

Sistem Kendali Penerbangan Quadrotor Pada Keadaan Melayang dengan Metode LQR dan Kalman Filter

Andi Dharmawan^{*1}, Ivan Fajar Arismawan²

¹Department of Computer Science and Electronics, FMIPA UGM

²Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta

e-mail: ^{*1}andi_dharmawan@ugm.ac.id, ²ivanfajar15@gmail.com

Abstrak

Quadrotor merupakan jenis UAV (Unmanned Aerial Vehicle) dengan empat baling-baling dan empat buah rotor. Quadrotor sebagai robot terbang memiliki keunggulan untuk lepas landas dan mendarat secara vertikal. Selain itu quadrotor juga memiliki kemampuan untuk terbang secara melayang mendekati keadaan stasioner. Namun quadrotor cukup sulit untuk dioperasikan, salah satunya adalah membuat quadrotor agar dapat terbang melayang dan mempertahankan keadaan stasioner terhadap kemiringan sudut euler (roll, pitch, dan yaw). Linear Quadratic Regulator (LQR) sebagai salah satu metode quadrotor yang memiliki keunggulan dalam mempertahankan kondisi di lapangan. Metode ini dapat digabungkan dengan algoritma kalman filter. Hal ini bertujuan untuk mengurangi kesalahan pengukuran dari hasil proses sensor fusion dan mempertahankan sudut euler (roll, pitch dan yaw).

Kalman filter bertujuan untuk mengurangi kesalahan pengukuran dari sensor fusion. Kemudian keluaran dari algoritma kalman filter menjadi masukan state bagi kendali LQR pada sudut roll dan sudut pitch. Masukan state tersebut dikalikan dengan negative feedback K_c sebagai proses sistem. Hasilnya diubah menjadi pulsa untuk memutar motor brushless sehingga quadrotor dapat terbang dengan stabil.

Hasil pengujian menunjukkan quadrotor ketika mempertahankan kestabilan terhadap sudut roll memiliki overshoot sebesar 0.35° dan terhadap sudut pitch memiliki overshoot sebesar 2° .

Kata kunci— UAV, robot, stasioner

Abstract

Quadrotor is a type of UAV (Unmanned Aerial Vehicle) with four propellers and four rotor. Quadrotor as flying robots has the advantage to take off and land vertically. In addition quadrotor also has the ability to fly hovered near a stationary state. However quadrotor had some difficulties to operate. One of these difficulties is to make quadrotor be able to fly and maintain the stationary state of the Euler angles (roll, pitch, and yaw). Linear Quadratic Regulator (LQR) as one of the modern control method which has the advantage of maintaining the conditions on the ground. This method can be combined with Kalman filter algorithm. It aims to reduce measurement error from the process sensor fusion and maintain Euler angles (roll, pitch and yaw).

Kalman filter aims to reduce the measurement error of the sensor fusion. Then the output of Kalman filter algorithm becomes the input state for control LQR the roll angle and pitch angle. Input state is multiplied with the negative feedback K_c as process systems. The results are converted into pulses to rotate the brushless motor so quadrotor can fly stably.

The test results showed quadrotor while maintaining stability against roll angle has overshoot of 0.35° and the pitch angle has overshoot of 2° .

Keywords— UAV, robot, stationery.

1. PENDAHULUAN

Quadrotor merupakan jenis UAV (Unmanned Aerial Vehicle) dengan 4 baling-baling dengan 4 buah rotor. Pemanfaatan *quadrotor* banyak diaplikasikan pada bidang militer saat awal dikembangkan. *Quadrotor* semakin meluas penggunaannya karena banyak yang memanfaatkan *quadrotor* untuk keperluan masing-masing contohnya pada bidang pendidikan.

Pengoprasian *quadrotor* dilakukan secara *remote* atau dikendalikan dari jarak jauh. Keunggulan dari *quadrotor* adalah *quadrotor* mampu lepas landas dan mendarat secara vertikal. Kemampuan terbang secara melayang ini juga salah satu keunggulan dari UAV jenis *quadrotor*.

Pengoprasian *quadrotor* memerlukan keahlian dari sang pilot yang memegang kendali. Salah satu kesulitan dari pengoprasian *quadrotor* adalah membuat *quadrotor* agar dapat terbang melayang dan mempertahankan sudut *euler* (*roll*, *pitch*, *yaw*).

Untuk membuat agar *quadrotor* melayang dan mempertahankan sudut *euler*, dibutuhkan sistem kendali yang handal. Sistem kendali harus mampu mempertahankan *quadrotor* pada saat kondisi melayang, tanpa sistem kendali membuat *quadrotor* agar dapat melayang sulit dilakukan. Sistem kendali yang dibutuhkan *quadrotor* untuk dapat melayang harus memiliki respon cepat supaya menjaga kondisi *quadrotor* berada dalam posisi melayang.

Quadrotor juga dilengkapi dengan sensor-sensor seperti *accelerometer 3 axis* dan *gyroscope 3 axis*. Dari sensor-sensor tersebut dapat digunakan untuk mengukur sudut yang dibentuk oleh badan pesawat dan sumbu bumi (sudut orientasi) ketika terbang. Sudut dibentuk berdasarkan proses *sensor fusion* dari sensor-sensor tersebut. Sudut ini nantinya akan dikendalikan agar *quadrotor* dapat melayang.

Linear Quadratic Regulator (LQR) sebagai salah satu metode untuk membuat sistem kendali *quadrotor* dapat digabungkan dengan kalman *filter*. Hal ini bertujuan untuk mengurangi kesalahan pengukuran dari hasil proses *sensor fusion* dan mempertahankan sudut *euler* (*roll*, *pitch* dan *yaw*) [1]. Respon kendali dari LQR juga baik [2]. Oleh karena itu, diharapkan *quadrotor* dapat terbang lebih stabil.

