

PEMODELAN DAN ANALISA YAW-LOCK INPUT MIXER TERHADAP FREE-RUDDER HOVERING MULTICOPTER

Helmy Mukti Himawan¹⁾, Panji Peksi Branjangan²⁾

¹ Teknik Mekatronika, Politeknik Kota Malang

email: helmymukti@poltekmalang.ac.id

² Teknik Mekatronika, Politeknik Kota Malang

email: panchee_art10@yahoo.com

Abstract

The main purpose of this research is to compare the response of a modeling and quadcopter implementation when it's in normal position and given a yaw-lock mixer input with some PID control parameters. Yaw is one of the movable axis parallel to the axis of the earth magnet. The yaw interference comes from the rotor imbalance as the main inertia generator, thus causing a drift effect on yaw. The yaw-lock mixer input system is installed to keep the yaw in one quadcopter heading position as well as the setpoint tracking. One of the advantages of keeping yaw stability is when mounted dolphin 2 fpv camera axis (roll-pitch), with free-rudder image control generated will consist stable.

The method used in this research are; the quadcopter is dynamically modeled so that hypotheses can be drawn about the quadcopter response. After obtained the quadcopter response formulation, then do the sampling of magnetometer data. Based on the formulation and the sampling data, a decision was made regarding the control technique that became the yaw-lock mixer input to the quadcopter board. The graphic result of reading control response then analyzed the efficiency and the level of "disturbance cancellation".

Keywords: quadcopter, yaw-lock, rudder, PID, inersia

1. PENDAHULUAN

Pada dekade terakhir, *Unmanned Aerial Vehicle* (UAV) menjadi topik yang menarik di beberapa organisasi riset. Jenis UAV yang lebih banyak digunakan sebagai obyek riset adalah quadrotor atau lebih dikenal dengan *quadcopter*. *Quadcopter* merupakan jenis UAV dengan menggunakan empat baling-baling sebagai penggerak utamanya^[1]. Berbagai aspek sangat mempengaruhi performa *quadcopter* antara lain desain, kontrol dan navigasi. Dari jenis beratnya, *quadcopter* termasuk salah satu jenis UAV yang lebih berat dari udara. Dari kondisi tersebut maka dengan keadaan lingkungan yang sama, *quadcopter* relatif lebih sulit dikendalikan daripada tipe UAV yang lebih ringan dari udara misalnya zeppelin atau balon udara. Desainer *quadcopter* lebih banyak menggunakan IMU (Inertial Measurement Unit) sebagai komponen kontrol *quadcopter*. Namun beberapa peneliti juga ada yang menggunakan komponen lain seperti vision-based control.

Vision control memungkinkan fpv / kamera yang berbasis *quadcopter* mempertahankan posisi yaw pada gimbal 2 axis (*roll-pitch*). Pada penelitian sebelumnya yang dilakukan Ostrowski disimpulkan bahwa kontrol

quadcopter berbasis vision merupakan kontrol yang efektif. Namun terdapat kekurangan yaitu ketika level cahaya berubah maka kontrol *quadcopter* juga akan terganggu^[2].

Penelitian yang menggunakan Image sebagai controller dapat bekerja secara efektif namun dengan unit pemroses data yang sama, image memiliki kelemahan respon yang cukup lebih lamban jika dibandingkan dengan IMU sensor unit^[3]. Penelitian yang menyajikan formulasi dynamic control *quadcopter* menggunakan data *quadcopter* yang sudah tersedia pada library simulasi maupun penelitian sebelumnya. Padahal tiap *quadcopter* memiliki perbedaan perilaku tergantung dari banyak faktor mulai dari internal maupun eksternal^[4]. Teknik untuk mengendalikan *quadcopter* yang cocok tergantung dari karakteristik *quadcopter* yang akan dikontrol^[5]. Dari beberapa penelitian tersebut, maka dapat disimpulkan bahwa obyek *quadcopter* memiliki perilaku dan respon kontrol yang berbeda. Dengan keadaan tersebut maka diperlukan kontrol sistem *quadcopter* dengan menggunakan parameter yang didapatkan dari model *quadcopter* sebenarnya.

