 dan *rise time* 2,253s di mana juga diberikan *noise* berupa *random number* dan diberi filter Kalman dengan respons seperti yang ditunjukkan oleh Gambar 9.

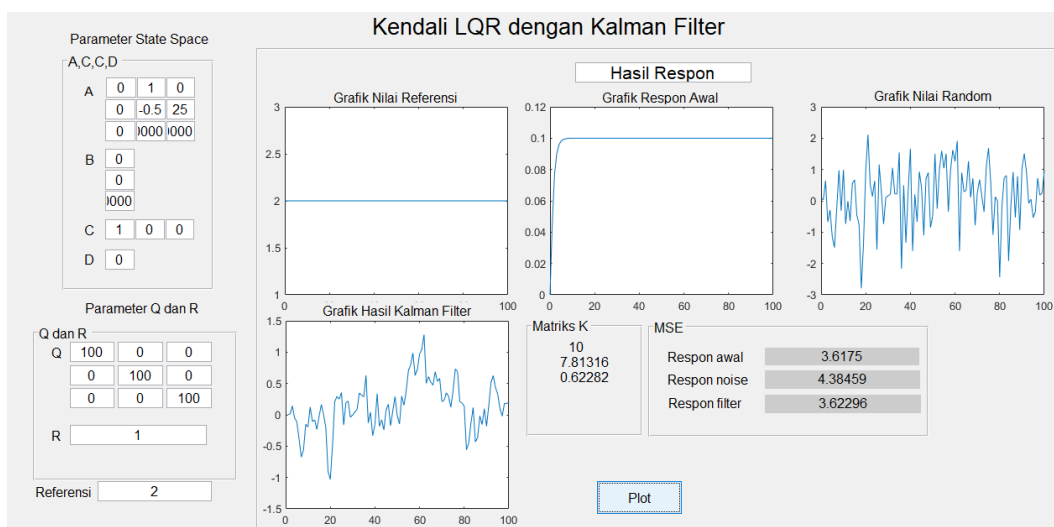
Variasi keenam ialah dengan indeks  $Q = [100 \ 0 \ 0; 0 \ 100 \ 0; 0 \ 0 \ 100]$  dan  $R = 1$ . Sistem *steady state* pada amplitudo 0,09 dan *rise time* 2,20 s di mana juga diberikan *noise* berupa *random number* dan diberi filter Kalman dengan respons seperti yang ditunjukkan oleh Gambar 10.



Gambar 8. Matriks Q = 5 dan R = 2



Gambar 9. Matriks Q Matriks Q = 10 dan R = 1



Gambar 10. Matriks Q bernilai 100



Tabel 2. Info sistem

Info	Variasi Q					
	0.1	0.5	1	5	10	100
Rise Time	5,15	3,072	2,68	2,30	2,25	2,06
Settling Time	9,380	5,557	4,87	4,19	4,08	3,96
Overshoot	0	0	0	0	0	0
Peak	3,16	1,414	0,99	0,44	0,31	0,09

Tabel 3. Hasil gain K

No.	Q	R	K	MSE		
				RA	RN	RF
1	0,1	0,5	0,447	0,305	1,26	0,628
			0,228 0,001			
2	1	0,5	1,414	1,791	3,301	2,55
			0,789 0,012			
3	1	2	0,707	0,625	1,434	0,670
			0,370 0,003			
4	5	2	1,581	1,972	3,062	2,432
			0,895 0,015			
5	10	1	3,162	2,871	3,588	2,972
			1,990 0,062			
6	100	1	10	3,617	4,384	3,622
			7,813 0,622			

Tabel 2 menunjukkan karakter sistem yang didapatkan setelah dilakukan enam variasi nilai Q. Keseluruhan gain K yang diperoleh dari variasi indeks Q dengan R bernilai satu ditunjukkan pada Tabel 3. Pemilihan enam variasi nilai Q dan R dipilih untuk menganalisis perbedaan nilai pengaruh nilai Q dan R tertinggi hingga terendah. Analisis dilakukan dengan melihat hasil MSE yang ditampilkan. Hasil tersebut menunjukkan bahwa semakin besar nilai Q yang diberikan maka perubahan gain K juga akan semakin besar. Jika mengacu pada hasil respons sistem, nilai Q akan mempengaruhi rise time dan juga peak overshoot (Mp) atau amplitudo sistem. Semakin besar nilai Q maka akan semakin cepat rise time sistem dan semakin kecil peak overshoot-nya.

Tabel 4 dan Tabel 5 menunjukkan hasil perhitungan MSE dengan nilai MSE terkecil yaitu pada variasi Q = 100, selain itu dengan melihat pada respons sistem yang memiliki rise time paling cepat dan peak overshoot kecil maka indeks Q sebesar 100 dipilih menjadi nilai indeks acuan. Dengan memberikan nilai Q = 100 dan R = 2 sudah menunjukkan perbedaan hasil respons optimal yang diharapkan yaitu dengan MSE terkecil dan sistem dapat mencapai rise time pada waktu 5,77 detik dengan nilai Q = 0,1 dan R = 2.

Tabel 4. Hasil perhitungan MSE (1)

	MSE		
	Q = 0,1	Q = 0,5	Q = 1
LQR	5,77	0,73	0,29
LQR dengan noise	0,36	0,31	0,35
Filter Kalman	0,20	0,10	0,09

Tabel 5. Hasil perhitungan MSE (2)

	MSE		
	Q = 5	Q = 10	Q = 100
LQR	0,03	0,02	0,0015
LQR dengan noise	0,37	0,33	0,32
Filter Kalman	0,07	0,06	0,05

#### IV. SIMPULAN

Berdasarkan hasil penelitian yang dilakukan, maka didapatkan bahwa kendali LQR dapat digunakan untuk optimisasi motor DC dengan cara membuat model *state space* sistem. Pada penelitian ini juga menentukan nilai parameter yang digunakan serta nilai indeks performansi Q dan R untuk mendapatkan gain k.

