

Penerapan PID (Proportional Integral Derivative) Untuk Mengontrol Mode Manual Drone Hexacopter

Mochammad Zaky ^{a,1,*}, Desyderius Minggu ^{a,2}, Gatut Sulistiyanto ^{a,3},

^aPoliteknik Angkatan Darat Pendem, Junrejo, Kota Batu, Jawa Timur, Indonesia

¹ kommel2504@gmail.com *

* Penulis Koresponden

INFO ARTIKEL

Histori Artikel

Pengajuan 4 Januari 2020

Diperbaiki 5 Maret 2020

Diterima 16 Mei 2020

Kata Kunci

Drone

Hexa Copter

PID

ABSTRAK

Meminimalisir kerugian materiil dan non-materiil dalam sebuah tugas operasi adalah hal mutlak dalam sebuah strategi agar tidak memberikan dampak yang mampu merusak sebuah strategi. Untuk menerapkannya adalah menggunakan teknologi yang memiliki sistem kontrol yang baik. Sistem kontrol yang sering digunakan untuk kontrol secara manual adalah pengendali PID, dimana sistem kontrol jenis ini memiliki referensi berdasarkan dari konstanta nilai – nilai Propotional, Integral dan Derivative. Penerapan PID ini berdasarkan dari nilai referensi, dimana nilai tersebut dihasilkan melalui trial and error. Agar mendapatkan nilai kombinasi yang sesuai sebelum hasil output ini dikirim ke ESC sebagai acuan pergerakan kecepatan motor brushless. Konstanta pengendali PID untuk mencapai kestabilan saat hexacopter menembak adalah $K_p=130$; $K_i=0$; $K_d=0,076$. Hasil dari pengujian yaitu nilai rise time sebesar 0,0526 detik, settling time sebesar 0,783, overshoot sebesar 0,205% dan steady state error 1%.

Ini adalah artikel akses terbuka di bawah lisensi [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/).



1. Pendahuluan

Teknologi informatika, terutama robotika di jaman modern ini sudah berkembang menjadi sebuah hal penting dalam kehidupan manusia. Robot merupakan peralatan elektro mekanik atau bio mekanik, yang merupakan gabungan peralatan yang menghasilkan gerakan otomatis atau gerakan berdasarkan kontrol dari pilot. Teknologi robotika mampu membantu peran manusia, seperti robot-robot yang digunakan pada industri kesehatan, pertanian, pertahanan, permainan, dan lain-lain. Maraknya penggunaan robot saat ini dikarenakan dalam kondisi tertentu mampu menjadi pekerja yang ideal, memiliki tingkat akurasi dan tingginya efisiensi daripada tenaga manusia, dan yang lebih penting biaya operasi yang rendah dengan output yang dihasilkan lebih tinggi.

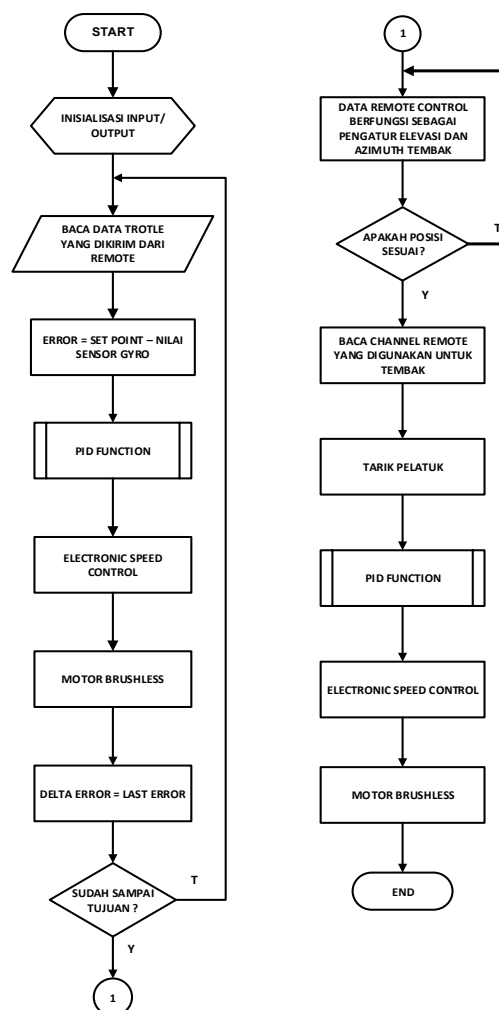
Unmanned Aerial Vehicle (UAV) merupakan salah satu jenis robot terbang yang dikendalikan secara manual oleh pilot atau otomatis, mampu membawa muatan seperti senjata ataupun lainnya. Penggunaan terbesar dari UAV adalah di bidang militer khususnya tugas operasi penyerbuan serta pengintaian markas-markas musuh. Pengintaian merupakan salah satu tugas operasi militer. Tujuan dari pengintaian adalah suatu gerakan rahasia untuk

