

## Pendaratan Otomatis *Quadcopter AR Drone* Menggunakan Metode *Linear Quadratic Regulator (LQR)*

Amroy Casro Lumban Gaol<sup>1</sup>, Gembong Edhi Setyawan<sup>2</sup>, Wijaya Kurniawan<sup>3</sup>

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>amroycasro@gmail.com, <sup>2</sup>gembong@ub.ac.id, <sup>3</sup>wjaykurnia@ub.ac.id

### Abstrak

Saat ini UAV (*Unmanned Aerial Vehicle*) jenis *quadcopter* menjadi salah satu teknologi *mobile robot* yang banyak digunakan oleh manusia, baik pada penggunaan komersial maupun digunakan sebagai objek untuk pengembangan teknologi *quadcopter*. UAV jenis *quadcopter* menjadi salah satu UAV yang paling banyak digunakan karena kemampuannya bermanuver ke segala arah, dan fleksibel. Salah satu hal yang menarik untuk dikembangkan pada *quadcopter* adalah kemampuan untuk melakukan pendaratan dengan aman, hal ini bertujuan untuk mengurangi risiko kerusakan fisik pada *quadcopter*. Penelitian ini adalah untuk membuat sistem pendaratan *quadcopter* menggunakan algoritma *LQR (Linear Quadratic Regulator)*, metode ini merupakan sistem kontrol optimal yang memperoleh sinyal *feedback* untuk sistem linear dengan meminimalkan *cost function* kuadrat. Pada penelitian ini *LQR* digunakan untuk melakukan kontrol *attitude* dan kontrol sudut *z* untuk melakukan pendaratan yang di implementasikan pada ROS (*Robot Operating System*), dengan memberikan nilai referensi sebesar 0. Hasil dari penelitian ini adalah menghasilkan algoritma pendaratan yang aman dengan mengetahui nilai *Settling Time* dari pendaratan. Pengujian dilakukan dengan menguji pendaratan dengan ketinggian yang bervariasi yaitu pada ketinggian 1m, 1.5m, 2m, 2.5m, dan 3m. Berdasarkan hasil pengujian yang dilakukan pada penelitian ini, menghasilkan *settling time* pendaratan sebesar 1.743 detik setiap pertambahan satu meter ketinggian.

**Kata kunci:** UAV, *quadcopter*, ROS, *LQR*, pendaratan

### Abstract

Currently *quadcopter*, one type of UAV (*Unmanned Aerial vehicle*), became one of the *mobile robot* technology that is widely used by humans, it's used for commercial and development of *quadcopter* purpose. The reason *quadcopter* is widely used because of its ability to maneuver in all direction and flexibility. One of the interesting thinks to develop is safe landing problem of *quadcopter*, due a safe landing can reduce the risk of physical damage of the *quadcopter*. The purpose of this research was to implement *LQR, Linear Quadratic Regulator* method, for safe landing the *quadcopter*, this method is an optimal control that produce *feedback* gains for linear system by minimize quadratic cost function. *LQR* approach is used to control the attitudes and altitude to perform *quadcopter* landing that are implemented in ROS, *Robot Operating System*. The result of this research is to produce *quadcopter* landing method by knowing the *settling time*. Landing test was performed with variety of altitude i.e. at 1m, 1.5m, 2m, 2.5m, and 3m, and based on the result of landing test, the average of *settling time* for landing safely is 1.75 second for increment one meter of altitude.

**Keywords:** UAV, *quadcopter*, ROS, *LQR*, landing

## 1 PENDAHULUAN

*Unmanned Aerial Vehicle* (UAV) atau lebih sering dikenal dengan pesawat tanpa awak adalah salah satu jenis *mobile robot* yang terdiri dari dua jenis yaitu bentuk pesawat *fixed wing* dan bentuk *copter*. Saat ini UAV banyak

digunakan untuk berbagai keperluan seperti pengambilan citra wilayah, sinematografi udara dan keperluan militer. UAV juga digunakan oleh akademisi maupun peneliti untuk melakukan pengujian dan pengembangan sistem yang dapat membantu pekerjaan sehari-hari (Navajas & Raad, 2015). UAV yang digunakan pada penelitian ini adalah jenis *Quadcopter*. UAV

jenis *quadcopter* memiliki empat baling-baling yang letaknya bersilang dan digunakan untuk melakukan pergerakan pada *quadcopter*. Pergerakan pada *quadcopter* terdiri dari 6 *degrees of freedom (DOF)* yaitu pergerakan rotasi dengan 3 sumbu yaitu (*Pitch, roll, yaw*) dan pergerakan translasi ( $x, y, z$ ).

