

Peningkatan Kestabilan Quadrotor menggunakan Kendali Linear Quadratic Regulator dengan Kompensasi Integrator dalam Mempertahankan Posisi

Oktaf Agni Dhewa¹, Faisal Fajri Rahani²

¹ Program Studi Sarjana Terapan Teknik Elektronika, Universitas Negeri Yogyakarta, Indonesia

² Program Studi Informatika, Universitas Ahmad Dahlan, Indonesia

INFORMASI ARTIKEL

Riwayat Artikel:

Dikirimkan 07 November 2022

Direvisi 14 November 2022

Diterima 30 November 2022

Kata Kunci:

UAV Quadrotor;
Position Holding Stability;
Optimal Control;
Integrator

Penulis Korespondensi:

Oktaf Agni Dhewa
Universitas Negeri Yogyakarta,
Jl. Colombo Yogyakarta No.1,
Karang Malang, Caturtunggal,
Kec. Depok, Kabupaten
Sleman, Daerah Istimewa
Yogyakarta, Indonesia
Email:
oktafagnidhewa@uny.ac.id

ABSTRACT / ABSTRAK

The quadrotor's ability to maintain position is a major requirement for the completion of various current missions. However, the large steady-state error (SSE) and multiple overshoots due to environmental disturbances cause flight instability. This condition makes Quadrotor unable to complete the mission optimally. Therefore, in this study applying a linear quadratic regulator control method, this research contributes to the addition of integrator compensation in handling the translational movement of the quadrotor. System model design testing is carried out by comparing quadrotor control using the LQR method without an Integrator and LQR with an Integrator. The value of $R=1$ for all states and $Q_x=0.87$, $Q_y=124.6$, $Q_{(v_x)}=1.77$, $Q_{(v_y)}=124.6$ and $Ki_x=0.004$, $Ki_y=0.002$ makes the SSE tendency that occurs 0.10 meters for the x-axis and -0.28 for the y-axis, while the multi-overshoot that occurs is 0.41 m for the maximum deviation and -1.35 m for the minimum deviation on the x-axis and 0.40 m maximum deviation and 0.47 m minimum deviations on the y axis. The test results show that the LQR control method with Integrator compensation is able to minimize and improve SSE and multiple overshoots that occur in quadrotor flights. In addition, it is able to significantly increase accuracy to 100% from 71.38% and precision to 37.71% from 35.91%.

Kemampuan quadrotor dalam mempertahankan posisi menjadi kebutuhan utama untuk penyelesaian berbagai misi saat ini. Namun, besarnya steady state error (SSE) dan multiple overshoot karena gangguan lingkungan menyebabkan ketidakstabilan gerak terbang. Kondisi tersebut menjadikan quadrotor tidak mampu menyelesaikan misi secara optimal. Maka dari itu, pada penelitian ini menerapkan sebuah metode kendali Linear Quadratic Regulator penelitian ini memiliki kontribusi dengan penambahan kompensasi Integrator dalam menangani pergerakan translasi quadrotor. Pengujian desain model sistem, dilakukan dengan membandingkan antara pengendalian quadrotor menggunakan metode LQR tanpa Integrator dan LQR dengan Integrator. Nilai $R=1$ untuk semua *state* serta $Q_x=0.87$; $Q_y=124.6$; $Q_{(v_x)}=1.77$; $Q_{(v_y)}=124.6$ dan $Ki_x=0.004$; $Ki_y=0.002$ menjadikan kecenderungan SSE yang terjadi sebesar 0,10 m untuk sumbu x dan -0,28 m untuk sumbu y, sedangkan multi overshoot yang terjadi sebesar 0,41 meter simpangan maksimal dan -1,35 m simpangan minimal pada sumbu x serta 0,40 m simpangan maksimal dan 0,47 meter simpangan minimal pada sumbu y. Hasil pengujian tersebut menunjukkan bahwa metode LQR dengan kompensasi Integrator mampu meminimalkan dan memperbaiki SSE maupun multiple overshoot yang terjadi pada penerbangan quadrotor. Selain itu juga mampu meningkatkan akurasi secara signifikan sebesar 100% dari 71,38% serta presisi sebesar 37,71% dari 35,91%.

This work is licensed under a [Creative Commons Attribution-Share Alike 4.0](https://creativecommons.org/licenses/by-sa/4.0/)



Sitasi Dokumen ini:

O. A. Dhewa and F. F. Rahani, "Peningkatan Kestabilan Quadrotor menggunakan Kendali Linear Quadratic Regulator dengan Kompensasi Integrator dalam Mempertahankan Posisi," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 4, no. 2, pp. 62-75, 2022, DOI: [10.12928/biste.v4i2.6808](https://doi.org/10.12928/biste.v4i2.6808).

1. PENDAHULUAN

Revolusi industri 4.0 telah menuntut kreativitas dan inovasi yang mengarah pada sistem cerdas maupun otomasi. Satu di antara teknologi yang berkembang pesat adalah *Unmanned Aerial Vehicle* (UAV). UAV sendiri merupakan pesawat terbang tanpa awak yang memiliki aksesibilitas penggunaan yang mudah dan efisien. Hal ini tentunya telah menarik perhatian para penggiat invensi di bidang industri, sipil maupun militer, dengan memanfaatkan UAV sebagai wahana bantu untuk menyelesaikan beragam misi. Seperti halnya *aerial surveillance, reconnaissance, transportasi barang, taxi drone* dan banyak lainnya [1].

Berbagai tipe UAV yang telah mengalami peningkatan baik pada pola bentuk fisik maupun sistem penerbangan untuk mencapai target operasional misi tertentu. Satu di antara UAV tersebut adalah *quadrotor* yang mana tipe UAV ini memanfaatkan 4 buah aktuator motor sebagai penggerak terbang. Lebih luasnya, *quadrotor* merupakan UAV yang memiliki kemampuan lepas landas dan mendarat secara vertikal (VTOL) bahkan pada area yang terbatas [2]. Selain itu, *quadrotor* juga memiliki kecepatan terbang yang konstan sehingga dapat melayang (*hovering*) di lokasi tertentu dengan gerak stasionernya [3].

Melihat lebih mendalam mengenai arsitektur sistem *quadrotor*, tentunya potensi kemampuan yang dimiliki *quadrotor* tidak serta merta ada dengan sendirinya melainkan dibangun melalui keterlibatan beberapa infrastruktur sistem yang cukup kompleks. Semua infrastruktur sistem *quadrotor* yang ada, terdapat satu sistem yang berperan penting sebagai faktor penentu dalam melakukan proses penerbangan. Sistem itu adalah sistem pengendalian gerak terbang *quadrotor*. Tanpa adanya sistem kendali yang terkelola dengan baik, maka *quadrotor* tidak mampu mempertahankan gerak rotasi maupun translasi sehingga menyebabkan jatuhnya pesawat.