memperoleh data tentang kedudukan, komposisi, kekuatan dan kegiatan musuh, serta keadaan kondisi wilayah berdasarkan 5 aspek medan (medan kritis, lapang tinjau, dan lapang tembak, lindung dan lindung tembak, rintangan dan jalan pendekat) serta sace (suhu, angin, cahaya dan endapan) sebelum penyergapan. Informasi yang didapat dari pengintaian adalah untuk membandingkan kekuatan, memberikan informasi pada pasukan belakang dan untuk mengetahui pola gelar pasukan musuh. Drone hexacopter ini dilengkapi dengan senjata pistol sehingga dapat digunakan untuk tugas operasi penyerbuan markas-markas musuh.

Drone ini nantinya diharapkan dapat meringankan tugas terutama untuk satuan-satuan di seluruh jajaran TNI-AD, khususnya dalam operasi pengintaian, penyerangan, dan penyerbuan. Agar bisa menjangkau kondisi-kondisi tersebut hexacopter dilengkapi dengan sistem kendali tembak jarak jauh. Dengan menerapkan metode PID dalam sistem kendali mode manual diharapkan hexacopter dapat terbang dan menembak secara stabil sehingga hasil dari perancangan alat ini dapat digunakan oleh personel TNI-AD dalam melaksanakan tugas penyerbuan markas musuh.

2. Metode penelitian

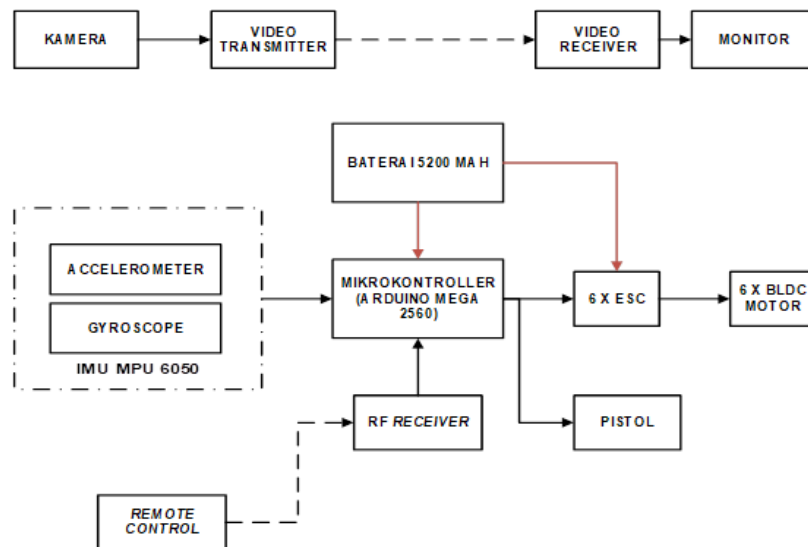
Perencanaan pembuatan *drone hexacopter* akan dijelaskan melalui *flowchart* yang digambarkan pada Gambar 2.1 sebagai berikut:



Gambar 2.1. Flowchart Sistem Penerbangan Hexacopter

Berikut ini adalah algoritma dari program pengiriman data pergerakan drone secara keseluruhan, yaitu:

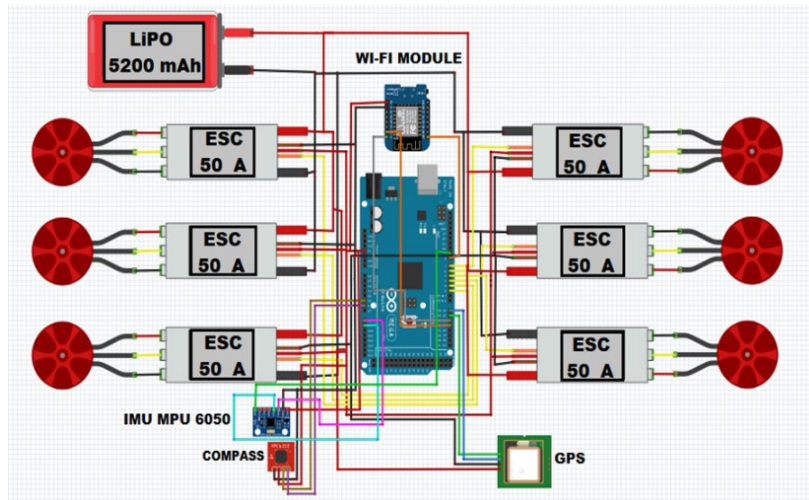
- a. *Drone, remote control*, dan layar monitor di aktifkan.
- b. Arduino Mega akan membaca dan menginisialisasi input atau output.
- c. *User* akan mengirimkan perintah kepada *drone* melalui *remote control*.
- d. Arduino akan membaca *set point* yang diterima kemudian membuat nilai *error*.
- e. Setelah mendapatkan nilai *error*, maka nilai tersebut akan digunakan sebagai inputan *PID function*.
- f. Data yang telah diolah pada *PID function* akan diterima oleh ESC dan diteruskan kepada *motor brushless DC*.
- g. Apakah *drone* telah sampai tujuan?
Jika *False* (F) atau belum, akan kembali ke *point C*.
Jika *True* (T) atau sudah, akan masuk ke *point H*
- h. Data *throttle* yang di dapatkan akan di kondisikan sebagai pengatur nilai elevasi dan nilai azimuth tembak.
- i. Apakah Posisi Tembak di dapatkan?
Jika *False* (F) atau belum, akan kembali ke *point H*
Jika *True* (T) atau sudah, maka akan masuk ke *point J*
- j. Arduino akan membaca *channel* inputan untuk posisi tembak.
- k. Pistol akan ditarik pelatuknya.
- l. *PID function* diaktifkan untuk menstabilkan *drone* saat menembak.
- m. Selesai.



Gambar 2.2 Blok Diagram Sistem Kerja *Hexacopter*

Pada Gambar 2.2. menjelaskan cara kerja *drone hexacopter* secara mendasar dengan IMU MPU 6050 yang terdiri dari *gyroscope* dan *accelerometer*, *Electronics Speed Control*, *remote control/transmitter* dan *receiver* yang diinisialisai dan diolah oleh Arduino Mega untuk mengontrol sistem penerbangan drone hexacopter. Monitor digunakan sebagai *hardware* untuk menampilkan posisi *drone* dengan kamera yang terpasang pada *drone*.

Untuk menjelaskan cara kerja dari Gambar 2.2 blok diagram digunakan untuk mempermudah dalam menyusun rangkaian sistem dari alat yang akan dibuat. Berdasarkan blok diagram diatas maka langkah-langkah dalam menyusun rangkaian minimum sistem dari alat ini dapat dilihat pada Gambar 2.3.



Gambar 2.3 Skema Perancangan Pada Arduino Mega 2560

Rancang bangun penerapan PID (*Proporsional Integral Derivative*) untuk mengontrol mode manual sistem penerbangan *drone hexacopter*. Alat ini bekerja berdasarkan suatu sistem secara menyeluruh dan terintegrasi dari masing-masing modul rangkaian arduino yang menerima data dari setiap sensor yang terpasang pada drone, diantaranya GPS mengirimkan koordinat posisi *drone*, ESC mengontrol kecepatan perputaran *motor brushless*, dan *receiver* menerima nilai referensi dari *remote control*.

Selain menggunakan *streaming* kamera pada monitor, GPS juga digunakan untuk mengirimkan data koordinat posisi drone pada saat diterbangkan secara manual dan Arduino akan membaca input dari ESC berupa nilai kecepatan putaran motor brushless, setelah membaca input dari ESC, maka Arduino bisa memberikan perintah untuk memutar *motor brushless* secara *clockwise* maupun *counter clockwise*

3. Hasil dan Analisis

Berdasarkan hasil penelitian yang sudah dilakukan, didapatkan data-data hasil pengujian. Pengujian yang dilakukan meliputi pengujian modul GPS NEO-6, Sensor *gyroscope*, sensor *compass* HMC5883, dan penentuan parameter konstanta sistem kontrol PID.

