

## SISTEM PENGENDALIAN *TWIN ROTOR MIMO* DENGAN KONTROLER PID BERBASIS LABVIEW 2014

**Achmad Tri Ofandi Mardianto**

S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia  
e-mail : achmadmardianto@mhs.unesa.ac.id

**Endryansyah**

Dosen Teknik Elektro, Fakultas Teknik, Universitas negeri Surabaya, Ketintang 60231, Indonesia  
e-mail : endryansyah@unesa.ac.id

### Abstrak

*Twin Rotor MIMO* merupakan salah satu sistem aerodinamik yang mirip dengan helikopter. Dari beberapa penelitian yang telah dilakukan sebelumnya memiliki beberapa kekurangan yaitu nilai kontroler PID (*Proportional-Integral-Derivative*) ke *hardware* tanpa melalui simulasi terlebih dahulu, sehingga *tuning* dilakukan tanpa mempertimbangkan spesifikasi dari *hardware* yang digunakan. Selain itu data respon sistem tidak secara *real-time* karena harus dipindahkan dahulu dari serial monitor ke MATLAB untuk dapat dilihat datanya. Tujuan penelitian ini adalah untuk mencari nilai kontroler PID dari hasil simulasi sistem berdasarkan spesifikasi dari *hardware* yang dipakai. Kemudian nilai tersebut akan diterapkan pada *hardware* dengan cara memasukannya kedalam *sketch Arduino IDE*. Selain itu, ketika *hardware* sedang menyala dapat ditampilkan pada laptop dengan *GUI LabVIEW*. Penelitian ini menggunakan *software LabVIEW2014* yang digunakan untuk mencari nilai PID dari hasil simulasi serta digunakan sebagai GUI dari *hardware* yang sedang menyala. Hasil penelitian menunjukkan bahwa nilai konstanta PID dari hasil simulasi dengan metode kedua *Ziegler-Nichols* didapatkan nilai  $K_{px}=0.04368$ ,  $K_{ix}=6.5397$ , dan  $K_{dx}=1.000273$ , serta  $K_{pz}=8$ ,  $K_{iz}=16.55$ ,  $K_{dz}=2.2025$ . Nilai *error steady state* terkecil dan *time respon* tercepat terdapat pada setpoint -13 untuk *Main* dan 60 untuk *Tail Rotor*, sedangkan yang terbesar dan terlama terdapat pada setpoint -15 *Main* dan 80 *Tail*.

**Kata Kunci:** *Twin Rotor MIMO*, Kontroler PID, *LabVIEW2014*

### Abstract

*Twin Rotor MIMO* is one of the aerodynamic systems similar to a helicopter. The previous studies related to this had some deficiencies such as the controller value PID (*proportional-Integral-Derivative*) to the hardware was without going through the simulation first so that tuning was done without considering Specifications of the used hardware. Besides that, the data response system was not real-time because it had to be moved from serial monitor to MATLAB in order to see the data. The purpose of this research is to look for the value of a PID controller from the simulation system results based on the specifications of the used hardware. Then, the value was applied to the hardware by pairing it into the sketch *Arduino IDE*. In addition, when the hardware was on, it could be displayed on a laptop with the *LabVIEW GUI*. This research used *LabVIEW2014* software which was used to look up the PID value of the simulated results and was used as a GUI of the currently burning hardware. The results showed that the PID constant value of the simulated result with the *Ziegler-Nichols* second method obtained  $K_{px} = 0.04368$ ,  $K_{ix} = 6.5397$ , and  $K_{dx} = 1.000273$ , as well as  $K_{pz} = 8$ ,  $K_{iz} = 16.55$ ,  $K_{dz} = 2.2025$ . The smallest steady-state error value and the fastest response time were at setpoint-13 for *Main* and 60 *Tail*, while the largest and the longest were in setpoint -15 *Main* and 80 *Tail*.

**Keywords:** *Twin Rotor MIMO*, PID Controller, *LabVIEW2014*

## PENDAHULUAN

*Twin Rotor multi input multi output (TRMS)* merupakan salah satu sistem aerodinamik yang mirip dengan model helikopter (Dipesh, 2011). Untuk dapat menyeimbangkannya dibutuhkan sebuah metode pengendali yang baik untuk mempertahankan posisi seimbang. Prinsip kerja pada *twin Rotor* dapat dijelaskan sebagai model sederhana dari helikopter tetapi memiliki perbedaan yaitu bagian menara.