Seperti yang telah sedikit disinggung sebelumnya, pengendalian gerak *quadrotor* dibedakan menjadi 2 kategori yaitu pengendalian gerak rotasi dan gerak translasi. Mekanisme alur pengendalian gerak *quadrotor* dimulai dari gerak rotasi dan diikuti gerak translasi dengan pengendalian yang diakomodir oleh sebuah metode kendali tertentu untuk mencapai keadaan stabil. Dengan mengacu beberapa penelitian yang dilakukan sebelumnya yang berfokus pada pengendalian sikap gerak rotasi *quadrotor*, wahana mampu memiliki kestabilan terbang yang baik dengan menerapkan metode sistem kendali *Linear Quadratic Regulator* (LQR) *full state feedback*. Penelitiannya menjelaskan bahwa metode LQR memberikan karakteristik regulator dalam sinyal kendali yaitu dengan memaksa sinyal kendali menuju ke nilai mendekati 0. Hal ini membuat sistem mampu mengatasi kesalahan melalui respon yang cepat. Pernyataan tersebut diperkuat dengan bukti hasil eksperimen penelitian yang diperoleh, dimana *quadrotor* mampu memberikan kesalahan minimal yang terjadi pada sistem dengan respon dari ketiga sudut orientasi baik *roll, pitch* dan *yaw* dalam mencapai keadaan tunaknya kurang dari 1 detik [4]. Selain itu juga, dibandingkan dengan pengendalian menggunakan kendali PID, kendali LQR memiliki kemampuan yang lebih baik dan *power full* dalam performansi indeks seperti pengelolaan overshoot, response time dan control precision pada kestabilan pergerakan pesawat terbang tanpa awak [5].

Namun, merujuk pada literatur penelitian lain mengenai kinerja dari kendali LQR, mengindikasikan bahwa metode ini kurang cukup baik dalam mengakomodasi gerak translasi sebuah objek dengan simpangan rata-rata yang masih besar terjadi saat sistem telah mencapai keadaan tunaknya [6]. Mengacu permasalahan yang telah disinggung, sebuah UAV tipe *quadrotor* tidak akan mampu menyelesaikan misinya dengan optimal secara akurat maupun presisi ketika masih mengalami penyimpangan dalam penerbangan. Penyimpangan sistem yang semakin besar akan diiringi dengan hasil *steady state error* atau simpangan rata-rata yang juga semakin besar. Di samping itu, adanya keadaan *steady state error* yang divergen dengan intensitas sesering mungkin akan memicu sistem untuk mengalami *multiple overshoot* yang mengakibatkan keadaan penerbangan tidak hanya bergeser tetapi juga mempengaruhi pesawat berubah haluan.

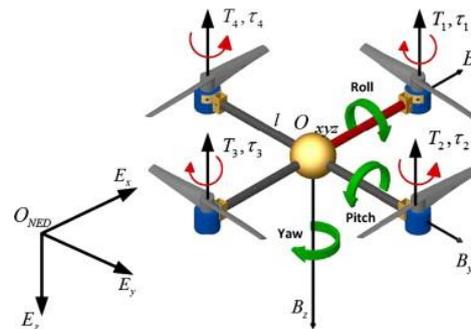
Oleh karena itu, agar *quadrotor* mampu menyelesaikan misi penerbangannya, permasalahan utama yang perlu diselesaikan sebelumnya adalah kestabilan yang tinggi *quadrotor* dalam mempertahankan posisinya atau *position holding*. Mempertahankan posisi penerbangan yaitu keadaan sikap *quadrotor* terbang di posisi titik tertentu pada sumbu x dan y bumi. Penelitian ini menggunakan sistem kendali LQR dalam menangani gerak rotasi pesawat yang tentunya terlibat dalam pergerakan translasi pesawat, dimana permasalahan *steady state error* dan *multiple-overshoot* yang kemungkinan besar terjadi akan diminimalkan serta diakomodasi oleh penanganan kompensasi *Integrator* [7]. Penelitian ini berkontribusi untuk memperbaiki masalah permasalahan *steady state error* dan *multiple-overshoot* dengan penggunaan integrator. Sehingga pergerakan translasi khususnya dalam kestabilan mempertahankan posisi pesawat mampu ditingkatkan dengan baik.

2. METODE

2.1. Pemodelan *Quadrotor*

Penelitian ini menggunakan *quadrotor* dengan konfigurasi terbang X yang memanfaatkan 4 buah motor yang terbagi menjadi 2 motor depan dan 2 motor belakang. Keempat motor tersebut dalam melakukan

pergerakan ke depan dan ke belakang dianggap sejajar sumbu bujur bumi, sedangkan pergerakan ke kanan dan kiri dianggap sejajar dengan sumbu lintang bumi. Berdasarkan prinsip tersebut, persamaan model dari *quadrotor* dengan konfigurasi terbang X dapat didasarkan pada persamaan *Newton – Euler*. Persamaan yang memerlukan dua kerangka acuan yaitu *North – East – Down* (NED) atau *North – West – Up* (NWU) *inertial* dan *body fixed frame*, dimana penelitian ini menggunakan kerangka acuan NED. Kerangka acuan inersia NED memiliki dua axis N dan E yang sejajar dengan arah utara dan timur serta satu axis D yang diarahkan ke bawah menuju pusat bumi, sehingga hukum pertama *Newton* akan berlaku pada kerangka acuan ini. Kerangka acuan lain yaitu *body fixed frame* ini ditunjukkan oleh arah sumbu x , y dan z yang memiliki titik pusat yang bertepatan dengan pusat gravitasi pada badan *quadrotor* [8]. Kedua kerangka acuan yang ditunjukkan pada Gambar 1.



Gambar 1. *Quadrotor* dengan kerangka acuan NED [8]

Sikap-sikap *quadrotor* yang berhubungan dengan rotasi terdapat 6 *state* yaitu: *Euler angle* [ϕ θ ψ] (*Roll – Pitch – Yaw*) dan kecepatan sudut disekitar tiap sumbu *body fixed frame* [p q r], untuk 6 *state* yang lain yaitu posisi dari *center of gravity* (CoG) [x y z] dan masing – masing komponen kecepatan linear [\dot{x} \dot{y} \dot{z}], relatif terhadap *body fixed frame*. Oleh karena itu, *quadrotor* memiliki 12 *state* yang menggambarkan 6 *degrees of freedom* (DOF) [9].

Pemodelan *quadrotor* terdiri atas pergerakan rotasi 3 sumbu (*roll*, *pitch*, *yaw*), dan pergerakan translasi 3 sumbu x , y , z bumi (*earth*) terhadap sumbu z badan (*body*) dengan referensi pergerakan translasi *quadrotor* yang digunakan dalam pemodelan adalah *earth inertial frame* (*E-frame*) [E_x E_y E_z] dan *body inertial frame* (*B-frame*) [B_x B_y B_z].

Selanjutnya, untuk pergerakan rotasi ϕ (*roll*) merupakan gerakan *quadrotor* mengelilingi sumbu x *body*, pergerakan rotasi θ (*pitch*) merupakan gerakan *quadrotor* mengelilingi sumbu y *body* dan pergerakan rotasi ψ (*yaw*) merupakan gerakan *quadrotor* mengelilingi sumbu z *body*.

Melalui pendekatan *Newton-Euler* dan Hukum *Newton* II, *quadrotor* memiliki gaya total yang ditentukan oleh gaya angkat keempat rotornya yang dapat dikalkulasi dengan Persamaan (1) [10].

$$F_T = u_1 = \sum_{i=1}^4 f_i \quad (1)$$

Pergerakan translasi sumbu x , y dan z sumbu bumi terhadap sumbu z *body frame* dari *quadrotor* dapat diperoleh dari Persamaan (2) sampai Persamaan (4)

$$\ddot{x} = \frac{F_T \cdot \theta}{m} \quad (2)$$

$$\ddot{y} = \frac{F_T \cdot \phi}{m} \quad (3)$$

$$\ddot{z} = \frac{F_T}{m} + g \quad (4)$$

Lebih lanjut, untuk pemodelan gerak rotasi *quadrotor* dapat digunakan hukum *Euler* yang keduanya merupakan hubungan dari torsi (τ) dan percepatan sudut (α) serta hubungan antara momentum angular (L) dan kecepatan putar (ω) didalamnya untuk setiap sumbu yang dialami sebuah pusat massa seperti ditunjukkan pada Persamaan (5) [11].

$$\tau = I \cdot \alpha + Lx\omega \sin \beta \quad (5)$$

Dengan β merupakan sudut yang dibentuk oleh momentum angular dan kecepatan putar. Sudut yang terbentuk pada gerak rotasi *quadrotor* sangat kecil dan tidak berpengaruh, sehingga sudut tersebut dapat diabaikan dengan memberikannya nilai nol. Dari hal tersebut menjadikan **Persamaan (5)** hanya dipengaruhi hubungan torsi, percepatan sudut dan inersia seperti pada **Persamaan (6)** dan **Persamaan (7)**.