Dari parameter *quadcopter* yang diperoleh, terdapat satu variabel yaitu yaw yang dapat

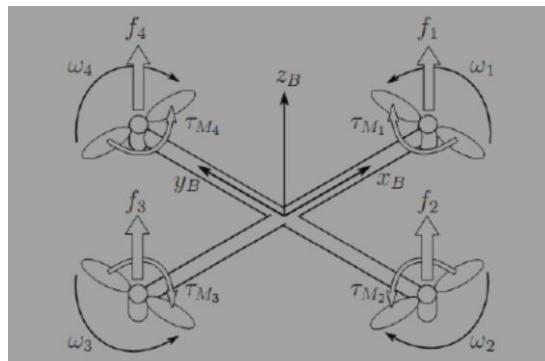
berubah pada saat terjadi gangguan internal maupun eksternal. Gangguan eksternal yaitu terjadi karena faktor alam misalnya; angin, tekanan udara, suhu, kelembaban, dll. Sedangkan gangguan internal dapat terjadi jika aktuator mengalami ketidakseimbangan parameter inersia.

Untuk mengatasi gangguan tersebut maka diperlukan unit umpan-balik untuk menjaga heading *quadcopter* pada orientasi yang diinginkan. Penggunaan magnetometer yang dipasang sebagai mixer sinyal input diharapkan dapat mengatasi gangguan tersebut.

2. TINJAUAN PUSTAKA

2.1 Quadcopter

Quadcopter atau quadrotor adalah helicopter dengan empat rotor. Rotor tersebut diarahkan ke atas dan diletakkan dalam formasi kotak dengan jarak tertentu ke pusat massa dari *quadcopter*. *Quadcopter* dikendalikan dengan cara menyesuaikan kecepatan sudut tiap rotor yang dihasilkan oleh motor elektrik. *Quadcopter* termasuk ke dalam UAV kecil karena strukturnya yang cukup sederhana. *Quadcopter* digunakan dalam proses pemantauan, search and rescue, inspeksi konstruksi dan beberapa aplikasi lainnya. Berhubungan dengan struktur *quadcopter* tersebut, maka saat ini penelitian di bidang robotik banyak yang menggunakan *quadcopter* sebagai objek penelitian. Analisis dan model dinamik merupakan titik awal dari keseluruhan pembelajaran tentang *quadcopter*. Beberapa kasus meningkatkan kompleksitas dan aerodinamika. Beberapa metode kontrol yang berbeda telah dipelajari seperti PID controller, LQR controller, Back Stepping dan kontrol nonlinear lainnya.



Gambar 1. Struktur dan gaya keseluruhan aktuator quadcopter

Saat ini, banyak studi tentang bagaimana memilih metode kontrol yang cocok

berdasarkan kondisi dengan efisiensi yang tinggi. Keempat gaya yang dihasilkan rotor seharusnya mengendalikan ketinggian dengan baik [6].

Model dari *quadcopter* yang digunakan sesuai dengan metode Newton-Euler dan persamaan (1) menyatakan persamaan total dari metode tersebut.

$$mV_B + V \times (mV_B) = R^T \cdot G + T_B \quad (1)$$

Di mana:

m : massa total

V_B : Kecepatan linier

V : Kecepatan Sudut

R : rotation matrix

G : Percepatan gravitasi

T_B : Jumlah Gaya total

Model tersebut mengacu pada hukum Newton dan selanjutnya dilakukan penurunan persamaan berdasarkan notasi dalam desain sketsa *quadcopter* yang akan dirancang. Berdasarkan Hukum newton kedua, gerakan translasional dapat dituliskan dalam formula [7]:

$$F = m(v + (\omega \times mv)) \quad (2)$$

Di mana:

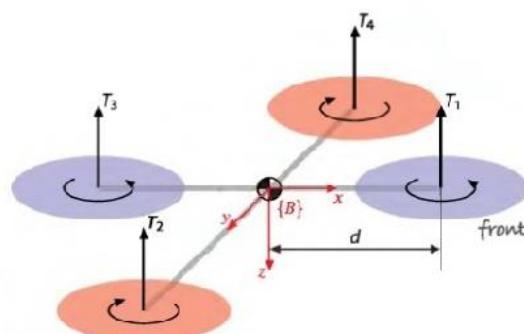
F : Gaya

m : massa

v : kecepatan linier

ω : kecepatan sudut

Notasi *quadcopter* berdasarkan model Corke's adalah sebagai berikut:



Gambar 2. Notasi *quadcopter* berdasarkan model Corke's

Dari sketsa notasi tersebut maka didapatkan:

$$F = F_g - F_{\text{thrust}} \quad (3)$$

$$F = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - wR_B \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \quad (4)$$

Di mana m adalah massa dari *quadcopter*, T adalah gaya vertikal *quadcopter*, dan wR_B adalah rotasi matrix dari *body-frame* atau *Inertial frame*. Dari persamaan di atas maka akan didapatkan formulasi model gerakan yaw yang dibutuhkan untuk memodelkan sistem *quadcopter* dalam kebutuhan analisis data.