Pada tahun 2012 Ramalakshmi, dan Manoharan melakukan penelitian tentang “*Non-linear Modeling and PID Control of Twin Rotor MIMO System*” dengan

menggunakan *Simulink MATLAB*. Tujuan dari penelitian tersebut adalah untuk mengetahui nilai *error* yang dihasilkan jika diberikan sebuah kendali. Dari penelitian tersebut dapat diambil kesimpulan bahwa *error* yang dihasilkan dapat berkurang ketika diberikan sebuah kendali.

Pada tahun 2015 Akash A. Patel, Prakash M. Pithadiya, dan Hardik V. Kannad melakukan penelitian tentang “*Control of Twin Rotor MIMO System (TRMS) Using PID Controller*” dalam penelitian ini bertujuan untuk mengimplementasikan *Twin Rotor MIMO System* dengan *Simulink MATLAB*. Dari penelitian ini didapatkan

kesimpulan bahwa dengan kendali PID masih memiliki osilasi pada sistem.

Pada tahun 2015 dilakukan penelitian tentang “*PSO-based Optimal PID Controller for Twin Rotor MIMO System*” oleh Mehdi dan Ghaida. Dalam penelitian ini bertujuan untuk mengimplementasikan kontroler PID menggunakan *tuning* PSO untuk mengatasi *error overshoot* pada *twin Rotor*. Dari hasil penelitian tersebut dapat diambil kesimpulan yaitu dari hasil pengujian untuk mengimplementasikan kontroler PID menggunakan metode PSO dengan nilai konstanta  $K_p = 2.57$ ,  $K_i = 7.32$  dan  $K_d = 6.94$  untuk horizontal, dan  $K_p = 2.99$ ,  $K_i = 0.004$  dan  $K_d = 1.7$  untuk vertikal.

Dari beberapa penelitian yang telah diuraikan tersebut dapat disimpulkan bahwa penelitian dilakukan dengan data respon sistem tidak dapat dilihat secara *real-time* karena harus dipindahkan dahulu dari *serial monitor* ke MATLAB untuk bisa dilihat datanya. Dari permasalahan tersebut maka penulis mengambil judul penelitian “Sistem Pengendalian *Twin Rotor MIMO* dengan Kontroler PID Berbasis LabVIEW2014”.

Pada penelitian ini digunakan *software* LabVIEW 2014 untuk mensimulasikan pemodelan sistem. Dari simulasi tersebut akan diketahui respon yang kemudian digunakan untuk mencari nilai parameter dari kontroler PID (*Proporsional-Integral-Derivatid*) sebagai pengendali dalam menyeimbangkan posisi sudut TRMS terhadap permukaan bidang datar, sehingga dapat mempertahankan posisinya dengan seimbang.

Nilai parameter PID yang telah diketahui akan diterapkan pada *hardware* alat dengan cara memasukkannya kedalam *sketch* Arduino IDE. Selain itu, ketika *hardware* alat sedang dinyalakan dapat ditampilkan dengan GUI LabVIEW. *Hardware* yang digunakan adalah Arduino UNO R3 sebagai mikrokontroler, MPU6050 sebagai *sensor*, motor driver sebagai aktuator, dan motor DC sebagai plant.

## KAJIAN PUSTAKA

### *Twin Rotor MIMO*

*Twin Rotor MIMO* digerakkan oleh dua motor DC. Terdapat dua baling-baling yang saling tegak lurus dan bergabung dengan balok yang diputar pada alasnya yang dapat berputar bebas di bidang horizontal dan vertikal. Posisi balok yang bergabung dapat di pindahkan dengan cara mengubah tegangan untuk mengendalikan kecepatan pada baling-baling (Rahideh, 2007).

### Sensor MPU6050 (IMU)

Sensor MPU6050 *Inertial Measurement Unit* (IMU) adalah sensor yang digunakan untuk mengukur sudut kemiringan atau keseimbangan. IMU terdiri dari dua buah alat ukur yaitu *accelerometer* dan *gyroscope*. *Accelerometer* digunakan sebagai sensor posisi dan perpindahan sedangkan *gyroscope* digunakan sebagai sensor sudut/gerak rotasi. Pada *Twin Rotor MIMO* digunakan modul MPU6050 yang didalamnya telah terdapat sensor *accelerometer* dan *gyroscope* (Raranda, 2017). Berikut Gambar 1. Sensor MPU-6050



Gambar 1. MPU-6050  
(Sumber: Invensense.com, 2019)

### Rotary Encoder

*Rotary encoder* adalah divais elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kendali.