Nilai indeks Q yang paling baik dari hasil penelitian yang telah dilakukan adalah [0,1 0 0; 0 0,1 0; 0 0 0,1] dengan R bernilai 0,5 yang menghasilkan karakteristik respons dengan rise time pada data ke-50. Mengacu pada hal tersebut, menunjukkan bahwa indeks Q dan R sangat berpengaruh terhadap respon sistem. Dengan semakin besar nilai Q dengan nilai R tertentu yang diberikan, maka rise time sistem akan semakin cepat, peak overshoot semakin kecil dan juga menghasilkan nilai error kecil yaitu sebesar 0,305. LQR dengan noise sebesar 1,241 dan setelah diberi filter Kalman nilai error sebesar 0,489, hal tersebut membuktikan bahwa kendali LQR dapat mengurangi kesalahan keadaan tunak dan meningkatkan kecepatan respon. Sistem juga diberi noise berupa gaussian white noise yang kemudian dapat direduksi oleh filter Kalman. Selain itu, penyesuaian juga dilakukan pada parameter Q sebesar 100 dan R sebesar 10 pada filter Kalman, sehingga dapat mereduksi noise lebih optimal. Hasil respons LQR, random number, dan filter Kalman ditampilkan pada GUI MATLAB.

#### DAFTAR PUSTAKA

- [1] Sofiah and M. Hurairah, "Jse-11," *Ranc. Bangun Pembangkit List. Altern. dengan Bantu. Pully dan Belt Mot. DC sebagai Penggerak Altern.*, pp. 10–17, 2020.
- [2] E. M. SARTIKA, M. MULIADY, R. SARJONO, and V. YUVENS, "Pengontrolan Kecepatan Rotor BLDC UAV Berdasarkan Hasil Identifikasi menggunakan Metode Regresi," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 9, no. 1, p. 114, 2021, doi: 10.26760/elkomika.v9i1.114.
- [3] Q. P. Syahna, E. Yuniarti, and E. Kurniawan, "Analisis Respon

- Sistem Kendali LQR (Linear Quadratic Regulator) Pada Simulasi Gimbal Kamera Dua Sumbu,” *Al-Fiziya J. Mater. Sci. Geophys. Instrum. Theor. Phys.*, vol. 2, no. 1, pp. 49–55, 2019, doi: 10.15408/fiziya.v2i1.11178.
- [4] F. F. Rahani and T. K. Priyambodo, “Penalaan Mandiri Full State Feedback LQR dengan JST Tiruan Pada Kendali Quadrotor,” vol. 9, no. 1, pp. 21–32, 2019, doi: 10.22146/ijeis.37212.
- [5] F. Y. Pratama and Endryansyah, “Rancang Bangun Pengendalian Kecepatan Brushless DC Motor Tipe A2212/10T 1400KV Menggunakan Kontroler PID Berbasis Labview,” *J. Tek. Elektro*, vol. 7, no. 3, pp. 157–166, 2018.
- [6] M. Ruswandi Djalal, T. Energi, J. Teknik Mesin, and P. Negeri Ujung Pandang Jalan Perintis Kemerdekaan, “Penalaan optimal kendali motor DC berbasis ant colony optimization,” *J. Teknol.*, vol. 12, no. 1, pp. 49–56, 2020, [Online]. Available: <https://dx.doi.org/10.24853/jurtek.12.1.49-56>
- [7] I. Hudati and U. G. Mada, “KENDALI MRAC PID PADA KECEPATAN MOTOR DC DENGAN,” vol. 13, no. 2, pp. 1–14, 2022.
- [8] K. FATHONI and A. B. UTOMO, “Perancangan Kendali Optimal pada Motor Arus Searah Tanpa Sikat melalui Metode LQRI,” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 7, no. 2, p. 377, 2019, doi: 10.26760/elkomika.v7i2.377.
- [9] E. N. Demirhan, K. Caglar Coskun, and C. Kasnakoglu, “LQI Control Design with LQG Regulator via UKF for a Fixed-Wing Aircraft,” in *2020 24th International Conference on System Theory, Control and Computing, ICSTCC 2020 - Proceedings*, Oct. 2020, pp. 25–30. doi: 10.1109/ICSTCC50638.2020.9259728.
- [10] H. Mudia, A. W. Putra, M. N. Faizi, and H. Amri, “Perancangan Kendali LQR-PD Untuk Pengendalian Sumbu Elevasi Gun Pada Turret-Gun Kaliber 20 Milimeter,” *INOVTEK - Seri Elektro*, vol. 2, no. 2, p. 71, 2020, doi: 10.35314/ise.v2i2.1402.
- [11] A. Casro, L. Gaol, G. E. Setyawan, and W. Kurniawan, “Pendaratan Otomatis Quadcopter AR Drone Menggunakan Metode Linear Quadratic Regulator (LQR),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 10, pp. 1028–1035, 2017, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [12] J. Saelao and D. Sukawat, “Kalman filter for estimate the state of shallow water model,” *Aust. J. Basic Appl. Sci.*, vol. 5, no. 1, pp. 31–37, 2011.
- [13] M. ARROFIQ, L. S. NUGROHO, F. FAHMIZAL, and E. APRIASKAR, “Sistem Kendali Eddy Current Brakes Dinamometer menggunakan Linear Quadratic Regulator (LQR),” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 9, no. 4, p. 923, 2021, doi: 10.26760/elkomika.v9i4.923.

# Design of Water Quality Monitoring System for Koi Fish Farming Using NodeMCU ESP32 and Blynk Application Based on Internet of Things

Dendi Rosandi<sup>1\*</sup>, Junaidi<sup>1</sup>, Donni Kis Apriyanto<sup>1</sup>, Arif Surtono<sup>1</sup>

<sup>1</sup>Department of Physics, Faculty of Mathematics and Natural Science, University of Lampung;  
junaidi.1982@fmipa.unila.ac.id, donni.kis@fmipa.unila.ac.id, arif.surtono@fmipa.unila.ac.id