2.2 Kontrol Proportional Integral Derivative (PID)

Kontrol PID selalu didasari atas tinjauan terhadap karakteristik yang diatur (*Plant*). Dengan demikian betapapun rumitnya suatu *Plant*, perilaku *Plant* tersebut harus diketahui terlebih dahulu sebelum penalaan parameter PI itu dilakukan.

- **Plant**

Fungsi transfer dari *Plant* orde 1 dituliskan dalam domain S seperti pada persamaan berikut [8]:

$$G(s) = \frac{K}{TpS + 1} \quad (5)$$

- **Controller**

Fungsi transfer dari *Controller* tipe PID dituliskan dalam domain S seperti pada persamaan berikut [8]:

$$Gc(s) = \frac{K_p(\tau_i s + 1)}{\tau_i s} \quad (6)$$

- **Hope**

Fungsi transfer dari hope yang berupa persamaan orde 1 dituliskan dalam domain S seperti pada persamaan berikut [8]:

$$Gh(s) = \frac{1}{Ths + 1} \quad (7)$$

- **Analisa Close Loop**

Persamaan *Plant* dan *Controller* digunakan untuk melakukan analisa *close loop*, yang kemudian digunakan untuk menentukan parameter kontrol PI. Rumus dasar untuk analisa *close loop* adalah [8]:

$$Gcl(s) = \frac{Gp(s).Gc(s)}{1 + Gp(s).Gc(s)} \quad (8)$$

Persamaan $Gc(s)$ dan $Gp(s)$ dimasukkan ke dalam persamaan $Gcl(s)$ [8]:

$$Gcl(s) = \frac{\frac{K}{TpS + 1} \times \frac{K_p(\tau_i s + 1)}{\tau_i s}}{1 + \frac{K}{TpS + 1} \times \frac{K_p(\tau_i s + 1)}{\tau_i s}} \quad (9)$$

$$Gcl(s) = \frac{KK_p(\tau_i s + 1)}{\tau_i S(\tau_i S + 1)KK_p(\tau_i S + 1)} \quad (10)$$

$$\tau_h = \frac{\tau_p}{KK_p}; K_p = \frac{\tau_p}{\tau_h K} \quad (11)$$

- **Implementasi Controller**

Fungsi transfer dalam domain s adalah sebagai berikut [8]:

$$Gcl(s) = \frac{K_p(\tau_i s + 1)}{\tau_i s} = \frac{C(s)}{E(s)} \quad (12)$$

persamaan diubah ke dalam domain t [8]:

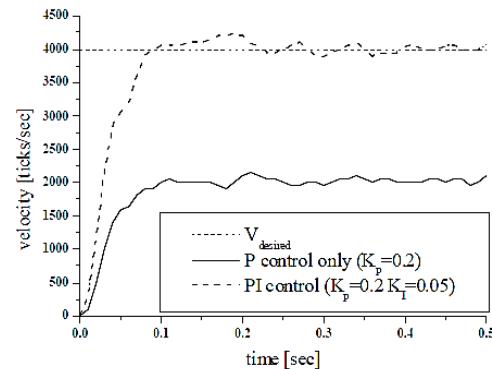
$$C(t) = K_p e(t) + \frac{K_p}{\tau_i} \int e(t) dt \quad (13)$$

Setelah itu persamaan diubah ke dalam persamaan beda [8]:

$$New_{op} = K_p * New_error$$

$$New_{oi} = \frac{K_p * Ts}{Ti} * (New_{error} + Last_{oi})$$

Dengan cara lain kegunaan *Controller* I pada *Controller* PI adalah untuk mengurangi *error steady-state* pada *Controller* P. Dengan sebuah tambahan persamaan integral. *Error steady-state* dapat dikurangi hingga zero "0" seperti pada gambar 8 di bawah ini.



