

# Sistem Kendali Kestabilan Quadcopter dengan Navigasi Manual Berbasis Pengendali PID

Sani Saffanah<sup>1</sup>, Kartono Wijayanto<sup>2</sup>, Adnan Rafi Al Tahtawi<sup>3</sup>

<sup>1,2,3</sup>Program Studi D4 Teknik Otomasi Industri, Jurusan Teknik Elektro, Politeknik Negeri Bandung  
Jl. Gegerkalong Hilir, Ciwaruga, Bandung, Indonesia  
sanisaffanah.99@gmail.com<sup>1</sup>

---

---

## Abstrak

*Quadcopter* merupakan salah satu jenis pesawat tanpa awak dengan jenis *rotary wing* menggunakan *propeller* untuk dapat terbang. Saat ini, *quadcopter* biasa digunakan untuk fotografi udara, pemetaan, dan lain-lain. Pada umumnya, permasalahan yang sering dihadapi ketika *quadcopter* mengudara yaitu adanya guncangan atau gerakan yang mengakibatkan *quadcopter* tidak stabil. Navigasi pada *quadcopter* juga diperlukan untuk mengetahui posisi ketika *quadcopter* mengudara. Pada Tugas Akhir ini merancang dan merealisasikan sistem kendali kestabilan menggunakan PID pada *quadcopter* dengan navigasi manual. Sistem ini dioperasikan menggunakan *remote control*. Sistem ini menggunakan dua buah mikrokontroler, dimana mikrokontroler pertama yaitu Ardupilot Mega 2.8 berfungsi sebagai *flight controller*, sedangkan mikrokontroler kedua menggunakan Arduino Nano sebagai penyimpanan data *logger* dari sensor. Sensor yang digunakan adalah sensor MPU6050 untuk membaca sudut roll, pitch, dan yaw ketika *quadcopter* mengudara. Adanya modul GPS digunakan untuk dapat mengetahui posisi dari *quadcopter*. *Quadcopter* berhasil dirancang menggunakan kendali PID dan Ardupilot sebagai *flight controller* dengan menggunakan metode *trial and error*. Berdasarkan hasil pengujian, pengendalian pergerakan sudut roll ( $\phi$ ) dan pitch ( $\theta$ ) mendapatkan nilai  $K_p=0,15$ ;  $K_i=0,05$ ; dan  $K_d=0,08$ . Pengendalian sudut yaw ( $\psi$ ) mendapatkan nilai  $K_p=0,2$ ;  $K_i=0,02$ ; dan  $K_d=0,00$ . Dengan menggunakan kendali PID tersebut, kontroler mampu menghasilkan respon yang stabil dengan estimasi *settling time* pada sudut roll, pitch, dan yaw sebesar 1,07 s; 0,41 s; dan 0,35 s.

**Kata kunci:** *Quadcopter*, Sistem Kendali, Kestabilan, PID

---

---

## I. PENDAHULUAN

UAV (*Unmanned Aerial Vehicle*) adalah robot penjelajah udara tanpa awak yang dapat terbang dengan menggunakan gaya aerodinamik untuk menghasilkan gaya angkat. *Quadcopter* merupakan salah satu jenis pesawat tanpa awak dengan tipe multirotor yang memiliki 4 buah rotor untuk dapat menggerakkan *propeller*. *Propeller* tersebut berfungsi untuk memberikan gaya angkat pada *quadcopter* sehingga dapat terangkat keatas. *Quadcopter* dapat terbang secara autonomous atau dioperasikan langsung dengan *remote control*. Saat ini, *quadcopter* biasa digunakan untuk fotografi udara, peliputan kemacetan, pemetaan, pertanian, bahkan digunakan di industri untuk mengantar sebuah barang, dan masih banyak lagi kegunaan lainnya. Permasalahan yang sering dihadapi yaitu saat *quadcopter* di udara, memungkinkan terjadinya

guncangan atau gerakan yang mengakibatkan *quadcopter* menjadi tidak stabil. Kestabilan pada *quadcopter* dapat terganggu dikarenakan perbedaan daya angkat pada keempat motor, yang disebabkan oleh berbagai faktor diantaranya adalah efek aerodinamis, distribusi berat, dan gangguan luar seperti kecepatan angin. Agar *quadcopter* dapat terbang secara optimal, maka stabilisasi terbang *quadcopter* perlu dipertahankan dengan menggunakan pengendali yang tepat. Untuk bisa mengetahui posisi dan arah dari *quadcopter* itu sendiri juga diperlukan sebuah navigasi yang cocok.

