



---

# PERANCANGAN PENGENDALI PID UNTUK GERAKAN *PITCH* DAN *ROLL* PADA *QUADCOPTER*

*Rosalia H. Subrata, Raymond Tarumasely & Calvin Dwianto S.*

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Trisakti

Jalan Kiai Tapa No. 1, Grogol, Jakarta Barat

*E-mail: rosalia@trisakti.ac.id*

## **ABSTRACT**

*Recently, Quadcopter is used a lot due to it's capability to perform difficult movements compared with other type of air craft and also for its capability to take off and land on narrow space vertically. The Proportional Integral Derivate (PID) Controller is expected to control the Quadcopter's angle movement pitch and role back to the original position. Design process is conducted by creating a mathematical model of the system to stabilize pitch and roll movement of the Quadcopter and handle other types of disturbance. The parameters of the PID controller are determined by using MATLAB v.13 program through simulation process. The parameters are used as references for Arduino Mega Microcontroller to perform pitch and roll movement of the Quadcopter. By using PID controller coefficient  $K_p = 0.0316$ ,  $K_i = 0.05$ ,  $K_d = 0.26$ , the controller is able to reduce the oscillation of the system with error rate about two degrees from the input.*

**Keywords:** *quadcopter, pitch, roll, PID*

## **ABSTRAK**

*Quadcopter saat ini banyak digunakan karena mempunyai kemampuan untuk melakukan gerakan yang sulit dilakukan oleh pesawat jenis lain, serta kemampuan untuk terbang dan mendarat secara vertikal di tempat yang sempit. Pengendali Proportional Integral Derivative (PID) untuk gerakan sudut pitch dan roll pada quadcopter ini diharapkan mampu mengembalikan posisi pitch dan roll sesuai dengan referensi yang dimasukkan ketika diberikan dorongan pada quadcopter. Proses perancangan dimulai dengan membuat model matematis sistem terlebih dahulu. Dibutuhkan sebuah sistem pengendali untuk mengatur gerakan sudut pitch dan roll pada quadcopter, sehingga dapat bergerak dengan stabil dan mampu mengatasi gangguan dari luar. Sebagai pengendali gerakan sudut pitch dan roll digunakan pengendali PID, yang parameternya ditentukan dengan program aplikasi MATLAB versi 13. Parameter PID ini diprogram ke mikrokontroler, dalam hal ini digunakan Arduino Mega. Osilasi pada sistem dapat diredam dengan koefisien PID  $K_p = 0,0316$ ,  $K_i = 0,05$ , dan  $K_d = 0,2$ . Besarnya error rate sekitar dua derajat dari input yang diberikan.*

**Kata kunci:** *quadcopter, pitch, roll, PID*

---

## 1. PENDAHULUAN

*Quadcopter (Quadrotor Helicopter)* merupakan salah satu jenis *unmanned aerial vehicle (UAV)* yang memiliki empat baling-baling yang digerakkan oleh empat *brushless* motor. Keempat motor ini dapat dikendalikan kecepatannya secara terpisah, dua motor berputar searah jarum jam dan dua motor lain berputar berlawanan arah jarum jam [1].

Helikopter merupakan salah satu alat transportasi udara yang menggunakan *rotary wing* untuk bergerak dan mampu untuk melayang di udara dalam posisi diam (*hovering*), namun memiliki kecepatan translasi yang lebih kecil dari alat transportasi *fixed wing*. Helikopter konvensional dengan satu rotor utama dan satu rotor di ekor memiliki banyak sifat yang sama dengan *quadcopter*. *Quadcopter* tidak mempunyai bagian yang bergerak kecuali baling-baling yang digerakkan oleh motor, sedangkan helikopter konvensional membutuhkan *hub* kompleks untuk memutar sumbu motor agar dapat menginduksi gerakan translasi [1].

*Quadcopter* mempunyai kelebihan pada mobilitas dan fleksibilitas untuk menjelajahi wilayah yang sempit [2]. *Quadcopter* mempunyai kemampuan melakukan manuver yang sulit dilakukan pesawat jenis lain, melayang di udara dalam posisi diam (*hovering*), serta terbang dan mendarat secara vertikal (VTOL, *vertical takeoff and landing*). Tidak seperti kendaraan di darat, pergerakan *quadcopter* memiliki sedikit gesekan, oleh karena itu harus ada redaman untuk menghentikan gerakannya dan menjaga kestabilannya [1]. *Quadcopter* saat ini banyak digunakan dalam bidang militer untuk memantau batas-batas wilayah maupun sebagai mata-mata dalam peperangan, bidang fotografi untuk mengambil gambar dari udara, bidang geografi untuk memantau cuaca di suatu wilayah, dan pada mitigasi bencana sebagai sarana untuk melihat kondisi di sekitar daerah bencana dari jarak jauh.