Penelitian tentang metode ini juga banyak dilakukan. Penerapan kendali LQG yang merupakan kombinasi LQR dan kalman *filter* pada MAV untuk kestabilan terbang [3]. Tidak hanya itu, pemodelan test bench *quadrotor* juga sudah dilakukan [4].

2. METODE PENELITIAN

2.1 Analisis Sistem

Quadrotor yang digunakan merupakan tipe + yang mempunyai komponen 6 *state* yang kan dikendalikan. State model tersebut adalah *yaw*, *pitch*, *roll*, kecepatan sudut sumbu z, kecepatan sudut sumbu y dan kecepatan sudut sumbu x yang ditunjukkan pada persamaan (1)

$$\begin{matrix} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \\ \dot{\mathbf{x}} \end{matrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \phi \\ \psi \\ p \\ q \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (1)$$

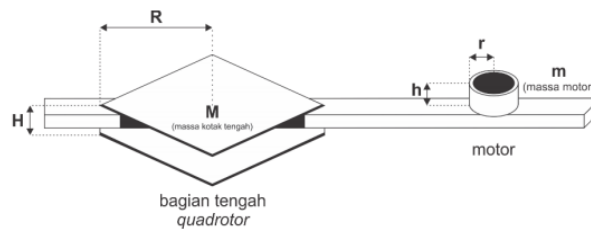
Inersia sumbu x, sumbu y dan sumbu z diberikan pada persamaan (2) sampai (4)

$$I_{xx} = \frac{mr^2}{4} + \frac{mh^2}{6} + 2mr^2 + \frac{MR^2}{4} + \frac{MH^2}{12} \quad (2)$$

$$I_{yy} = \frac{mr^2}{4} + \frac{mh^2}{6} + 2mr^2 + \frac{MR^2}{4} + \frac{MH^2}{12} \quad (3)$$

$$I_{zz} = \frac{MR^2}{4} + 4mr^2 \quad (4)$$

Dimana nilai-nilai komponen tersebut diperlihatkan sesuai Gambar 1



Gambar 1 Persamaan inersia x,y,z [5]

Pada Gambar 1 tersebut komponen H adalah tinggi dari CM (Center of Mass), R adalah jari-jari body tengah *quadrotor*, M adalah massa dari body tengah *quadrotor*, h adalah tinggi motor, r adalah jari-jari motor, dan m adalah massa motor.

Penerapan kendali LQR dan Kalman *filter* pada *quadrotor* membutuhkan nilai **Q**, nilai **R** serta nilai **Q_f** dan nilai **R_f**. untuk mendapatkan hasil yang optimal. Nilai **Q** dan **R** pada kendali LQR dibutuhkan untuk memperoleh *gain feedback* LQR yang disimbolkan dengan **K_c**. Untuk mencari nilai **A** dan **B** pada Kalman *filter* membutuhkan nilai **K_f** yang dicari dari penalaan nilai **Q_f** dan nilai **R_f** ditunjukkan dengan persamaan (5) dan (6) [1].

$$A^k = [A - K_f * C] \quad (5)$$

$$y = C * x \quad (6)$$

dimana nilai **K_f** dapat dicari dengan persamaan (7). Kalman *filter* berperan sebagai *estimator* dengan cara mengolah data sekarang dengan data yang sebelumnya untuk memperkirakan keadaan yang akan datang. Data yang akan dijadikan masukan untuk Kalman *filter* sebanyak dua data yaitu *pitch* dan *roll* ke dalam persamaan (8) pada landasan teori untuk menghasilkan nilai \hat{x} .

$$K_f = PC^T R_f^{-1} \quad (7)$$

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f(y - \hat{y}) \quad (8)$$

Nilai \hat{x} kemudian dikalkulasikan dengan *gain feedback* kendali LQR (**K_c**) yang menjadi nilai masukan **u**. Sehingga persamaan yang akan diberikan ke motor ditunjukkan pada persamaan (9) sampai (11)

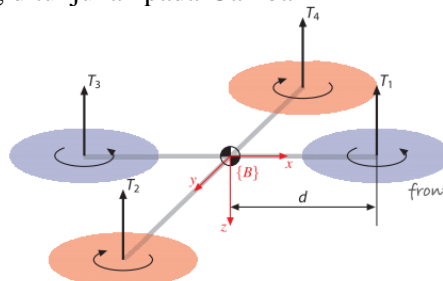
$$U_2 = K_{c00} * \hat{x}_0 + K_{c03} * gx \quad (9)$$

$$U_3 = K_{c11} * \hat{x}_1 + K_{c14} * gy \quad (10)$$

$$U_4 = K_{c22} * yaw_deg + K_{c25} * gz \quad (11)$$

dimana U_2 untuk *roll*, U_3 untuk *pitch*, dan U_4 untuk *yaw*.

Analisis peletakan sensor *accelerometer* dan *gyroscope* juga diperlukan untuk menentukan *thrust* dan *torque* pada *quadrotor*. Penelitian ini menggunakan *quadrotor* tipe + dengan peletakan motor yang ditunjukkan pada Gambar 2



Gambar 2 *Quadrotor* tipe + dengan peletakan motor [6]

Baling-baling 1 dan baling-baling 3 yang berposisi di depan dan belakang berputar melawan arah jarum jam (*counter clockwise*) sedangkan baling-baling 2 yang berposisi di kanan dan kiri dan baling-baling 4 berputar searah jarum jam (*clockwise*).

Peletakan sensor *accelerometer* dan *gyroscope* diposisikan sedemikian rupa sehingga motor 1 dan motor 3 berada di depan dan belakang adalah sumbu x (*roll*), motor 2 dan motor 4 berada di kanan dan kiri adalah sumbu y (*pitch*). Ketika motor 3 posisi lebih rendah dibanding motor 1 maka nilai sudut *roll* akan positif begitu sebaliknya. Pada sudut *pitch* akan bernilai positif apabila posisi motor 4 lebih rendah dibanding motor 2 begitu juga sebaliknya.

