



Terbit online pada laman web jurnal : <http://jurnal.sttp-yds.ac.id>

SAINSTEK (e-Journal)

| ISSN (Print) 2337-6910 | ISSN (Online) 2460-1039 |



Klik di sini dan tuliskan Kategori Artikel

ANALISA RANCANGAN KESEIMBANGAN MENGGUNAKAN SENSOR IMU TYPE – MPU6050 PADA QUADCOPTER

Fadhli Palaha¹, Jamar Teles², Yolnasdi³

¹ Sekolah Tinggi Teknologi Pekanbaru, Jl. Dirgantara, No.4, Pekanbaru, Indonesia

² Sekolah Tinggi Teknologi Pekanbaru, Jl. Dirgantara, No.4, Pekanbaru, Indonesia

³ Sekolah Tinggi Teknologi Pekanbaru, Jl. Dirgantara, No.4, Pekanbaru, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 26 November 2020

Revisi Akhir: 11 Desember 2020

Diterbitkan Online: 31 Desember 2020

KATA KUNCI

Quadcopter, PID, MPU-6050

KORESPONDENSI

Telepon: "+62 (0761) 61815"

E-mail: fadhlyyy@yahoo.com

A B S T R A C T

Kontroler Proporsional Integral Derivatif (PID) merupakan kontroler yang memiliki stabilitas yang baik dengan tingkat error dan overshoot yang kecil. Penelitian ini bertujuan untuk mencapai nilai keseimbangan dan kehandalan quadcopter pada saat terbang. Quadcopter yang dirancang diputar oleh motor brushless direct current (BLDC) yang dikontrol oleh Arduino. Keluaran dari Arduino berupa PWM menjadi masukan untuk penggerak motor, dimana kecepatan keluaran dari motor akan dideteksi oleh sensor MPU-6050. Setelah itu akan terjadi pengulangan proses, sampai kecepatan memenuhi nilai set point.

Penentuan hasil parameter kontroler PID ini didapatkan dengan menggunakan rumus-rumus sederhana dan proses trial and error. Dimana hasil parameter kontroler PID diperoleh nilai $K_p = 1.3$, $K_i = 0.05$ dan $K_d = 15$. Dengan nilai K_p , K_i dan K_d tersebut sistem dapat berjalan dengan baik dengan mempertahankan kecepatan putaran motor mendekati setpoint. Hasil pengujian menunjukkan bahwa quadcopter dengan pengendali PID dapat hovering dengan stabil. Meski respon sikap pada kondisi transien masih memiliki overshoot. Hal ini dapat diatasi salah satunya dengan menggunakan skema pengendali yang kokoh.

1. PENDAHULUAN

Quadcopter pertama kali dikembangkan oleh dua bersaudara bernama Louis Breguet and Jacques Breguet dibawah bimbingan Professor Charles Richet pada tahun 1907 di Prancis, yang mana mereka beri nama Breguet Richet Gyroplane No.1 Breguet-Richet-1907 (Donald Norris, 2014). Perkembangan Quadcopter terus berlanjut hingga saat ini, dimana pesawat tanpa awak atau dikenal dengan nama lain *Unnamed Aerial Vehicles* (UAV). Quadcopter mempermudah pada sistem transportasi udara tanpa awak dengan beban minimum dan daerah yang sulit terjangkau oleh manusia. Awalnya quadcopter banyak dikembangkan pada bidang militer, dan sekarang bisa dipelajari dan dikembangkan ke berbagai bidang seperti

pertanian, keamanan, area bencana, weather, dan Aerial photography.

Quadcopter terdiri dari 4 propeller yang di putar oleh 4 motor Brushless Direct Current (BLDC) yang akan di disain berdasarkan penggunaannya. BLDC motor tidak menggunakan sikat atau brush untuk kontak pergantian magnet tetapi dilakukan secara komutasi elektronis. Terdapat empat gerakan dasar pada quadcopter yaitu gerakan altitude (throttle), gerakan sudut (roll, pitch), dan gerakan sudut (yaw) (Subrata, Rosalia H, 2017).

Jika quadcopter ingin dimonitor maka quadcopter harus dilengkapi dengan automatic navigation system menggunakan Global Positioning System (GPS). Penggunaan GPS tentu saja dilengkapi dengan visualisasi koordinat posisi yang akan mempermudah pengguna

untuk memahami jarak *quadcopter*. Hal yang terpenting adalah kendali keseimbangan yang terdapat pada *quadcopter* tersebut. Dalam *quadcopter* sudah memiliki *controller* yang lengkap dengan *gyroscope*, *accelerometer*, magnetometer, dan GPS. Kelengkapan sensor ini tentu memberikan kemudahan untuk mengendalikan *quadcopter*.

Salah satu rancangan dasar suatu *quadcopter* meliputi sensor dan *controller* yang dapat mengatur keseimbangan pergerakan *quadcopter*. Yang mana penggunaan IMU MPU-6050 sensor dapat memberikan data mentah/dasar keseimbangan, apabila di integrasikan pada *quadcopter*. IMU sensor yang memberikan nilai X, Y, Z kemudian direalisasikan untuk mendapatkan sudut roll, pitch, dan yaw. Dengan dibantu penerapan PID *controller* dapat memperhalus pergerakan, mempercepat respon, dan menghasilkan keseimbangan. Penelitian ini bertujuan untuk mencapai nilai keseimbangan dan kehandalan *quadcopter* pada saat terbang.