\*Correspondence: dendirosandi85@gmail.com

*Intisari – Penelitian tentang rancang bangun sistem pemantauan kualitas air budidaya ikan koi telah dilakukan menggunakan NodeMCU ESP32 berbasis Internet of Things (IoT) yang terhubung dengan perangkat Android melalui aplikasi Blynk. Sistem ini terdiri dari empat sensor kualitas air, yaitu jumlah padatan terlarut, tingkat keasaman, tingkat amonia, dan besar suhu. Sistem ini dibuat agar kualitas air ikan koi selalu terpantau secara langsung lewat Android. Keuntungan menggunakan perangkat Android sebagai penampil sistem pemantauan adalah memudahkan pengguna untuk memantau kondisi air di lokasi manapun. Hasil dari sistem ini ditampilkan dalam bentuk pemantauan pada aplikasi Blynk dan pada liquid crystal display (LCD) 20x4, dengan notifikasi tentang ambang batas kualitas air. Hasil penelitian menunjukkan bahwa sensor memiliki akurasi sebesar 94,92% untuk jumlah padatan terlarut, 98,88% untuk tingkat keasaman, 90,49% untuk tingkat amonia, dan 98,77% untuk suhu, dengan waktu tunggu selama 5 menit untuk tampilan hasil pemantauan. Tingkat akurasi yang tinggi menunjukkan alat ini dapat digunakan untuk melakukan pemantauan kualitas air budidaya ikan koi.*

*Kata kunci: Blynk, ESP32, IoT, pemantauan, kualitas air*

**Abstract -** A research study on the design and development of a water quality monitoring system for koi fish farming has been conducted using the Internet of Things (IoT) based NodeMCU ESP32, connected to an Android device through the Blynk application. This system consists of four water quality sensors, namely dissolved solids, acidity level, ammonia level, and temperature. The purpose of this system is to ensure direct monitoring of the koi fish water quality through an Android device. The advantage of using an Android device as the monitoring interface is that it allows users to monitor the water conditions from any location. The system's results are displayed through the Blynk application and a 20x4 liquid crystal display (LCD) screen, with notifications indicating the threshold limits of water quality. The research findings indicate that the sensors have an accuracy of 94.92% for dissolved solids, 98.88% for acidity level, 90.49% for ammonia level, and 98.77% for temperature, with a waiting time of 5 minutes for the monitoring results to be displayed. The high level of accuracy demonstrates that this device can be used effectively for monitoring the water quality in koi fish farming.

**Keywords:** Blynk, ESP32, IoT, monitoring, water quality

## I. INTRODUCTION

The quality of the aquarium water for breeding koi fish is crucial to consider. Water conditions that do not meet the requirements can become a source of diseases, such as *Myxobolus sp.*, which can be highly dangerous to the growth of koi fish. The water quality considered to be good for breeding koi fish is at an optimal temperature range of 30 to 32°C [1]. Other parameters that need to be considered are changes in the water's potential hydrogen (pH) level, which represents the acidity of the water. The pH level of aquarium water that is good for koi fish is around 6 to 7 [1]. Additionally, the level of total dissolved solids in the water must not exceed 400 nephelometric turbidity units (NTU) or 333 parts per million (ppm) [2]. The maximum limit of ammonia in the aquarium is 0,3 ppm. If the water quality does not meet the specified levels, the disease *Myxobolus sp.* will develop [3].

The internet is part of the technological development that has rapidly progressed in society. The internet can now be used as a communication medium and control devices from a distance as long as the devices are connected and interconnected. The development of Internet communication networks, the Internet of Things (IoT), makes objects interconnected through Internet communication and exchange data, turning it into information [4].

Based on the explanation, this research builds a water quality monitoring system based on IoT connected to

Android through the Blynk application. This system is designed to enable real-time monitoring of the water quality for koi fish directly through an Android device. The advantage of using an Android device as the monitoring interface is that it facilitates users to monitor the water conditions from any location. This monitoring system will be implemented in the koi fish aquarium, by measuring four water quality parameters such as total suspended solids, acidity level, ammonia level, and temperature. This system uses a NodeMCU ESP32 microcontroller, which is equipped with wireless fidelity (Wi-Fi) communication that is installed in the module that will be connected to the IoT system.

## II. BASIC THEORY

In this research, four water quality parameter sensors are used with the NodeMCU ESP32 as the microcontroller. The four water quality parameter sensors used include sensors for measuring total dissolved solids (TDS), acidity, ammonia, and temperature.

### A. NodeMCU ESP32

Compared to the Arduino Uno, ESP32 does not require any additional devices to access Wi-Fi or Bluetooth. Both wireless networks are already built-in within the ESP32 module.

The advantages of the ESP32 microcontroller compared to the ESP8266 include the use of a NodeMCU Xtensa Dual Core 32-bit LX6 with 600 Dhrystone Million Instructions per

Second (DMIPS), while the ESP8266 still uses a NodeMCU Xtensa Single-core with 32-bit L106. In terms of Bluetooth and Wi-Fi, the ESP32 has been integrated as a system-on-chip, whereas the ESP8266 is still separate. This means that in terms of the tools required, the ESP32 is superior compared to the ESP8266, as the ESP8266 requires additional devices for the same research purposes. Additionally, the ESP32 has the most general-purpose input/output (GPIO) pins at 32 pins compared to the ESP8266 which only has 17 GPIO pins [5].

#### B. Total Dissolved Solid (TDS) Sensor (DFRobot)

The selection of the DFRobot TDS sensor is based on its ability to measure dissolved solids ranging from 0 to 1000 ppm, with an operating temperature of up to 80°C according to its datasheet. The TDS sensor uses the electrical conductivity method, wherein two probes are submerged in a liquid or solution. The signal processing circuit will produce an output indicating the conductivity of the solution [6]. The electrode that is given a voltage source will conduct an electrical current. The conductance of a sample is proportional to the ions in the sample solution. The signal conditioning module subsequently converts the conductance value into voltage [7].

TDS is a physical parameter of raw water, the measure of dissolved substances, both organic and inorganic, in the solution. This includes the amount of material in the water, which can be carbonates, bicarbonates, chlorides, sulfates, phosphates, nitrates, calcium, magnesium, sodium, organic ions, and other ions [8].