Tabel 3.1 Hasil Pengukuran Modul GPS NEO-6M

Koordinat GPS		Koordinat Modul GPS		Error GPS (m)
Latitude	Longitude	Latitude	Longitude	
-7.936989 ^o	112.634564 ^o	-7.936992 ^o	112.634564 ^o	0.3
-7.937335 ^o	112.635229 ^o	-7.937332 ^o	112.635225 ^o	0.5
-7.937717 ^o	112.634990 ^o	-7.937718 ^o	112.634994 ^o	0.4
-7.938189 ^o	112.634582 ^o	-7.938195 ^o	112.634582 ^o	0.6
-7.938177 ^o	112.634272 ^o	-7.938178 ^o	112.634277 ^o	0.5
Rata-Rata Error				0.4

Tabel 3.1 adalah tabel data hasil pengukuran modul GPS Neo-6M untuk menentukan koordinat posisi pergerakan hexacopter. Dari percobaan tersebut dihasilkan nilai rata-rata error dari pengujian modul GPS adalah 0.4 m. Oleh karena itu modul GPS tersebut dalam keadaan baik karena toleransi error dari modul GPS Neo-6M adalah 2.5 meter.

Tabel 3.2. Hasil Pengukuran Sensor *Gyroscope*

Sudut Sensor ($^{\circ}$)	Output Sudut ($^{\circ}$)	Sudut Error ($^{\circ}$)	Error (%)
Roll 10 $^{\circ}$	10.5	0.5	4.7
Roll 20 $^{\circ}$	30.6	0.6	1.96
Roll 30 $^{\circ}$	32	2	6.25
Roll -10 $^{\circ}$	-10.38	0.38	3.7
Roll -20 $^{\circ}$	-20.2	0.2	0.9
Roll -30 $^{\circ}$	-30	0	0
Pitch 10 $^{\circ}$	10.4	0.4	3.84
Pitch 20 $^{\circ}$	20	0	0
Pitch 30 $^{\circ}$	31	1	3.22
Pitch -10 $^{\circ}$	-10.8	0.8	7.4
Pitch -20 $^{\circ}$	-20.21	0.21	1.03
Pitch -30 $^{\circ}$	-30.77	0.77	2.5
Rata-Rata Error			2.95

Pada tabel 3.2. dihasilkan nilai rata-rata error pengujian modul *gyroscope* MPU 6050 adalah sebesar 2.95 %, sedangkan untuk nilai toleransi error minimum adalah -3% dan maksimum sebesar 3%. Dengan demikian modul *gyroscope* MPU 6050 yang digunakan dalam kondisi baik.

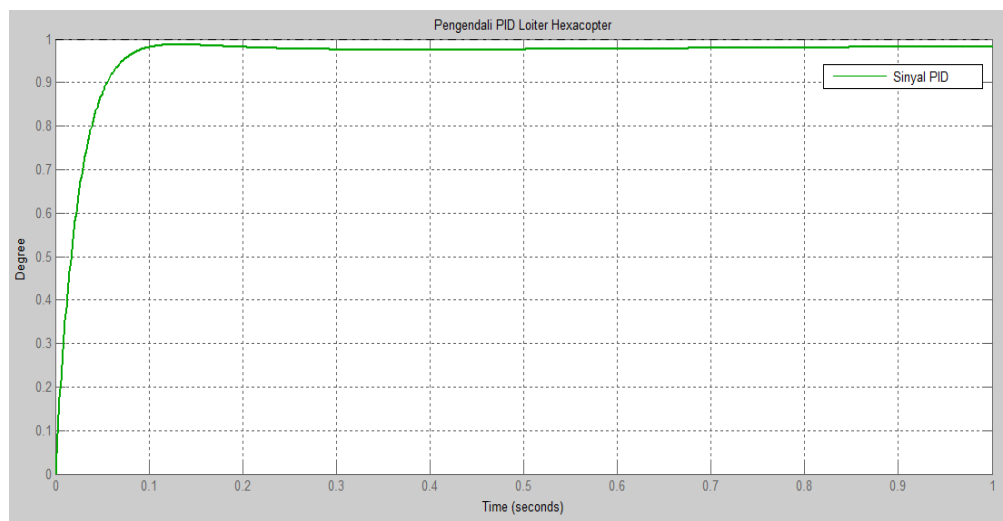
Tabel 3.3. Hasil Pengukuran Sensor *Compass* HMC5883