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \dot{\omega} \tag{7}$$

Hubungan τ (torsi) dengan gaya angkat atau (*thrust*) dapat dituliskan dengan **Persamaan (8)**.

$$\begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} F_\phi \\ F_\theta \\ F_\psi \end{bmatrix} r \tag{8}$$

sehingga model gerak rotasi pada sumbu x , y dan z pada sumbu bumi atau disebut dengan sudut orientasi *roll*, *pitch* dan *yaw* dapat diperoleh melalui **Persamaan (9)** sampai **Persamaan (11)**.

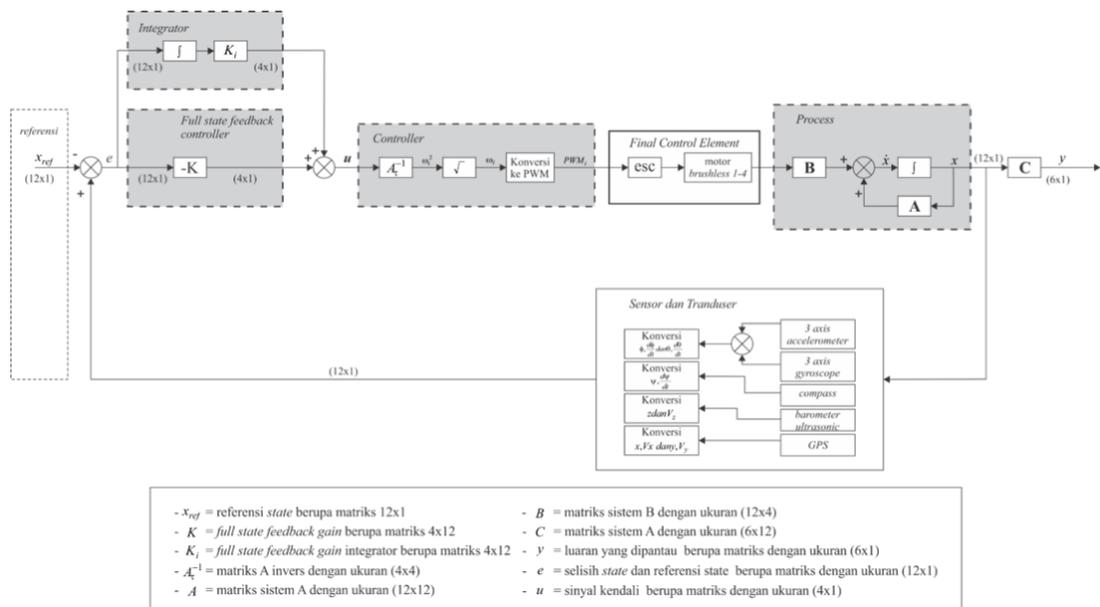
$$\ddot{\phi} = \frac{\tau_\phi}{I_{xx}} \tag{9}$$

$$\ddot{\theta} = \frac{\tau_\theta}{I_{yy}} \tag{10}$$

$$\ddot{\psi} = \frac{\tau_\psi}{I_{zz}} \tag{11}$$

2.2. Rancangan Sistem Kendali

Anatomi sistem kendali LQR dengan kompensasi Integrator pada pengendalian mempertahankan posisi ditunjukkan melalui diagram blok pada **Gambar 2**.



Gambar 2. Diagram blok sistem kendali *quadrotor* menggunakan metode LQR dan kompensasi Integrator

Sistem kendali ini menggunakan konsep *quadratic* dengan 12 *state* pengendalian. *State-state* tersebut diperoleh dari pemodelan *quadrotor* yang ditunjukkan pada **Persamaan (12)** dan **Persamaan (13)** [12].

$$\begin{bmatrix} \dot{x} \\ \dot{V}_x \\ \dot{y} \\ \dot{V}_y \\ \dot{z} \\ \dot{V}_z \\ \dot{\phi} \\ \dot{\omega}_\phi \\ \dot{\theta} \\ \dot{\omega}_\theta \\ \dot{\psi} \\ \dot{\omega}_\psi \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{u_1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{u_1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_A \begin{bmatrix} x \\ V_x \\ y \\ V_y \\ z \\ V_z \\ \phi \\ \omega_\phi \\ \theta \\ \omega_\theta \\ \psi \\ \omega_\psi \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix}}_B \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (12)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_C \begin{bmatrix} x \\ V_x \\ y \\ V_y \\ z \\ V_z \\ \phi \\ \omega_\phi \\ \theta \\ \omega_\theta \\ \psi \\ \omega_\psi \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_D \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (13)$$

Berdasarkan [Persamaan \(12\)](#) *state space*, pada matriks konstan B terdapat komponen I_{xx} , I_{yy} dan I_{zz} yang merupakan inersia sumbu x , y , z *quadrotor*. Setiap inersia tersebut dihitung melalui [Persamaan \(14\)](#) sampai [Persamaan \(16\)](#).

$$I_{xx} = \sum_{j=1}^n (I_{G_{xxj}} + m_j(y_j^2 + z_j^2)) \quad (14)$$

$$I_{yy} = \sum_{j=1}^n (I_{G_{yyj}} + m_j(x_j^2 + z_j^2)) \quad (15)$$

$$I_{zz} = \sum_{j=1}^n (I_{G_{zzj}} + m_j(x_j^2 + y_j^2)) \quad (16)$$

Dengan $I_{G_{xxj}}$, $I_{G_{yyj}}$ dan $I_{G_{zzj}}$ merupakan inersia pada posisi x , y dan z dari titik acuan benda ke- j wahana, x_j , y_j dan z_j merupakan jarak *center of gravity* terhadap setiap benda ke- j bagian dari wahana ke poros putar wahana pada sumbu x , y dan z , sedangkan m_j adalah massa dari masing-masing benda bagian wahana ke- j [13].

Perhitungan *thrust* dan *torque*, terdapat konstanta yang berhubungan dengan masing-masing gaya. Konstanta tersebut adalah konstanta b yang merupakan konstanta *thrust* dan konstanta k yang merupakan konstanta *drag*. Konstanta b dan k dapat dijabarkan melalui perhitungan pada [Persamaan \(17\)](#) dan [Persamaan \(18\)](#) [14].

$$b = C_T \rho A R^2 \quad (17)$$

$$k = C_D \rho A R^3 \quad (18)$$

dengan ρ adalah densitas/massa jenis udara (1,184 kg/m³), A adalah luas area yang terbentuk dari baling-

baling ketika berputar, R adalah jari-jari *quadrotor* yang diukur dari titik tengah massa *quadrotor* hingga titik tengah *motor brushless*.

Hubungan gaya dengan parameter konstanta dan kecepatan putar untuk setiap sumbunya, maka akan didapatkan persamaan untuk masukkan proses kendali LQR yaitu ditunjukkan pada [Persamaan \(19\)](#).