Rotasi baling-baling dan posisi peletakan IMU seperti itu, mengakibatkan *quadrotor* memiliki *thrust* yang ditunjukkan pada persamaan (12)

$$T_i = b\omega^2, \quad i = 1,2,3,4 \quad (12)$$

dimana i adalah motor ke- i . Gerak translasi *quadrotor* ditunjukkan pada persamaan (13)

$$m\dot{v} = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} - {}^oR_B \begin{pmatrix} 0 \\ 0 \\ T \end{pmatrix} \quad (13)$$

dimana m adalah massa *quadrotor*, \dot{v} adalah turunan pertama dari kecepatan, T adalah *thrust* total, dan oR_B adalah koordinat *frame* bumi.

Torsi x-axis *quadrotor* yang merupakan *rolling torque* diberikan pada persamaan (14) dan (15)

$$\tau_x = dT_4 - dT_2 \quad (14)$$

$$\tau_x = db(\omega_4^2 - \omega_2^2) \quad (15)$$

dimana d adalah jari-jari *quadrotor* yang diukur dari titik tengah pusat massa hingga titik tengah motor.

Seperi halnya torsi x-axis, untuk torsi y-axis yang merupakan *pitching torque* dapat dijabarkan pada persamaan (16) dan (17)

$$\tau_y = dT_1 - dT_3 \quad (16)$$

$$\tau_y = db(\omega_1^2 - \omega_3^2). \quad (17)$$

Torsi yang diaplikasikan pada setiap baling-baling oleh motor yang melawan gaya gesek udara dapat diperoleh dengan persamaan (18)

$$Q_i = k\omega_i^2 \quad (18)$$

dimana k adalah faktor seperti b , maka total torsi reaksi z-axis ditunjukkan pada persamaan (19) dan (20).

$$\tau_z = Q_1 - Q_2 + Q_3 - Q_4 \quad (19)$$

$$\tau_z = k\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2. \quad (20)$$

Torsi yaw dapat dikendalikan dengan memanfaatkan perkiraan koordinat dari keempat motor. Berdasarkan persamaan Euler untuk pergerakan berputar dari *quadrotor* pada sumbu yaw memiliki persamaan (21).

$$J\dot{\vec{\omega}} = -\vec{\omega} \times J\vec{\omega} + \Gamma \quad (21)$$

dimana J adalah 3 x 3 matriks inersia dari *quadrotor*, $\vec{\omega}$ adalah vektor kecepatan sudut, dan $\Gamma = (\tau_x, \tau_y, \tau_z)^T$ adalah torsi dari *quadrotor*.

Gerak *quadrotor* dihasilkan dari integrasi persamaan dinamik dari persamaan (13) sampai (21), dimana gaya dan momen gaya pada *quadrotor* ditunjukkan pada persamaan (22)

$$\begin{pmatrix} T \\ \Gamma \end{pmatrix} = \begin{pmatrix} -b & -b & -b & -b \\ 0 & -db & 0 & db \\ db & 0 & -db & 0 \\ k & -k & k & -k \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} = A \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} \quad (22)$$

Jadi fungsi dari kecepatan motor seperti pada persamaan (23)

$$\begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} = A^{-1} \begin{pmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} \quad (23)$$

Pengendalian setiap motor yang menggunakan kendali LQR dapat menggunakan persamaan (23) dengan T tidak dikendalikan maka komponen $-b$ tidak diikuti sertakna dengan

torsi x, y, dan z [6] adalah masukkan dari persamaan (9) hingga (11). Fungsi kecepatan setiap motor ditunjukkan pada persamaan (24).

$$\begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} = \begin{pmatrix} 0 & db & k \\ -db & 0 & -k \\ 0 & -db & k \\ db & 0 & -k \end{pmatrix} \begin{pmatrix} U_2 \\ U_3 \\ U_4 \end{pmatrix} \quad (24)$$

Sehingga pulsa yang diberikan untuk kendali setiap motor ditunjukkan persamaan (25) – (28).

$$\text{motor depan} = U_3 * db + U_4 * k \quad (25)$$

$$\text{motor kanan} = U_2 * -db + U_4 * k \quad (26)$$

$$\text{motor belakang} = U_3 * -db + U_4 * k \quad (27)$$

$$\text{motor kiri} = U_2 * db + U_4 * k \quad (28)$$

Dalam mencari konstanta b dan k dapat diperoleh dari persamaan *thrust* dan *torque* ditunjukkan pada persamaan (29) dan (30) [7].

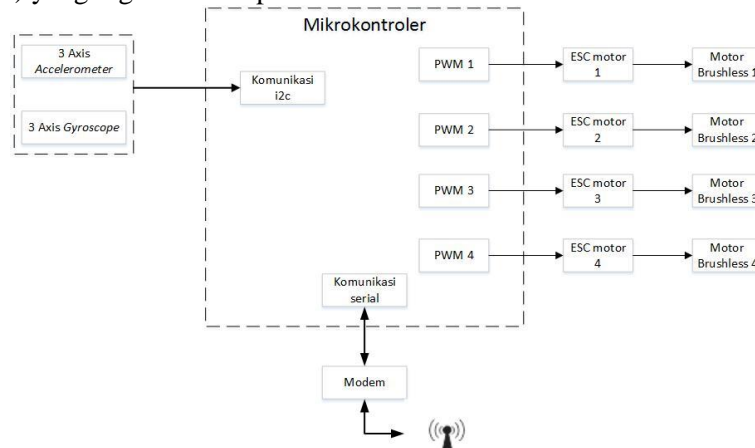
$$T = C_T \rho A R^2 \omega^2 \quad (29)$$

$$Q = C_Q \rho A R^3 \omega^2. \quad (30)$$

Dari persamaan (29) dan (30) diperoleh konstanta $b = C_T \rho A R^2$ dan konstanta $k = C_Q \rho A R^3$, dimana T adalah *thrust*, Q adalah *torque*, C_T dan C_Q adalah *thrust* tak berdimensi dan koefisien *torque*, ρ adalah densitas udara ($1,184 \text{ kg/m}^3$), A adalah luas area *quadrotor* ketika berputar ($2\pi^2$), R adalah jari-jari *quadrotor* yang diukur dari titik tengah pusat massa hingga titik tengah motor *brushless*, dan ω adalah kecepatan putar *propeller* (rps).