2. TINJAUAN PUSTAKA

2.1 Drone

Drone atau pesawat tanpa awak atau dalam bahasa lainnya disebut *Unmanned Aerial Vehicle* (UAV) adalah merupakan sebuah wahana mesin terbang yang di kendalikan dari jarak jauh oleh pilot (Bimo Jati Utomo, 2015). *Drone* mampu mengendalikan dirinya sendiri, menggunakan hukum aerodinamika untuk mengangkat dirinya dan mampu mengangkat beban-bekan tertentu. Dahulu orang mengenal *drone* atau pesawat tanpa awak ini digunakan oleh militer untuk memata-matai musuh di daerah konflik. Namun karena cara kerja *drone* saat itu masih sederhana maka tidak sepenuhnya berhasil ada yang terjebak angin dan berubah arah.

2.2 Macam-macam Drone

Drone memiliki 2 versi yaitu :

1. Versi pertama adalah dikendalikan oleh pilot secara manual dari jarak jauh dengan menggunakan sistem radio kontrol.
2. Versi kedua adalah dikendalikan secara otomatis oleh program yang telah ditentukan sebelum terbang.

Berdasarkan baling-baling terdapat 2 jenis *Drone* yaitu :

a. Fixed Wing Drone

Jenis *Drone* ini adalah berbentuk pesawat biasa yang sering kita lihat dan ini digunakan untuk proses yang cepat serta memiliki daya jangkauan yang lebih cepat serta lebih luas, biasanya digunakan untuk *mapping*. *Drone* jenis *Fixed Wing* memiliki energi lebih irit baterai karena hanya menggunakan satu *propeller* seperti yang terlihat pada gambar 2.4.



Gambar 2.1 Fixed Wing Drone

b. Multicopter Drone

Multicopter Drone ini memiliki kestabilan yang cukup bagus dan mempunyai daya angkut beban yang besar sehingga bagi pengguna dapat mengkolaborasi penggunaan sesuai dengan kebutuhan seperti dengan menambahkan kamera di bagian sisi tertentu dari *drone* tersebut. Gambar 2.2 adalah contoh *multicopter drone*.



Gambar 2.2 Multicopter Drone

Seiring kemajuan teknologi maka *drone* juga berkembang dengan beberapa variasi berdasarkan jumlah *propeller*, sehingga jenis-jenis *drone* ini dinamai berdasarkan *propeller* tersebut, diantaranya (Imas Tri Setyadewi dan Yudha Agung Nugroho (2018), :

- 1 *Tricopter* : *Drone* yang memiliki 3 motor, dan 3 *propeller*.
- 2 *Quadcopter* : *Drone* yang memiliki 4 motor, dan 4 *propeller*.
- 3 *Hexacopter* : *Drone* yang memiliki 6 motor, dan 6 *propeller*.
- 4 *Octacopter* : *Drone* yang memiliki 8 motor, dan 8 *propeller*.

Setelah didapat nama-nama *drone* sesuai jumlah *propellernya* tersebut, maka setiap penyebutan nama wahana tersebut melekatlah nama dari jumlah *propeller* tersebut.

2.3 Motor Brushless Direct Current (BLDC)

Brushless direct current (BLDC) adalah salah satu dari sekian banyak jenis motor yang umum digunakan. Sesuai namanya, BLDC motor yang tidak menggunakan sikat atau *brush* untuk kontak pergantian magnet tetapi dilakukan secara komutasi elektronis.

Secara umum motor BLDC terdiri dari dua bagian yaitu, *rotor* ; bagian yang bergerak yang terbuat dari permanen magnet dan *stator*; bagian yang tidak bergerak, yang terbuat dari kumparan 3 fasa. Walaupun merupakan motor listrik *synchronous* AC 3 fasa, motor ini tetap disebut dengan BLDC karena pada implementasinya BLDC menggunakan sumber DC sebagai sumber energi utama yang kemudian diubah menjadi tegangan AC dengan menggunakan inverter 3 fasa. Tujuan dari pemberian tegangan AC 3 fasa pada *stator* BLDC adalah menciptakan medan magnet putar *stator* untuk menarik magnet *rotor*. Gambar 2.3 merupakan salah satu contoh Motor BLDC yang digunakan pada *Quadcopter*.



Gambar 2.3 Brushless DC Motor

2.4 IMU (Inertial Measurement Unit)

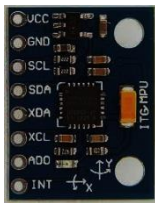
IMU (*Inertial Measurement Unit*) merupakan suatu unit dalam modul elektronik yang mengumpulkan

data kecepatan angular dan akselerasi linear yang kemudian dikirim ke *Central Processing Unit* (CPU) untuk mendapatkan data keberadaan dan pergerakan suatu benda. IMU terdiri dari kombinasi *accelerometer* (sensor percepatan) dan *gyroscope* (sensor kecepatan angular). *Accelerometer* digunakan untuk mengukur percepatan suatu benda dan *gyroscope* digunakan untuk mengukur kecepatan rotasi dari suatu benda.

Inertial Measurement Unit (IMU) sangat berguna pada sistem navigasi dan kendali. IMU yang terdiri dari tiga sensor *accelerometer* dan tiga sensor *gyroscope* yang masing-masing ditempatkan pada tiga sumbu (x, y, dan z) dan saling tegak lurus.

Sensor IMU yang digunakan pada tulisan ini yaitu MPU-6050 yang mempunyai 6 buah output, 3 output berupa nilai *accelerometer* dan 3 output dari *gyroscope*. MPU-6050 juga mempunyai PIN yang digunakan sebagai output dari suhu sekitar. Berikut PINOUT dari MPU-6050 :

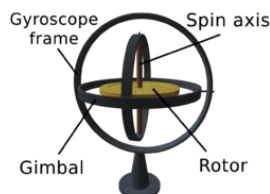
- 1) VCC -> +3.3V DC
- 2) SCL (Serial Clock) untuk I2C clock.
- 3) SDA (Serial Data) untuk I2C data.
- 4) XDA -> NC
- 5) XCL -> NC
- 6) AD0 untuk output sensor suhu
- 7) INT untuk Interrupt Arduino
- 8) Ground.