Sudut Sensor <i>Compass</i> HMC5883 Acuan	Sudut Error Sensor <i>Compass</i>	Error %
0 $^{\circ}$	0 $^{\circ}$	0
90 $^{\circ}$	90.4 $^{\circ}$	0.4
180 $^{\circ}$	180.6 $^{\circ}$	0.3
270 $^{\circ}$	270.3 $^{\circ}$	0.1
359 $^{\circ}$	359.6 $^{\circ}$	0.1
Rata-Rata Error		0.28

Fungsi sensor *compass* HMC5883 bertujuan untuk mengetahui derajat pada tiap arah mata angin. Tabel 3.3. adalah hasil pengujian dari modul *compass* HMC5883. Nilai rata-rata *error* yang dihasilkan dari pengujian tersebut adalah 0.28 %, sedangkan toleransi nilai *error* sebesar 0.5 % sehingga modul *compass* HMC 5883 dalam kondisi baik.

Tabel 3.4. Hasil Parameter PID Posisi *Loiter*

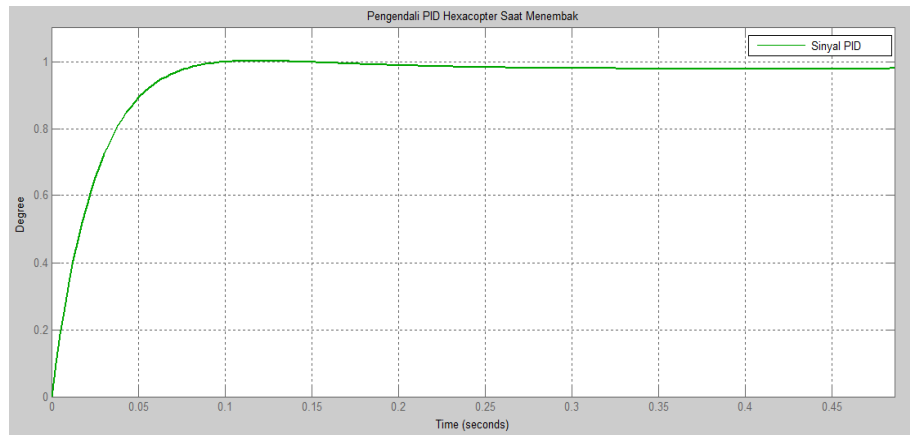
Kp	Ki	Kd	Rise Time (s)	Settling Time (s)	Overshoot (%)	Steady State Error (%)
130	0	0.076	0.0491	0.573	0	0.19

Gambar 3.1 Grafik Respon *Hexacopter* pada Saat *Loiter*

Berdasarkan grafik pada gambar 3.1. dan tabel 3.4, kombinasi konstanta PID ini menjadi kombinasi PID yang diterapkan pada sistem terbang hexacopter dengan nilai $K_p = 130$, $K_i = 0$, $K_d = 0.076$. Kombinasi konstanta PID ini menunjukkan respon sistem yang terbaik dari beberapa konstanta percobaan sebelumnya. Sistem kontrol PID ini memiliki waktu untuk mencapai nilai *steady state error* yang paling kecil sebesar 0.19% walaupun adanya gangguan kemiringan akibat angin. Nilai *rise time* dan *settling time* yang dihasilkan lebih kecil daripada percobaan penentuan konstanta PID sebelumnya yaitu *rise time* sebesar 0.0491 detik dan *settling time* sebesar 0.573 detik. Untuk nilai lewatan maksimum (*overshoot*) adalah 0.19%, sehingga *hexacopter* stabil pada saat loiter di udara.