2.2 Arsitektur Sistem

Arsitektur sistem *quadrotor* yang dirancang dilengkapi dengan sensor 3 aksis *accelerometer* dan 3 aksis *gyroscope*. Sensor-sensor tersebut digunakan untuk pembacaan sudut *yaw*, *pitch*, *roll* dan kecepatan sudut sumbu x, kecepatan sudut sumbu y, dan kecepatan sudut z. Sebagai aktuator *quadrotor* digunakan 4 buah motor *brushless* dengan 4 buah ESC (*Electric Speed Controller*) yang digambarkan pada Gambar 3

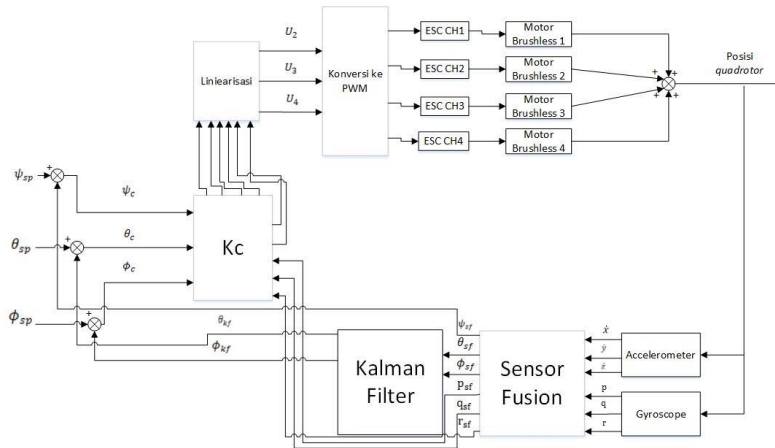


Gambar 3 Arsitektur sistem

Dari Gambar 3, Arduino sebagai mikrokontroler menerima inputan dari sensor *accelerometer* dan *gyroscope* untuk diolah menjadi data sudut *yaw*, *pitch*, *roll*. Nilai pembacaan tersebut kemudian dibandingkan dengan nilai *set point*. Hasil perbandingan tersebut kemudian diubah menjadi pulsa PWM (*Pulse width Modulation*) lalu dioutputkan ke motor *brushless*.

2.3 Rancangan Sistem Kendali

Rancangan sistem kendali yang akan diterapkan dalam penelitian ini digambarkan pada Gambar 4 Diagram sistem kendali



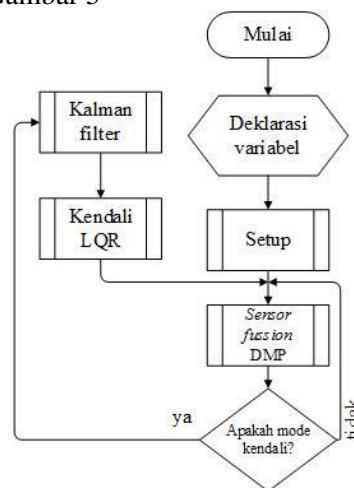
Gambar 4 Diagram sistem kendali

Pada sistem kendali, sistem menerima inputan dari sensor *accelerometer* dan *gyroscope*. Data dari sensor sensor tersebut kemudian diolah dengan sensor *fusion* sehingga menghasilkan data sudut *yaw* (ψ), *pitch* (θ), dan *roll* (ϕ), kecepatan sudut sumbu x (p), kecepatan sudut sumbu y (q) dan kecepatan sudut sumbu z (r). Hasil sensor *fusion* kemudian masuk ke *Kalman filter*. Pada kalman filter ini terjadi operasi perhitungan antara data lalu dan data yang sekarang untuk memprediksi data yang akan datang.

Setelah *Kalman filter*, hasilnya akan dikalikan dengan penguatan umpan balik sebagai kendali LQR yang nanti akan menjadi input sistem. Selanjutnya masukkan proses sistem dibandingkan dengan nilai yang diinginkan, dimana nilai yang diinginkan dalam penelitian ini adalah sudut 0° untuk masing-masing sudut orientasi. Perbedaan nilai yang terjadi akan diproses diubah ke PWM untuk diberikan ke sistem yang sebelumnya dilinearkan yaitu motor brushless.

2.4 Rancangan Perangkat Lunak

Perangkat lunak yang dirancang dimaksudkan untuk sistem dapat bekerja sesuai dengan yang diinginkan. Perangkat lunak yang dirancang meliputi mengakses sensor, aktuator, maupun perangkat lainnya. Perangkat lunak yang digunakan adalah arduino dengan pemrograman bahasa C++ dengan IDE arduino (*Integrated Development Enviroment*) sebagai *compilernya*. Bagan alir program utama yang bertujuan untuk membuat *quadrotor* mampu mempertahankan sudut *eulernya* diperlihatkan pada Gambar 5



Gambar 5 Bagan alir program utama

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengujian Motor Brushless

Pengujian yang pertama yaitu untuk mencari lebar pulsa *dead zone* dari setiap motor. Pencarian ini dilakukan untuk mengetahui nilai titik awal motor mulai berputar. Nilai pulsa *dead zone* berpengaruh pada kecepatan putar motor ketika diberikan pulsa. Apabila nilai *dead zone* salah satu motor lebih besar dari yang lain, maka motor tersebut akan berputar lebih lambat dengan motor yang lain jika diberikan lebar pulsa yang sama. Berdasarkan eksperimen yang dilakukan maka didapatkan lebar pulsa *dead zone* dari tiap-tiap motor yang digunakan yang ditunjukkan dalam Tabel 1