#### C. Acidity Level Sensor

The pH sensor is designed to detect hydrogen ions ( $H^+$ ) using a thin, spherical glass electrode containing a 0.1 M HCl solution. A silver wire is dipped into the HCl solution, forming an AgCl compound. The exchange of hydrogen ions with  $H_3O^+$  from the sample solution results in a potential difference between the electrode (glass wall and silver wire) with (1).

$$E = (RT/2.303F) \log_{10}(H_3O^+) \quad (1)$$

with,

$E$	potential difference (V)
$R$	molar gas constant (8,314 J/mol K)
$T$	solution temperature (K)
$F$	the Faraday constant (96,485.3 C/mol)
$H_3O^+$	ion activity of the measured sample. [9]

This type of pH meter is widely used in quantitative chemical analysis. The probe on the pH sensor is used to measure the activity of hydrogen ions surrounding the thin-walled glass bulb at its tip. There is an increase of about 0.06 volts per pH unit measured and displayed as the measured pH value reading [10].

#### D. MQ-135 Ammonia Sensor

The MQ-135 sensor is a type of chemical sensor that is sensitive to compounds such as ammonia ( $NH_3$ ), nitrous oxide ( $NO_x$ ), alcohol, benzene, smoke (CO), carbon dioxide ( $CO_2$ ), and others. This sensor works by receiving changes in

resistance (analog) when exposed to the gas. It offers good durability for use as a pollution hazard indicator due to its practicality and low power consumption. The adjustment of the sensor's sensitivity is determined by the resistance value of the MQ-135, which varies for different gas concentrations. Therefore, when using this component, it is crucial to adjust the sensitivity [11].

The MQ-135 sensor has high sensitivity and low cost. This sensor consists of a micro aluminum oxide ( $Al_2O_3$ ) ceramic tube, a tin dioxide ( $SnO_2$ ) sensitive layer, and nickel-chromium serving as a heating coil. The sensor has six pins, with four dedicated to the signal and electrodes, and the remaining two for the heating coil. The tin dioxide semiconductor is the sensitive part of the sensor that has low conductivity in clean air. The operating principle of this sensor is based on its resistance variation when in contact with the gas to be sensed. The magnitude of the sensor's output signal depends on the gas concentration, its nature, and the type of metal oxide used for the sensor surface [12].

The semiconductor particles are heated at high temperatures in clean air, resulting in their absorption of oxygen on the particle surfaces and capturing free electrons. Donor electrons in tin dioxide are attracted towards the absorbed oxygen on the sensing material surface in clean air, then preventing the flow of electric current. For the sensor that comes into contact with reducing gas, the density of absorbed oxygen on the sensing material decreases due to the reaction with the reducing gas. This release of electrons into tin dioxide allows current to flow freely through the sensor [13].

#### E. DS18B20 Temperature Sensor

The DS18B20 temperature sensor is a digital sensor that operates using a single wire, commonly referred to as a one-wire bus. This one-wired protocol allows data and ground to be connected to the microcontroller using only one wire [14]. features a unique silicon serial number, enabling multiple DS18B20 sensors to be connected to a single bus in a circuit. The DS18B20 sensor can measure temperature in various locations, both in dry rooms and in water [15]. The DS18B20 temperature sensor is a high-accuracy sensor with an accuracy of  $\pm 0.5^\circ C$  in the temperature range of  $-10^\circ C$  to  $85^\circ C$ . It is widely used for temperature monitoring system applications [16].

### III. METHODOLOGY

This research was carried out in several stages, including system design, testing, and data collection.

#### A. System Design

The input consists of four parameter sensors: TDS sensor, pH sensor, ammonia sensor, and temperature sensor. The input values will be read by the ESP32, which is connected to the power supply. The input values will be converted into unit values. The parameter values in the unit form will be sent to Android and displayed in the Blynk application as the output. They will also be sent directly to the 20x4 liquid crystal display (LCD) screen as the output of the system. The block diagram of the system design can be seen in Figure 1.

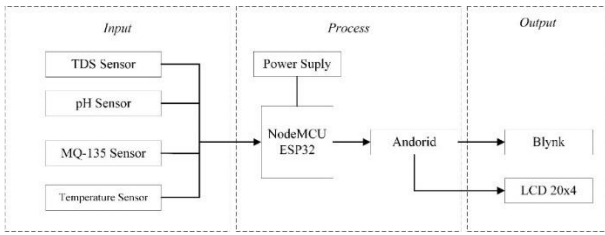


Figure 1. Block diagram of system design

The designed system includes parameter reading, processing, and data transmission to the Blynk app and LCD. In the parameter reading stage, sensors read physical parameters and convert them into electric signals (analog and digital signals). Sensors that produce analog signals will be converted into digital signals using the NodeMCU ESP32 processor. The digital values of the sensor readings and conversions will be displayed on the LCD, then the data will be sent to the Android interface application, the Blynk app, through the wireless network using Wi-Fi. The circuit diagram of the monitoring system is shown in Figure 2.

**B. System Testing**

The system testing process involves conducting tests on each component using software programs. If each component functions according to the software program instructions, the system operates effectively and data can be collected. However, during sensor testing, measurements are performed to assess accuracy and precision.

There are four parameters tested on the sensor including dissolved solids, acidity level, ammonia concentration, and temperature. The measurement data is used to calculate the error, accuracy, and precision of the monitoring system using (2)-(4).

$$\%Error = \left| \frac{Y - X_n}{Y} \right| \times 100\% \quad (2)$$

$$\%Accuracy = \left( 1 - \left| \frac{Y - X_n}{Y} \right| \right) \times 100\% \quad (3)$$

$$\%Precision = \left( 1 - \left| \frac{X_n - \bar{X}_n}{\bar{X}_n} \right| \right) \times 100\% \quad (4)$$

with,

- Y reference parameter value
- $X_n$  measured parameter value n-th
- $\bar{X}_n$  the average value of n measured parameters [17]

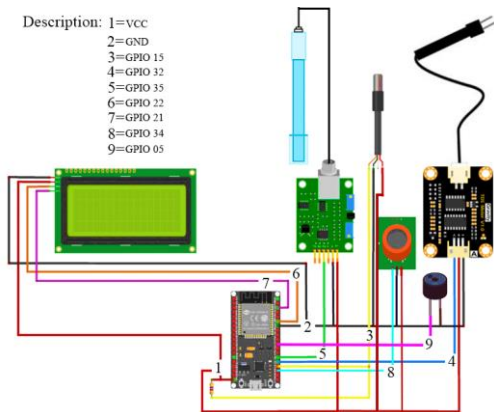


Figure 2. Circuit diagram for water quality monitoring

**C. Data Collection**

Data collection is performed by running the entire system, both hardware and software, and implementing the monitoring system instrumentation in a koi breeding aquarium. Data is collected by measuring the dissolved solids, acidity level, ammonia level, and water temperature. Measurements are taken in real-time, where measurements are taken for 65 minutes at three different times.