Perancangan pengendali untuk gerakan sudut *pitch* dan *roll* pada *quadcopter* diharapkan mampu mengembalikan posisi *pitch* dan *roll* sesuai dengan referensi yang diberikan ketika mendapat gangguan dari luar. Tidak ada sensor yang dapat mengukur sudut secara langsung, oleh karena itu sudut diestimasi menggunakan

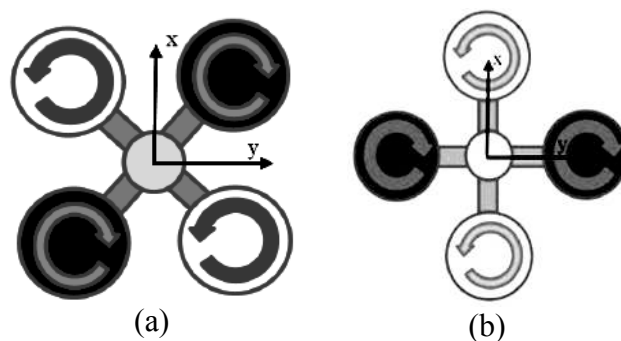
*accelerometer* dan atau sensor *gyroscope*. Sudut *roll* dikendalikan dengan cara menambah atau mengurangi kecepatan salah satu motor kiri atau kanan. Sudut *pitch* dikendalikan dengan cara menambah atau mengurangi kecepatan salah satu motor depan atau belakang. Karena *quadcopter* membutuhkan ketelitian dan *error rate* yang kecil, maka sistem ini menggunakan pengendali *Proportional Integral Derivative* (PID).

Hal terpenting yang dibahas dalam makalah ini adalah pemodelan dan penentuan parameter PID sistem. Parameter PID ditentukan dengan bantuan program MATLAB karena MATLAB memiliki fungsi *pidtune* yang menggunakan pendekatan *crossover frequency*.

## 2. KAJIAN PUSTAKA

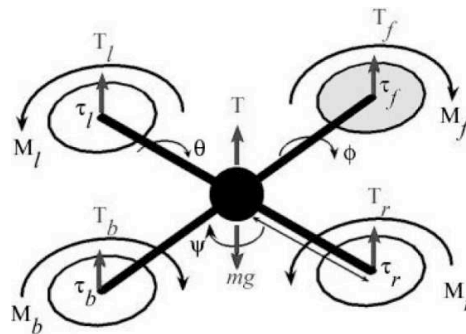
### 2.1 Gerakan Dasar *Quadcopter*

*Quadcopter* memiliki empat buah baling-baling yang digerakkan oleh empat motor *brushless*. Kecepatan keempat motor ini dapat dikendalikan secara terpisah. Ada dua jenis pengaturan letak motor relatif terhadap sistem koordinat *body frame*, yaitu konfigurasi "x" dan konfigurasi "+" yang diperlihatkan pada Gambar 1 [3].



Gambar 1 (a) Konfigurasi "x" dan (b) Konfigurasi "+"

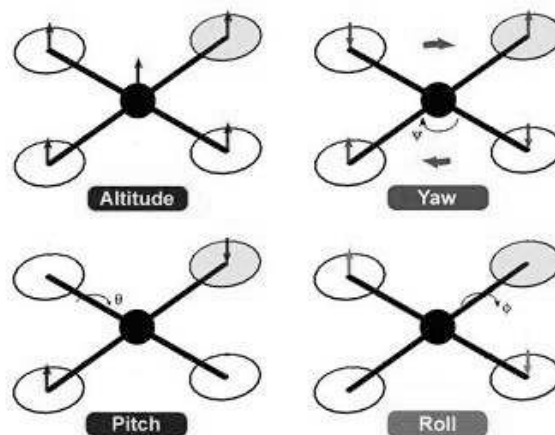
Gerakan yang dilakukan *quadcopter* terdiri dari gerakan translasi dan gerakan rotasi. Gerak translasi terdiri dari gerak di sumbu x, y dan z. Gerak rotasi terdiri dari gerak *roll* ( $\phi$ ), *pitch* ( $\theta$ ) dan *yaw* ( $\psi$ ). Konsep gerakan motor *quadcopter* diilustrasikan dalam *free body diagram quadcopter* pada Gambar 2 [4].



Gambar 2 Konsep Gerakan Motor *Quadcopter*

Pasangan motor depan ( $M_f$ ) dan motor belakang ( $M_b$ ) berputar searah jarum jam, sedangkan pasangan motor kanan ( $M_r$ ) dan motor kiri ( $M_l$ ) berputar berlawanan arah jarum jam. Konfigurasi ini dirancang untuk menyeimbangkan gaya tarik dan gaya dorong yang dihasilkan masing-masing pasangan motor. *Quadcopter* dapat terbang melayang di udara dengan stabil karena ada pergerakan yang menghasilkan gaya dorong serta gaya tarik yang melawan gaya gravitasi.

Kecepatan setiap motor memiliki pengaruh dalam gerakan *quadcopter*. Agar dapat bergerak naik diperlukan kecepatan yang sama dan cukup besar pada keempat motornya. Untuk melakukan *hovering* keempat motornya harus mempunyai kecepatan yang sama. Kombinasi kecepatan motor menghasilkan gerakan *quadcopter* yang berbeda. Gambar 3 [4] memperlihatkan gerakan dasar *quadcopter* dengan mengubah-ubah kecepatan dari keempat motornya.



Gambar 3 Gerakan Dasar *Quadcopter*



Gerakan dasar *quadcopter* terdiri dari gerakan naik turun, berputar pada sumbu datar, dan ke samping kanan dan kiri pada sumbu z.

1. Gerakan naik turun

Untuk bergerak terbang ke atas dan ke bawah, *quadcopter* perlu menaikkan atau menurunkan kecepatan putar setiap motor dengan jumlah yang sama. Menaikkan kecepatan akan mengakibatkan *quadcopter* terbang ke atas, dan menurunkan kecepatan akan mengakibatkan *quadcopter* terbang ke bawah.