Gambar 3. Respon Kontrol [8]

Ketika menggunakan $e(t)$ pada fungsi error, maka rumus PI adalah [8]:

$$R(t) = K_p \times [e(t) + \frac{1}{\tau_i} \times \int_0^t e(t) dt] \quad (12)$$

Di mana $Q_i = \frac{K_p}{\tau_i}$, maka rumus PI menjadi:

$$R(t) = K_p \times e(t) + Q_i \times \int_0^t e(t) dt \quad (13)$$

Cara yang lebih tepat dalam *Controller* PI dengan merubah ke nilai diskrit sebagai berikut [8]:

$$R(n) = K_p \times e_n + Q_i \times \Delta t \sum_{i=1}^n \frac{e_i + e_{i-1}}{2} \quad (14)$$

Sekarang dapat mengisarkan dari penjumlahan R_{n-1} , maka nilai sebelumnya [8]:

$$R_n - R_{n-1} = K_p(e_n - e_{n-1}) + Q_i \times \Delta t \left(\frac{e_n + e_{n-1}}{2} \right) \quad (15)$$

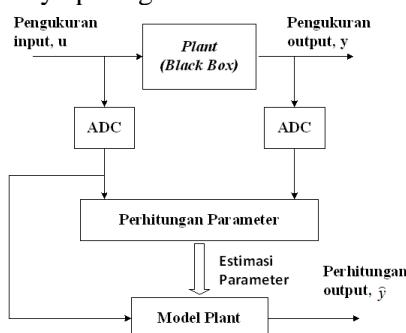
Maka:

$$R_n = R_{n-1} + K_p(e_n - e_{n-1}) + K_i \left(\frac{e_n + e_{n-1}}{2} \right) \quad (16)$$

Untuk mendapatkan sistem yang diinginkan, maka Kedua parameter gain harus ditetapkan secara *optimal* (biasa disebut *tuning mode*). Adapun metode *tuning* yang akan digunakan dalam menentukan nilai parameter adalah metode Ziegler-Nichols. *Tuning* parameter *Controller PID* selalu didasari atas tinjauan terhadap karakteristik yang diatur (*Plant*). Karena penyusunan model matematik *Plant* tidak mudah, digunakan data yang berupa kurva keluaran pendekatan model. Berdasar pendekatan itu *tuning Controller PID* dapat dilakukan, seperti yang dilakukan Ziegler Nichols dengan metode kurva reaksi dan osilasi.

2.3 Identifikasi dengan Pemodelan RLS

Ada dua cara untuk membangun model dari sebuah sistem yaitu pertama menggunakan pendekatan ilmu pengetahuan murni dengan memperhitungkan hukum-hukum fisik yang berlaku dan Menggunakan pendekatan eksperimental. Dalam banyak hal hampir tidak mungkin, untuk membangun model hanya menggunakan hukum-hukum fisik yang berkait. Beberapa parameter ternyata harus ditentukan menggunakan hasil dari eksperimen. Pendekatan ini disebut dengan identifikasi sistem, ada banyak metode untuk menganalisa data eksperimen, diantaranya dengan menggunakan metode *Recursive Least Square* atau biasa disebut dengan metode identifikasi RLS yang dapat dilihat block diagramnya pada gambar 2.



Gambar 4. Block Diagram Proses Identifikasi

Hasil estimasi untuk k observasi akan digunakan untuk memperoleh estimasi untuk observasi k+1. Persoalan yang sama akan berlaku bila jumlah parameter tidak diketahui sebelumnya. Cara ini disebut dengan cara atau

metode *Recursive Least-Squares* (RLS). Dengan algoritma RLS, maka akan diperoleh kesalahan komputasi yang lebih kecil dibanding jika kita menggunakan metode pengambilan *sampling* dengan cara lain. Kelemahan RLS adalah memerlukan proses komputasi yang lebih kompleks dari metode RLS adalah berusaha meminimalisasi nilai *cost Function* ($\epsilon(n)$), yang didefinisikan sebagai penjumlahan dari kuadrat error. Kita dapat menjabarkan nilai *cost Function* sebagai berikut [9]:

$$\epsilon(n) \sum_{i=1}^n \beta(n,i) |e(i)|^2 \quad (17)$$

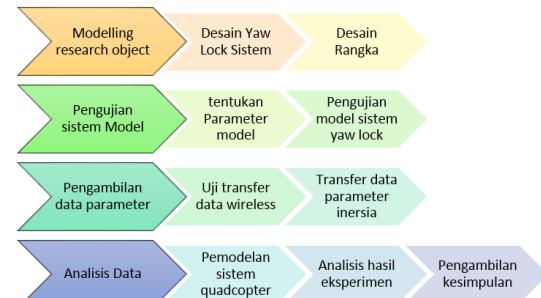
$\beta(n,i)$ melambangkan *forgetting factor* yang nilainya antara 0 dan 1. Bentuk yang biasa digunakan untuk *forgetting factor* adalah bentuk *exponensial*.