Gambar 2.4 MPU-6050 Module

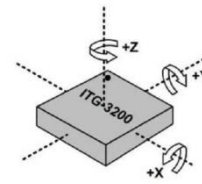
2.4.1 Sensor Gyroscope

Sensor *Gyroscope* adalah perangkat yang digunakan untuk mengukur atau mempertahankan orientasi, yang berlandaskan pada prinsip-prinsip momentum sudut secara mekanis. Pada dasarnya *gyroscope* mekanik berbentuk seperti sebuah roda berputar atau cakram di mana poros bebas untuk mengambil setiap orientasi seperti yang terlihat pada gambar 2.5.



Gambar 2.5 Gyroscope Mekanik

Sensor *gyroscope* dapat mendeteksi gerakan sesuai gravitasi, atau gerakan pengguna. Sebelum digunakan, sensor *gyroscope* terlebih dahulu dilakukan menggunakan bantuan bandul guna mendapatkan nilai faktor kalibrasi. Output dari sensor *gyroscope* berupa kecepatan sudut dari arah 3 sumbu, yaitu sumbu x (kanan - kiri), sumbu y (depan - belakang), dan sumbu z (atas - bawah) seperti yang terlihat pada gambar 2.6 dibawah ini.



Gambar 2.6 Sumbu sudut Gyroscope

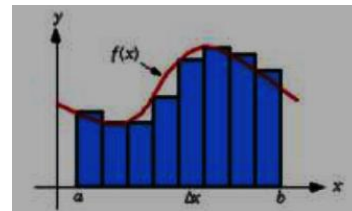
2.4.2 Accelerometer

Accelerometer adalah sensor yang digunakan untuk mengukur percepatan atau perubahan kecepatan terhadap waktu. Sensor ini dipasang bersama benda yang akan diukur akselerasinya, seperti mengukur perubahan kecepatan roket yang meluncur atau digunakan untuk analisis getaran (*vibration analysis*) pada mesin, serta digunakan untuk mendeteksi gerak dan kemiringan pada *smart phone*. Percepatan merupakan suatu keadaan berubahnya kecepatan terhadap waktu. Bertambahnya suatu kecepatan dalam suatu rentang waktu disebut percepatan (*acceleration*). Namun jika kecepatan semakin berkurang daripada kecepatan sebelumnya, disebut perlambatan (*deceleration*).

Berubahnya arah pergerakan suatu benda akan menimbulkan percepatan pula. Untuk memperoleh data jarak dari sensor *accelerometer*, diperlukan proses integral ganda terhadap keluaran sensor.

$$\vec{s} = \int (\int (\vec{a}) dt) dt \dots\dots\dots (1)$$

Proses penghitungan ini dipengaruhi oleh waktu cuplik data, sehingga jeda waktu cuplik data (dt) harus selalu konstan dan dibuat sekecil mungkin secara sederhana, integral merupakan luas daerah di bawah suatu sinyal selama rentang waktu tertentu. Untuk lebih jelasnya dapat dilihat pada Gambar 2.7



Gambar 2.7 Proses integral terhadap suatu sinyal

Accelerometer dapat mendeteksi status aktivitas gerakan (*active/inactive*) dengan membandingkan percepatan / akselerasi pada sumbu manapun dengan ambang batas sensitivitas yang dapat diatur melalui kode program. Tersedia juga pendeteksi ketukan (*tap sensing*) yang dapat mendeteksi ketukan tunggal maupun ganda pada berbagai arah. Modul ini juga dapat mendeteksi gerak jatuh bebas (*free-fall sensing*).

2.5 Electronic Speed Controller (ESC)

Electronic Speed Controller (ESC) berperan penting dalam proses pengendalian kecepatan dan arah putar motor. Pemilihan ESC tergantung dari besar arus maksimal dari motor ESC dapat digunakan dengan spesifikasi arus maksimum 30 ampere. Untuk menggerakkan *propeller* BLDC yang dikendalikan kumutasinya oleh *Electronic Speed Controller* (ESC).



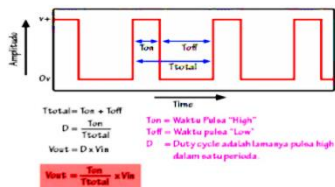
Gambar 2.8 ESC Motor BLDC

Quadcopter mempunyai dua buah arah motor yang berputar searah jarum jam (*Clock Wise*) dan dua buah motor yang berputar dengan berlawanan arah jarum jam (*Counter Clock Wise*). Untuk mendapatkan konfigurasi arah putar yang berbeda, BLDC hampir mirip dengan motor DC konvensional yaitu mengatur polaritasnya. BLDC dapat diatur arah putarnya dengan menukar urutan tiga kabel yang dimilikinya, dengan menukar urutan kabel sama saja mengatur urutan komutasinya.

2.6 Pulse Wide Modulation (PWM)

Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam suatu periode, untuk mendapatkan tegangan rata-rata yang berbeda. Aplikasi PWM berbasis mikrokontroler biasanya berupa pengendalian kecepatan motor DC, pengendalian motor servo, pengaturan nyala terang LED dan lain sebagainya.

Lebar pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. Artinya, Sinyal PWM memiliki frekuensi gelombang yang tetap namun *duty cycle* bervariasi (antara 0% hingga 100%), (Prayogo, R. 2012).



Gambar 2.9 Sinyal PWM dan Rumus (Prayogo, R. 2012).