Pada penelitian sebelumnya permasalahan pendaratan dengan otomatis sudah pernah dilakukan dengan beberapa metode. Misalnya metode yang digunakan yaitu *Non Linear Backstepping Robust Control* untuk mengontrol posisi *quadcopter* dengan memanfaatkan sensor *lidar* untuk mendeteksi ketinggian dilakukan oleh (Beck, Lesueu, Charland-Arcand, & Akhrif, 2016). Pada penelitian sebelumnya juga telah dilakukan pengendalian pendaratan menggunakan metode *Proportional Integral derivative (PID)* dengan memanfaatkan *computer vision*. *Quadcopter* akan mendarat pada *landing pattern* pada saat kamera yang terintegrasi pada *quadcopter* mendeteksi *landing pattern* ketika *quadcopter hovering* di sekitar *pattern* (Barták, Hraško, & Obdržálek, 2014). Pendaratan menggunakan *landing pattern* memunculkan permasalahan baru, yaitu jika tidak ada *pattern* ketika hendak mendarat, atau jika kondisi pencahayaan yang minim yang mengakibatkan kegagalan mendarat karena pembacaan citra dari kamera yang tidak dapat di proses. Dengan menggunakan *computer vision* sebagai pendeteksi *ground* memerlukan kamera yang handal dan tentunya biaya yang cukup besar sedangkan untuk pengolah citra digital yang akan di proses juga memerlukan komputasi yang cukup kompleks sehingga masalah ini juga akan berhubungan dengan permasalahan konsumsi daya.

Pada penelitian ini dibutuhkan sistem *quadcopter open source* yang dapat dikembangkan perangkat lunaknya. Penelitian ini fokus untuk mengembangkan perangkat lunak sistem bukan untuk mengembangkan fisik dari *quadcopter* sendiri, maka dari itu peneliti menggunakan *Parrot Ar Drone* sebagai objek perangkat kerasnya. *Ar Drone* adalah salah satu jenis *quadcopter* yang dapat di kembangkan secara bebas karena *quadcopter* ini sudah dilengkapi dengan beberapa sensor yaitu, kamera depan, kamera belakang, *gyroscope*, ultrasonik dan pada dasarnya *quadcopter* ini sudah dilengkapi dengan mekanisme sistem stabilisasi agar pengembang dapat fokus mengembangkan algoritma *high level control* (Barták, Hraško, & Obdržálek, 2014). Pada

penelitian ini sensor ultrasonik yang terintegrasi pada *quadcopter Ar Drone 2.0* digunakan untuk mendeteksi *ground*, dan hasil dari pembacaan jarak ketinggian ini digunakan untuk mengontrol pendaratan *quadcopter* ini.

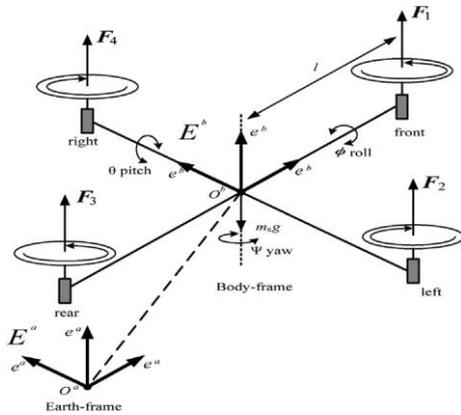
Algoritma sistem kontrol yang digunakan adalah *Linear Quadratic Regulator (LQR)*. *LQR* merupakan sebuah metode sistem kontrol optimal, yang memperoleh sinyal *feedback* untuk sistem linear dengan meminimalkan *cost function* kuadrat. Sistem optimal yaitu nilai parameter yang dipilih sedemikian rupa sehingga pemilihan indeks kinerja minimum tergantung dengan keadaannya (Ogata K. , Teknik Kontrol Automatik jilid 2, 1997). Desain kontrol optimal menggunakan *LQR* didasarkan pada indeks kinerja kuadrat yang dapat mengurangi penentuan unsur matriks  $K$ . Matriks  $K$  pada sistem kendali ini dihasilkan dari perhitungan menggunakan persamaan Riccati yang membutuhkan matriks pembobotan  $Q$  yang semi definit positif, matriks pembobotan  $R$  yang definit positif dan keadaan (*state*) sebagai nilai masukan. Jika algoritma sistem kendali *LQR* dibandingkan dengan algoritma *PID*, maka *LQR* lebih baik dari *PID* karena sistem kontrol *LQR* mencari nilai minimal kesalahan dengan berdasarkan indeks performa sistem, pada penelitian yang dilakukan Badri (2012), mengatakan bahwa metode *LQR* lebih baik dibandingkan dengan metode *PID* karena *LQR* memiliki respons yang lebih cepat dalam mencapai sinyal referensi. Kelebihan dari *LQR* jika dibandingkan dengan *PID* yaitu salah satunya, semua nilai *state quadcopter* dapat di umpan balik.