Tabel 3.5. Hasil Parameter PID Saat Menembak

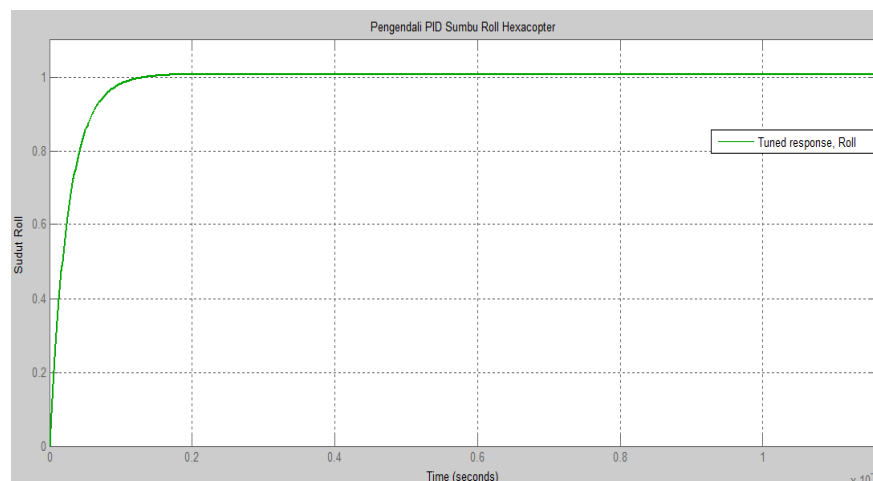
Kp	Ki	Kd	Rise Time (s)	Settling Time (s)	Overshoot (%)	Steady State Error (%)
130	0	0.076	0.0526	0.783	0.205	1

Gambar 3.2. Grafik Respon PID *Hexacopter* Saat Menembak

Berdasarkan grafik pada gambar 3.2. dan tabel 4.5. kombinasi PID yang diimplementasikan langsung pada sistem tembak terbang *hexacopter* dengan nilai $K_p = 130$, $K_i = 0$, $K_d = 0,076$. Sistem kontrol PID ini mencapai nilai *steady state error* yang paling kecil sebesar 1% walaupun adanya gangguan kemiringan akibat tekanan *recoil* pistol saat menembak. Nilai *rise time* sebesar 0.0526 detik dan *settling time* sebesar 0.783 detik. Nilai lewatan maksimum (*overshoot*) adalah 0.205%, sehingga *hexacopter* akan berusaha menyetabilkan dirinya saat adanya efek *recoil* pistol.

Tabel 3.6. Hasil Parameter PID *Hexacopter* Pergerakan Roll

Kp	Ki	Kd	Rise Time (s)	Settling Time (s)	Overshoot (%)	Steady State Error (%)
13	1310.6	0.07	0.057	0.98	0.723	1.01

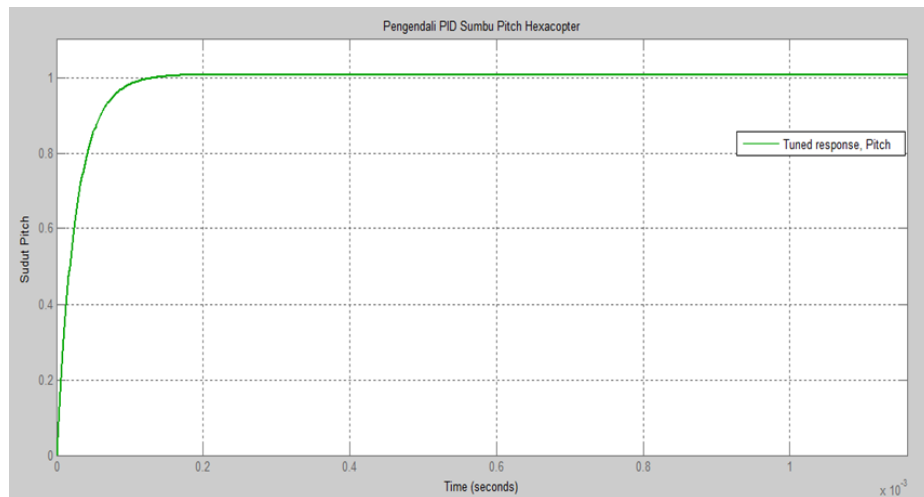
Gambar 3.3. Grafik Respon PID *Hexacopter* pada Pergerakan Roll

Pada gambar 3.3 dan tabel 4.6 kombinasi PID yang diimplementasikan langsung pada sistem terbang *hexacopter* bergerak *roll* ke kanan dengan nilai $K_p = 13$, $K_i = 1310.6$, $K_d = 0,07$ mencapai nilai *steady state error* sebesar 1.01%. Nilai *rise time* sebesar 0.057 detik dan *settling time* sebesar 0.98 detik. Untuk nilai lewatan maksimum (*overshoot*) adalah 0.723%,

sehingga hexacopter akan berusaha bergerak roll ke kanan dan kiri secara stabil saat adanya gangguan angin.