Tabel 1 Nilai *dead zone* tiap motor

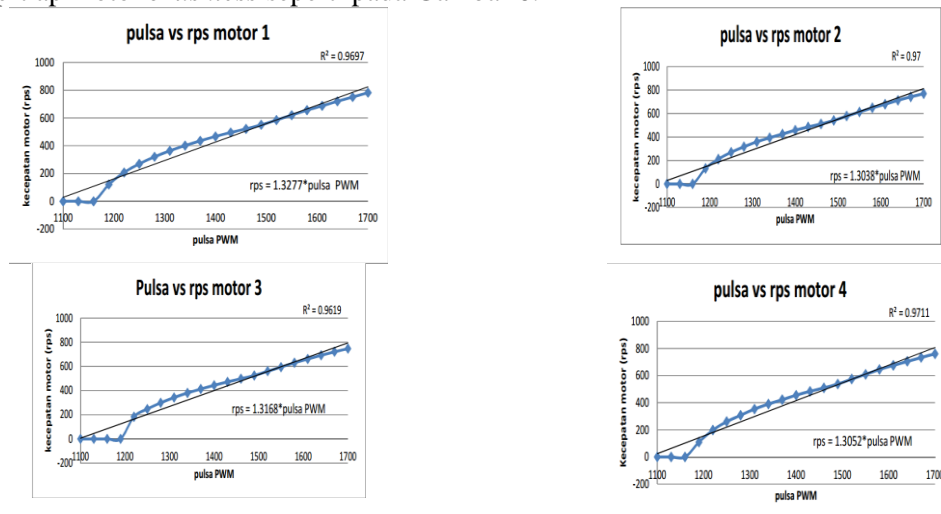
Motor	Lebar pulsa <i>dead zone</i>
motor 1 (depan)	1187
motor 2 (kanan)	1181
motor 3 (belakang)	1191

Dead zone adalah tegangan minimum yang dibutuhkan motor untuk berputar. Pada Tabel 1 didapatkan bahwa nilai *dead zone* yang paling kecil adalah motor 2 (kanan). Secara logika, motor 2 akan berputar lebih cepat dari ketiga motor yang lain ketika diberikan lebar pulsa yang sama karena memiliki *dead zone* yang paling kecil. Sedangkan pada motor 3 (belakang) akan berputar lebih lambat dibanding ketiga motor jika diberikan lebar pulsa yang sama karena nilai *dead zone* motor 3 paling besar. Maka dilakukan pengujian variasi lebar pulsa PWM yang hasilnya diperlihatkan pada Tabel 2.

Tabel 2 Hasil pengujian kecepatan motor

Masukan pulsa PWM	1200	1400	1500	1600	1700
Keluaran motor 1 (depan) $\left(\frac{rad}{second}\right)$	160.94	426.48	559.25	692.02	824.79
Keluaran motor 2 (kanan) $\left(\frac{rad}{second}\right)$	160.76	421.52	551.90	682.28	812.66
Keluaran motor 3 (belakang) $\left(\frac{rad}{second}\right)$	136.86	400.22	531.9	663.58	795.26
Keluaran motor 4 (kiri) $\left(\frac{rad}{second}\right)$	154.74	415.78	546.3	676.82	807.34

Maka dari itu diperlukan perlakuan yang berbeda dari tiap motor dengan nilai lebar pulsa yang sama. Tindakan yang dilakukan adalah dengan melinearkan grafik pulsa PWM vs rps tiap-tiap motor *brushless* seperti pada Gambar 6.



Gambar 6 Hasil linearisasi motor *brushless*

Konstanta linearitas motor *brushless* adalah m . Konstanta m kemudian digunakan sebagai pembagi rps yang dimasukkan pada program, yaitu variabel w_0 . Jadi pulsa yang akan dikirimkan ke tiap motor adalah

$$\text{pulsa motor ke } -i = \left(\frac{w_0 + w_c}{m_i} \right) + \text{dead zone}$$

Dengan w_c adalah hasil dari perhitungan kendali LQR dan m_i adalah konstanta linearitas motor *brushless* ke- i , sehingga motor dapat berputar dengan kecepatan yang hampir sama ketika terjadi perubahan lebar pulsa.

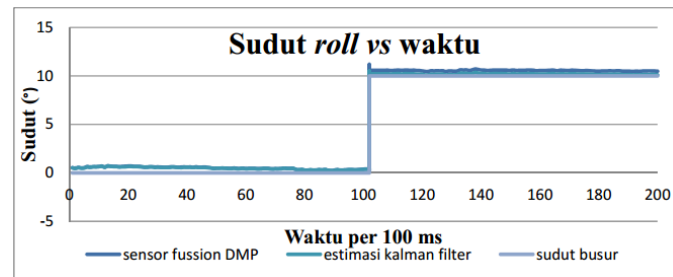
3.2 Hasil Pengujian Kalman Filter

pengujian dilakukan dengan memvariasikan nilai Q_f dan R_f . Selanjutnya dilakukan pengambilan data sudut roll dan sudut pitch ketika masing-masing membentuk sudut 0° dan sudut 10° . Variasi nilai dilakukan dari nilai $Q_f = 1$ [8]. Hasil variasi nilai Q_f dan R_f pada sudut *roll* diberikan pada Tabel 3.