**IV. RESULTS AND DISCUSSION**

**A. Realization of Monitoring System**

The IoT-based water quality monitoring instrumentation system using NodeMCU ESP32 and Blynk application has been realized with the results shown in Figure 3. The box used for this hardware device has a dimension of (18 × 11 × 6) cm made of plastic.

**B. Testing of Monitoring Sensor**

1) *Total Dissolved Solid Sensor*: The supporting media used is a plastic glass sample container with a volume of 250 ml. The testing mechanism in this research is a digital measuring tool TDS Meter and an analog TDS DFRobot sensor alternately placed in the sample container which previously was given distilled water added with ten samples of coffee powder.

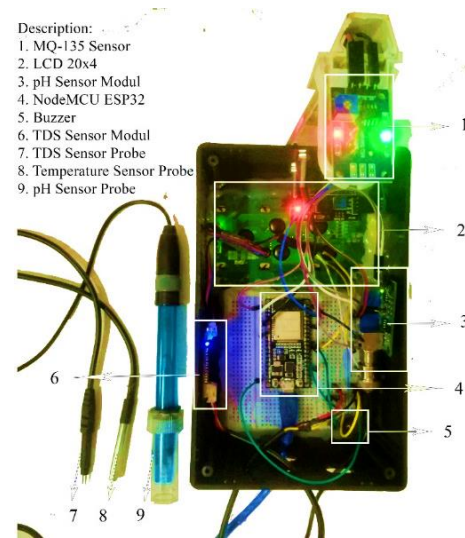


Figure 3. Realization of the monitoring system instrumentation tool

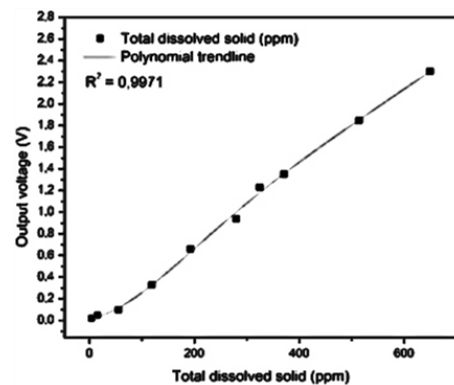


Figure 4. The characteristics of the TDS sensor

Table 1. The results of testing the fifth-degree polynomial equation for total dissolved solids in the Analog TDS DFRobot sensor

TDS Sample (ppm)	TDS (ppm) on the TDS DFRobot Sensor			Average (ppm)	Error (%)
	1	2	3		
(4.00 ± 0.08)	4.00	4.00	6.00	(4.67 ± 1.63)	16.67
(15.00 ± 0.32)	18.00	16.00	17.00	(17.00 ± 1.41)	13.33
(55.00 ± 1.00)	45.00	50.00	48.00	(47.67 ± 3.56)	13.33
(119.00 ± 2.38)	121.00	119.00	120.00	(120.00 ± 1.41)	0.84
(193.00 ± 3.88)	192.00	194.00	199.00	(195.00 ± 5.10)	1.38
(280.00 ± 5.62)	270.00	281.00	266.00	(272.33 ± 10.98)	2.98
(325.00 ± 6.48)	336.00	324.00	330.00	(330.00 ± 8.49)	1.74
(371.00 ± 7.36)	370.00	368.00	371.00	(369.67 ± 2.16)	0.36
(514.00 ± 10.30)	513.00	515.00	514.00	(514.00 ± 1.41)	0.13
(650.00 ± 13.00)	650.00	650.00	651.00	(650.33 ± 0.82)	0.05

Figure 4 shows the TDS sensor characteristics with an R2 value of 0.997 and a polynomial trend line. A 5th-degree polynomial equation is obtained to measure TDS in ppm as shown in (5), with V as the output voltage (V).

$$TDS = -2.97315 + (502.27811 \times V) + (-497.64512 \times V^2) + (396.01364 \times V^3) + (-136.34245 \times V^4) + (17.51903 \times V^5) \tag{5}$$

The polynomial equation used will make the sensor readings more accurate. The obtained fifth-degree polynomial equation is entered into the monitoring instrumentation system program to ensure that the readings from the TDS DFRobot analog sensor match the calibrated measuring tool. The average error of the sensor is 5.08%, accuracy is 94.92%, and precision is 96.75%.

2) *pH Sensor*: The supporting media used is a 250 ml plastic glass sample container. The testing mechanism in this research is distilled water that has been previously given 1-10 drops of vinegar to obtain different pH values and output voltage. The results of testing the output voltage of the analog pH sensor compared to the pH value read by the digital ATC pH meter are shown in Figure 5.

Figure 5 shows the characteristic of the pH sensor with an R2 value of 0.985. Then, with a linear trend line, a linear equation was obtained to calculate the pH value shown in (6).

$$pH = (-2.98934) \times V + 13.19243 \tag{6}$$

The testing of the sensor that has the equation incorporated into the program to ensure the accuracy and precision levels are in line with the calibrated digital ATC pH meter. The average error of the sensor is 1.12%, accuracy is 98.88%, and precision is 99.81%

3) *MQ-135 Ammonia Sensor*: The testing process of the analog MQ-135 sensor was carried out in the laboratory room with the ammonia (NH3) concentration detected by the analog pH sensor compared to the sample PPM ammonia value using (7)-(8).

$$NH_3 = m/V \tag{7}$$

$$M = (m/Mr) \times (1000/V) \tag{8}$$

with,  
*NH<sub>3</sub>* ammonia concentration (ppm)  
*m* mass of ammonia substance (mg)  
*V* volume of the room (L)  
*M* ammonia molarity (mol/L)  
*m* mass of ammonia substance (mg)  
*Mr* relative molecular mass of ammonia (g/mol)

The supporting media used is a 600 ml Erlenmeyer flask as the sample container. The testing mechanism in this research is an MQ-135 analog sensor placed on top of the sample container. The results of the analog sensor MQ-135 output voltage testing are compared to the results of the liquid ammonia sample ppm value shown in Figure 6. Figure 6 shows the R2 value of 0.982 with a polynomial trend line. The polynomial equation of power 3 to calculate the ammonia concentration in ppm is shown in (9).