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} F_T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ bl((\omega_3^2 + \omega_4^2) - (\omega_1^2 + \omega_2^2)) \\ bl((\omega_1^2 + \omega_4^2) - (\omega_2^2 + \omega_3^2)) \\ k(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (19)$$

dengan ω_1 adalah kecepatan sudut rotor depan kanan, ω_2 adalah kecepatan sudut rotor belakang kanan, ω_3 adalah kecepatan sudut rotor belakang kiri dan ω_4 adalah kecepatan sudut rotor depan kiri.

Kendali LQR merupakan pengembangan metode HJB (Hamilton-Jacobi-Bellman). Tujuan kendali LQR yaitu mencari *gain* matriks \mathbf{K} sebagai *feedback gain* untuk mengoptimalkan sistem kendali. Pemilihan kendali LQR sendiri dikarenakan metode kendali ini meminimalisir *cost function*. Kendali LQR pada sistem *quadrotor* diterapkan pada Persamaan model *state space*. Kendali LQR bekerja berdasarkan matriks \mathbf{Q} dan \mathbf{R} yang tepat untuk dapat mempertahankan kestabilan terbang *quadrotor*. Proses pembobotan matriks \mathbf{Q} dan \mathbf{R} dilakukan untuk mendapatkan respon kendali sesuai dengan spesifikasi kendali yang diinginkan [15].

Nilai \mathbf{Q} berbanding lurus dengan nilai variabel bantu dari persamaan *Riccati* (\mathbf{P}) untuk penentuan nilai \mathbf{K} . Nilai \mathbf{P} ini sebanding dengan penguatan nilai \mathbf{K} , sehingga semakin besar nilai \mathbf{Q} maka semakin besar pula nilai \mathbf{K} . Penalaan \mathbf{Q} dan \mathbf{R} diperoleh dari Persamaan LQR dimulai dari nilai 1, kemudian akan ditambah atau dikurangi sesuai dengan kebutuhan sistem, namun konstanta \mathbf{Q} dan \mathbf{R} memiliki fungsi yang berlawanan, apabila nilai \mathbf{R} besar, maka *feedback* menjadi tidak terlalu mempengaruhi sistem, oleh karena itu penalaan hanya dilakukan pada matriks \mathbf{Q} , di mana matriks \mathbf{R} diberi nilai 1 agar nilai \mathbf{R} tidak mempengaruhi kendali [16]. Hubungan matriks \mathbf{Q} untuk setiap masing-masing *state* yang dikendalikan ditunjukkan [Persamaan \(20\)](#).

$$\mathbf{Q} = \begin{bmatrix} Q_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_{v_x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Q_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{v_y} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_{v_z} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Q_\phi & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{\omega_\phi} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{\omega_\theta} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_\psi & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Q_{\omega_\psi} \end{bmatrix} \quad (20)$$

Metode kendali LQR merupakan metode yang dikhususkan sebagai regulator, dimana bekerja pada pemodelan sistem linier yang ditunjukkan pada [Persamaan \(21\)](#) dan [Persamaan \(22\)](#).

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (21)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \quad (22)$$

dengan sinyal kendali \mathbf{u} dapat dihitung melalui [Persamaan \(23\)](#) [17],

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad (23)$$

dan menjadikan $\dot{\mathbf{x}}$ dalam bentuk keadaan tertutup memiliki [Persamaan \(24\)](#).

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} \quad (24)$$

dengan \mathbf{x} merupakan *state* sistem, \mathbf{u} adalah sinyal masukkan kendali, \mathbf{y} adalah sinyal keluaran kendali, \mathbf{A} adalah matriks yang memetakan hubungan antara *state* dan turunan *state*, \mathbf{B} adalah matriks yang memetakan hubungan antara sinyal kendali dan turunan *state*, \mathbf{C} adalah matriks yang memetakan hubungan antara sinyal kendali dan keluaran sinyal kendali dan \mathbf{K} adalah *full state feedback gain*.

Masalah yang muncul dalam sistem kendali ini adalah bagaimana jika pengguna atau sistem otonom ingin mengubah referensi dari *state*. Pengubahan referensi dengan besaran fisik dari *state* tidak sama dengan besaran referensi masukan dapat dilakukan menggunakan referensi *state* (x_{ref}).

Referensi *state* dapat berguna dalam mempertahankan keadaan wahana dengan posisi tertentu seperti pada posisi ketinggian maupun melakukan pergerakan translasi terhadap sumbu x dan y bumi di posisi tertentu. Adanya referensi *state* mengubah [Persamaan \(21\)](#) dengan melibatkan [Persamaan \(23\)](#) menjadi [Persamaan \(25\)](#).

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{B} \left(-\mathbf{K}(\mathbf{x} - \mathbf{x}_{ref}) \right) \quad (25)$$

Berdasarkan perhitungan sinyal kendali $\mathbf{x} - \mathbf{x}_{ref}$ merupakan sebagai kesalahan terhadap referensi yang diinginkan yang dinamakan *error*. *Error* membuat sistem akan menghasilkan *steady state error* atau *offset* terhadap referensi maupun *overshoot* yang semakin besar karena perubahan nilai keadaan yang berubah secara langsung khususnya untuk mempertahankan gerak translasi sumbu x , y dan z wahana terhadap sumbu bumi. Hal tersebut dapat diatasi dengan mengintegrasikan *error* yang terjadi dan dikalikan dengan sebuah *gain* atau yang disebut komponen *Integrator* [12]. Dalam kendali LQR dengan referensi *state* dapat ditambahkan komponen *Integrator* yang mengubah [Persamaan \(25\)](#) menjadi [Persamaan \(26\)](#).

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{B} \left(-\mathbf{K}(\mathbf{x} - \mathbf{x}_{ref}) + \mathbf{K}_i \int (\mathbf{x} - \mathbf{x}_{ref}) \right) \quad (26)$$

Dengan \mathbf{u} merupakan sinyal kendali yang menjadi masukan proses dengan kalkulasi perkalian antara *feedback gain* dan *state* sistem yang dikendalikan. Komponen *Integrator* akan menjadikan sinyal kendali \mathbf{u} seperti, dimana terdapat empat masukan kendali yaitu u_1 yang merupakan masukan gaya vertikal, u_2 masukan untuk torsi rotasi *roll*, u_3 masukan untuk torsi rotasi *pitch* dan u_4 masukan untuk torsi rotasi *yaw*. Sinyal-sinyal kendali tersebut dijabarkan dalam bentuk matriks seperti [Persamaan \(27\)](#).