Gambar 2. Rotary Encoder  
(Sumber: Autonics.com, 2018)

### Kontroler PID

Kontroler PID secara luas digunakan dalam sistek kendali industry karena mengurangi jumlah nilai parameter yang akan disetel (Astrom, 1994). Kontroler *proporsional* yang ditambahkan dengan kontroler integral dan derivatif hubungannya adalah  $u(t)$  sebagai keluaran dari kontroler dan  $e(t)$  adalah sinyal error (Ogata, 1985). Aksi kontroler proporsional integral derivatif tersebut didefinisikan sebagai:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (1)$$

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (2)$$

Keterangan:

|                    |  |
|--------------------|--|
| $e(t)$             | : Sinyal error                                       |
| $u(t)$             | : Sinyal keluaran                                    |
| $K_p$              | : Konstanta proporsional                             |
| $T_i$              | : Waktu <i>integral</i>                              |
| $T_d$              | : Waktu <i>derivative</i>                            |
| $t$                | : Waktu  |
| $dt$               | : Turunan waktu                                      |
| $\int_0^t e(t) dt$ | : <i>Integral</i> sinyal <i>error</i> terhadap waktu |
| $s$                | : Penunjuk persamaan transformasi <i>laplace</i>     |
| $\frac{de(t)}{dt}$ | : Turunan sinyal <i>error</i> terhadap waktu         |

**Metode Tuning Ziegler-Nichols**

**1) Metode Kedua Ziegler-Nichols**

Metode ini dapat diaplikasikan pada plant dengan respon step pada sistem. Aturan Ziegler-Nichols digunakan untuk mencari nilai kontroler PID. Ada dua metode pada aturan tuning Ziegler-Nichols yaitu metode pertama dan metode kedua. Pada penelitian ini menggunakan metode kedua, berikut tabel aturan tuning Ziegler-Nichols metode kedua:

Tabel 1. Aturan tuning Ziegler-Nichols Metode Kedua

| Tipe Kendali | Kp           | Ti                    | Td            |
|--------------|--------------|-----------------------|---------------|
| P            | $0,5K_{cr}$  | $\infty$              | 0             |
| PI           | $0,45K_{cr}$ | $\frac{1}{1,2}P_{cr}$ | 0             |
| PID          | $0,6K_{cr}$  | $0,5P_{cr}$           | $0,125P_{cr}$ |

(Sumber: Ogata, 1985)

Keterangan:

Kcr : Konstanta kritis

Pcr : Waktu kritis

**2) Metode Tuning Ziegler-Nichols Termodifikasi**

Modifikasi metode tuning Ziegler-Nichols menggunakan Chien HronesReswick (CHR) yaitu algoritma tuning yang menekankan pada pengaturan setpoint. Selain itu, salah satu spesifikasi kualitatif pada kecepatan respon dan overshoot dapat ditampung. Dibandingkan dengan persamaan metode Ziegler-Nichols yang lama, metode ini menggunakan waktu konstan T dari plant.

Tabel 2. Metode Modifikasi Tuning Ziegler-Nichols

| Kontroler | Kp     | Ti   | Td   |
|-----------|--------|------|------|
| P         | 0.7/a  | 0    | 0    |
| PI        | 0.6/a  | T    | 0    |
| PID       | 0.95/a | 1.4T | 0.4T |

(Sumber: Manoj, 2014)

**METODE PENELITIAN**

**Pendekatan Penelitian**

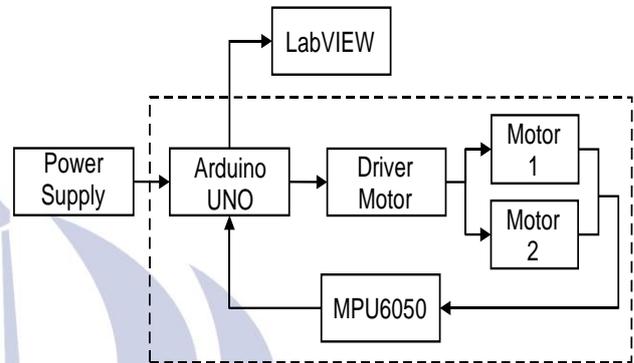
Penelitian ini menggunakan pendekatan kuantitatif. Yaitu berpijak pada apa yang disebut dengan fungsionalisme struktural, realisme, potivisme, behaviourisme dan empirisme yang intinya menekankan pada hal-hal yang bersifat konkrit, uji empiris dan fakta-fakta yang nyata. (Sarwono,2006).