2. Gerakan berputar pada sumbu datar (sumbu x dan y)

Untuk dapat berputar pada sumbu x (gerakan *roll*) dilakukan dengan cara menambah atau mengurangi kecepatan salah satu motor kiri atau kanan secara seimbang. Untuk berputar pada sumbu y (gerakan *pitch*) perubahan kecepatan dilakukan pada pasangan motor depan dan belakang. Perubahannya adalah salah satu anggota pasangan motor dikurangi kecepatannya dan anggota pasangan motor lainnya dinaikkan kecepatannya dengan selisih kecepatan yang sama, sedangkan pasangan motor lainnya dibiarkan dengan kecepatan tetap. Dengan pengaturan seperti ini, *quadcopter* akan bergerak memutar dari arah pasangan motor yang berkecepatan lebih rendah, ke arah pasangan motor yang berkecepatan lebih tinggi.

3. Gerakan ke samping kanan dan kiri pada sumbu z ( $\psi$ )

Gerakan ini dilakukan dengan menurunkan kecepatan satu pasang motor depan-belakang atau kiri-kanan, dan menaikkan kecepatan satu pasang motor lainnya. Dengan pengaturan seperti ini, *quadcopter* akan bergerak berputar ke arah perputaran pasangan motor yang lebih lambat kecepatannya dibandingkan pasangan yang lainnya.

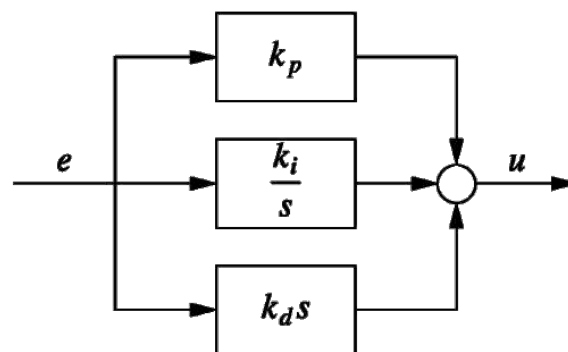
## 2.2 Pengendali PID

Pengendali PID merupakan pengendali yang terdiri dari tiga jenis pengendali, yaitu pengendali *Proportional*, *Integral*, dan *Derivative*. Pengendali PID merupakan salah satu pengendali yang banyak digunakan pada sistem kendali industri. Diagram blok pengendali PID diperlihatkan pada Gambar 4 [5].

*Transfer function* pengendali PID untuk sistem kontinu dinyatakan dengan Rumus (1) berikut ini [5]

$$\frac{U(s)}{E(s)} = K_p + K_d s + K_i \frac{1}{s} \quad (1)$$

dengan *input* pengendali adalah sinyal *error*  $e(t)$ , *output*-nya adalah sinyal kendali  $u(t)$ , dan  $K_p$ ,  $K_d$ , serta  $K_i$  merupakan koefisien pengendali.