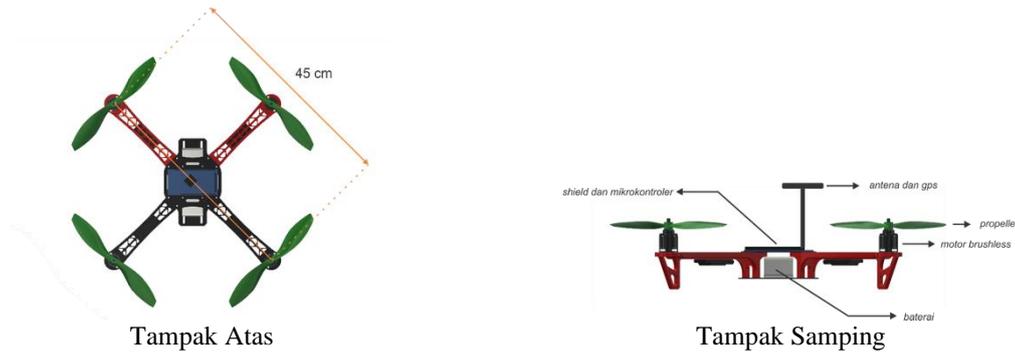
$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = - \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} & K_{17} & K_{18} & K_{19} & K_{110} & K_{111} & K_{112} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} & K_{27} & K_{28} & K_{29} & K_{210} & K_{211} & K_{212} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} & K_{37} & K_{38} & K_{39} & K_{310} & K_{311} & K_{312} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} & K_{47} & K_{48} & K_{49} & K_{410} & K_{411} & K_{412} \end{bmatrix} \begin{bmatrix} x \\ V_x \\ y \\ V_y \\ z \\ V_z \\ \varphi \\ \omega_\varphi \\ \theta \\ \omega_\theta \\ \psi \\ \omega_\psi \end{bmatrix} + \begin{bmatrix} Ki_{11} & Ki_{12} & Ki_{13} & Ki_{14} & Ki_{15} & Ki_{16} & Ki_{17} & Ki_{18} & Ki_{19} & Ki_{110} & Ki_{111} & Ki_{112} \\ Ki_{21} & Ki_{22} & Ki_{23} & Ki_{24} & Ki_{25} & Ki_{26} & Ki_{27} & Ki_{28} & Ki_{29} & Ki_{210} & Ki_{211} & Ki_{212} \\ Ki_{31} & Ki_{32} & Ki_{33} & Ki_{34} & Ki_{35} & Ki_{36} & Ki_{37} & Ki_{38} & Ki_{39} & Ki_{310} & Ki_{311} & Ki_{312} \\ Ki_{41} & Ki_{42} & Ki_{43} & Ki_{44} & Ki_{45} & Ki_{46} & Ki_{47} & Ki_{48} & Ki_{49} & Ki_{410} & Ki_{411} & Ki_{412} \end{bmatrix} \begin{bmatrix} x \\ V_x \\ y \\ V_y \\ z \\ V_z \\ \varphi \\ \omega_\varphi \\ \theta \\ \omega_\theta \\ \psi \\ \omega_\psi \end{bmatrix} \quad (27)$$

Nilai \mathbf{u} dikalikan dengan nilai A_τ^{-1} untuk mencari kecepatan motor ω kemudian dikonversi menjadi nilai *Pulse Width Modulation* (PWM) ke setiap motor. Gaya dan momen gaya gerakan *quadrotor* pada *quadrotor* ditunjukkan dengan [Persamaan \(28\)](#) [18].

$$\begin{bmatrix} F_T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} -b & -b & -b & -b \\ -lb & -lb & lb & lb \\ lb & -lb & -lb & lb \\ k & -k & k & -k \end{bmatrix} \quad (28)$$

2.3. Desain Mekanik *Quadrotor*

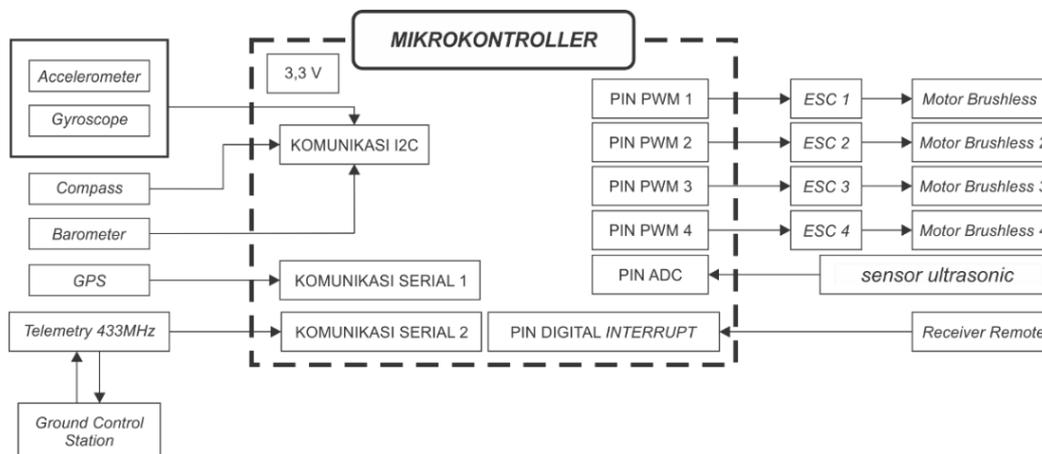
Kerangka mekanik ini terdiri dari *center plate* atas, *center plate* bawah dan 4 lengan yang terpasang pada sisi-sisi *center plate*. Di antara *center plate* atas dan bawah terdapat ruangan yang ditempati oleh baterai yang berbentuk balok berukuran dan modul *telemetry*. Bagian atas dari *center plate* atas digunakan untuk menempatkan *shield* mikrokontroler serta antena penyangga GPS dan modul GPS itu sendiri. Bagian lain *frame* yaitu arm dimanfaatkan sebagai penempatan motor dan esc, dimana setiap *motor brushless* terpasang baling-baling (*propeller*) ukuran 10x5 dengan tipe hqprop. Secara detail ditunjukkan pada [Gambar 3](#).



Gambar 3. Desain mekanik quadrotor

2.4. Arsitektur Elektronik Quadrotor

Masukkan dari sistem *quadrotor* berupa beberapa sensor pengukur data sebagai nilai yang dibutuhkan dalam sistem terutama pada sistem kendali *quadrotor*. Sensor-sensor tersebut adalah *accelerometer* dan *gyroscope* yang menghasilkan data berupa nilai sudut *roll*, *pitch* dari proses *sensor fusion* menggunakan metode DMP. Data sudut *yaw* diperoleh dari *compass*, dimana pada penelitian ini modul yang dimanfaatkan yaitu HMC5883L. Sensor berikutnya adalah *barometer* dan *ultrasonic* yang digunakan untuk menyediakan data ketinggian. Sensor *barometer* digunakan ketika *quadrotor* terbang di atas ketinggian satu meter dan memiliki data mentah yang cukup *fluktuatif* sehingga untuk mendapatkan nilai ketinggian yang cukup stabil dilakukan penyaringan data tekanan menggunakan SISO tapis *Kalman*. Modul sensor *barometer* yang digunakan adalah MS5611. Kemudian digunakan sensor GPS modul Ublok LEA-6H dengan akurasi hingga $0,01^0$. Sensor ini dimanfaatkan untuk menyediakan data *longitude* dan *latitude* yang nantinya di ubah dalam bentuk meter. Metode penyaringan data GPS juga digunakan tapis *Kalman*. Sensor yang terakhir adalah rangkaian pembagi tegangan yang dibaca melalui *port* ADC sensor tegangan untuk mengetahui tegangan baterai secara *real time* sehingga mampu menambahkan pulsa ke aktuator ketika tegangan baterai menurun yang menjadikan *quadrotor* memiliki gaya angkat yang sama. Selanjutnya, untuk perangkat keras yang digunakan dalam mengolah algoritme pada penelitian ini adalah mikrokontroler ARM Cortex-M4 dengan kecepatan kalkulasi komputasi atau *rated speed* 96 MHz dan dapat ditingkatkan melalui *overclocking* hingga 120 MHz. Selain itu mikrokontroler ini memiliki *flash memory* sebesar 512 KBytes, *Bandwidth* 192 Mbytes/sec dan *cache* sebesar 256 Bytes. Hubungan komponen satu dengan yang lainnya ditunjukkan pada Gambar 4.