Tabel 3 Variasi nilai Q_f dan R_f sudut *roll* pada kalman filter

Q_f	R_f	K_f	A_k
[1 0]	1	[0.8695 0]	[-0.8695 0]
[0.9 0]	1	[0.8179 0]	[-0.8179 0]
[1.1 0]	1	[0.9550 0]	[-0.9550 0]
[1.2 0]	1	[0.9695 0]	[-0.9695 0]
[1.3 0]	1	[1.0112 0]	[-1.0112 0]

Dari variasi pada Tabel 3 diperoleh hasil terbaik untuk sudut *roll* adalah $Q_f = 1.2$ yang disajikan pada



Gambar 7 Pengujian sudut *roll* pada kalman filter

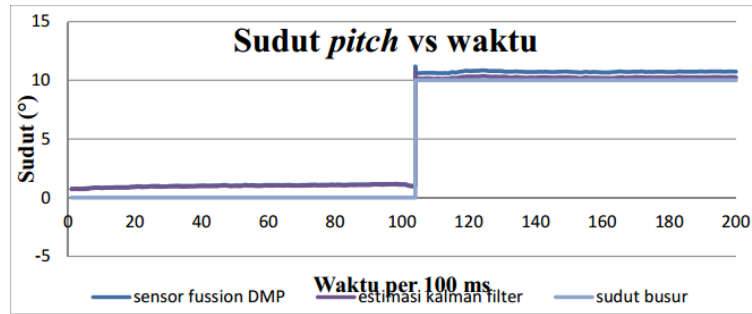
Hasil pengujian pada Gambar 7 adalah bacaan pada sudut $0^\circ = 0.47^\circ$ dengan standar deviasi 0.13° dan bacaan pada sudut $10^\circ = 9.94^\circ$ dengan standar deviasi 0.44° . estimator kalman filter mampu mengurangi kesalahan dalam pembacaan sudut *roll*.

Pengujian dilanjutkan ke sudut *pitch*. Variasi nilai Q_f dan R_f yang dilakukan mendapatkan hasil yang sama dengan pengujian yang dilakukan pada sudut *roll*. Hal ini dikarenakan sumbu x (*roll*) dan sumbu y (*pitch*) simetris. Variasi nilai Q_f dan R_f ditunjukkan dengan Tabel 4.

Tabel 4 Variasi nilai Q_f dan R_f sudut *pitch* pada kalman filter

Q_f	R_f	K_f	A_k
[0 1]	1	[0 0.8695]	[0 -0.8695]
[0 0.9]	1	[0 0.8179]	[0 -0.8179]
[0 1.1]	1	[0 0.9550]	[0 -0.9550]
[0 1.2]	1	[0 0.9695]	[0 -0.9695]
[0 1.3]	1	[0 1.0112]	[0 -1.0112]

Dari variasi pada Tabel 4 Variasi nilai Q_f dan R_f sudut *pitch* pada kalman filter diperoleh hasil terbaik untuk sudut *pitch* adalah $Q_f = 1.1$ yang disajikan pada Gambar 8.



Gambar 8 Hasil pengujian sudut *pitch* pada kalman *filter*

Hasil pengujian pada Gambar 8 adalah bacaan pada sudut $0^\circ = 0.98^\circ$ dengan standar deviasi 0.1 dan bacaan pada sudut $10^\circ = 10.05^\circ$ dengan standar deviasi 0.30°. Dari eksperimen yang diperoleh, estimator kalman *filter* mampu mengurangi kesalahan dalam pembacaan sudut *pitch*.

3.3 Hasil Pengujian Kendali LQR

3.3.1 Variasi bobot Q dan R

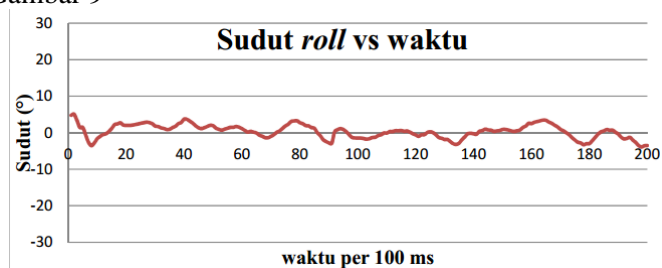
Variasi bobot matriks Q dan R berpengaruh terhadap kestabilan *quadrotor* dalam mempertahankan sudut *euler*. Pada penelitian ini, variasi nilai matriks Q dan R dilakukan secara terpisah pada sudut *roll*, *pitch* dan sudut *yaw* sehingga nilai Q tidak diberikan pada satu kesatuan komponen matriks 6x6 melainkan dua komponen matriks 2x6 untuk masukan pada sudut *roll*, sudut *pitch* dan sudut *yaw*.

Dari beberapa variasi nilai Q dan R kemudian didapatkan nilai K_c seperti yang ditunjukkan pada Tabel 5 untuk sudut *roll*

Tabel 5 Variasi bobot Q dan R pada sudut *roll*

Q	R	K_c
$\begin{bmatrix} 0,1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	1	$[0,3162 \ 0 \ 0 \ 1,0004 \ 0 \ 0]$
$\begin{bmatrix} 700 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	1	$[26,4575 \ 0 \ 0 \ 1,0332 \ 0 \ 0]$
$\begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	1	$[31,6228 \ 0 \ 0 \ 1,0396 \ 0 \ 0]$
$\begin{bmatrix} 1200 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	1	$[34,6410 \ 0 \ 0 \ 1,0433 \ 0 \ 0]$
$\begin{bmatrix} 1700 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	1	$[41,2311 \ 0 \ 0 \ 1,0513 \ 0 \ 0]$

Dengan variasi nilai tersebut, didapatkan hasil yang terbaik pada sudut *roll* adalah Q dengan nilai 1000 dimana *quadrotor* mampu mempertahankan sudut *roll* pada 0.35° . Seperti yang ditunjukkan pada Gambar 9



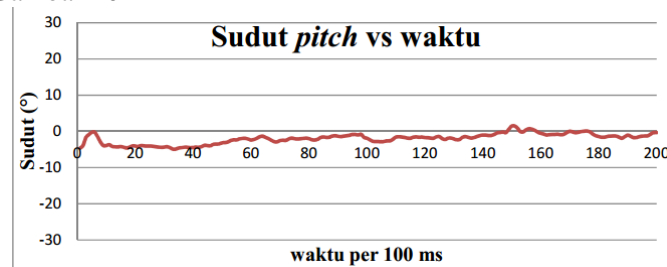
Gambar 9 Hasil pengujian sudut *roll* dengan kendali LQR

Pengujian berikutnya dilakukan pada sudut *pitch*. Sama dengan pengujian sudut *roll*, perbedaannya saat pengujian statis *quadrotor* sumbu yang dimatikan yaitu sumbu x sehingga *quadrotor* mampu membentuk sudut *euler pitch*. Dilakukan pula variasi nilai Q dan R seperti pada pengujian sudut *roll*. Dari beberapa variasi nilai Q dan R kemudian didapatkan nilai K_c seperti yang ditunjukkan pada Tabel 6.