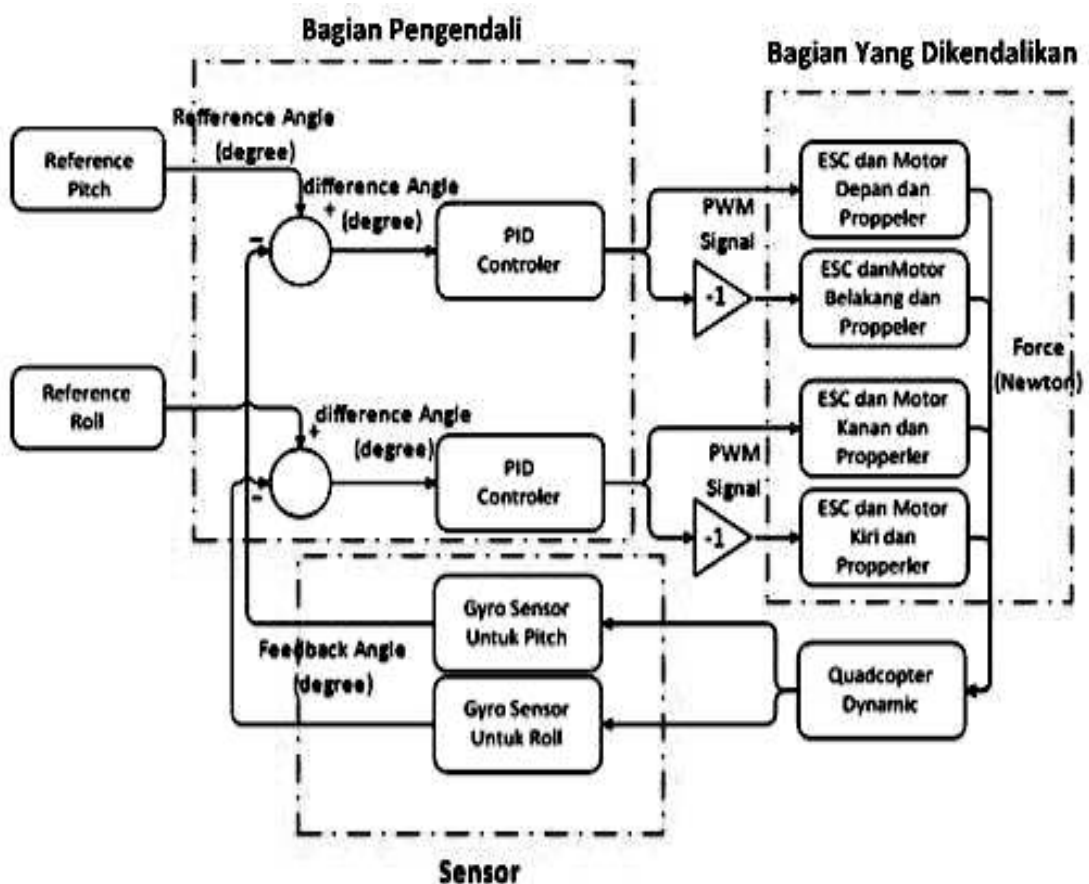
Gambar 4 Diagram Blok Pengendali PID

### 3. METODE PENELITIAN

Komponen-komponen yang dibutuhkan dalam merancang pengendali PID *quadcopter* yaitu (1) empat buah motor *brushless*; (2) *electronic speed controller* (ESC); (3) empat buah *propeller* (baling-baling); (4) *frame*; (5) *microcontroller* (Arduino Mega); dan (6) sensor *inertial measurement unit* (IMU).

Secara umum komponen-komponen tersebut terdiri dari tiga bagian yaitu bagian yang dikendalikan, sensor, dan bagian pengendali seperti terlihat pada Gambar 5. Bagian yang dikendalikan terdiri dari motor, ESC, dan *propeller*. *Frame* merupakan rangka *quadcopter* tempat untuk memasang motor, ESC dan baling-baling. Motor *brushless* yang dilengkapi ESC (sebagai *driver* motor) berfungsi sebagai penggerak baling-baling untuk menghasilkan gaya angkat [6]. Kecepatan putaran setiap baling-baling dapat diatur secara terpisah. Variasi kecepatan motor dapat menghasilkan gaya angkat dan menghasilkan gerakan *quadcopter*. Sensor

*Inertial Measurement Unit* (IMU) terdiri dari sensor *accelerometer* dan *gyroscope* untuk mengetahui posisi serta lintasan yang dilalui *quadcopter*. Dalam penelitian ini, sensor hanya digunakan untuk pembacaan *gyroscope* saja dengan alasan penelitian lebih ditekankan pada pengendalian gerakan *pitch* dan *roll*. Pengendali PID ini kemudian diprogram pada *microcontroller* (Arduino Mega) untuk mengatur gerakan sudut *roll* dan *pitch* ketika diberi dorongan sehingga sesuai referensi yang diberikan.

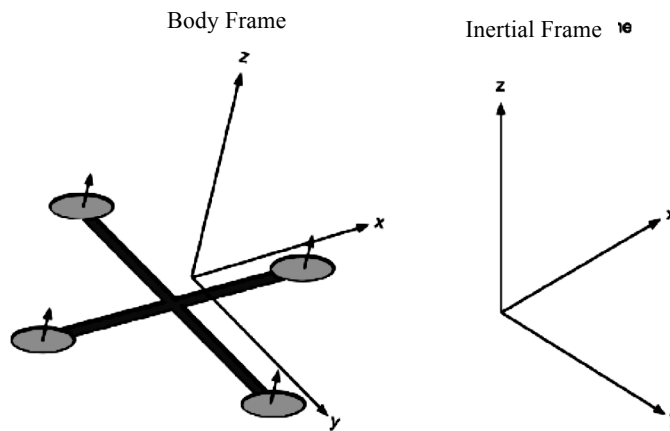


Gambar 5 Diagram Blok Sistem

### 3.1 Pemodelan Sistem

#### 3.1.1 *Quadcopter*

Konfigurasi *frame quadcopter* ada 2 macam yaitu konfigurasi “x” dan plus (+) [3]. Konfigurasi *frame quadcopter* pada penelitian ini menggunakan konfigurasi *frame plus (+)*, *body frame* dan *inertial frame* diperlihatkan pada Gambar 6.



Gambar 6 Sistem Koordinat *Body Frame*

Sistem koordinat  $x$ ,  $y$  dan  $z$  pada Gambar 6 akan menjadi patokan dari putaran sistem dengan posisi  $x = (x, y, z)$ , kecepatan rotasi  $\dot{x} = (\dot{x}, \dot{y}, \dot{z})$ , dan sudut *roll*, *pitch*, dan *yaw* dari *quadcopter*  $(\phi, \theta, \psi)$ . Hubungan gerak rotasi dapat diturunkan dengan menggunakan *state variable* yang sesuai, yaitu sudut *frame quadcopter*  $(\phi, \theta, \psi)$  dan kecepatan *angular body frame*  $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ . Persamaan gerak *quadcopter* yang sesuai dengan *free body diagram* pada Gambar 2 dapat dinyatakan dengan Persamaan (2) berikut [4]:

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{j_y - j_z}{j_x} \dot{\psi} \dot{\theta} \\ \frac{j_z - j_x}{j_y} \dot{\phi} \dot{\psi} \\ \frac{j_x - j_y}{j_z} \dot{\theta} \dot{\phi} \end{bmatrix} + \begin{bmatrix} \frac{1}{j_x} \tau_\phi \\ \frac{1}{j_y} \tau_\theta \\ \frac{1}{j_z} \tau_\psi \end{bmatrix} \quad (2)$$