Gambar 4. Arsitektur elektronik quadrotor

2.5. Rancangan Pengujian

Pengujian ini dilakukan ketika kestabilan gerak rotasi dan ketinggian telah didapat. Pengujian mempertahankan gerak translasi dilakukan pada nilai referensi titik koordinat bumi tertentu, berdasarkan pembacaan koordinat pertama kali pada saat *quadrotor* mendapatkan sinyal *trigger* dari *remote* peneliti. Setelah mendapatkan nilai referensi, peneliti memvariasikan nilai K yang diperoleh dari variasi nilai Q melalui bantuan simulasi MATLAB.

Hal tersebut dilakukan secara berulang hingga *quadrotor* mampu mempertahankan kestabilan gerak translasi sumbu x dan y horizontal. Performansi terbaik yang didapatkan dibandingkan antara pengendalian

Tabel 4. Perbandingan gerak translasi LQR dan LQR Integrator dalam mempertahankan posisi sumbu Y

Pengujian ke -	LQR			LQR Integrator		
	Simpangan maks.	Simpangan min.	SSE	Simpangan maks.	Simpangan min.	SSE
1	3,21	0,80	2,07	0,40	0,47	-0,28
2	-0,03	-1,93	-1,19	1,95	2,19	0,21
3	0,14	-2,99	-1,36	2,28	-1,96	-0,51
4	4,09	0,57	2,45	1,98	-1,52	-0,70
5	4,96	0,48	2,10	2,27	-1,79	0,98
Rata - rata	2,47	-0,61	0,82	1,52	-0,52	-0,06

Merujuk hasil analisa kesalahan yang dihasilkan *quadrotor* dalam mengendalikan gerak translasi menyatakan bahwa sistem kendali LQR dengan penambahan komponen *Integrator* mampu menekan kesalahan yang terjadi pada *quadrotor* ketika menggunakan kendali LQR tanpa adanya *Integrator* [20]. Hal tersebut terlihat pada simpangan maksimal dan minimum baik sumbu x dan y . Dari kelima pengujian, *quadrotor* mampu mempertahankan dirinya untuk tidak melewati batasan toleransi kesalahan 2,5 meter ke arah depan, belakang, kanan maupun kiri.

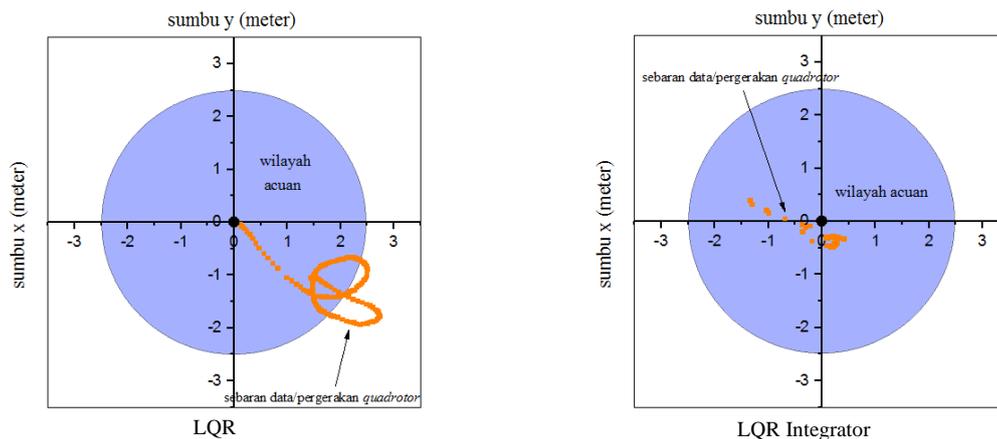
Hasil juga membuktikan, dengan memanfaatkan kendali LQR yang ditambah *Integral* mampu memenuhi kebutuhan sistem gerak translasi *quadrotor*. Kalkulasi kesalahan berupa kecenderungan *steady state error* dari kelima pengujian memiliki rata-rata -0,06-meter ke arah belakang untuk sumbu x dan -0,15-meter ke arah kiri. Dengan diperkuat melalui hasil kalkulasi rata-rata kesalahan yang lebih besar dalam penggunaan kendali LQR yaitu sebesar 0,95 untuk sumbu x dan 0,82 untuk sumbu y yang melebihi batas acuan toleransi penelitian.

Selanjutnya, apabila data – data terbang hasil perolehan pengujian dilihat dari pandangan lain yaitu keakurasian dan kepresisian *quadrotor* terhadap acuan toleransi, gerak terbang translasi *quadrotor* dengan memanfaatkan kendali LQR maupun LQR dan *Integrator*, maka akan didapatkan hasil analisa seperti Tabel 5.

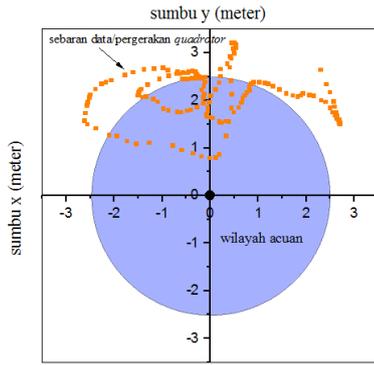
Tabel 5. Perbandingan akurasi dan presisi LQR dan LQR Integrator pengendalian mempertahankan posisi

Pengujian ke -	LQR		LQR Integrator	
	Akurasi (%)	Presisi (%)	Akurasi (%)	Presisi (%)
1	62	23,81	100	45,45
2	90	30,18	100	35
3	80	45,89	100	38,62
4	66	32,84	100	37,93
5	58,67	46,82	100	31,54
Rata - rata	71,33	35,91	100	37,71

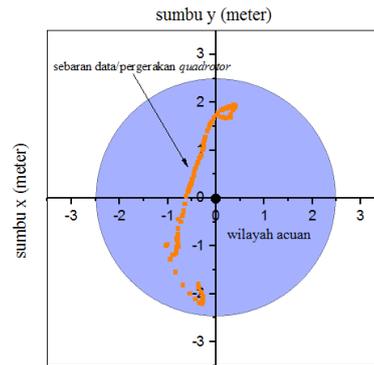
Hasil tersebut dibuktikan dengan Gambar 5 hingga Gambar 9 yang merupakan perbandingan sebaran data *quadrotor* dengan LQR serta LQR dan *Integrator*.

**Gambar 5.** Hasil pengendalian *quadrotor* mempertahankan posisi pengujian ke - 1

Berlandaskan Tabel 5 dan visualisasi Gambar 5 hingga Gambar 9, pengendalian terbang dalam mempertahankan gerak translasi dengan memanfaatkan LQR dan penambahan komponen *Integrator* mampu meningkatkan secara signifikan akurasi *quadrotor* terhadap acuan yang ada. Acuan pengendalian gerakan anti translasi adalah jangkauan toleransi 2,5-meter ke arah luar dari referensi yang diinginkan.