Tabel 6 Variasi nilai Q dan R pada sudut *pitch*

Q	R	K _c
$\begin{bmatrix} 0 & 0,1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	1	[0,3162 0 0 1,0004 0 0]
$\begin{bmatrix} 0 & 700 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	1	[26,4575 0 0 1,0332 0 0]
$\begin{bmatrix} 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	1	[31,6228 0 0 1,0396 0 0]
$\begin{bmatrix} 0 & 1200 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	1	[34,6410 0 0 1,0433 0 0]
$\begin{bmatrix} 0 & 1700 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	1	[41,2311 0 0 1,0513 0 0]

Dengan variasi nilai tersebut, didapatkan hasil yang terbaik pada sudut *pitch* adalah Q dengan nilai 1200 dimana quadrotor mampu mempertahankan sudut *roll* pada -2° . Seperti yang ditunjukkan pada Gambar 10



Gambar 10 Hasil pengujian sudut *pitch* dengan kendali LQR

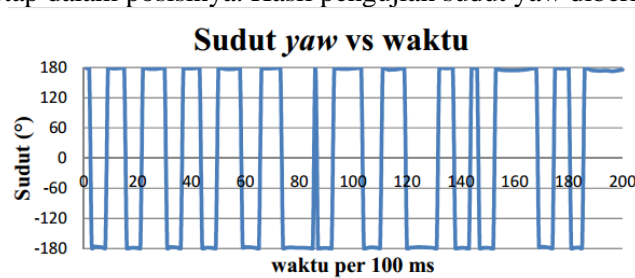
Pengujian selanjutnya adalah pengujian pada sudut *yaw*. Pada sudut *yaw*, tidak diterapkan kalman *filter* jadi pada sudut *yaw* murni kendali LQR saja. Pengujian ini dilakukan dengan menggantung *quadrotor* di titik tengah lalu keempat motor dinyalakan, *quadrotor* akan berputar membentuk sudut *euler yaw* dan diamati bagaimana sikapnya jika diberikan variasi nilai Q dan R.

Dari beberapa variasi nilai Q dan R didapatkan variasi nilai terbaik seperti yang ditunjukkan dalam Tabel 7.

Tabel 7 Variasi nilai Q dan R pada sudut *yaw*

Q	R	K _c
$\begin{bmatrix} 0 & 0 & 0,1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	1	[0 0 0 0,3162 0 1,0005]
$\begin{bmatrix} 0 & 0 & 700 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	1	[0 0 26,4575 0 0 1,0379]
$\begin{bmatrix} 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	1	[0 0 31,6228 0 0 1,0451]
$\begin{bmatrix} 0 & 0 & 1500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	1	[0 0 38,7298 0 0 1,0550]
$\begin{bmatrix} 0 & 0 & 1500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 500 \end{bmatrix}$	1	[0 0 38,7298 0 0 22,3632]

Dari hasil variasi nilai Q dan R didapatkan hasil terbaik pada sudut *yaw* yaitu nilai Q = 1500,500 dan R = 1 dengan hasil quadrotor mampu mempertahankan sudut *yaw* pada sudut -179° dan 179° tidak menjadi masalah karena kedua sudut tersebut berimpit sehingga *quadrotor* hampir tetap dalam posisinya. Hasil pengujian sudut *yaw* diberikan pada Gambar 11



Gambar 11 Hasil pengujian sudut *yaw* dengan kendali LQR

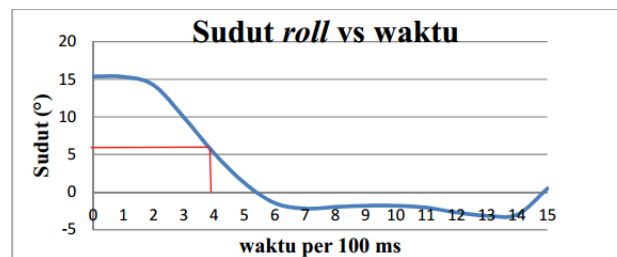
3.3.2 Hasil pengujian kecepatan respon

Pengujian terakhir adalah pengujian kecepatan respon sistem. Pengujian dilakukan pada dua sudut *euler* yaitu sudut *roll* dan sudut *pitch*. Pengujian dilakukan dengan cara memberikan gangguan kemudian diamati seberapa cepat respon sistem untuk kembali ke keadaan semula yang dinyatakan dengan *rise time* atau juga *response time*. Dengan pengujian ini, telah mewakili keadaan *quadrotor* ketika terbang.