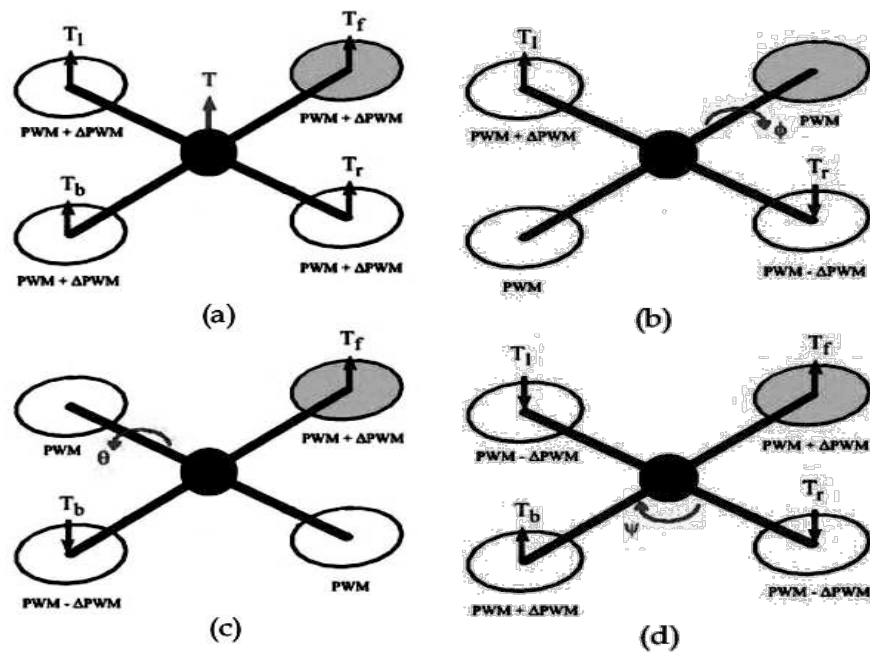
dengan

$$j_x = j_y = j_z = \frac{2M_0 r^2}{5} + 2l^2 m \quad (3)$$

$M_0$  adalah massa pusat *body frame*,  $r$  adalah jari-jari pusat *body frame*,  $l$  adalah lengan *frame*,  $m$  adalah massa sebuah motor,  $\tau_\phi$  adalah torsi *roll*,  $\tau_\theta$  adalah torsi *pitch* dan  $\tau_\psi$  adalah torsi *yaw*.



Untuk menggerakkan *quadcopter*, keempat motor baling-baling memerlukan gaya dan torsi. Gerak *quadcopter* dapat dikendalikan dengan mengatur kecepatan putaran masing-masing motor seperti diperlihatkan pada Gambar 7. Momen adalah torsi yang dihasilkan dalam mencapai gerakan *roll*, *pitch* dan *yaw*.



Gambar 7 Arah gerakan (a) *Up*, (b) *Roll*, (c) *Pitch*, (d) *Yaw*

Total gaya dorong *quadcopter* adalah jumlah gaya dorong yang dihasilkan setiap baling-baling [4]:

$$T = T_f + T_r + T_b + T_l \quad (4)$$

Torsi putaran *roll*, *pitch* dan *yaw* adalah:

$$\tau_\phi = l(T_l - T_r) \quad (5)$$

$$\tau_\theta = l(T_f - T_b) \quad (6)$$

$$\tau_\psi = \tau_f + \tau_b - \tau_r - \tau_l \quad (7)$$

Gaya dorong dan torsi dari sebuah baling-baling yang berputar pada kecepatan putar  $\omega$  adalah:

$$T = c_T \omega^2 \quad (8)$$

$$Q = c_Q \omega^2 \quad (9)$$

dengan  $c_T$  dan  $c_Q$  adalah koefisien dorong dan torsi yang berbeda-beda pada setiap baling-baling yang digunakan.

### 3.1.2 Motor, ESC dan Baling-Baling

Pemodelan pada penelitian ini dilakukan dengan pendekatan *transfer function*. Model *transfer function* dari ESC, motor dan baling-baling telah diteliti pada [6]:

$$\frac{T(z)}{N(z)} = \frac{0,615}{z^2 - 0,3108z + 0,6253} \quad (10)$$

Untuk mendapatkan model *transfer function* gerak rotasi, diturunkan persamaan fisik sebagai berikut:

$$T_1 l - T_2 l = J\theta \quad (11)$$

Bila  $T_2 = -T_1$  maka:

$$T_1 = \frac{J}{2l} \theta \quad (12)$$

*Transfer function* dalam fungsi diskrit adalah sebagai berikut:

$$\frac{\theta(z)}{T(z)} = \frac{0,01613z}{z^2 - 2z + 1} \quad (13)$$

### 3.2 Perancangan Pengendali PID

*Transfer function* pengendali PID adalah seperti rumus berikut [5]:

$$PID(s) = K_p + K_d s + K_i \frac{1}{s} \quad (14)$$

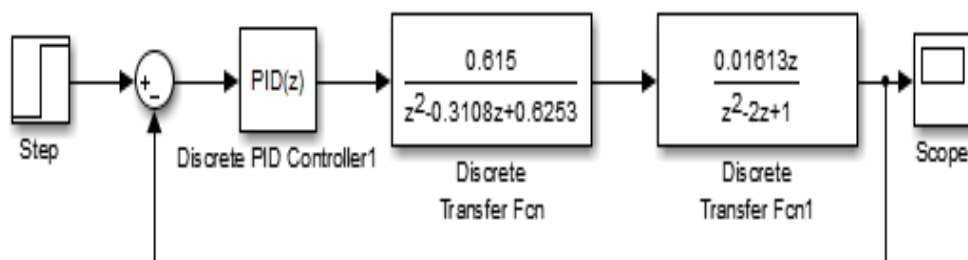
Dalam sistem diskrit menjadi:

$$PID(z) = K_p + \frac{K_i z}{z-1} + \frac{K_d(z-1)}{z} \quad (15)$$

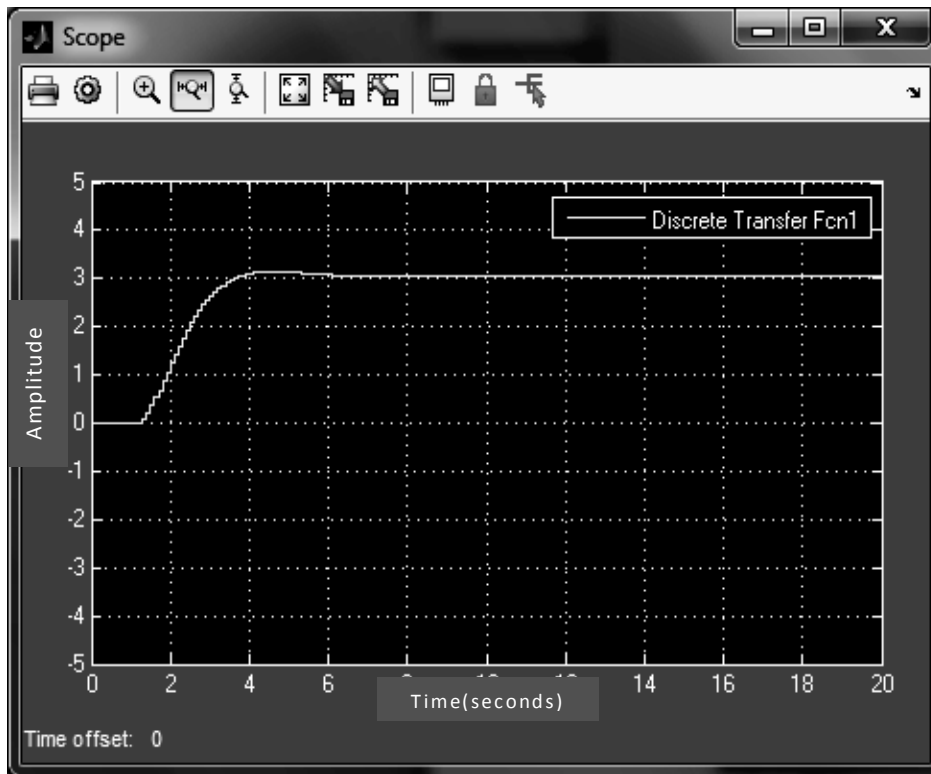
$$= \frac{(K_p + K_i + K_d)z^2 - (K_p + 2K_d)z + K_d}{z^2 - z}$$

Koefisien  $K_p$ ,  $K_d$ , dan  $K_i$  diperoleh dengan menggunakan fungsi *pidtune* pada program MATLAB, fungsi *pidtune* ini menggunakan pendekatan *crossover frequency*. *Pidtune* akan memilih *crossover frequency* dari 0,52 – 1,0 rad/sec untuk mendapatkan *time response* yang lebih baik.

Diagram blok pada Gambar 8 diperoleh dengan menggunakan *simulink*. Respons sistem terhadap waktu yang diperlihatkan pada Gambar 9 dihasilkan menggunakan fungsi PID *tuner* dengan koefisien PID  $K_p = 0,0148$ ,  $K_i = 0,00015$ ,  $K_d = 1,078$ , dan  $N(\text{koefisien filter}) = 2,05$ . Terlihat bahwa sistem stabil pada detik keenam.



Gambar 8 Diagram Blok Pitch dan Roll



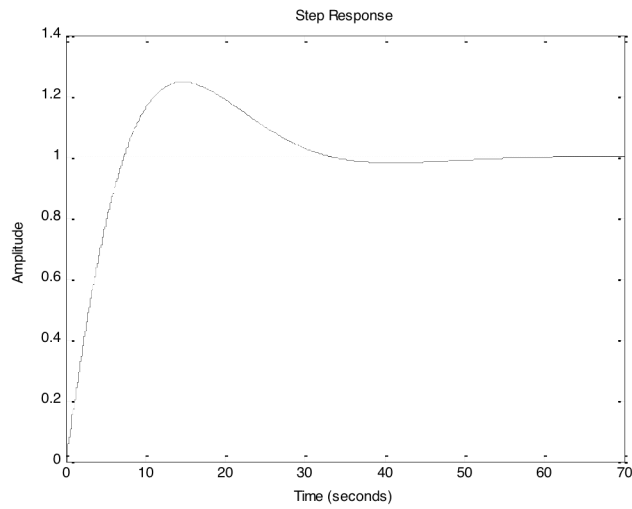
Gambar 9 Hasil Respon Sistem dengan PID

Jika tidak menggunakan filter pada *Derivative*-nya, dengan menggunakan fungsi *pidtune* dengan *sample time* 0,1 second sebagai berikut:

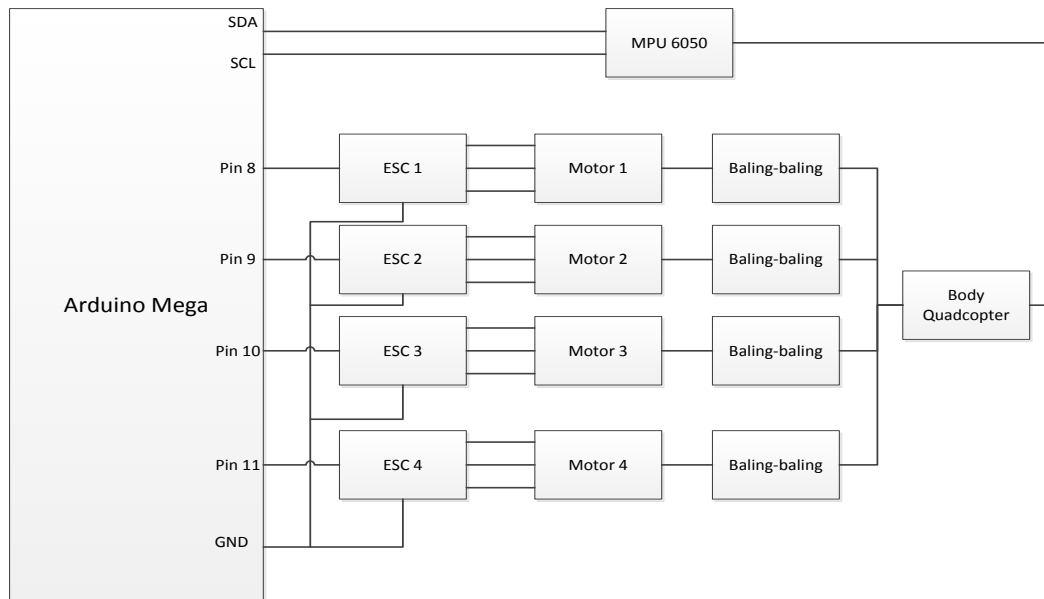
```
G1 =tf([0.615],[1 -0.3208 0.6253],0.1);  
G2 =tf([0.01613 0],[1 -2 1],0.1);  
pid = pidtune(G1*G2,'pid')  
G = pid*G1*G2;  
trans = feedback(G,1);  
step(trans)
```

diperoleh koefisien PID  $K_p = 0,0316$ ,  $K_i = 0,05$ ; dan  $K_d = 0,26$ . Jika diberi input *unit step* maka akan dihasilkan *output* seperti pada Gambar 10.

Pengendali PID yang dirancang ini selanjutnya diprogram ke dalam mikrokontroler berbasis Arduino Mega. Gambar blok diagram sistem keseluruhan dapat dilihat pada Gambar 11.



Gambar 10 Output Sistem dengan PID yang Dihasilkan Fungsi *Pidtune*



Gambar 11 Blok Diagram Sistem Keseluruhan

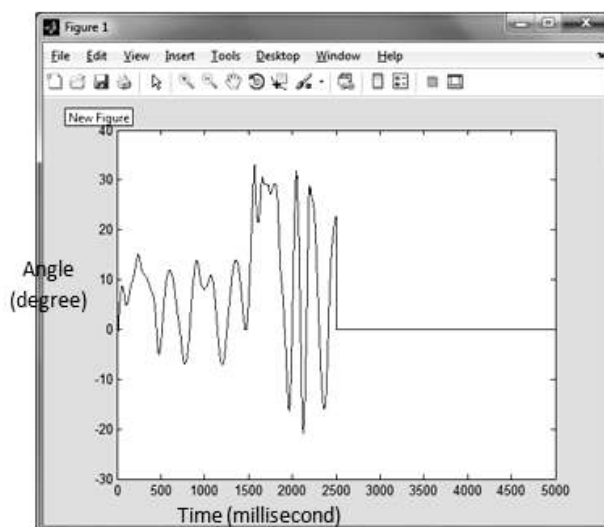
#### 4. HASIL DAN PEMBAHASAN

Rancangan fisik *quadcopter* pada penelitian ini dapat dilihat pada Gambar 12. Setelah dilakukan perancangan pengendali PID maka dilakukan uji coba pada *quadcopter* hasil rancangan dengan menggunakan Arduino Mega sebagai mikrokontroler.