Pemodelan dan perancangan sistem kendali kestabilan pada *quadcopter* telah diteliti oleh peneliti sebelumnya, seperti pemodelan sistem kendali yang bekerja pada saat *quadcopter* terbang melayang menggunakan sistem kendali PD [1] dengan Nilai  $K_p=25$  dan  $K_d=35$  yang kemudian disimulasikan di MATLAB. Pemodelan lain juga

dibuat namun menggunakan pengendali PID [2] yang kemudian diimplementasikan pada Ardupilot. Dirancang juga sebuah *quadcopter* menggunakan Arduino Mega dengan metode PI [3] yang sebelumnya sudah dilakukan pemodelan menggunakan fungsi *pidtune* dengan pendekatan *crossover frequency*, namun pengendalian ini memfokuskan pada gerakan pitch dan roll dari *quadcopter*. Hasil nilai parameter PID yang didapatkan dari penelitian ini adalah  $K_p = 0,0316$ ;  $K_i = 0,05$ ; dan  $K_d = 0,5$ . Arduino Uno [4] juga digunakan sebagai kontroler pada *quadcopter* dan berhasil membuat *quadcopter* terbang dengan stabil. Penelitian lainnya mengenai sistem kendali kestabilan pada *quadcopter* dengan mempertahankan ketinggian tertentu juga dibahas di beberapa jurnal. Metode PI [5] digunakan untuk pemodelan sistem pada *quadcopter* agar dapat mempertahankan ketinggian sesuai dengan apa yang diinginkan (*altitude hold*). Sedangkan adapula yang memodelkan dan mengimplementasikan sistem kendali ketinggian *quadcopter* menggunakan PID Ziegler Nichols [6] dengan metode osilasi. Penambahan sensor SRF02 [7] digunakan untuk memastikan ketinggian pada *quadcopter* dan menggunakan kontroler Arduino pro micro sebagai flight controller.

Berdasarkan penelitian sebelumnya, masalah utama yang masih terjadi yaitu kestabilan *quadcopter* yang perlu diteliti kembali untuk mendapatkan hasil yang maksimal. Adanya navigasi pada *quadcopter* juga diperlukan untuk bisa mengetahui posisi dan arah dari *quadcopter*. Navigasi pada *quadcopter* ini berfungsi untuk mengetahui posisi *quadcopter*, ketika *quadcopter* hilang kendali. Maka, penulis memutuskan untuk merancang dan membangun sistem kendali kestabilan menggunakan PID berbasis mikrokontroler pada *quadcopter* agar mendapatkan hasil terbang yang stabil, juga adanya navigasi manual pada sistem *quadcopter*. Perbedaan penelitian ini dengan penelitian yang sudah dikemukakan diatas yaitu rancang bangun *quadcopter* dilakukan tanpa membuat model terlebih dahulu. Pergerakan kestabilan dari *quadcopter* juga dipantau melalui sensor MPU6050 yang dikontrol oleh Arduino Nano.

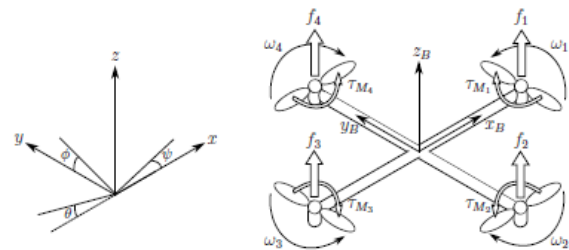
## II. METODE PENELITIAN

Pembahasan dalam penelitian ini bertujuan untuk mendapatkan kendali kestabilan yang optimal pada *quadcopter* menggunakan metode PID.

### II.1 Analisis Sistem

*Quadcopter* yang digunakan menggunakan model X yang disusun secara simetris dengan 4 buah rotor. Data masukan mengenai orientasi *quadcopter* didapat dari nilai sumbu x dan sumbu y yang berisi informasi kemiringan relatif dari *quadcopter*. Data tersebut yang nantinya akan mempengaruhi kecepatan dari motor. Sedangkan navigasi manual pada *quadcopter* berfungsi untuk mengetahui posisi dari *quadcopter*. Data yang dikirimkan dari GPS yaitu berupa data longitude dan latitude.