Perhitungannya rumus perhitungan PWM dapat dilihat pada persamaan (Prayogo, R. 2012):

$$T_{Total} = T_{on} + T_{off} \quad (2)$$

$$D = \frac{T_{on}}{T_{Total}} \quad (3)$$

$$V_{out} = D \times V_{in} \quad (4)$$

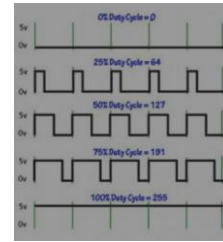
Dimana :

T_{on} = Waktu Pulsa *High*

T_{off} = Waktu Pulsa *Low*

D = *Duty Cycle* adalah lama pulsa *high* dalam satu periode

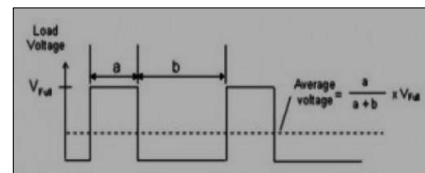
Misalkan suatu PWM memiliki resolusi 8 bit berarti PWM ini memiliki variasi perubahan nilai sebanyak $2^8 = 256$ variasi mulai dari 0 – 255 perubahan nilai yang mewakili *duty cycle* 0 - 100% dari keluaran PWM tersebut.

Gambar 2.10 Persentase *Duty Cycle*

Dengan cara mengatur lebar pulsa “on” dan “off” dalam satu periode gelombang melalui pemberian besar sinyal referensi output dari suatu PWM akan didapat *duty cycle* yang diinginkan. *Duty cycle* dari PWM dapat dinyatakan pada persamaan (Prayogo, R. 2012):

$$Duty\ Cycle = \frac{T_{on}}{(T_{on} + T_{off})} \times 100\% \quad (5)$$

Duty cycle 100% berarti sinyal tegangan pengatur motor dilewatkan seluruhnya. Jika tegangan catu 100V, maka motor akan mendapat tegangan 100V. Pada *duty cycle* 50%, tegangan pada motor hanya akan diberikan 50% dari total tegangan yang ada, begitu seterusnya.



Gambar 2.11 Pengontrolan Tegangan Pulsa PWM (Prayogo, R. 2012).

Dengan menghitung *duty cycle* yang diberikan, akan didapat tegangan output yang dihasilkan. Sesuai dengan rumus yang telah dijelaskan pada persamaan (Prayogo, R. 2012):

$$Average\ Voltage = \frac{a}{a+b} \times V_{full} \quad (6)$$

Dimana :

Average Voltage = Tegangan output pada motor yang dikontrol oleh PWM

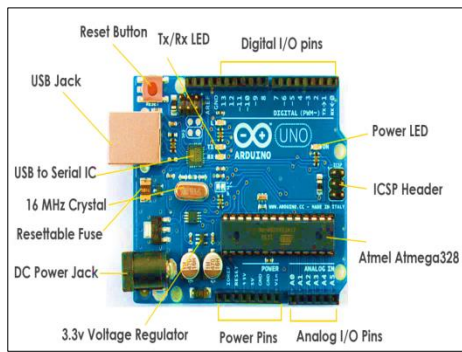
a = Nilai *duty cycle* saat kondisi sinyal “on”

b = Nilai *duty cycle* saat kondisi sinyal “off”

V_{full} = Tegangan maksimum pada motor

2.7 Arduino Uno sebagai Controller

Arduino uno adalah suatu perangkat *prototype* elektronik berbasis mikrokontroler yang fleksibel dan *open source*, perangkat keras dan perangkat lunaknya mudah digunakan. *Board Arduino Uno* menggunakan mikrokontroler ATmega328 seperti yang terlihat pada gambar 2.12. Secara umum letak pin-pin terminal I/O pada berbagai board Arduino posisinya sama dengan letak pin-pin terminal I/O dari Arduino Uno yang mempunyai 14 pin digital yang dapat diset sebagai I/O, 6 pin input Analog.



Gambar 2.12 Board Arduino Uno

Perangkat lunak yang ada pada Arduino dipergunakan untuk membuat program yang menggunakan bahasa pemrograman C/C++ yang akan dioptimasi. Pada *software* Arduino juga telah dilengkapi tutorial dan contoh yang dapat dimodifikasi sesuai dengan kebutuhan, instalasi, dan pengaturan *software* Arduino.

2.8 Kajian Penulisan

- Maulana Yusuf, Adnan Rafi Al Tahtawi, **2017**. Implementasi Sistem Kendali Lepas Landas Quadrotor Menggunakan Pengendali Proporsional Integral Derivatif (PID) yang penelitiannya membahas tentang pengendalian PID yang dirancang untuk quadrotor agar mampu bergerak vertikal dengan tujuan lepas landas dengan stabil.
- Effendi Dodi Arisandi, **2014**. Penelitiannya membahas tentang Kemudahan Pemrograman Mikrokontroler Arduino Pada Aplikasi Wahana Terbang dengan system open source.
- Ryan Nathanael Soenjoto Widodo, Petrus Santoso, Handry Khoswanto, **2013**. Pada kajiannya membahas Aplikasi Android Untuk Remote Control Quadcopter aplikasi Android untuk mengontrol *quadcopter* buatan Parrot yang memiliki empat motor penopang, yaitu AR.Drone 2.0. Berdasarkan pengujian aplikasi dapat mengontrol AR.Drone 2.0 dengan memanfaatkan layar sentuh dan sensor *accelerometer* dengan baik.