## 2 UAV QUADCOPTER

*Quadcopter* termasuk jenis *VTOL (vertical take-off and landing)* bergerak ke segala arah serta dapat melayang di udara (*hover*). Dengan kemampuan tersebut, *quadcopter* dapat digunakan untuk melakukan *tracking* terhadap objek yang ditangkap oleh kamera *quadcopter*. Secara umum *quadcopter* memiliki pengaturan motor depan dan motor belakang yang berputar searah jarum jam, sedangkan motor kiri dan motor kanan berputar berlawanan arah jarum jam.

Motor-motor tersebut bersifat independen, yaitu kecepatannya dapat berbeda untuk satu motor dengan motor yang lain. Dengan mengatur kecepatan dari setiap motor, maka pergerakan dari *quadrotor* dapat berubah.

Quadrotor akan terbang dengan syarat gaya angkat dari quadrotor lebih besar dari gaya gravitasi. Pada kondisi titik berat yang seimbang dan karakteristik motor yang sama, kondisi hover tercapai saat semua motor memiliki kecepatan yang besarnya sama. Quadrotor memiliki kelebihan untuk dapat bergerak ke segala arah. Terdapat istilah yang digunakan pada pergerakan keempat motor pada quadcopter, antara lain Throttle yaitu menaikkan dan menurunkan posisi quadcopter, roll yaitu pergerakan quadcopter pada sumbu X (kiri atau kanan), pitch yaitu pergerakan quadcopter pada sumbu Y (depan atau belakang), dan yaw yaitu pergerakan quadcopter pada sumbu Z atau dengan kata lain rotasi ke kiri atau ke kanan yang dilakukan quadcopter.



Gambar 2.1 Pergerakan Quadcopter

2.1 MODEL QUADCOPTER

Model matematika quadcopter yang digunakan pada penelitian ini adalah persamaan matematika yang dikembangkan oleh Bouabdallah (2007), di mana pergerakan yang akan dikontrol adalah berdasarkan pergerakan sudut roll, pitch, yaw dan ketinggian.

$$\begin{aligned} \dot{\phi} &= \dot{\theta} \psi \frac{I_y - I_z}{I_x} - \dot{\theta} \Omega \frac{J_r}{I_x} + \frac{l}{I_x} U2 \\ \dot{\theta} &= \dot{\phi} \psi \frac{I_z - I_x}{I_y} - \dot{\theta} \Omega \frac{J_r}{I_x} + \frac{l}{I_y} U3 \\ \ddot{\psi} &= \dot{\phi} \dot{\theta} \frac{I_x - I_y}{I_z} + \frac{l}{I_z} U4 \\ \ddot{z} &= -g + (\cos \phi \cos \theta) \frac{1}{m_r} U1 \end{aligned} \tag{1}$$

Berdasarkan dari model matematis yang sudah dijabarkan di atas selanjutnya persamaan tersebut diubah menjadi sebuah fungsi yang selanjutnya akan berubah menjadi :

$$\begin{aligned} \dot{x} &= f(X, U) \\ &= \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \psi a_1 - \dot{\theta} a_2 \Omega_r + b_1 U2 \\ \dot{\theta} \\ \dot{\theta} \psi a_3 - \dot{\theta} a_4 \Omega_r + b_2 U3 \\ \dot{\psi} \\ \dot{\phi} a_5 + b_3 U4 \\ \dot{z} \\ -g + (\cos \phi \cos \theta) \frac{1}{m_r} U1 \end{pmatrix} \end{aligned} \tag{2}$$