Tabel 3.7 Hasil Parameter PID *Hexacopter* Pergerakan *Pitch*

Kp	Ki	Kd	Rise Time (s)	Settling Time (s)	Overshoot (%)	Steady State Error (%)
13	1310.6	0.07	0.0514	0.672	0.132	1.01

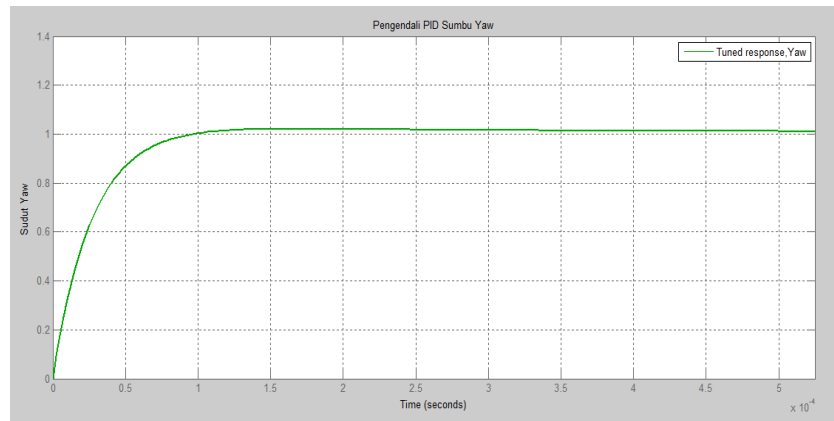


Gambar 3.4 Grafik Respon PID *Hexacopter* pada Pergerakan *Pitch*

Penjelasan gambar 3.5. dan tabel 3.7, kombinasi konstanta PID sistem terbang *hexacopter* bergerak *pitch* dengan nilai $K_p = 13$, $K_i = 1310.6$, $K_d = 0,07$ mencapai nilai *steady state error* sebesar 1.01%. Nilai *rise time* sebesar 0.0514 detik, *settling time* sebesar 0.672 detik, nilai lewatan maksimum (*overshoot*) adalah 0.132%, sehingga *hexacopter* akan berusaha bergerak *pitch* depan dan belakang secara stabil.

Tabel 3.8. Hasil Parameter PID *Hexacopter* Pergerakan *Yaw*

No.	Kp	Ki	Kd	Rise Time (s)	Settling Time (s)	Overshoot (%)	Steady State Error (%)
1.	17.11	141	0.097	0.0534	0.21	2.13	1.02



Gambar 3.5 Grafik Respon PID *Hexacopter* pada Pergerakan Yaw

Berdasarkan grafik pada gambar 3.5 dan tabel 3.8. kombinasi konstanta PID dengan nilai $K_p = 17.11$, $K_i = 1410.6$, $K_d = 0.097$. Sistem kontrol PID ini mencapai nilai *steady state* sebesar 1.02%. Nilai *rise time* sebesar 0.0534 detik dan *settling time* sebesar 0.21 detik. Untuk nilai lewatan maksimum (*overshoot*) adalah 2.13%, sehingga *hexacopter* akan berusaha bergerak *yaw* kanan dan kiri secara stabil saat adanya gangguan angin.