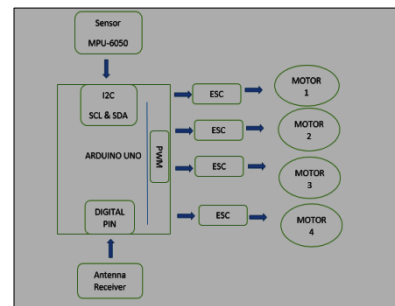
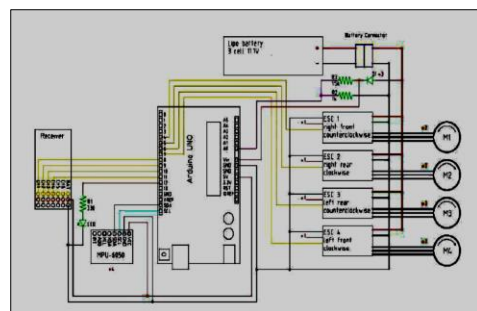
3 METODOLOGI

Spesifikasi *quadcopter* yang akan digunakan memiliki komponen motor *brushless*, ESC, *propeller*, *frame*, baterai, *remote turnigy*, sensor MPU-6050 dan arduino uno. Dimana untuk mengontrol motor *brushless* diperlukan metode PID dengan *output* PWM untuk mengatur kecepatan motor, sehingga perputaran motor dapat dikendalikan dengan stabil. Tabel 3.1 bentuk perputaran motor dan pergerakan quadcopter yang akan dirancang.

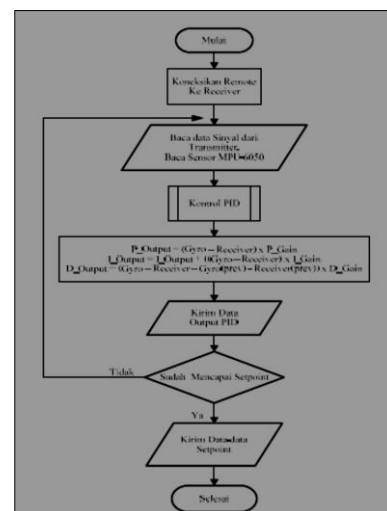
Tabel 3.1 Tabel pergerakan *quadcopter*

Gerakan	Motor 1 CW 1	Motor 2 CCW 2	Motor 3 CW3	Motor 4 CCW4
Naik	Cepat	Cepat	Cepat	Cepat
Turun	Lambat	Lambat	Lambat	Lambat
Mundur	Cepat	Cepat	Lambat	Lambat
Maju	Lambat	Lambat	Cepat	Cepat
Kanan	Lambat	Cepat	Cepat	Lambat
Kiri	Cepat	Lambat	Lambat	Cepat
Rotasi Kanan	Lambat	Cepat	Lambat	Cepat
Rotasi Kiri	Cepat	Lambat	Cepat	Lambat

Pada gambar 3.2 berikut ini juga merupakan perancangan yang berupa blok diagram terdiri dari blok masukan catu daya, sensor, sinyal RC, blok proses berupa arduino dan blok keluaran berupa ESC dan motor *brushless*.

Gambar 3.1 Blok diagram sistem pada *quadcopter*Gambar 3.2 Skematik *Quadcopter*

Pada tahapan ini menggunakan 100system C yang digunakan sebagai 100system pemrograman untuk di *upload* ke 100system100 uno, dengan menggunakan analisa logika PID dan menggunakan *software* 100system100 IDE. Berikut merupakan *flowchart* dari 100system *quadcopter* yang akan dirancang :

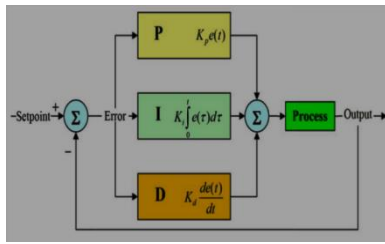


Gambar 3.3 Rancangan Flowchart Quadcopter

3.1 Rancangan *Propotional Integral Differential* (PID) *controller*

Propotional Integral Differential (PID) *controller* merupakan suatu metode yang digunakan untuk mengurangi *error* pada sistem antara keadaan yang diinginkan dengan keadaan *real time*. Bagian-bagian

kontroler PID berupa P, I, D masing-masing bertujuan untuk mempercepat reaksi sebuah sistem dan meminimalkan *error*.

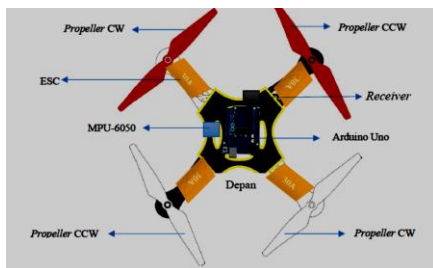


Gambar 3.4 Sistem Kerja PID

4 HASIL DAN PEMBAHASAN

4.1 Perancangan Hardware

Berikut merupakan sketsa desain dari *quadcopter* yang dibangun.

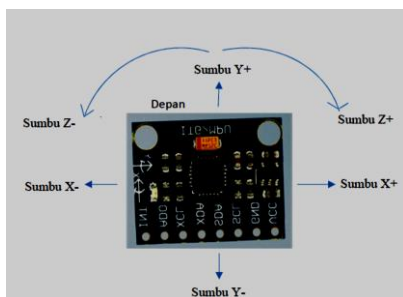


Gambar 4.1 Desain Quadcopter

Dari gambar 4.1 menjelaskan bahwa gambaran umum desain *quadcopter* terdiri dari 4 buah *propeller* yang dipasang pada 4 buah motor dengan masing-masingnya diatur dengan arah yang berbeda yaitu, 2 buah *propeller* diatur sebagai perputaran searah jarum jam (CW) dan 2 buah *propeller* diatur sebagai perputaran berlawanan arah jarum jam (CCW).