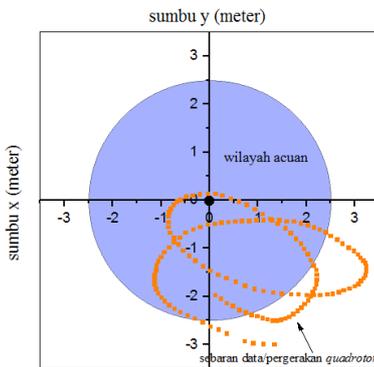


LQR

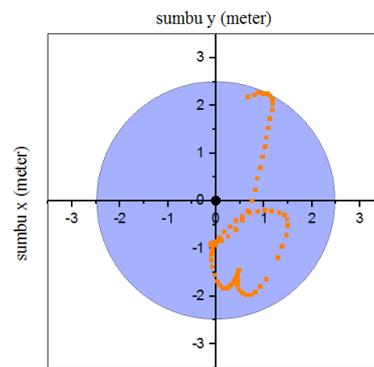


LQR Integrator

Gambar 6. Hasil pengendalian *quadrotor* mempertahankan posisi pengujian ke - 2

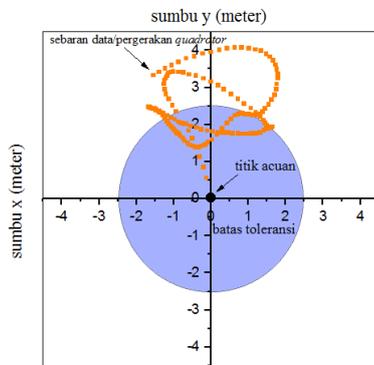


LQR

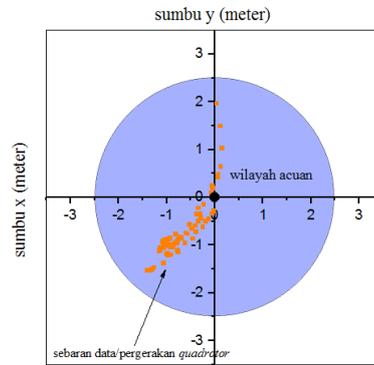


LQR Integrator

Gambar 7. Hasil pengendalian *quadrotor* mempertahankan posisi pengujian ke - 3

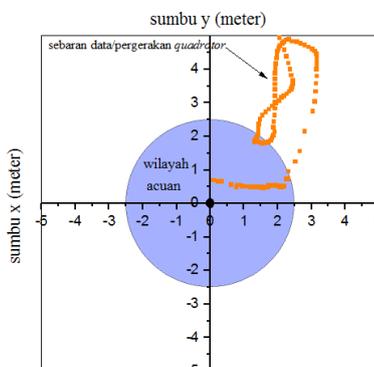


LQR

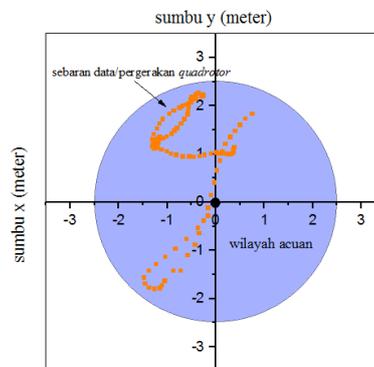


LQR Integrator

Gambar 8. Hasil pengendalian *quadrotor* mempertahankan posisi pengujian ke - 4



LQR



LQR Integrator

Gambar 9. Hasil pengendalian *quadrotor* mempertahankan posisi pengujian ke - 5

Perhitungan akurasi berdasarkan gerak terbang translasi *quadrotor* diperoleh dari banyak data yang mencerminkan posisi *quadrotor* berada di dalam batas toleransi yang pada Gambar 5 ditunjukkan pada lingkaran berwarna biru muda. Banyak data tersebut dibagi dengan nilai 150 dan dikalikan 100% atau dapat dituliskan ke bentuk Persamaan (29).

$$Akurasi = \frac{\text{jumlah data posisi berada di dalam toleransi}}{150} \times 100\% \quad (29)$$

Angka 150 muncul berasal dari pengambilan cuplikan data terbang pengujian *quadrotor* yang mana setiap penerbangan dilakukan selama 15 detik dan menghasilkan sebanyak 150 data. Hasil kalkulasi membuktikan bahwa dengan menggunakan kendali LQR dan *Integrator* sebagai sistem pengendalian *quadrotor* mampu meningkatkan akurasi menjadi 100% dari 71,33% akurasi akomodasi kendali LQR saja. Hal ini tentunya LQR dan *Integrator* membuat sistem *quadrotor* hanya membutuhkan energi lebih kecil dibandingkan kendali LQR.

Akan tetapi, akurasi yang tinggi belum tentu memiliki kepresisian yang tinggi juga. Presisi sendiri merupakan kedekatan perbedaan nilai setiap sebaran data dari sebuah kumpulan data. Pada kasus pengendalian gerak anti translasi ini, kepresisian yang diberikan kendali LQR dan *Integrator* pada sistem tidak begitu signifikan yaitu hanya meningkat pada kisaran 1,8 % dari kendali LQR. Hal ini dikarenakan *Integrator* memang memiliki sifat yang meningkatkan sistem lebih responsif, sehingga *quadrotor* dapat direduksi kesalahannya namun tidak berarti dengan sifat peredaman yang baik.

Tingkat kepresisian diperoleh dari langkah perhitungan rata – rata setiap jarak posisi terhadap referensi dan standar deviasinya. Satu data diwakili oleh 2 data perpindahan x dan y dengan referensi perpindahan x dan y sama dengan nol. Maka jarak sebuah titik posisi terhadap titik referensi dapat dikalkulasi melalui persamaan pitagoras pada Persamaan (30).

$$\text{jarak posisi terhadap referensi} = \sqrt{x^2 + y^2} \quad (30)$$

Kemudian, rata – rata dari setiap jarak tersebut sebagai pembanding dari standar deviasi setiap posisi terhadap referensi yang dikalikan dengan 100 %, seperti Persamaan (31).

$$Presisi = \frac{\text{standar deviasi jarak posisi terhadap referensi}}{\text{rata – rata jarak posisi terhadap referensi}} \times 100\% \quad (31)$$

Keakurasian dan kepresisian yang tinggi tersebut, memang tidak dicerminkan pada penelitian ini. Masih terdapat simpangan – simpangan yang terjadi cukup ketika *quadrotor* berusaha menstabilkan dirinya pada posisi tertentu. Hal ini diakibatkan karena gangguan berupa kecepatan angin yang mengalami perubahan signifikan sewaktu-waktu dapat menggeser *quadrotor* dari keadaan regulatornya. Selain itu, laju perubahan pembacaan GPS termasuk pada kategori frekuensi rendah yang tidak berimbang dengan perubahan perpindahan *quadrotor* yang sebenarnya saat itu. Kondisi tersebut menjadi *quadrotor* tidak mampu menghasilkan respons dalam menciptakan torsi yang sesuai untuk menanggulangi secara cepat dan kembali ke jalur yang diinginkan ketika mendapat gangguan angin yang cukup kuat. Namun demikian, respons yang dicerminkan pada penelitian ini telah memenuhi standar toleransi pengendalian yang baik ketika terbang di lapangan terbuka. Terlihat dari beberapa kali percobaan *quadrotor* tidak melebihi batas toleransi acuannya.