Struktur *quadcopter* ditunjukkan pada Gambar 1 yang termasuk kecepatan sudut yang sesuai, torsi, dan gaya yang diciptakan oleh empat motor [8][9].



Gambar 1. Bingkai Inersia dan Tubuh *Quadcopter* [9]

Ketika *quadcopter* terbang dalam ruang tiga dimensi, terdapat dua sistem koordinat pada *quadcopter*. Koordinat pertama yaitu bingkai inersia seperti yang ditunjukkan pada gambar sebelah kiri. Pada model tersebut menggambarkan kerangka koordinat bumi atau koordinat titik referensi yang digunakan untuk *quadcopter*. Gambar sebelah kanan menunjukkan koordinat kedua, merupakan kerangka tubuh yang berarti koordinat yang bergerak bersama-sama dengan tubuh *quadcopter*. Pada Gambar 1. memperlihatkan bahwa motor bagian depan (penghasil gaya  $f_1$ ) dan belakang (penghasil gaya  $f_3$ ) berputar berlawanan arah jarum jam, sedangkan motor bagian kanan (penghasil gaya  $f_2$ ) dan kiri (penghasil gaya  $f_4$ ) berputar searah jarum jam yang membuat semua baling-baling berputar pada kecepatan yang sama.

Sistem yang bekerja meliputi input, proses, dan output. Input pada sistem penerbangan ini adalah sensor IMU (*gyroscope* dan *accelerometer*) dan receiver. Input bertugas untuk memberikan masukan yang nanti akan diproses oleh mikrokontroler. Input pada kontroler yang digunakan untuk mengetahui arah pergerakan dari *quadcopter* yaitu sensor IMU. Pada Ardupilot sudah terdapat sensor IMU didalamnya. Pada bagian proses, digunakan Ardupilot sebagai flight controller, sedangkan pada output terdapat 4 ESC (*Electronic Speed Control*) yang masing-masing terhubung pada motor brushless dc. ESC digunakan

untuk mengendalikan kecepatan motor BLDC yang bekerja secara cepat untuk menghidupkan atau mematikan pulsa ke motor. ESC dikatan juga sebagai driver motor dengan mengeluarkan pulsa berupa sinyal PWM (*Pulse Width Modulation*).

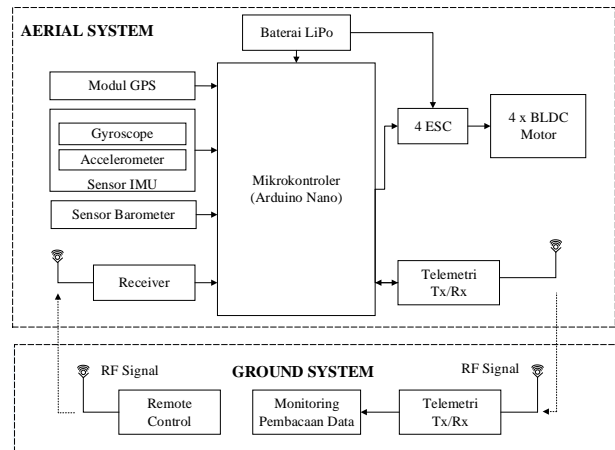
Sedangkan untuk Arduino Nano sendiri berfungsi untuk membaca kemiringan sudut yang dihasilkan dari *quadcopter*. Arduino Nano membaca respon *quadcopter* ketika posisi terbang, ini dikarenakan Ardupilot tidak bisa membaca data ketika *quadcopter* mengudara. Input yang masuk ke Arduino Nano yaitu sebuah receiver, sensor MPU6050, dan sebuah modul GPS. Modul GPS digunakan untuk navigasi yang menyediakan informasi lokasi dan receiver digunakan untuk menerima sinyal dari *remote control*. Sinyal dari *remote control* ini yang nantinya akan diteruskan ke mikrokontroler. Sedangkan untuk mendapatkan informasi mengenai pembacaan sudut kemiringan dan juga lokasi dari *quadcopter*, maka digunakan telemetry.

## II.2 Perancangan Sistem

Perancangan sistem dilakukan, agar penelitian yang dikerjakan sesuai dengan perencanaan yang mendasar. Sistem yang akan dirancang pada penelitian ini yaitu sistem kendali sikap terbang *quadcopter*.