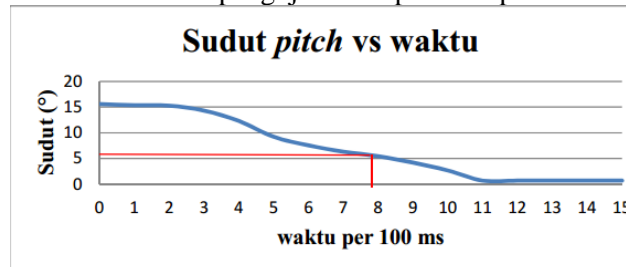
Response time adalah waktu yang dibutuhkan sistem untuk mencapai keadaan 63,2% dari keadaan stabilnya dilambangkan dengan τ [9]. Untuk *rise time* ditentukan pada kisaran 5%-95% [10], dituliskan dengan persamaan (31).

$$tr = \tau * \ln 19 \quad (31)$$

Berdasarkan eksperimen yang dilakukan pada sudut *roll* dan sudut *pitch*, diberikan gangguan 15° kemudian diamati kecepatan responnya. Dilakukan pada sudut *roll* dan sudut *pitch* karena sikapnya sama sehingga dapat dibandingkan. Hasil pengujian kecepatan respon diberikan pada Gambar 12 untuk sudut *roll* dan Gambar 13 pada sudut *pitch*



Gambar 12 Grafik pengujian kecepatan respon sudut *roll*



Gambar 13 Grafik pengujian kecepatan respon sudut *pitch*

Dari 12 dan 13 sistem mencapai 63,2% pada sudut *roll* dicapai dalam waktu 0,038 detik pada sudut $5,66^\circ$ sedangkan pada sudut *pitch* dicapai dalam waktu 0,077 detik pada sudut $5,7^\circ$. Dari data tersebut, maka *rise time* dapat dicari dengan persamaan (31). Didapatkan *rise time* sebagai berikut

$$rise\ time\ roll = 0,038 * \ln 19 = 0,11\ \text{detik}$$

$$rise\ time\ pitch = 0,077 * \ln 19 = 0,23\ \text{detik}$$

Kecepatan respon sudut *roll* lebih cepat dari pada kecepatan respon sudut *pitch* namun dengan data *rise time* yang didapat baik pada sudut *roll* maupun sudut *pitch*, mampu menangani gangguan ketika terbang dan membuat *quadrotor* mampu mempertahankan sudut *euler*.

4. KESIMPULAN

1. Telah berhasil dirancang dan dibuat sistem kendali LQR dan kalman *filter* pada *quadrotor* untuk mempertahankan sudut *euler* (*roll* dan *pitch*) serta sudut *yaw* dengan menggunakan kendali LQR.
2. Hasil dari penalaan kalman *filter* pada sudut *roll* yaitu $Q_f = 1.2$ dan $R_f = 1$ serta pada sudut *pitch* $Q_f = 1.1$ dan $R_f = 1$. Nilai eror pembacaan sudut *roll* dengan *sensor fusion* pada sudut 0° adalah $0,49^\circ$ dan pada sudut 10° adalah $0,29^\circ$ sedangkan dari estimasi kalman filter eror pembacaan sudut *roll* dengan pada sudut 0° adalah $0,47^\circ$ dan pada sudut 10° adalah $-0,06^\circ$. Nilai eror yang dihasilkan dari pengukuran sudut *pitch* dengan *sensor fusion* pada sudut 0° adalah $1,02^\circ$ dan pada sudut 10° adalah $0,53^\circ$ sedangkan dari estimasi kalman *filter*

pada sudut 0° adalah 0.98° dan pada sudut 10° adalah 0.05° . Dari hasil tersebut estimator kalman *filter* berhasil mengurangi kesalahan pengukuran bacaan sudut.

3. Hasil dari penalaan pada kendali LQR diperoleh hasil dengan $Q = 1000$ $R = 1$ pada sudut roll yang membuat *quadrotor* mampu mempertahankan sudut euler pada sudut 0.35° , untuk sudut pitch nilai $Q = 1200$ dan $R = 1$ yang mampu mempertahankan sudut euler pitch pada sudut -2° serta pada sudut yaw nilai $Q = 1500,500$ dan $R = 1$ dimana *quadrotor* dapat mempertahankan sudut euler yaw pada sudut 179° dan 179° .
4. *Rise time* pada sudut *roll* = 0,11 detik dan pada sudut *pitch* = 0,23 detik maka *quadrotor* dapat menangani gangguan dengan cepat

5. SARAN

1. Penambahan *state* kecepatan sudut dan sudut *yaw* pada estimator kalman *filter*.
2. Penambahan *state* kendali posisi translasi agar *quadrotor* lebih mudah dikendalikan.

DAFTAR PUSTAKA

- [1] Miguel, J. and Domingues, B. "Quadrotor prototype," Universidade Tecnica de Lisboa, 2009.
- [2] Mohammadbagheri, A., Zaeri, N., and Yaghoobi, M. "Comparison Performance Between PID and LQR Controllers for 4- leg Voltage-Source Inverters," vol. 7, pp. 230–234, 2011.
- [3] Minh, L. D. and Ha, C. "Modeling and Control of Quadrotor MAV Using Vision-based Measurement," pp. 1–6, 2010.
- [4] Hoffmann, F., Goddemeier, N., and Bertram, T., "Attitude estimation and control of a quadcopter," pp. 1072–1077, 2010.
- [5] Dhewa, O. A. "Implementasi Metode LQR (Linear Quadratic Regulator) pada Quadrotor dengan Penalaan Q dan R untuk Keadaan Hovering," Universitas Gadjah Mada, 2014.
- [6] Corke, P. *Robotics , Fundamental Algorithms in MATLAB ®*, vol. 73. Berlin: Springer, 2011.
- [7] Prouty, R., 2002, *Helicopter Performance, Stability, and Control*, Krieger Publishing Company.
- [8] Rodliyah, D. "Perancangan sistem kendali optimal multivariabel linear quadratic gaussian (lqg) pada kapal fpb 38 untuk meningkatkan performansi manuvering," 2011.
- [9] Katsuhiko, O. *Modern Control Engineering (Ogata 3rd Edition)*, 3 rd editi. Minnesota, 1996.
- [10] Kardono, "Perancangan dan Implementasi Sistem Pengaturan Optimal LQR untuk Menjaga Kestabilan Hover pada Quadcopter," vol. 1, no. 1, pp. 1–6, 2012.