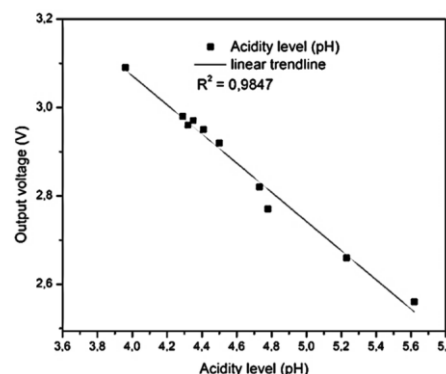


Figure 5. Graph of pH value testing against voltage output (V) of the sensor

Table 2. Results of the linearity test of the acidity level (pH) on the analog pH sensor

pH sample	pH on pH sensor			Average	Error (%)
	1	2	3		
(5.62 ± 0.2)	5.37	5.37	5.38	(5.37 ± 0.01)	4.39
(5.23 ± 0.2)	5.24	5.29	5.30	(5.28 ± 0.03)	0.89
(4.78 ± 0.2)	4.89	4.90	4.88	(4.89 ± 0.01)	2.30
(4.73 ± 0.2)	4.75	4.78	4.77	(4.77 ± 0.02)	0.78
(4.32 ± 0.2)	4.33	4.33	4.34	(4.33 ± 0.01)	0.31
(4.41 ± 0.2)	4.37	4.38	4.37	(4.37 ± 0.01)	0.83
(4.50 ± 0.2)	4.48	4.48	4.49	(4.48 ± 0.01)	0.37
(4.29 ± 0.2)	4.30	4.29	4.28	(4.29 ± 0.01)	0.16
(4.35 ± 0.2)	4.30	4.31	4.33	(4.31 ± 0.02)	0.84
(3.96 ± 0.2)	3.97	3.95	3.98	(3.97 ± 0.02)	0.34

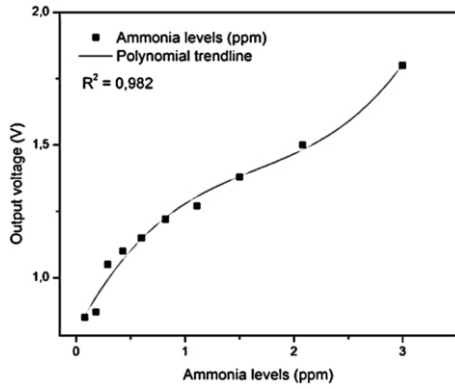


Figure 6. Graph of ammonia concentration testing against output voltage (V) of the sensor

$$NH_3 = (-6.08955 \times V^3) + (25.42202 \times V^2) + (-30.90137 \times V) + 11.77048 \quad (9)$$

The test was then performed again on the sensor that had been programmed to ensure the level of accuracy and precision in accordance with the samples whose ammonia concentration was known. The results of the MQ-135 sensor test for measuring ammonia concentration are shown in Table 3, with three repeat tests. The average error of the sensor is 9.51%, accuracy is 90.49%, and precision is 93.77%.

4) *DS18B20 Temperature Sensor*: The DS18B20 sensor can be used to measure temperature parameters in aquatic environments because it is waterproof. The sensor will be compared with the HTC-2 temperature-measuring device. The supporting media used is a 600 ml Erlenmeyer flask for filling with 400 ml of water, and an electric stove that functions as a heater for the water temperature. The results of the temperature sensor test, with three repeat tests, are shown in Table 4. The average error is 1.18%, accuracy is 98.79%, and precision is 99.71%.

C. Data Collection

Data collection was carried out by implementing the system in the koi fish farming place. The realized system is shown in Figure 7.

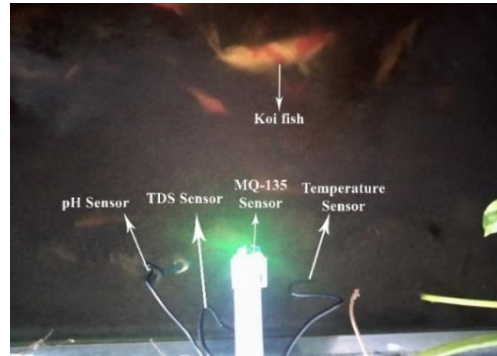


Figure 7. Display of the 4 parameter sensors

Table 3. Results of the testing of the polynomial equation of power 3 for ammonia (NH<sub>3</sub>) concentration on the analog MQ-135 sensor

Ammonia sample (ppm)	Ammonia Concentration (ppm) on the Analog MQ-135 Sensor			Average	Error (%)
	1	2	3		
0.08	0.07	0.08	0.06	(0.07 ± 0.01)	13.58
0.18	0.19	0.17	0.18	(0.18 ± 0.01)	4.55
0.29	0.35	0.27	0.40	(0.34 ± 0.07)	21.84
0.43	0.39	0.48	0.55	(0.47 ± 0.08)	16.39
0.60	0.67	0.71	0.59	(0.66 ± 0.06)	10.56
0.82	0.85	0.81	0.87	(0.84 ± 0.03)	3.75
1.11	1.15	1.18	1.16	(1.16 ± 0.02)	5.28
1.50	1.52	1.55	1.60	(1.56 ± 0.04)	3.78
2.08	2.28	2.20	2.05	(2.18 ± 0.12)	5.67
3.00	2.80	3.46	3.21	(3.16 ± 0.33)	9.67

Table 4. Results of temperature measurement of the DS18B20 sensor against the HTC-2