### II.2.1 Arsitektur Sistem

Sitem ini menggunakan dua buah mikrokontroler, dimana mikrokontroler pertama yaitu Ardupilot berfungsi sebagai flight controller. Sedangkan mikrokontroler kedua menggunakan Arduino Nano sebagai penyimpanan data *logger* dari sensor. Hal ini dilakukan, karena Ardupilot tidak bisa menyimpan dan membaca data dari sensor kemiringan yang sudah ada pada Ardupilot itu sendiri, sehingga Arduino Nano digunakan untuk pembacaan sensor ketika *quadcopter* sedang mengudara. Secara keseluruhan, sistem dapat diilustrasikan oleh arsitektur sistem yang ada pada Gambar 2.



Gambar 2. Arsitektur Sistem

Pada Gambar 2, sistem dibagi menjadi dua bagian, yaitu sistem di udara (*aerial system*) dan di darat (*ground system*). Pada bagian aerial sistem merupakan *quadcopter* itu sendiri dan dirancang menggunakan dua buah mikrokontroler. Mikrokontroler yang digunakan sebagai flight controller yaitu Ardupilot, dimana didalamnya terdapat sensor IMU 6 DOF (*accelerometer* dan *gyroscope*). Sedangkan pada mikrokontroler kedua terdapat Arduino Nano yang berfungsi sebagai pemantauan kendali dan pemantauan data terhadap gerakan dari *quadcopter* itu sendiri. Telemetry yang terhubung pada Arduino Nano berfungsi untuk merekam data gerakan dari *quadcopter* yang nantinya akan dikirimkan ke perangkat laptop di *ground system*. Daya yang diperoleh sistem berasal dari baterai LiPo 3 sel.

Pada bagian *ground system*, terdapat *remote control* yang nantinya akan dioperasikan oleh pilot. *Remote control* akan terhubung dengan receiver pada *quadcopter* melalui frekuensi radio. Sedangkan monitoring pembacaan data dilihat pada perangkat laptop yang terhubung dengan telemetry. Data yang terbaca yaitu berupa sudut kemiringan dan posisi (*longitude* dan *latitude*) dari *quadcopter* tersebut. Komunikasi berlangsung secara satu arah. Adapun komponen yang digunakan dalam perancangan sistem ini ditunjukkan pada Tabel 1.

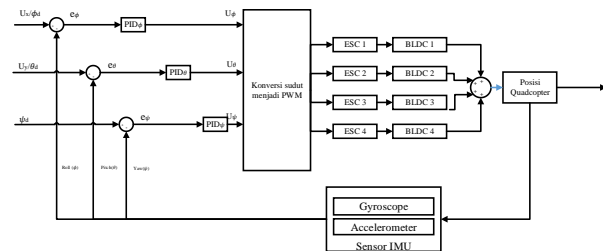
Tabel 1. Spesifikasi Perangkat Keras *Quadcopter*

No.	Komponen	Spesifikasi
1.	Controller	Ardupilot Mega 2.8
		Arduino Nano
2.	Sensor IMU	GY-521 MPU-6050.

3.	Modul GPS	GPS UBLOX NEO-6M GY-NEO6MV2
4.	Receiver	Receiver FS-iA6
5.	Transmitter	Transmitter FS-i6X
3.	ESC	Hobbywing Skywalker 50A
4.	Motor BLDC	Motor Brushless DC 2212 920KV
5.	Baterai Lipo	Baterai Turnigy 3300mah 3S 30C Lipo Pack
6.	RF Modules	APC 220
7.	Propeller	propeller 10x4,5

Dimana,  $u(t)$  yaitu kontrol input,  $e(t)$  adalah selisih antara kondisi yang diinginkan atau  $x_c(t)$  dengan kondisi sekarang atau  $x(t)$ .  $K_p$ ,  $K_i$ , dan  $K_d$ , semuanya adalah nilai non-negatif, menunjukkan koefisien untuk istilah kontroler proportional, integral, dan derivatif.