Proses adaptif algoritma RLS mempunyai step-step sebagai berikut [9]:

- Inisialisasi bobot awal filter dan $P(0)$. Nilai $P(0) = \delta - 1$ merupakan matriks identitas, δ merupakan konstanta positif yang nilainya kecil.
- Menghitung nilai estimasi error: $\xi(n) = d(n) - wH(n-1).u(n)$ (18)
- Menghitung nilai gain vector: $k(n) = \frac{\lambda^{-1}P(n-1).u(n)}{1 + \lambda^{-1}.u^H(n).P(n-1).u(n)}$ (19)
- Update nilai invers matrik autokorelasi: $P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}k(n)uH(n)P(n-1)$ (20)
- Update nilai bobot filter: $w(n) = w(n-1) + \mu.k(n).\xi^*(n)$ (21)
- Perbesar nilai n, lalu kembali ke step 2

3. METODE PENELITIAN

Untuk Sebagai tahapan penelitian yang akan dilakukan dapat diamati pada diagram berikut:



Gambar 5. Tahapan Penelitian

Pada tahap awal penelitian pemodelan pembuatan *Quadcopter* yaitu melakukan *modelling research* antara lain melakukan desain untuk yaw lock sistem dan desain rangka pada *Quadcopter*. Setelah melakukkan tersebut maka langkah selanjutnya melakukkan uji coba sebuah model dengan melakukan pemodelan matematis dari *Quadcopter* tersebut dengan menentukan parameter model dan melakukan uji model yang telah dilakukannya dengan software pemodelan yang akan dipakai.

Pemodelan *Quadcopter*

Quadcopter Dari hasil pemodelan dengan MATLAB yang telah dilakukan diatas maka akan didapatkan fungsi transfer dalam bentuk laplace seperti dibawah ini dengan memasukkan nilai spesifikasi *buck converter*. Fungsi Transfer dalam bentuk laplace

$$\frac{V_{out}}{D} = \frac{V_{in}/L.C}{s^2 + s\left(\frac{1}{R_{load}.C} + \frac{R_L}{L}\right) + \left(\frac{1}{L.C} + \frac{R_L}{R_{load}.L.C}\right)} \quad (22)$$

Dengan memasukkan parameter seperti pada tabel 1. Maka setelah itu kita akan mendapatkan grafik output dengan melakukan plot fungsi transfer:



Gambar 6. Grafik digambarkan dengan MATLAB (sumbu x: waktu, sumbu y: nilai)

Untuk perbaikan hasil maka digunakan metode Ziegler-Nichols identifikasi motor harus bisa menampilkan kurva reaksi yang berbentuk tidak berbentuk kurva S sehingga tidak dapat menggunakan metode 1, tetapi harus menggunakan metode 2 dari Ziegler-Nichols dengan mencari *ultimate gain* (K_u). Dengan melakukan metode experimental sampai mendapatkan nilai K_u yaitu 0,488. Dan nilai dari *ultimate period* (P_u) adalah 0,0793. Setelah mendapatkan nilai dari K_u dan P_u maka kita dapat menentukan berapa nilai dari K_p , K_i , dan K_d sebagai berikut:

- Parameter K_p

$$K_p = 0,6 \times K_u$$

$$K_p = 0,6 \times 0,488 = 0,2928$$

- Parameter K_i
 $T_i = 0,5 \times P_u = 0,5 \times 0,0793 = 0,03965$
 $K_i = K_p/t_i = 0,2928 / 0,03965 = 7.3846$
- Parameter K_d
 $T_d = 0,125 \times K_p = 0,0366$
 $K_d = K_p \times T_d$
 $= 0,2928 \times 0,0366$
 $= 0,0107$

Sehingga nilai masing-masing parameter didapatkan sebagai berikut:

- $K_p=0,2928$
- $K_i=7.3846$
- $K_d=0,0107$

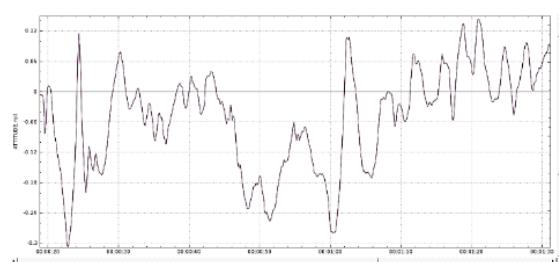
Metodologi yang digunakan dalam penelitian ini adalah merancang dan membuat basis data dan aplikasi. Pada tahap pembuatan aplikasi dan sistem basis data akan dilakukan dengan cara sebagai berikut :

- a. Identifikasi kebutuhan sistem
 - b. Membuat rancangan diagram alir (*Flowchart*)
 - c. Membuat rancangan DFD (*Data Flow Diagram*)
 - d. Membuat ERD (*Entity Relationship Diagram*)
 - e. Mengubah diagram alir (*Flowchart*) ke dalam tata bahasa pemrograman berbasis java
 - f. Mengimplementasikan program pada Netbeans IDE 8.0 menjadi sebuah aplikasi
 - g. Pengujian aplikasi
- Merevisi program jika diperlukan.

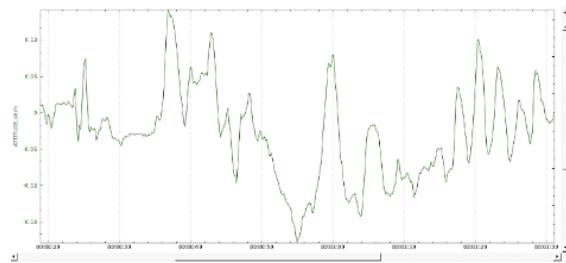
4. HASIL DAN PEMBAHASAN

4.1 Hasil Pemodelan *Quadcopter*

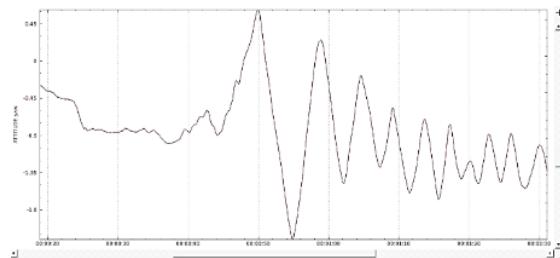
Pada Sensor IMU terdiri dari beberapa axis yaitu Roll Pitch Yaw. Pengukuran sensor dilakukan agar proses pergerakan Yaw dapat terukur dengan baik dengan membandingkan gerakan roll pitch yaw pada satu domain range yang sama.



Gambar 7. Grafik Roll dari pergerakan quadcopter board APM



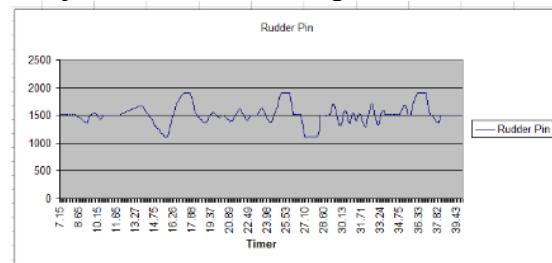
Gambar 8. Grafik Pitch dari pergerakan quadcopter board APM



Gambar 9. Grafik Yaw dari pergerakan quadcopter board APM

4.2 Hasil Pemodelan *Pulse Width Modulation* pada *Rudder Pin*

Rudder PIN merupakan sinyal yang menentukan YAW respon. Jadi setiap perubahan *duty cycle* pada rudder PIN menentukan orientasi quadcopter pada sumbu vertical. Rudder pin diperoleh dari sinyal *receiver* yang berupa PWM 50Hz dengan *duty cycle* 5-10%. Sehingga secara teori diperoleh *duty cycle* 1ms-2ms atau 1000us-2000us. Dengan hasil percobaan PIN Rudder yang dihubungkan dengan unit *telemetry*, maka didapatkan hasil sesuai dengan Grafik 10.