Di mana :

$$\begin{cases} a_1 = \frac{I_y - I_z}{I_x} & a_5 = \frac{I_x - I_y}{I_z} \\ a_2 = -\frac{J_r}{I_x} & b_1 = \frac{l}{I_x} \\ a_3 = \frac{\psi(I_z - I_x)}{I_y} & b_2 = \frac{l}{I_y} \\ a_4 = \frac{J_r}{I_x} & b_3 = \frac{l}{I_z} \end{cases} \tag{3}$$

Dan input adalah :

$$\begin{cases} U1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U2 = b(-\Omega_2^2 + \Omega_4^2) \\ U3 = b(\Omega_1^2 - \Omega_3^2) \\ U4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{cases} \tag{4}$$

Dari persamaan quadcopter di atas kemudian dituliskan menjadi state space quadcopter di mana \$\dot{x} = f(x, u)\$ di mana \$x\$ adalah state vector dan \$u\$ adalah input vector.

$$\begin{aligned} x &= [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ z \ \dot{z}]^T \\ u &= [U_1 \ U_2 \ U_3 \ U_4]^T \end{aligned} \tag{5}$$

Persamaan di atas akan dimasukkan pada state space dengan :

$$\begin{aligned} \dot{x} &= A_x + B_u \\ y &= C_x + D_u \end{aligned} \tag{6}$$

Selanjutnya berdasarkan persamaan (1) dan persamaan (7) akan menghasilkan matriks sebagai berikut :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dot{\theta} \psi a_1 - \dot{\theta} a_2 \Omega_r & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \psi a_3 - \dot{\theta} a_4 \Omega_r & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \dot{\phi} a_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & b1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & b2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b3 \\ 0 & 0 & 0 & 0 \\ (\cos \phi \cos \theta) \frac{1}{m_r} & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{pmatrix} \text{ dan}$$

$$D = \begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}$$

Terdapat beberapa parameter konstan yang sudah ditentukan dan dihitung oleh peneliti yang lain. pada penelitian ini yang menjadi acuan nilai konstan adalah penelitian (Sun, 2012) .

**Tabel 2.1** Parameter Ar Drone

Parameter	Values
$I_{xx}$	2.2383 [kgm2]
$I_{yy}$	2.9858 [kgm2]
$I_{zz}$	4.8334 [kgm2]

### 3 DESAIN SISTEM KONTROL

Sistem kendali yang digunakan pada penelitian ini adalah sistem kendali optimal *Linear Quadratic Regulator*.

*Linear Quadratic Regulator (LQR)* Merupakan sebuah metode sistem kontrol optimal, yang memperoleh sinyal *feedback* untuk sistem linear dengan meminimalkan *cost function* kuadratis (Ogata K. , 2010). *Plant* diasumsikan bersifat sistem linear, dalam bentuk persamaan keadaan dan fungsi obyektif adalah fungsi kuadratik dari keadaan *plant* dan sinyal *input* kendali.

Persamaan model matematis dibutuhkan untuk menganalisis sistem pengendali optimal, karena persamaan inilah nantinya yang mewakili unjuk kerja dari sistem. Persamaan ini di representasikan menjadi persamaan ruang keadaan (*state space*), dan untuk mendapatkan sistem *quadcopter* dalam model matematis sistem ini dinyatakan dalam bentuk persamaan *differential* orde pertama (Ogata K. , Teknik Kontrol Otomatik jilid 1, 1997).

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (7)$$

Di mana :

$x_{nx1}$  = *state* sistem

$u_{mxn}$  = *input* sistem

$y$  = *output* sistem

$A$  = Matriks sistem  $A_{n \times n}$

$B$  = Matriks sistem  $B_{n \times m}$

Matriks  $A$  dan matriks  $B$  harus terkendali (*controllable*) dengan *full rank n* (Ogata K. , 2010).