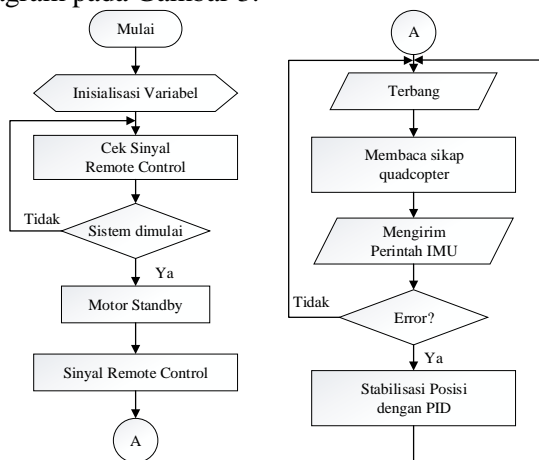
*Quadcopter* memiliki enam buah state yang dikelompokkan kedalam dua bagian, diantaranya yaitu posisi ( $x, y, z$ ) dan sikap ( $\theta, \phi, \psi$ ). Pada penelitian Tugas Akhir ini menggunakan sistem kendali PID untuk menstabilkan *quadcopter* pada saat terbang melayang. Diagram blok sistem kendali PID pada *quadcopter* dapat dilihat pada Gambar 4.



Gambar 4. Diagram Blok Sistem Kendali PID *Quadcopter*

### II.2.2 Algoritma Pemrograman *Quadcopter*

Secara garis besar, algoritma pemrograman dari *quadcopter* digambarkan pada flowchart diagram pada Gambar 3.



Gambar 3. Algoritma Pemrograman *Quadcopter* Mode Manual

### II.2.3 Sistem Kendali Kestabilan PID

Sebuah kendali PID adalah kendali untuk menentukan presisi tidaknya suatu sistem instrumentasi dengan karakteristik adanya umpan balik (feedback controller) pada sistem. Kendali PID (proportional, integral, derivatif) merupakan salah satu pengendali yang bertujuan untuk memperbaiki kinerja suatu sistem. Persamaan (1) merupakan persamaan dari kontrol PID.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (1)$$

Untuk pengendalian sikap *quadcopter* menggunakan sensor IMU (*gyroscope* dan *accelerometer*) yang mampu mendeteksi sudut dan kecepatan sudut sikap roll ( $\phi$ ), pitch ( $\theta$ ), dan yaw ( $\psi$ ). Tiga buah sikap *quadcopter* ini perlu diatur agar dapat terbang dengan stabil, yaitu sudut roll, pitch, dan yaw. Perancangan pengendali untuk gerakan sudut roll dan pitch pada *quadcopter*, diharapkan mampu mengembalikan posisi sudut roll dan pitch sesuai dengan referensi yang diberikan ketika mendapat gangguan dari luar.

Ketika mendapat gangguan dari luar, agar *quadcopter* mampu mengembalikan posisi roll dan pitch sesuai dengan referensi yang diberikan, maka diperlukan perancangan kendali untuk gerakan sudut roll dan pitch. Algoritma kendali PID pada sistem kestabilan *quadcopter* meliputi sudut roll dan pitch, karena sudut yaw hanya mengikuti. Referensi yang digunakan untuk setpoint kendali PID saat *quadcopter* di udara yaitu sudut nol roll dan nol pitch. Apabila terjadi perubahan kemiringan pada saat *quadcopter* mengudara, maka akan menghasilkan *error* pitch dan roll. *Error* tersebut yang nantinya akan diproses oleh kontroler untuk bisa memperbaiki sistem, sehingga menghasilkan sinyal PWM untuk menggerakkan sudut pitch dan roll. Sudut roll dikendalikan dengan cara menambah atau mengurangi kecepatan salah satu motor kanan atau motor kiri. Sudut pitch

dikendalikan dengan cara menambah atau mengurangi kecepatan salah satu motor belakang atau motor depan.

### III. HASIL DAN PEMBAHASAN

#### III.1 Implementasi Sistem

Implementasi pada perangkat keras ini dilakukan berdasarkan hasil perancangan sebelumnya. Hasil dari perancangan *quadcopter* dapat dilihat pada Gambar 5.