HTC-2 (°C)			DS18B20 (°C)			Average		Error (%)
1	2	3	1	2	3	HTC-2	DS18B20	
28.40	28.50	28.70	28.60	28.80	29.00	(28.53±0.50)	(28.80±0.20)	0.93
29.20	29.50	29.60	29.50	29.90	29.90	(29.43±0.50)	(29.77±0.23)	0.79
30.50	30.50	30.50	30.80	30.90	30.90	(30.50±0.50)	(30.87±0.06)	1.20
31.50	31.60	31.50	31.80	31.90	31.90	(31.53±0.50)	(31.87±0.06)	1.06
32.40	32.60	32.50	32.80	33.00	32.90	(32.50±0.50)	(32.90±0.10)	1.23
33.30	33.40	33.50	33.80	33.80	33.90	(33.40±0.50)	(33.83±0.06)	1.30
34.30	34.50	34.50	34.80	34.90	34.90	(34.43±0.50)	(34.87±0.06)	1.26
35.50	35.60	35.50	36.10	36.00	35.90	(35.53±0.50)	(36.00±0.10)	1.31
36.30	36.50	36.30	36.90	37.00	36.70	(36.37±0.50)	(36.87±0.15)	1.37
37.50	37.20	37.30	38.20	37.60	37.70	(37.33±0.50)	(37.83±0.32)	1.34



Figure 8. Display of water quality parameters on the LCD



Figure 9. Results of water quality parameter measurement on the Blynk IoT interface application

The dimensions of the aquarium are (500×130×100) cm. The measurement results are displayed on the LCD in real-time, the Blynk IoT interface application, and stored in an Excel format (.xlsx). The display of the measurement results in the first 5 minutes on the LCD is shown in Figure 8.

The Blynk IoT interface application can run after the NodeMCU ESP32 is connected to the Wi-Fi. The Blynk IoT interface application can also automate notifications in the Android system that can inform the monitoring system user if there are values from the parameters outside the established limits of the parameters that are in line with the optimal water conditions for koi breeding. The data collection process in this study was conducted with time allocations in the morning, afternoon, and late afternoon in a 5-minute interval, as in [18]. The results of the system measurement in the Blynk IoT interface can be seen in Figure 9.

Figure 9 shows the results of the measurement system, indicating that the Blynk IoT interface application can perform readings of data sent from the ESP32 to the Blynk server and displayed in the form of streaming data in the Blynk IoT application. The measurement of three different times, namely morning, afternoon, and late afternoon, in the form of a graph, is shown in Figure 10.

Figure 10 shows the monitoring data from four parameters at three different times: morning (07.00-08.00), afternoon (13.00-14.00), and late afternoon (16.00-17.00).

The monitoring data indicate that the water quality is in a good state, as the water quality parameters are still within the normal range for the koi habitat. However, a temperature below 30 °C is not optimal for the koi environment.

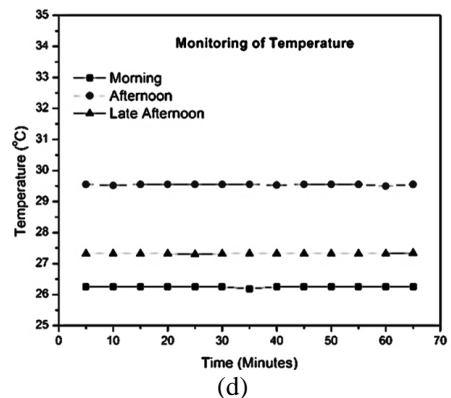
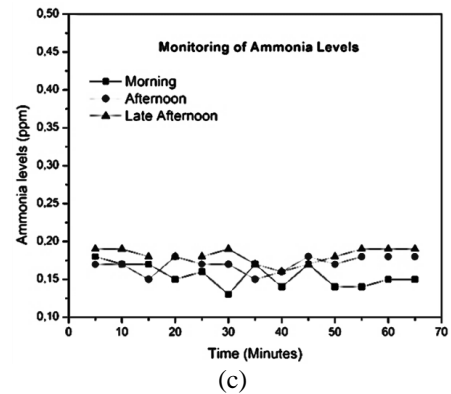
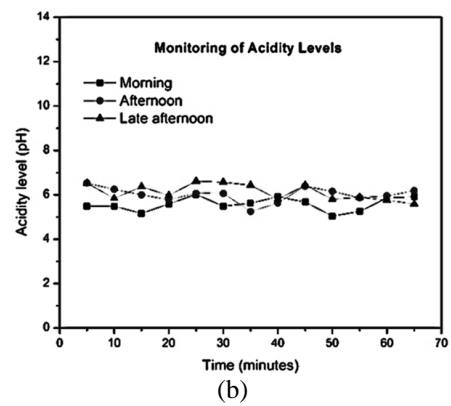
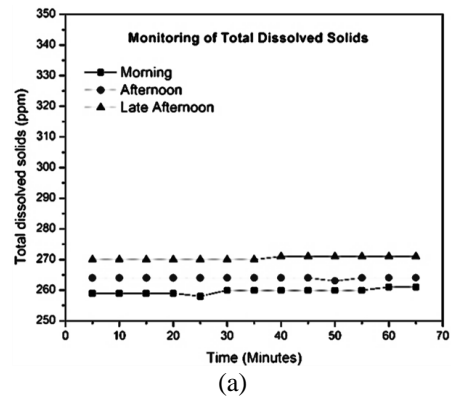


Figure 10. Data monitoring of water quality parameters, (a) TDS sensor, (b) pH sensor, (c) ammonia sensor, (d) temperature sensor



## V. CONCLUSIONS

Based on the test results and discussions that have been conducted, it can be concluded that the water quality monitoring system, which measures four parameters, namely total dissolved solids, acidity level, ammonia level, and temperature, using IoT technology connected to a smartphone through the Blynk application, has been successfully implemented. The device has been calibrated with an error of 5.08% for the TDS sensor, 1.12% for the pH sensor, 9.51% for the ammonia sensor, and 1.22% for the temperature sensor. The low error values indicate that the design of the monitoring system can be used in koi fish farming aquariums.