Dengan persamaan :

$$Co = [B \ AB \ A^2 \ \dots \ A^{n-1}B] \quad (8)$$

Berdasarkan persamaan (7) sinyal *feedback* yang digunakan pada sistem kontrol ini adalah matriks  $K$  dengan kendali optimal yaitu:

$$u = -Kx \quad (9)$$

Sehingga menghasilkan persamaan baru yaitu :

$$\dot{x} = (A - BK)x + BKx \quad (10)$$

Seerti yang sudah disebutkan sebelumnya untuk mendapatkan sinyal *feedback* dengan meminimalkan *cost function*, yaitu waktu integral dari bentuk kuadratis pada vektor keadaan  $x$  dan vektor masukan  $u$  seperti persamaan berikut :

$$J = \frac{1}{2} \int_{\infty}^0 (x^t Qx + u^t uR) dt \quad (11)$$

Dimana  $Q$  adalah matriks semi definit positif dan  $R$  adalah vektor variabel kontrol atau matriks definit positif, berdasarkan persamaan tersebut variasi parameter dari masalah perancangan *LQR* dapat di tentukan. Langkah pertama yang dilakukan adalah memilih matriks bobot nilai  $Q$  dan  $R$ .

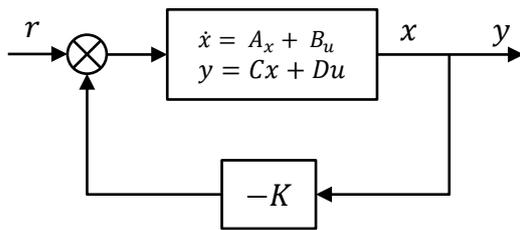
Selanjutnya berdasarkan persamaan (9) Optimal *full-state feedback*  $K$  didapatkan dari persamaan *matrix algebraic Riccati equation (MARE)*:

$$u = -R^{-1}B^T Px \quad (12)$$

Di mana  $P$  didapatkan dari persamaan *Riccati equation*:

$$PA + A^T P + Q - PBR^{-1}B^T = 0 \quad (13)$$

Selanjutnya blok diagram dari *full-state feedback* direpresentasikan seperti



Gambar 3.1 Diagram blok LQR

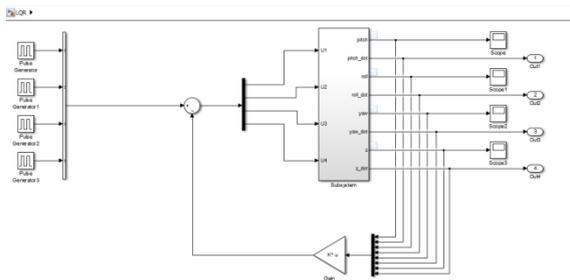
#### 4 PERANCANGAN DAN IMPLEMENTASI

Metode pendaratan menggunakan LQR ini akan di implementasikan pada quadcopter dengan framework ROS (Robot Operating System), tetapi sebelum implementasi pada quadcopter, dilakukan pengujian pada perangkat lunak yang dapat melakukan komputasi kompleks, dalam hal ini peneliti menggunakan Matlab.

##### 4.1 Perancangan

Pada bagian ini perancangan dilakukan menggunakan bantuan matlab, perangkat lunak ini membantu untuk melakukan simulasi diagram blok dan penentuan matriks pembobotan Q dan matriks R.

Gambar 4.1 merupakan diagram blok yang digunakan untuk menguji model matematik quadcopter AR Drone dan melakukan trial and error pemilihan matriks Q dan matriks R.



Gambar 4.1 Diagram Blok pada Matlab

Pemilihan matriks Q dan matriks R dilakukan dengan trial and error, tetapi Penentuannya menggunakan Bryson's rule (Navajas & Raad, 2015) sebagai acuan dasar, yaitu :

$$Q_{ii} = \frac{1}{\max \text{ acceptable value of } x_i^2}$$

$$R_{ii} = \frac{1}{\max \text{ acceptable value of } U_i^2} \quad (14)$$

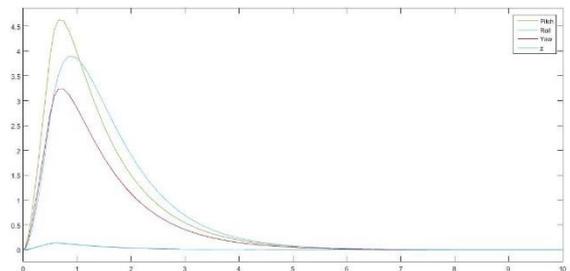
Berdasarkan trial and error yang sudah dilakukan matriks Q dan matriks R di dapatkan

$$Q = \begin{bmatrix} 0.0005 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0005 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0005 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0005 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0005 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0005 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0625 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0625 \end{bmatrix}$$