Gambar 5. Implementasi *Quadcopter*

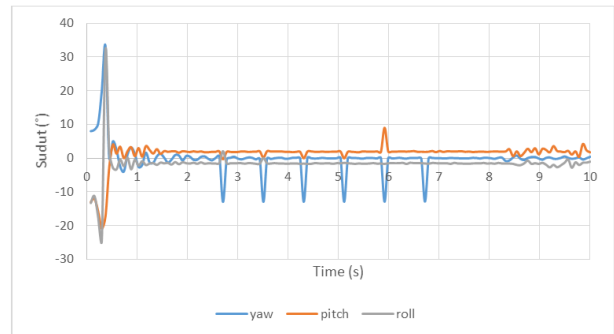
#### III.2 Pengujian Pengendali PID

Pengujian ini dilakukan untuk mengetahui respon kendali kestabilan pada *quadcopter*. Respon kendali pada stabilitas *quadcopter* merupakan hasil dari pengolahan sudut *accelerometer* yang dipadukan dengan kecepatan sudut *gyroscope*. Pengujian ini dilakukan dengan cara mengebalikan posisi *quadcopter* pada saat level atau saat kemiringan  $0^\circ$  dengan menggunakan kendali PID. Adapun nilai PID yang digunakan dapat dilihat pada Tabel 2. Konstanta PID ini didapatkan dari nilai default pada antarmuka Mission Planner dan telah dilakukan tuning dengan metode trial and error untuk bisa mendapatkan hasil yang optimal. Tabel 2 menunjukkan nilai parameter PID yang digunakan.

Tabel 2. Tabel Parameter PID

Konstanta	Roll( $\theta$ )	Pitch( $\phi$ )	Yaw( $\psi$ )
P	0,15	0,15	0,2
I	0,05	0,05	0,02
D	0,08	0,08	0,00

Dari Tabel 2, terlihat bahwa nilai konstanta PID yang ditentukan hanya dilakukan pada tiga state, yaitu saat sikap roll, pitch, dan yaw. Hasil dari konstanta PID tersebut, diimplementasikan pada sistem, sehingga didapatkan respon yang dapat dilihat pada Gambar 6.

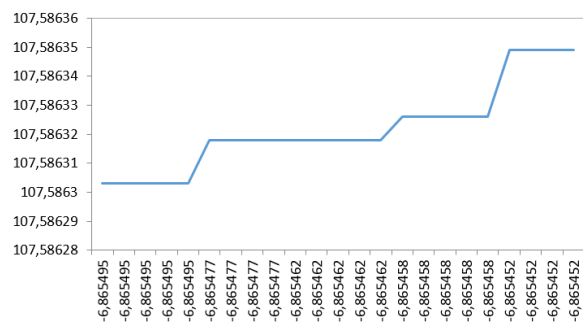


Gambar 6. Respon Sikap Kestabilan *Quadcopter* Menggunakan PID

Dapat dilihat dari grafik diatas bahwa overshoot maksimum pada gerakan roll yaitu sebesar  $32,19^\circ$ , pitch sebesar  $3,98^\circ$ , dan yaw sebesar  $33,15^\circ$ . Pengendalian pergerakan dalam gerakan roll dan pitch dengan  $K_p=0,15$ ;  $K_i=0,05$ ; dan  $K_d=0,08$  mendapatkan hasil respon steady state dengan settling time sebesar 1,07s dan 0,41s. Sedangkan pengendali pergerakan dalam gerakan yaw dengan  $K_p=0,2$ ;  $K_i=0,02$ ; dan  $K_d=0$  mendapatkan hasil respon steady state dengan settling time sebesar 0,35s.

#### III.3 Pengujian Navigasi Manual

Pengujian ini dilakukan untuk dapat mengetahui posisi *quadcopter* saat terbang, apakah posisi dalam keadaan stabil atau tidak. Pengujian dilakukan dengan cara menerbangkan *quadcopter* dengan *remote control*, kemudian *quadcopter* akan terbang dari titik satu ke titik lainnya. Pengujian yang dilakukan yaitu mengoperasikan *quadcopter* dengan jarak 10 meter. Rute yang dilewati oleh *quadcopter* memiliki tiga titik koordinat, dimana rute tersebut memiliki jalur lurus. Dalam pengujian ini, masing-masing koordinat berjarak 5 meter. Dari pengujian navigasi manual ini, didapatkan hasil seperti pada Gambar 7.



Gambar 7. Pengujian Navigasi Secara Manual

Garis biru merupakan posisi *quadcopter* ketika sedang mengudara. Dari hasil pengujian diatas,



dapat dikatakan bahwa modul GPS dapat mendeteksi posisi dari *quadcopter*. GPS dapat mendeteksi posisi dari *quadcopter* dengan mengirimkan data berupa data *longitude* dan *latitude*, namun dapat dilihat bahwa posisi dari *quadcopter* tidak stabil. Hal ini terjadi karena *quadcopter* diterbangkan secara manual menggunakan *remote control*.