Gambar 10. Grafik Pembacaan Rudder PIN pada Quadcopter

4.3 Hasil Pemodelan *Respon Kontrol Rudder-Compass* pada *Objek Quadcopter*

Pada implementasi kontrol digunakan kontrol PID dengan tuning trial & error. Kontrol digital membentuk persamaan PID sebagai berikut:

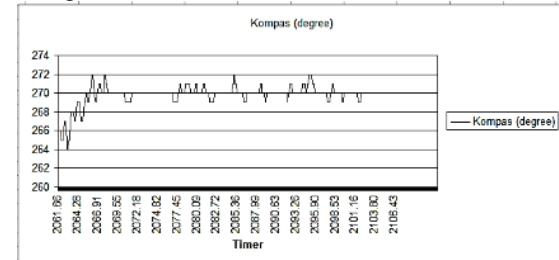
$$u(t_k) = u(t_{k-1}) + K_p \left[\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t} \right) e(t_k) + \left(-1 - \frac{2T_d}{\Delta t} \right) e(t_{k-1}) + \left(\frac{2T_d}{\Delta t} \right) e(t_{k-2}) \right] \quad (23)$$

Di mana:

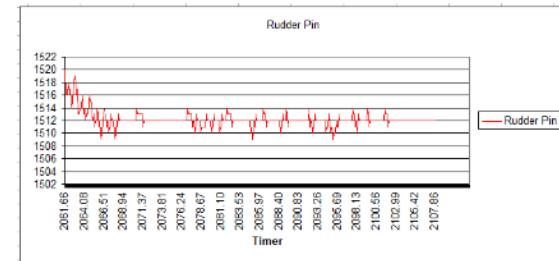
$$T_i = K_p / K_i$$

$$T_d = K_d / K_p$$

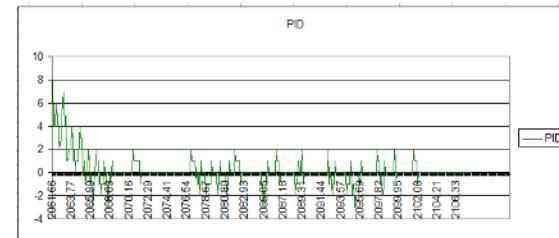
K_p adalah konstanta proporsional, K_i adalah konstanta integral, dan K_d adalah konstanta derivative. Dari hasil formula tersebut diwujudkan dalam algoritma bahasa C dengan *sampling time* ±50ms. Dari hasil kontrol PD dengan konstanta $K_p=1$ dan $K_d=1$ diperoleh grafik hasil pengukuran sebagai berikut:



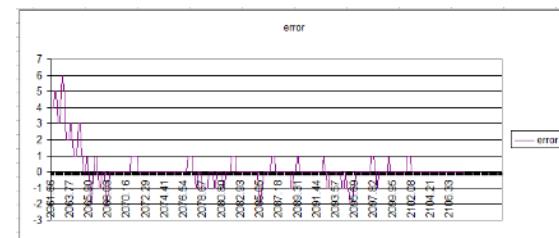
Gambar 11. Grafik PID Bearing Kompas



Gambar 12. Grafik Respon Kontrol PID Rudder PIN



Gambar 13. Grafik nilai PID dengan $K_p = 1$, $K_i = 0$, $K_d = 1$

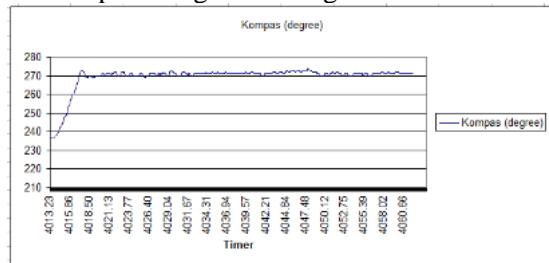


Gambar 14. Grafik Nilai Error dari *TakeOff* hingga *Hovering*

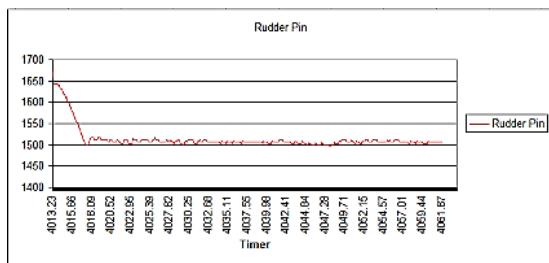
Dengan diperoleh hasil kontrol pada $K_p = 1$ dan $K_d = 1$, maka selanjutnya dilakukan pengubahan nilai masing-masing konstanta untuk memperoleh respon kontrol sesuai dengan yang diinginkan. Respon kontrol yang diinginkan adalah dengan mencapai *setting*

time secepat mungkin dan menghindari adanya osilasi.