Dan

$$R = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$

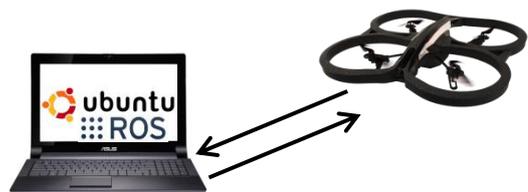
Dan menghasilkan respon sinyal seperti Gambar 4.2:



Gambar 4.2 Respon sinyal simulasi Q dan R

##### 4.2 Implementasi

Implementasi algoritma LQR dilakukan berdasarkan diagram blok yang sudah di rancang pada Gambar 3.1 kode program yang dilakukan menggunakan bahasa pemrograman python node yang diprogram pada ROS.



Gambar 4.3 Diagram Implementasi program

Berdasarkan algoritma yang sudah dirancang, maka tahapan proses sistem kendali adalah sebagai berikut :

1. Membaca *desire value* sinyal(referensi)
2. Membaca nilai *output* berupa nilai sensing yang disebut *state vector*
3. Menghitung *gain K* dengan menggunakan metode (*Linear Quadratic Regulator*)
4. Membandingkan *gain feedback* dengan sinyal referensi
5. Mengirim sinyal kontrol pada *plant*
6. Melakukan perulangan dari tahap 1.

**5 HASIL PENGUJIAN**

Untuk menguji respon sistem dan kinerja dari kontrol *attitude quadcopter* yang telah dibangun maka pengujian melakukan pengujian sistem dengan menguji *attitude* saat hovering maupun saat pendaratan. Pengujian pertama yang dilakukan adalah menguji kontrol *attitude* saat *hovering* dengan gangguan maupun tanpa gangguan.

Pengujian ini bertujuan hanya untuk menguji bagaimana kendali *attitude* dari sistem saat *state* dari *quadcopter* adalah *hovering*, jika pada pengujian ini *quadcopter* dapat melakukan kendali *attitude* dengan baik berdasarkan *set point*, maka kendali *attitude* ini menjadi acuan untuk kendali *attitude* saat pendaratan.

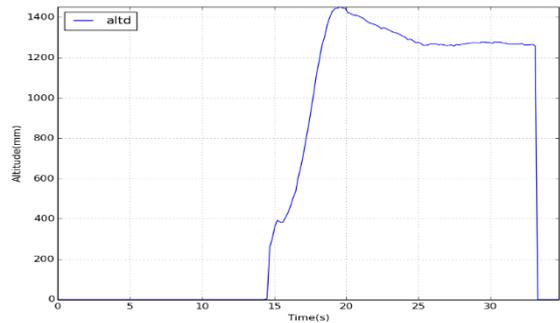
**5.1 Pengujian Pendaratan**

Pengujian dilakukan dengan dua tahapan, yaitu pengujian yang tidak menggunakan algoritma pendaratan, dan pengujian menggunakan algoritma pendaratan LQR. Pengujian dengan LQR yang ditampilkan pada *paper* ini hanya pengujian pada ketinggian 3 meter. perbedaan dari sistem ini akan ditunjukkan dengan analisa grafik.

**A. Pendaratan tanpa algoritma**

Pengujian ini dilakukan dengan tahapan sebagai berikut :

1. *Quadcopter* diterbangkan setinggi  $\pm 1.2$  meter dengan posisi acak
2. Pada saat ketinggian  $\pm 1.2$  meter dengan *hovering state*, pendaratan *quadcopter* dilakukan tidak menggunakan metode pendaratan apa pun .