4.2 Pengujian dan Hasil nilai sudut pada sensor MPU-6050 dengan sumbu X, Y, Z

Pada tahap pengujian ini dilakukan proses pengambilan data dengan variasi pergerakan sebagai berikut:



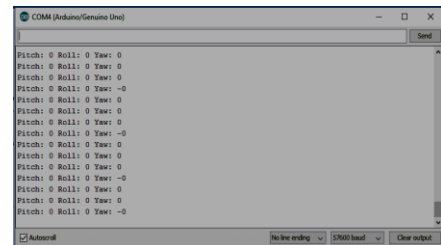
Gambar 4.7 Sumbu pada sensor MPU-6050

Pada gambar 4.7 menunjukan tiap-tiap sumbu yang terdapat pada sensor MPU-6050 yaitu sumbu X, Y, dan Z. Pada kondisi sumbu X+ *quadcopter* mengalami kemiringan ke kanan, pada kondisi sumbu X- *quadcopter* mengalami kemiringan ke kiri, pada kondisi Y+ *quadcopter* menghadap ke atas, pada kondisi Y- *quadcopter* menghadap ke bawah, pada kondisi Z+

quadcopter berotasi ke kanan, pada kondisi Z- *quadcopter* berotasi ke kiri.

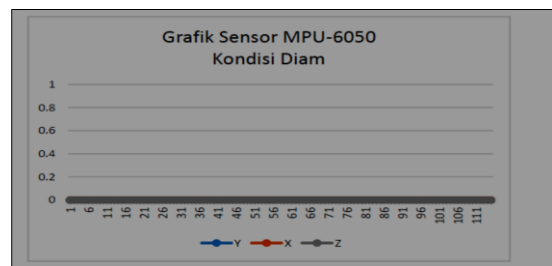
Tabel 4.1 hasil Data quadcopter diam

No	Y	X	Z
1	Pitch: -0	Roll: 0	Yaw: -0
2	Pitch: -0	Roll: 0	Yaw: -0
3	Pitch: 0	Roll: 0	Yaw: -0
4	Pitch: 0	Roll: -0	Yaw: -0
5	Pitch: 0	Roll: -0	Yaw: -0
6	Pitch: 0	Roll: 0	Yaw: -0
7	Pitch: 0	Roll: 0	Yaw: 0
8	Pitch: -0	Roll: -0	Yaw: 0
9	Pitch: 0	Roll: 0	Yaw: -0
10	Pitch: 0	Roll: 0	Yaw: -0
11	Pitch: 0	Roll: -0	Yaw: -0



Gambar 4.8 Data sensor MPU-6050 pada kondisi diam

Pada gambar 4.8 menunjukan pengambilan data sensor MPU-6050 dalam kondisi *quadcopter* diam, data tersebut diambil menggunakan serial monitor pada arduino IDE.



Gambar 4.9 Data Grafik sensor MPU-6050 pada kondisi diam

Dari gambar 4.9 menjelaskan mengenai sensor MPU-6050 masih dalam keadaan stabil dan belum mengalami kemiringan.

4.3 Pengaruh perubahan nilai X, Y, Z pada sensor MPU-6050 terhadap pulsa motor *brushless* saat kondisi diam.

- A. Pengujian dengan pemberian variasi nilai PID pada sumbu X
 Gyro input X = 0
 $K_p = 1.3$
 $K_i = 0.05$
 $K_d = 15$
 Setpoint = 0

Error = gyro input X - Setpoint
 Error = 0 - 0 = 0
 Last error = error
 PID output = $K_p \times \text{error} + K_i \times \text{error} + K_d (\text{error} - \text{last error})$
 PID output = $1.3 \times 0 + 0.05 \times 0 + 15 \times (0 - 0)$
 PID output = 0

- B. Pengujian dengan pemberian variasi nilai PID pada sumbu Y
 Gyro input Y = 0
 $K_p = 1.3$
 $K_i = 0.05$
 $K_d = 15$
 Setpoint = 0
 Error = gyro input Y - Setpoint
 Error = 0 - 0 = 0
 Last error = error
 PID output = $K_p \times \text{error} + K_i \times \text{error} + K_d (\text{error} - \text{last error})$
 PID output = $1.3 \times 0 + 0.05 \times 0 + 15 \times (0 - 0)$
 PID output = 0

- C. Pengujian dengan pemberian variasi nilai PID pada sumbu Z
 Gyro input Z = 0
 $K_p = 2$
 $K_i = 0.02$
 $K_d = 0.1$
 Setpoint = 0
 Error = gyro input Z - Setpoint
 Error = 0 - 0 = 0
 Last error = error
 PID output = $K_p \times \text{error} + K_i \times \text{error} + K_d (\text{error} - \text{last error})$
 PID output = $2 \times 0 + 0.02 \times 0 + 0.1 \times (0 - 0)$
 PID output = 0

Pulsa untuk Esc 1 (Depan_kanan - CCW)
 Pulsa awal Throttle = 1150
 $\text{ESC}_1 = \text{Throttle} - \text{PID_output_Y} + \text{PID_output_X} - \text{PID_output_Z}$
 $\text{ESC}_1 = 1150 - 0 + 0 - 0 = 1150$
 Pulsa untuk Esc 2 (Belakang_kanan - CW)
 $\text{ESC}_2 = \text{Throttle} + \text{PID_output_Y} + \text{PID_output_X} + \text{PID_output_Z}$
 $\text{ESC}_2 = 1150 + 0 + 0 + 0 = 1150$
 Pulsa untuk Esc 3 (Belakang_kiri - CCW)
 $\text{ESC}_3 = \text{Throttle} - \text{PID_output_Y} - \text{PID_output_X} - \text{PID_output_Z}$
 $\text{ESC}_3 = 1150 + 0 - 0 - 0 = 1150$
 Pulsa untuk Esc 4 (Depan_kiri - CW)
 $\text{ESC}_4 = \text{Throttle} - \text{PID_output_Y} - \text{PID_output_X} + \text{PID_output_Z}$
 $\text{ESC}_4 = 1150 - 0 - 0 + 0 = 1150$