## REFERENCES

- [1] E. E. Barus, R. K. Pingak, and A. C. Louk, "Otomatisasi Sistem Kontrol pH dan Informasi Suhu Pada Akuarium Menggunakan Arduino Uno dan Raspberry Pi 3," *J. Fis. Fis. Sains dan Apl.*, vol. 3, no. 2, pp. 117–125, 2018, doi: 10.35508/fisa.v3i2.612.
- [2] I. G. H. Putrawan, P. Rahardjo, and I. G. A. P. R. Agung, "Sistem Monitoring Tingkat Kekeruhan Air dan Pemberi Pakan Otomatis Pada Kolam Budidaya Ikan Koi Berbasis NodeMCU," *Maj. Ilm. Teknol. Elektro*, vol. 19, no. 1, p. 1, 2019, doi: 10.24843/mite.2020.v19i01.p01.
- [3] H. Azmi, D. R. Indriyanti, and N. Kariada, "Identifikasi Ektoparasit pada Ikan Koi (*Cyprinus carpio* L) di Pasar Ikan Hias Jurnatan Semarang," *Unnes J. Life Sci.*, vol. 2, no. 2, pp. 64–70, 2013, ISSN 2252-6277.
- [4] B. Artono and R. G. Putra, "Penerapan Internet Of Things (IoT) Untuk Kontrol Lampu Menggunakan Arduino Berbasis Web," *J. Teknol. Inf. dan Terap.*, vol. 5, no. 1, pp. 9–16, 2019, doi: 10.25047/jtit.v5i1.73.
- [5] A. Setiawan and A. I. Purnamasari, "Pengembangan Smarth Home Dengan Microcontrollers ESP23 Dan MC-38 Door Magnetic Switch Sensor Berbasis Internet of Things (IoT) Untuk Meningkatkan Deteksi Dini Keamanan Perumahan," *J. Resti (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 3, p. Halaman 541-457, 2019, ISSN 2580-0760.
- [6] Y. Irawan, A. Febriani, R. Wahyuni, and Y. Devis, "Water Quality Measurement and Filtering Tools using Arduino Uno, PH Sensor and TDS Meter Sensor," *J. Robot. Control*, vol. 2, no. 5, pp. 357–362, 2021, doi: 10.18196/jrc.25107357.
- [7] M. Martani, "Perancangan dan Pembuatan Sensor TDS pada Proses Pengendapan CaCO<sub>3</sub> dalam Air dengan Metode Pelucutan Elektron dan Medan Magnet," vol. 17, no. 3, 2014, ISSN : 1410-9662.
- [8] R. Afrianita, T. Edwin, and A. Alawiyah, "Analisis Intrusi Air Laut dengan Pengukuran Total Dissolved Solids (TDS) Air Sumur Gali di Kecamatan Padang Utara," *J. Dampak*, vol. 14, no. 1, p. 62, 2017, doi: 10.25077/dampak.14.1.62-72.2017.
- [9] Suryono, *Teknologi Sensor : Konsep Fisis dan Teknik Akuisisi Data Berbasis Mikrokontroler 32 Bit ATSAM3X8E (ARDUINO DUE)*, 1st ed. Semarang: Undip Press, 2018.
- [10] N. Sitorus, "Pendeteksi pH Air Menggunakan Sensor pH Meter V1.1 Berbasis Arduino Nano," *Fisika*, Universitas Sumatera Utara, Medan, 2017.
- [11] A. A. Rosa, B. A. Simon, and K. S. Lieanto, "Sistem Pendeteksi Pencemar Udara Portabel Menggunakan Sensor MQ-7 dan MQ-135," *Ultim. Comput.*, vol. XII, no. 1, pp. 23–28, 2020, ISSN 2355-3286.
- [12] A. T. Ajiboye, J. F. Opadiji, A. O. Yusuf, and J. O. Popoola, "Analytical determination of load resistance value for MQ-series gas sensors: MQ-6 as case study," *Telkomnika (Telecommunication Comput. Electron. Control)*, vol. 19, no. 2, pp. 575–582, 2021, doi: 10.12928/TELKOMNIKA.v19i2.17427.
- [13] M. Yusro and A. Diamah, *Sensor & Transducer, Teori dan Aplikasi*. Jakarta: Fakultas Teknit Universitas Negeri Jakarta, 2019.
- [14] O. Bondarenko, S. Kininmonth, and M. Kingsford, "Under Water Sensor Network, Oceanography and Plankton Assemblages," *IEEE*, vol. 3, no. 1, pp. 657–662, 2007, doi: 10.1109/ISSNIP.2007.4496921.
- [15] S. Alam, A. Y. M. A. Kadir, and E. Elihami, "Sistem Otomatis Sirkulasi Udara pada Tambak Udang," *JUTKEL J. Telekomun. Kendali dan List.*, vol. 2, no. 1, pp. 1–10, 2020, ISSN 2721-9372.
- [16] R. Pratiwi, "Penelitian Sumber Panas dengan Metode Tomografi Menggunakan Sensor Thermometer Digital DS18B20," *Fisika*, Universitas Indonesia, Depok, 2009.
- [17] L. D. Jones and A. F. Chin, *Electronics Instruments and Measurements*. Upper Saddle River: Prentice-Hall, 1991.
- [18] E. Rohadi *et al.*, "Sistem Monitoring Budidaya Ikan Lele Berbasis Internet Of Things Menggunakan Raspberry Pi," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 6, p. 745, 2018, doi: 10.25126/jtiik.2018561135.

## Acknowledgment

We sincerely thank the experts and colleagues who have committed to reviewing the Jurnal Listrik, Instrumentasi, dan Elektronika Terapan Volume 4 No. 1 of 2023. The experts and colleagues who have contributed to the journal are listed below.

<b>Name</b>	<b>Affiliation</b>
Brahmantya Aji Pramudita, S.Si., M.Eng.	Universitas Telkom
Dr. Sahirul Alam, S.T., M.Eng.	Universitas Gadjah Mada
Fakih Irsyadi, S.T., M.T.	Universitas Gadjah Mada
Ir. Prapto Nugroho, S.T., M.Eng., D.Eng., IPM.	Universitas Gadjah Mada
Muhammad Khairuddin, S.ST., M.Sc.	Politeknik Negeri Malang
Rendra Dwi Firmansyah, S.Si., M.Eng.	Lembaga Ilmu Pengetahuan Indonesia
Sinta Uri El Hakim, S.ST., M.Eng.	Universitas Gadjah Mada