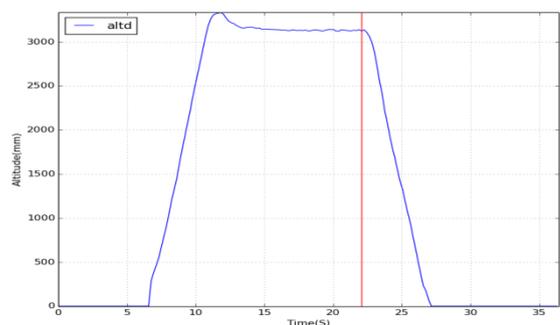
**Gambar 5.1** Pendaratan tanpa algoritma

Perubahan ketinggian dari ketinggian 1.2 meter sampai dengan ketinggian 0 atau mendarat hanya memerlukan waktu kurang dari 1 detik yaitu 0.211 detik, angka ini menunjukkan cepatnya pendaratan yang menyebabkan terjadinya benturan antara *quadcopter* dengan lantai. Tentunya hal ini sangat tidak baik bagi ketahanan dari fisik atau *body frame* pada *quadcopter*.

**B. Pengujian pendaratan dengan LQR**

Tahapan – tahapan pengujian pendaratan yang dilakukan yaitu :

1. *Quadcopter* melakukan take-off terlebih dahulu sebelum menambah ketinggian *quadcopter*.
2. *Quadcopter* diterbangkan dengan ketinggian yang bervariasi, tujuannya adalah untuk mengetahui kinerja dari sistem pada ketinggian yang berbeda.
3. Algoritma pendaratan LQR dijalankan setelah *quadcopter* pada posisi hovering.
4. Pada pengujian dengan menggunakan gangguan saat pendaratan diberikan gangguan dengan mendorong *quadcopter* ke atas / vertikal.
5. Respon sistem di rekam pada data log pada ROS.



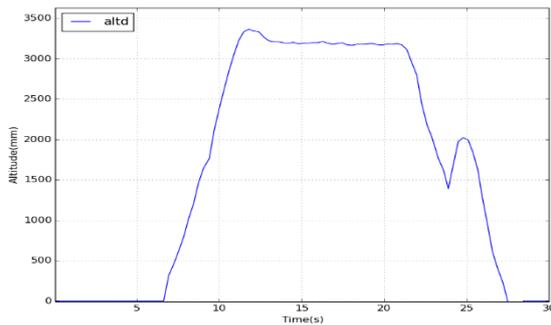
**Gambar 5.2** Pendaratan dengan LQR

Gambar 5.2 menunjukkan waktu yang dibutuhkan saat melakukan pendaratan dari

ketinggian 2.5 meter dengan *Settling Time* sebesar 5.1 detik.

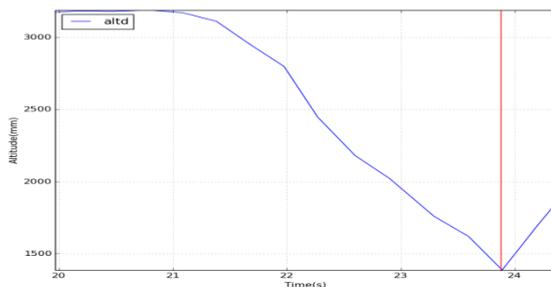
C. Pendaratan dengan LQR diberi gangguan

Gambar 5.3 merupakan hasil respon sistem ketika melakukan pendaratan dengan algoritma LQR pada ketinggian 3 meter dengan gangguan. Bentuk dari gangguan yang dimaksud adalah, ketika *quadcopter* melakukan pendaratan menggunakan LQR pada ketinggian 3 meter, pengujian memberikan gangguan pada *quadcopter* berupa mendorong *quadcopter*



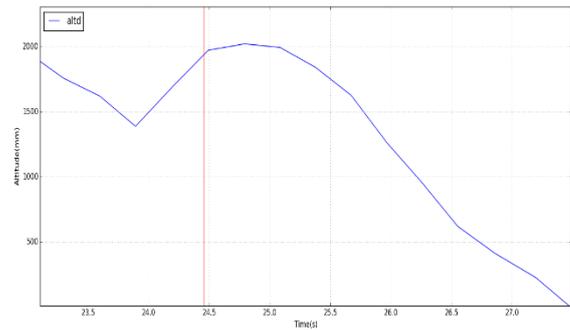
Gambar 5.3 Pendaratan dengan gangguan

Pada Gambar 5.3 menunjukkan adanya perubahan ketinggian saat berada di ketinggian 1.4 meter karena diberikan dorongan ke arah atas dan ketinggiannya (*Overshoot*) mencapai 2 meter, sehingga menghasilkan *settling time* yang lebih lama dari yang sebelumnya ketika mendarat dari ketinggian 3 meter.