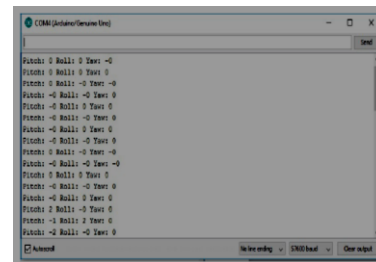
Pada perhitungan diatas dapat disimpulkan bahwa ketika nilai X, Y, Z pada sensor MPU-6050 tidak mengalami perubahan maka nilai PID tidak akan memberikan perubahan pada nilai pulsa yang diterima oleh ESC yang berfungsi untuk memutar motor *brushless*, pada kondisi ini pulsa untuk memutar motor masih dalam kondisi diam.

4.4 Data sensor MPU-6050 pada kondisi quadcopter hidup.

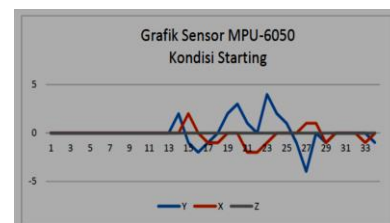
Dari tabel 4.2 berikut dapat disimpulkan bahwa masing-masing nilai pada sumbu X, Y, Z yang telah bernilai variatif maka dapat dipastikan Quadcopter telah mulai hidup.

Tabel 4.2 Data quadcopter hidup

No	Y	X	Z
1	Pitch: 0	Roll: 0	Yaw: -0
2	Pitch: 0	Roll: 0	Yaw: 0
3	Pitch: 0	Roll: -0	Yaw: -0
4	Pitch: -0	Roll: -0	Yaw: 0
5	Pitch: -0	Roll: 0	Yaw: 0
6	Pitch: -0	Roll: -0	Yaw: 0
7	Pitch: -0	Roll: 0	Yaw: 0
8	Pitch: -0	Roll: -0	Yaw: 0
9	Pitch: 0	Roll: -0	Yaw: -0
10	Pitch: -0	Roll: -0	Yaw: -0
11	Pitch: 0	Roll: 0	Yaw: 0
12	Pitch: -0	Roll: -0	Yaw: 0
13	Pitch: -0	Roll: 0	Yaw: 0



Gambar 4.10 Data sensor MPU-6050 pada kondisi hidup



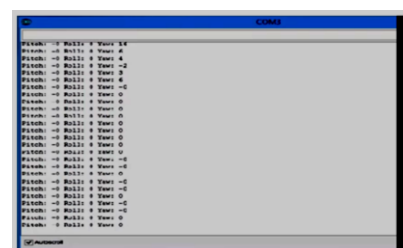
Gambar 4.11 Data Grafik sensor MPU-6050 pada kondisi hidup

4.5 Data sensor MPU-6050 pada kondisi quadcopter terbang.

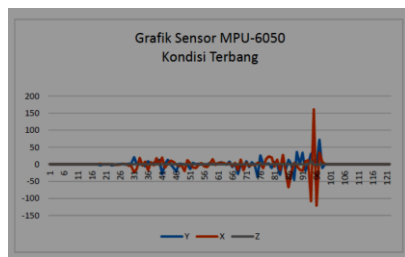
Dari tabel 4.3 berikutnya dapat pula disimpulkan bahwa masing-masing nilai pada sumbu X, Y, Z yang memiliki nilai cukup bervariasi hingga signifikan maka dapat dipastikan *Quadcopter* dapat terbang dengan cukup baik.

Tabel 4.3 data Quadcopter kondisi terbang

No	Y	X	Z
1	Pitch: -0	Roll: 0	Yaw: -0
2	Pitch: -0	Roll: 0	Yaw: -0
3	Pitch: 0	Roll: 0	Yaw: -0
4	Pitch: 0	Roll: -0	Yaw: -0
5	Pitch: 0	Roll: -0	Yaw: -0



Gambar 4.12 Data sensor MPU-6050 pada kondisi terbang



Gambar 4.13 Data Grafik sensor MPU-6050 pada kondisi terbang

4.6 Pengaruh perubahan nilai X, Y, Z pada sensor MPU-6050 terhadap pulsa motor BLDC saat kondisi terbang.

A. Pengujian dengan pemberian variasi nilai PID pada sumbu X

Gyro input X = 26

Kp = 1.3

Ki = 0.05

Kd = 15

Setpoint = 0

Error = gyro input X - Setpoint

Error = 26 - 0 = 26

Last error = error

PID output = Kp x error + Ki x error + Kd (error - last error)

PID output = 1.3 x 26 + 0.05 x 26 + 15 x (26--26)

PID output = -35.1

B. Pengujian dengan pemberian variasi nilai PID pada sumbu Y

Gyro input Y = 2

Kp = 1.3

Ki = 0.05

Kd = 15

Setpoint = 0

Error = gyro input Y - Setpoint

Error = 2 - 0 = 2

Last error = error

PID output = Kp x error + Ki x error + Kd (error - last error)

PID output = 1.3 x 2 + 0.05 x 2 + 15 x (2-2)

PID output = 2.7

C. Pengujian dengan pemberian variasi nilai PID pada sumbu Z

Gyro input Z = 0

Kp = 2

Ki = 0.02

Kd = 0.1

Setpoint = 0

Error = gyro input Z - Setpoint

Error = 0 - 0 = 0

Last error = error

PID output = Kp x error + Ki x error + Kd (error - last error)