4. Kesimpulan

- a. Untuk GPS yang digunakan adalah jenis NEO-6 sudah bekerja dengan baik dimana nilai rata-rata error sebesar 0.4 meter, sedangkan sensor gyroscope MPU 6050 menghasilkan nilai rata-rata error sebesar 2.95%, masih dalam batas toleransi yang diijinkan. Kemudian untuk sensor compass jenis HMC5883 sudah bekerja dengan baik dimana nilai rata-rata error sebesar 0.28% dan untuk batas nilai toleransi error sebesar 0.5%.
- b. Pada saat *hexacopter* lepas landas agar mendapatkan respon sistem yang stabil menuju posisi loiter maka dibutuhkan inputan konstanta PID, yaitu $K_p=130$; $K_i=0$; $K_d=0,076$ dengan nilai *rise time* sebesar 0,0491 detik, *settling time* sebesar 0,573 detik, *overshoot* sebesar 0% dan *steady state error* 0,19%. Pada saat *hexacopter* melakukan tembakan dengan konstanta PID yang sama saat *hexacopter* berada pada posisi loiter, maka nilai *overshoot* sebesar 0,205%, nilai *rise time* sebesar 0,0526 detik, *settling time* sebesar 0,783 detik dan *steady state error* 1%.
- c. *Hexacopter* mampu terbang dengan stabil yaitu pada sudut pitch dengan sudut step response, nilai konstanta PID-nya adalah $K_p=13$, $K_i=1310.6$ dan $K_d=0.07$ dengan nilai *rise time* sebesar 0.0514 detik, *settling time* sebesar 0.672, *overshoot* sebesar 0.132% dan *steady state error* 1.01%. Pada sudut roll diinputkan konstanta PID, yaitu $K_p=13$, $K_i=1310.6$ dan $K_d=0.07$. Dari grafik terlihat bahwa sistem dapat bergerak secara stabil dengan nilai *rise time* sebesar 0.057 detik, *settling time* sebesar 0.98, *overshoot* sebesar 0.723% dan *steady state error* 1.01%. Pada pergerakan yaw diinputkan konstanta PID, yaitu $K_p=17.11$, $K_i=1410.6$ dan $K_d=0.097$ dengan nilai *rise time* sebesar 0.0534 detik, *settling time* sebesar 0.21, *overshoot* sebesar 2.13% dan *steady state error* 1.02%.

Daftar Pustaka

- [1] Amin, Z., Meldi, (2018). Pengidentifikasi dan Pencarian Manusia Berbasis Citra Menggunakan Unmanned Aerial Vehicle. *Universitas Andalas*. Metal: Jurnal Sistem Mekanik dan Termal Vol. 02 No. 02 (2018) hal. 50-61.
- [2] Dharmawan, A. (2009). Pengendalian Motor Brushless DC Dengan Metode PWM Sinusoidal Menggunakan ATMEGA16. Universitas Indonesia, Depok.

- [3] T. A. S, A. Rabi', D. Minggu, And I. Mujahidin, "Frequency Hopping Video Real Time Untuk Pengamanan Data Pengintaian Operasi Inteligence Tni," *Jasiek (Jurnal Apl. Sains, Informasi, Elektron. Dan Komputer)*, 2019.
- [4] Habibi, G. Akbar. (2013). Perancangan Dan Analisis Otomasi Sistem Kendali Quadcopter Melalui Koordinat Dengan Global Positioning System Tracker. *Institut Teknologi Telkom*.
- [5] A. E. Pambudi, L. Maajid, J. Rohman, And I. Mujahidin, "Aplikasi Penggunaan Joystick Sebagai Pengendalian Remote Control Weapon Station (RCWS) Senjata Mesin Ringan (Smr)," *Jasiek (Jurnal Apl. Sains, Informasi, Elektron. Dan Komputer)*, Vol. 1, No. 2, Pp. 98-105, 2019.
- [6] Majid, M, (2016). Implementasi Arduino Mega 2560 untuk Kontrol Miniatur Elevator Barang Otomatis. *Universitas Negeri Semarang*.
- [7] Putra, Bagus T.P. (2019). Sistem Kendali Senjata Robot Tempur Dengan Metode Proportional Integral Derivative (PID). *Universitas Merdeka Malang*.
- [8] Putra, R.P. (2016). Aplikasi GPS APM2.5 NEO-6M Pada Robot Terbang Pendeteksi Asap. *Politeknik Negeri Sriwijaya*.
- [9] Ramzy, M.D. (2016). Pengontrol Motor Brushless 2300 KV Menggunakan ESC 12A Pada Drone Quadcopter Berbasis Flight Controller Pixhawk. *Politeknik Negeri Sriwijaya*.
- [10] E. Endrayana, D. H. S. Wahyuni, N. Nachrowie, And I. Mujahidin, "Variasi Ground Plane Antena Collinear Pada Pemancar Televisi Analog Dengan Frekuensi Uhf 442 Mhz," *Jasiek (Jurnal Apl. Sains, Informasi, Elektron. Dan Komputer)*, Vol. 1, No. 2, Pp. 149-156, 2019.
- [11] Ridhwan, M. T. (2012). Rancang Bangun Penggerak Daya Motor Brushless DC 350W/48V. *Politeknik Negeri Bandung, Bandung*.
- [12] Rohimah. (2016). Pengamanan Pendeteksi Sayatan Menggunakan Teknologi Global Positioning System (GPS) Dengan Sensor LDR Via Short Masaage Service (SMS). *Politeknik Negeri Sriwijaya*.