Gambar 12 Fisik *Quadcopter*

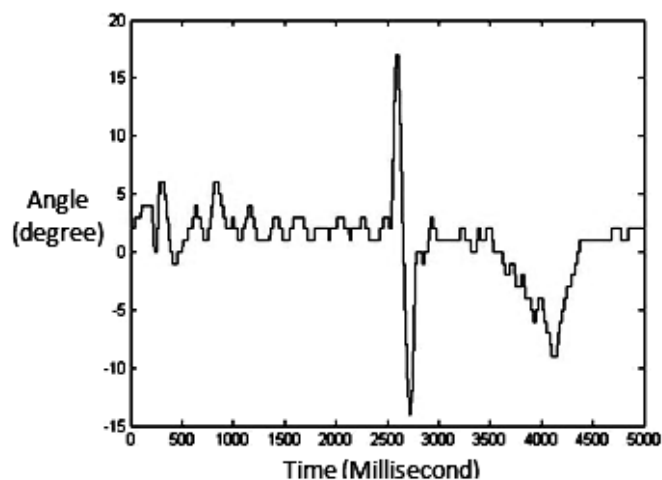
Pada saat kondisi *quadcopter* datar, sudut yang dibaca oleh *gyroscope* tidak menunjukkan sudut nol derajat, melainkan sudut *pitch* sebesar 5 derajat, sedangkan sudut *roll* -8 derajat.

Pada Gambar 13 dapat dilihat respon gerakan *pitch* yang dibaca dari *gyroscope* (dalam derajat) terhadap waktu (dalam *millisecond*) dengan nilai  $K_p = 0,0148$ ,  $K_i = 0,00015$ ,  $K_d = 1,078$ , dan  $N(\text{koefisien filter}) = 2,05$ . Pada Gambar 13 dapat dilihat bahwa respon yang sebenarnya tidak sesuai dengan hasil simulasi. Gerakan sistem hanya terdeteksi dari detik pertama hingga detik ke 2,5, setelah itu terjadi *error* pada Arduino dan Arduino tiba-tiba berhenti mengirimkan data.

Gambar 13 Respon Gerakan *Pitch* Terhadap Waktu dengan Pengendali PID Menggunakan Filter

Jika diobservasi dari awal hingga detik ke 2,5, dapat dilihat bahwa sistem tidak stabil dan ketika diberi gangguan berupa dorongan dari luar, pengendali tidak mampu mengembalikan kestabilannya.

Hasil pengujian tanpa menggunakan filter dengan koefisien PID yaitu  $K_p = 0,0316$ ,  $K_i = 0,05$ , dan  $K_d = 0,26$  yang dihasilkan PID *tuner* berupa respon gerakan *pitch* (derajat) terhadap waktu (*millisecond*) dapat dilihat pada Gambar 14. Pada Gambar 14 tampak *oscilasi* kemiringan yang dialami oleh sistem lebih sedikit, dengan *error* sekitar 2 derajat. Dari detik ke-nol hingga detik ke-1 sistem sedang melakukan penyesuaian ketika *quadcopter* diangkat. Dari detik ke-1,5 sampai detik ke-2,5 sistem stabil. Pada detik ke-3,5 – 4,5 pada salah satu lengan *quadcopter* diberikan dorongan kecil. Dapat dilihat dari Gambar 14 bahwa sistem mampu mengembalikan posisinya ke posisi semula.



Gambar 14 Respon Gerakan *Pitch* dengan Pengendali PID Tanpa Filter

## 5. KESIMPULAN

Dari hasil pengujian dan analisis perancangan pengendali PID untuk gerakan *pitch* dan *roll* pada *quadcopter* ini dapat ditarik kesimpulan sebagai berikut:

1. Dengan koefisien PID  $K_p = 0,0316$ ,  $K_i = 0,05$  dan  $K_d = 0,26$  untuk mengendalikan gerakan *pitch* dan *roll*, sistem pengendali *quadcopter* mampu meredam osilasi dan menghasilkan *error rate*  $\pm 2$  derajat dari *input*.

2. Dengan koefisien PID tanpa menggunakan filter pada *derivative*-nya, sistem pengendali *quadcopter* mampu meredam osilasi hingga menuju satu *state* tertentu pada detik ke-1 hingga 2,5.
3. Koefisien pengendali PID yang dirancang tanpa filter mampu mengembalikan kondisi sudut *pitch* dan *roll* ketika diberi gangguan dari luar pada detik ke-3,5 hingga 4,5.

#### DAFTAR PUSTAKA

- [1] Andreas Vikane Hystad, Joakim Brobakk Lehn. "Model, Design and Control of a Quadcopter." Master Thesis in Cybernetics and Robotics, Norwegian University of Science and Technology Department of Engineering Cybernetics, 2015.
- [2] Gembong Edhi Setyawan, Eko Setiawan, Wijaya Kurniawan. "Sistem Kendali Ketinggian Quadcopter Menggunakan PID". *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, Vol. 2, No. 2, hlm. 125-131, Oktober 2015.
- [3] S. Norouzi Ghazbi, Y. Aghli, M. Alimohammadi, A. A. Akbari. "Quadrotors Unmanned Aerial Vehicles: A Review." *International Journal On Smart Sensing And Intelligent Systems*, Vol. 9, No. 1, hlm 309-333, 2016.
- [4] Syed Ali Raza, Wail Gueaieb. "Intelligent Flight Control of an Autonomous Quadrotor", dalam *Motion Control*, Federico Casolo, Ed. New York City: InTech, 2010, hlm. 245-264.
- [5] S. Palani. *Control Systems Engineering*. 2<sup>nd</sup> edition. New Delhi: Tata McGraw Hill, 2010, hlm. 5.83.
- [6] Jose C. V. Junior, Julio C. De Paula, Gideon V. Leandro, Marlio C. Bonfim. "Stablility Control of a Quad-Rotor Using PID Controller." *Brazilian Journal Of Instrumentation and Control*, hlm. 15-20, November 2013.