## IV. KESIMPULAN DAN SARAN

### IV.1 Kesimpulan

Dari hasil pengujian sistem kendali kestabilan *quadcopter* dengan navigasi manual ini, dapat disimpulkan bahwa :

1. *Quadcopter* dapat terbang dan melakukan perintah *remote control* yang dioperasikan oleh pilot, yaitu throttle, roll, pitch, dan yaw.
2. *Quadcopter* dapat melakukan penerbangan yang stabil, dengan menggunakan nilai kendali PID sebagai berikut :
  - Konstanta kendali PID pada sumbu putar X (roll) menggunakan :  
Konstanta Proporsional = 0,15  
Konstanta Integral = 0,05  
Konstanta Derivative = 0,08  
Dengan pengendal PID tersebut didapatkan respon steady state dengan settling time sebesar 1,07 s.
  - Konstanta kendali PID pada sumbu putar Y (pitch) menggunakan :  
Konstanta Proporsional = 0,15  
Konstanta Integral = 0,05  
Konstanta Derivative = 0,08  
Dengan pengendal PID tersebut didapatkan respon steady state dengan settling time sebesar 0,41 s.
  - Konstanta kendali PID pada sudut yaw menggunakan :  
Konstanta Proporsional = 0,2  
Konstanta Integral = 0,02  
Konstanta Derivative = 0,00  
Dengan pengendal PID tersebut didapatkan respon steady state dengan settling time sebesar 0,35 s.
3. GPS dapat mendeteksi posisi dari *quadcopter* dengan mengirimkan data berupa data *longitude* dan *latitude*.

### IV.2 Saran

Adapun saran-saran yang penulis berikan untuk pengembangan kedepannya yaitu :

1. Membuat sebuah altitude hold, dimana *quadcopter* dapat terbang pada ketinggian tertentu yang dilakukan dengan cara mengunci level ketinggian.
2. Membuat *quadcopter* agar dapat terbang secara *autonomous*.
3. Memilih modul GPS yang lebih akurat.
4. Melakukan *propeller* balancing agar didapatkan hasil kestabilan yang lebih optimal.

## REFERENSI

- [1] A. S. Priambodo, A. I. Cahyadi dan S. Herdjunto, "Perancangan Sistem Kendali PD untuk Kestabilan Terbang," Departemen Teknik Elektro dan Teknologi Informasi, FT UGM, pp. 117-121, 2017.
- [2] V. Praveen dan A. S. Pillai, "Modeling and simulation of *quadcopter* using PID controller," IJCTA, vol. 9, no. 15, pp. 7151-7158, 2016.
- [3] R. H. Subrata, R. Tarumasely dan C. Dwianto S, "Perancangan Pengendali PID untuk Gerakan Pitch dan Roll pada *Quadcopter*," JETri, vol. 14, no. 2, pp. 1-16, 2017.
- [4] A. R. Al Tahtawi dan M. Yusuf, "Low-cost quadrotor hardware design with PID control system as flight controller," TELKOMNIKA, vol. 17, no. 4, pp. 1923-1930, 2019.
- [5] H. Ubaya, B. Tutuko dan B. Richardson, "Pengaturan Gerakan Hover pada *Quadcopter* dengan Menggunakan Metode PI Ziegler Nichols," pp. A91-197.
- [6] G. E. Setyawan, E. Setiawan dan W. Kurniawan, "Sistem Kendali Ketinggian *Quadcopter* Menggunakan PID," Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), vol. 2, no. 2, pp. 125-131, 2015.
- [7] W. K. Juang dan L. L. U. Tung, "Pembuatan Model *Quadcopter* yang Dapat Mempertahankan Ketinggian Tertentu," Jurnal Teknik Elektro, vol. 9, no. 2, pp. 49-55, 2016.
- [8] A. R. Al Tahtawi, E. Andika, M. Yusuf and W. N. Harjanto, "Pengembangan Low-cost Quadrotor dengan Misi Waypoint Tracking Berbasis Pengendali PID," ELKOMNIKA : Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika, vol. 8, no. 1, pp. 189-202, 2020.
- [9] T. Luukkonen, "Modelling and control of *quadcopter*," 2011.
- [10] K. Ogata, Modern Control Engineering Fifth Edition.