PID output = 2 x 0 + 0.02 x 0 + 0.1 x (0-0)

PID output = 0

Pulsa untuk Esc 1 (Depan_kanan - CCW)

Pulsa awal Throttle = 1150

ESC_1 = Throttle - PID_output_Y + PID_output_X - PID_output_Z

ESC_1 = 1150 - 2.7 + (35.1) - 0 = 1182.4

Pulsa untuk Esc 2 (Belakang_kanan - CW)

ESC_2 = Throttle + PID_output_Y + PID_output_X + PID_output_Z

ESC_2 = 1150 + 2.7 + (35.1) + 0 = 1187.8

Pulsa untuk Esc 3 (Belakang_kiri - CCW)

ESC_3 = Throttle + PID_output_Y - PID_output_X - PID_output_Z

ESC_3 = 1150 + 2.7 - (35.1) - 0 = 1117.6

Pulsa untuk Esc 4 (Depan_kiri - CW)

ESC_4 = Throttle - PID_output_Y - PID_output_X + PID_output_Z

ESC_4 = 1150 - 2.7 - (35.1) + 0 = 1112.2

Pada perhitungan diatas dapat disimpulkan bahwa ketika nilai X lebih besar dari Y, pada sensor MPU-6050 mengalami perbedaan dan perubahan maka nilai PID akan memberikan perubahan pada nilai pulsa yang diterima oleh ESC yang berfungsi untuk memutar motor brushless, pada kondisi ini pulsa untuk memutar motor ESC 3 dan ESC 4 makin lambat dan masih dalam kondisi pelan sehingga terlihat motor brushless bergerak pelan. Motor yang paling cepat berputar adalah ESC 1 dan 2, karena sudut kemiringan terlalu miring kearah kanan X+ dengan nilai sudut 35.1 dan sensor MPU-6050 berusaha mencapai titik setpoint secara perlahan.

Dapat disimpulkan bahwa semakin besar perubahan sudut sensor MPU-6050 maka akan semakin cepat dan besar pulsa yang diberikan kepada ESC, sehingga dengan pemberian pulsa berdasarkan kemiringan tersebut dapat mengontrol perputaran motor brushless agar dapat mencapai titik keseimbangan yang baik. Dan sebaliknya semakin kecil perubahan sudut sensor MPU-6050, maka pulsa yang dikirimkan ke ESC semakin sedikit dan menyebabkan putaran motor lambat.

5 KESIMPULAN DAN SARAN

Pada penelitian ini telah didapatkan rancangan desain *quadcopter* menggunakan arduino uno untuk nilai keseimbangan mengacu pada parameter *pith*, *roll*, dan *yaw* dapat mencapai kestabilan dan berusaha mencapai setpoint yaitu nilai 0. Dengan pengujian keseimbangan *quadcopter* dengan mengaplikasikan PID controller dimana parameter kontroler PID dengan nilai Kp = 1.3, Ki = 0.05 dan Kd = 15 didapat berdasarkan *trial* dan *error* yang dilakukan, sehingga respon keseimbangan *quadcopter* yang dikirimkan sensor IMU sangat baik dan cepat. Itu teridentifikasi dengan data keseimbangan yang terekam.

DAFTAR PUSTAKA

- [1] Donald Norris, 2014. *Build Your Own Quadcopter*. New York: Mc Graw Hill Education.
- [2] Heri Andrianto & Aan Darmawan, 2016. *Arduino belajar cepat dan Pemrograman*. Bandung: Informatika.
- [3] Abdul Kadir, 2012. *Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya Menggunakan Arduino*. Yogyakarta: Andi Offset.
- [4] Subrata, Rosalia H, 2017. *Perancangan Pengendalian PID untuk Gerakan Pitch dan Roll pada Quadcopter*. ISSN 1412-0372. Malang: Universitas Trisakti.

- [5] Jati U, Bimo, 2016. Rancang bangun UAV (*Unmaned Aerial Vehicle*) model *quadcopter* dengan menggunakan algoritma *propotional integral derivative*. ISSN 2442-5826. Bandung: Universitas Telkom.
- [6] Laksono, Heru Dibyo, 2015. *Metoda-Metoda untuk Analisa Kestabilan Sistem Kendali dengan Matlab*. Padang: Andalas university press.
- [7] ambabu, S, 2007. *Modeling And Control Of A Brushless Dc Motor*.
- [8] Imas Tri Setyadewi dan Yudha Agung Nugroho (2018), ANALISIS PEMILIHAN PROPELLER MESIN PESAWAT TANPA AWAK LSU 03 (PROPELLER ENGINE SELECTION ANALYSIS FOR UNMANNED AIRCRAFT LSU 03), *Majalah Sains dan Teknologi Dirgantara Vol. 13 No. 1 Juni 2018:23-30*
- [9] Christin P.R. Tuuk, Vecky C. Poekoel, Jane Litouw (2018), Implementasi Pengendali PID Untuk Kestabilan Posisi Terbang Wahana Tanpa Awak, *Jurnal Teknik Elektro dan Komputer Vol. 7 No. 1 (2018), ISSN : 2301 - 8402*
- [10] Bimo Jati Utomo¹ (2015), RANCANG BANGUN UAV (*UNMANNED AERIAL VEHICLE*) MODEL QUADCOPTER DENGAN MENGGUNAKAN ALGORITMA PROPORTIONAL INTEGRAL DERIVATIVE , e-Proceeding of Applied Science : Vol.1, No.1 April 2015 | Page 57, ISSN : 2442-5826