Gambar 5.4 Respon sistem saat gangguan

Pada Gambar 5.4 menunjukkan *settling time* yang dihasilkan sebelum gangguan yaitu dari ketinggian 3.2 meter sampai 1.4 meter dibutuhkan sebesar 3.062 detik. Sedangkan setelah adanya gangguan hingga mencapai ketinggian dari 1.4 meter sampai 2 meter, Gambar 5.5 menunjukkan *settling time* untuk mencapai *desire value* adalah sebesar 2.65 detik, sehingga total *settling time* yang dibutuhkan adalah sebesar 5.6 detik.



Gambar 5.5 Landing dari gangguan

Selain pengujian dengan ketinggian 3 meter di atas, pengujian dengan ketinggian 1m, 1.5m, 2m, 2.5m telah dilakukan dan Tabel 2 adalah hasil dari pengujian.

Tabel 2 hasil pengujian

No.	Pengujian	<i>Settling Time</i>
1.	Respon sistem saat ketinggian 1 meter	2.30 detik
2.	Respon sistem saat ketinggian 1.5 meter	2.72 detik
3.	Respon sistem saat ketinggian 2 meter	3.36 detik
4.	Respon sistem saat ketinggian 2.5 meter	3.95 detik
5.	Respon sistem saat ketinggian 3 meter	5.1 detik
6.	Respon sistem saat ketinggian 3 meter dengan gangguan	5.6 detik

$$\begin{aligned}
 \text{Mean} &= \sum \frac{t}{n} \\
 &= \frac{2.3+2.72+3.36+3.95+5.1}{10} \\
 &= 1.743s
 \end{aligned}$$

6 KESIMPULAN

Berdasarkan analisa dari hasil yang telah dilakukan pada penelitian ini dapat diambil kesimpulan yaitu:

- Telah didapatkan Matriks  $Q_{8 \times 8}$  dengan parameter berdasarkan state *quadcopter* yaitu pitch, pitch dot, roll, roll dot, yaw, yaw dot, z dan z dot.
- Telah didapatkan Matriks  $R_{4 \times 4}$  dengan parameter berdasarkan kontrol masukan yaitu U1 sebagai pergerakan translasi atas dan bawah, U2 pergerakan ke kiri dan ke kanan, U3 untuk pergerakan ke depan dan ke

belakang, U4 untuk pergerakan rotasi terhadap sumbu Z.

- c. Berdasarkan pengujian yang sudah dilakukan, didapatkan rata-rata waktu yang dibutuhkan untuk mencapai nilai referensi pendaratan yang di inginkan yaitu 1.743 detik setiap penambahan satu meter ketinggian.

#### DAFTAR PUSTAKA

- Badri, U. (2012). KONTROL OPTIMAL PADA MOTOR DC MENGGUNAKAN METODE LINEAR QUADRATIC REGULATOR (LQR). Surabaya: 2012.
- Barták, R., Hraško, A., & Obdržálek, D. (2014). *A Controller for Autonomous Landing of AR.Drone. IEEE.*
- Beck, H., Lesueu, J., Charland-Arcand, G., & Akhrif, O. (2016). Autonomous takeoff and landing of a quadcopter. *2016 International Conference on Unmanned Aircraft Systems (ICUAS) June 7-10, 2016. . Arlington, VA USA: IEEE.*
- Bouabdallah, S. (2007). *Design and control of quadrotors with application to autonomous flying.*
- Navajas, G. H., & Raad, J. A. (2015). *Concurrent Design Optimization And Control Of a Custom Designed Quadcopter. 16th International Conference on Research and Education in Mechatronics 2015. IEEE.*
- Ogata, K. (2010). *Modern Control Engineering. New Jersey.*
- Sun, Y. (2012). *MODELING, IDENTIFICATION AND CONTROL OF A QUAD-ROTOR DRONE USING LOW-RESOLUTION SENSING. Urbana: University of Illinois at Urbana-Champaign,.*