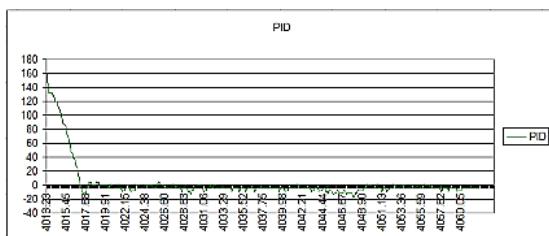
Eksperimen berikutnya adalah memberikan K_i nilai 1 dan menaikkan nilai K_p dan K_d masing-masing menjadi 2. Dengan mengubah nilai masing-masing konstanta PID tersebut maka diperoleh grafik sebagai berikut.



Gambar 15. Nilai sensor Kompas pada $K_p=2, K_i=1, K_d=2$



Gambar 16. Respon kontrol rudder pin pada $K_p=2, K_i=1, K_d=2$



Gambar 18. Nilai PID respon pada $K_p=2, K_i=1, K_d=2$

Dari hasil kontrol tersebut diperoleh data PID yang memiliki respon lebih baik dari pada kontrol PD. Dengan menambahkan K_i sebagai parameter PID maka respon *overshoot* dapat diminimalisasi dan diperoleh respon kontrol yang halus.

Dari keterangan grafik tersebut dapat dianalisis bahwa baik kontrol PD maupun PID dapat memberikan respon kontrol dengan stabil pada rudder. Hal ini disebabkan parameter gangguan pada YAW tidak sebesar *roll pitch* dan *altitude*. Namun dengan adanya eksperimen kontrol rudder terhadap sinyal yaw dapat menjadi acuan untuk kontrol sinyal quadcopter yang lainnya.

5. KESIMPULAN

Dari uji coba simulasi yang telah dilakukan maka dapat diambil beberapa kesimpulan. Pertama uji coba yang dilakukan dari hasil simulasi didapatkan hasil bahwa sistem telah dapat dilakukan pengiriman data secara realtime dari sensor dan joystick yaitu dengan menggunakan telemetry unit sinyal radio 915MHz dan pemrograman mikrokontroler untuk memisahkan sinyal yang dikirim ke board control dan telemetry unit. Kedua bahwa untuk mengetahui respon kontrol YAW dapat dilakukan dengan mengamati perilaku sinyal YAW dan Joystick rudder dalam satu domain waktu yang sama dengan membandingkan kedua sinyal hasil pengukuran tersebut. Ketiga respon kontrol PID pada sinyal rudder diperoleh hasil optimal pada $K_p = 2$ $K_i = 1$ dan $K_d=2$, dengan setpoint 270° dan sinyal rudder center pada 1512us.%.

6. REFERENSI

- [1] Gupte, Shweta, dkk. 2012. "A Survey of Quad-Rotor Unmanned Aerial Vehicles". 2012 Proceedings of IEEE Southeastcon.
- [2] Ostrowski, J. P. and Taylor, C. J. 2015. "Control of a Quad rotor Helicopter Using Dual Camera Visual Feedback". IEEE International Conference on vol 1.
- [3] Sathiyabama, G., dkk. 2014. "Controller Design for Quadcopter Using Labview With Image Processing Techniques". International Journal for Research and Development in Engineering (IJRDE).
- [4] Nemati, A., and Kumar, M. 2014. "Modeling and Control of a Single Axis Tilting Quadcopter". IEEE:American Control Conference (ACC).
- [5] Argentim, L. M., dkk. 2013. "PID , LQR and LQR-PID on a Quadcopter Platform". IEEE: Informatics, Electronics & Vision (ICIEV). 2013 International Conference.
- [6] Fatan, M., dkk. 2013. "An Adaptive Neuro PID for Controlling the Altitude of Quadcopter Robot Output of Plant". MIN-Fakultät Fachbereich Informatik - Universität Hamburg.
- [7] Tunay, H., dkk. 2013. "Auto-Tuning Quadcopter Using Loop Shaping". pp. 111–115.
- [8] Gunterus, Frans. 1994. *Pengenalan Metode Ziegler-Nichols pada Perancangan Controller pada PID*. Jakarta: PT. Elex Media Komputindo.