4. KESIMPULAN

Dari analisis hasil pengujian dari penelitian yang dilakukan dapat disimpulkan bahwa sistem kendali Linear Quadratic Regulator dengan Kompensasi Integrator mampu meningkatkan akurasi kestabilan penerbangan dalam pengendalian mempertahankan posisi *quadrotor*. Walaupun masih belum begitu signifikan dalam meningkatkan kepresisian mempertahankan posisi, pesawat telah mampu dengan baik meminimalkan SSE maupun Multi-overshoot yang terjadi. Hasil pengujian menunjukkan dengan nilai terbaik $\mathbf{R} = 1$ untuk semua *state* serta \mathbf{Q} terbaik $Q_x = 0,87$; $Q_y = 124,6$; $Q_{v_x} = 1,77$; $Q_{v_y} = 124,6$ yang menghasilkan $K_x = 0,93$; $K_y = 1,33$; $K_{v_x} = 11,67$; $K_{v_y} = 11,67$ dan $Ki_x = 0,004$; $Ki_y = 0,002$ menjadikan kecenderungan SSE yang terjadi sebesar 0,10 meter untuk sumbu x dan -0,28 untuk sumbu y , sedangkan multi-overshoot yang terjadi sebesar 0,41 meter simpangan maksimal dan -1,35 meter simpangan minimal untuk sumbu x serta 0,40 meter simpangan maksimal dan 0,47 meter simpangan minimal untuk sumbu y . LQR Integrator maupun meningkatkan keakurasian kestabilan mencapai 100% dari 71,33%, serta meningkatkan kepresisian 37,71% dari 35,91 yang dilakukan LQR dari spesifikasi minimum kebutuhan sistem. Ke depan diperlukan pendekatan dengan kendali adaptif dalam mengakomodasi kepresisian penerbangan mempertahankan posisi. Metode kecerdasan buatan menjadi opsi paling memungkinkan untuk dikolaborasikan dengan anatomi kendali LQR untuk mengoptimalkan kestabilan yang dihasilkan.

REFERENSI

- [1] M. S. Mahmoud, M. O. Oyediji, and Y. Xia, "Path planning in autonomous aerial vehicles," in *Advanced Distributed Consensus for Multiagent Systems*, Elsevier, 2021, pp. 331–362, <https://doi.org/10.1016/B978-0-12-821186-1.00018-0>.
- [2] J. Jaykumar and S. D. Gowda, "Drone types and its applications-A survey," *International Research Journal of Engineering and Technology*, vol. 7, no. 6, 2020, <https://www.irjet.net/archives/V7/i6/IRJET-V7I61362.pdf>.
- [3] O. A. Dhewa, A. Dharmawan, and T. K. Priyambodo, "Model of linear quadratic regulator (LQR) control method in hovering state of quadrotor," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 3, 2017, <https://jtec.utem.edu.my/jtec/article/view/1589>.
- [4] R. Dainelli, P. Toscano, S. F. di Gennaro, and A. Matese, "Recent Advances in Unmanned Aerial Vehicle Forest Remote Sensing—A Systematic Review. Part I: A General Framework," *Forests*, vol. 12, no. 3, p. 327, Mar. 2021, <https://doi.org/10.3390/f12030327>.
- [5] A. Houari, I. Bachir, D. K. Mohame, and M. K. Mohamed, "PID vs LQR controller for tilt rotor airplane," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, p. 6309, Dec. 2020, <https://doi.org/10.11591/ijece.v10i6.pp6309-6318>.
- [6] A. Dharmawan, A. Ashari, and A. E. Putra, "Translation Movement Stability Control of Quad Tiltrotor Using LQR and LQG," *International Journal of Intelligent Systems and Applications*, vol. 10, no. 3, pp. 10–21, Mar. 2018, <https://doi.org/10.5815/ijisa.2018.03.02>.
- [7] O. A. Dhewa, T. K. Priyambodo, A. Nasuha, and Y. M. Mustofa, "Enhancement of Stability on Autonomous Waypoint Mission of Quadrotor using LQR Integrator Control," *IIUM Engineering Journal*, vol. 23, no. 1, 2022, <https://doi.org/10.31436/iiumej.v23i1.1803>.
- [8] M. Eatemadi, "Mathematical Dynamics, Kinematics Modeling and PID Equation Controller of QuadCopter," *International Journal of Applied Operational Research*, vol. 7, no. 1, pp. 77-85, 2017, http://ijorlu.liu.ac.ir/browse.php?a_id=503&sid=1&slc_lang=en.
- [9] L. R. García Carrillo, A. E. Dzul López, R. Lozano, and C. Pégard, *Quad Rotorcraft Control*, Springer London, 2013, <https://doi.org/10.1007/978-1-4471-4399-4>.
- [10] G. P. Ribbe, N. A. Arriola, and M. R. Jr., "Modeling and Implementation of Quadcopter Autonomous Flight Based on Alternative Methods to Determine Propeller Parameters," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 5, pp. 727–741, 2020, <https://doi.org/10.25046/aj050589>.
- [11] W. Giernacki, J. Gośliński, J. Goślińska, T. Espinoza-Fraire, and J. Rao, "Mathematical Modeling of the Coaxial Quadrotor Dynamics for Its Attitude and Altitude Control," *Energies (Basel)*, vol. 14, no. 5, p. 1232, Feb. 2021, <https://doi.org/10.3390/en14051232>.
- [12] K. Ogata, *Modern control engineering*, Prentice-Hall, 2010, <https://books.google.co.id/books?id=Ca4rAAAAQBAJ&hl=en>.
- [13] M. Haidekker, *Linear Feedback Controls*. Elsevier, 2020, <https://doi.org/10.1016/C2018-0-00115-2>.
- [14] R. C. Hibbeler, *Engineering Mechanics: Dynamics*, 15th ed. Pearson, 2021, <https://books.google.co.id/books?id=tOFRjXB-XvMC&hl=en>.
- [15] F. Ahmad, P. Kumar, A. Bhandari, and P. P. Patil, "Simulation of the Quadcopter Dynamics with LQR based Control," *Mater Today Proc*, vol. 24, pp. 326–332, 2020, <https://doi.org/10.1016/j.matpr.2020.04.282>.
- [16] L. Martins, C. Cardeira, and P. Oliveira, "Linear Quadratic Regulator for Trajectory Tracking of a Quadrotor," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 176–181, 2019, <https://doi.org/10.1016/j.ifacol.2019.11.195>.
- [17] A. Ma'arif and N. R. Setiawan, "Control of DC Motor Using Integral State Feedback and Comparison with PID: Simulation and Arduino Implementation," *Journal of Robotics and Control (JRC)*, vol. 2, no. 5, 2021, <https://doi.org/10.18196/jrc.25122>.
- [18] R. Gill and R. D'Andrea, "Propeller thrust and drag in forward flight," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, Aug. 2017, pp. 73–79, <https://doi.org/10.1109/CCTA.2017.8062443>.
- [19] A. B. Zakaria and A. Dharmawan, "Sistem Kendali Penghingar Rintangan Pada Quadrotor Menggunakan Konsep Linear Quadratic," *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, vol. 7, no. 2, p. 219, Oct. 2017, <https://doi.org/10.22146/ijeis.25503>.
- [20] A. Winursito and G. N. P. Pratama, "LQR state feedback controller with precompensator for magnetic levitation system," *J Phys Conf Ser*, vol. 2111, no. 1, p. 012004, Nov. 2021, <https://doi.org/10.1088/1742-6596/2111/1/012004>.

BIOGRAFI PENULIS



Oktaf Agni Dhewa menyelesaikan pendidikan sarjana di program studi elektronika dan instrumentasi universitas gadjah mada pada tahun 2014 dan pendidikan master di program studi magister ilmu computer universitas gadjah mada pada tahun 2018. Saat ini penulis 1 adalah dosen tetap di program studi sarjana terapan teknik elektronika universitas negeri yogyakarta. Bidang penelitiannya adalah sistem kendali, robotika, dan internet of things.



Faisal Fajri Rahani menyelesaikan pendidikan sarjana di program studi elektronika dan instrumentasi universitas gadjah mada pada tahun 2016 dan pendidikan master di program studi magister ilmu computer universitas gadjah mada pada tahun 2019. Saat ini penulis 2 adalah dosen tetap di program studi informatika universitas ahmad dahlan. Bidang penelitiannya adalah sistem kendali, robotika, dan kecerdasan buatan.