Pada penelitian ini menggunakan software LabVIEW2014 dan Arduino IDE 1.8.5. Software LabVIEW2014 akan digunakan untuk mencari fungsi alih dari Twin Rotor MIMO dan mendesain kontrolernya, yaitu kontroler PID (Proporsional-Integral-Derivative) dengan

tuning Ziegler-Nichols metode kedua serta menampilkan respon sistem dari hardware Twin Rotor MIMO. Sedangkan software Arduino IDE digunakan untuk memprogram hardware alat.

**Desain Sistem**

Desain sistem dari Twin Rotor MIMO Dengan Kontroler PID Berbasis LabVIEW2014 ditunjukkan pada Gambar 4.

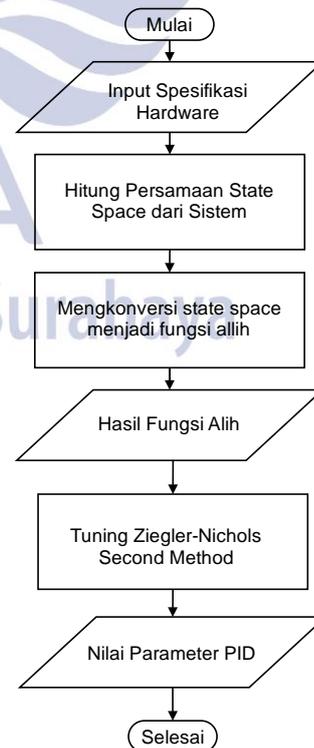


Gambar 4. Desain Sistem Twin Rotor MIMO

Pada Gambar 4. merupakan desain sistem Twin Rotor MIMO terdiri dari sensor MPU6050, Arduino, Driver Motor dan Motor DC. Data yang dihasilkan kemudian akan ditampilkan dengan GUI (Graphical User Interface) pada software LabVIEW 2014. Data tersebut akan ditampilkan secara real time oleh LabVIEW 2014, serta data tersebut dapat disimpan dalam bentuk file.

**Rancang Bangun Software LabVIEW2014**

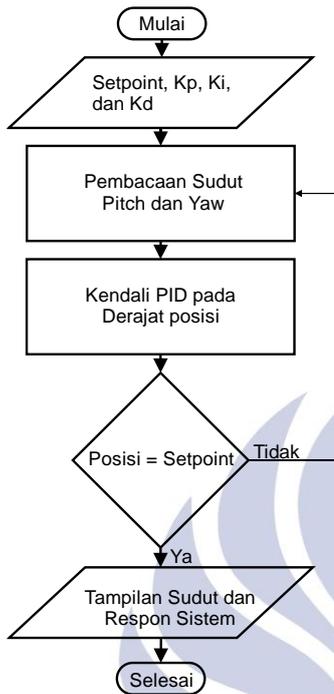
Berikut adalah desain algoritma untuk simulasi sistem Twin Rotor MIMO dengan software LabVIEW 2014:



Gambar 5. Diagram Alir Simulasi Twin Rotor MIMO

**Rancang Bangun Software Arduino IDE**

Pada Gambar 6. menunjukkan rancang bangun dari software Arduino IDE. Berikut adalah rancangan diagram alir dari perancangan software Arduino IDE :



Gambar 6. Diagram Alir Software Arduino IDE

**HASIL DAN PEMBAHASAN**

Pada penelitian ini pengujian dilakukan dengan dua tahap yaitu pengujian simulasi dari fungsi alih dari Twin Rotor MIMO dan pengujian terhadap GUI (Graphic User Interface) hasil dari pembacaan sensor MPU6050 dan Rotary Encoder yang akan ditampilkan oleh laptop atau PC.

**1. Pengujian Simulasi Sistem**

Pada pengujian simulasi sistem akan dicari pemodelan dari Twin Rotor MIMO, kemudian akan dirancang kontroler PID dan respon dari sistem tersebut akan ditampilkan.

**Pemodelan Twin Rotor MIMO**

Untuk pemodelan pada Twin Rotor MIMO harus diketahui spesifikasi dari hardware. Berikut adalah Gambar 7. spesifikasi Twin Rotor MIMO dan Tabel 2. spesifikasi hardware.



Gambar 7. Tampilan Twin Rotor MIMO

Tabel 3. Spesifikasi Hardware Twin Rotor MIMO

| No  | Nama Alat            | Simbol   | Spesifikasi | Satuan |
|-----|----------------------|----------|-------------|--------|
| 1.  | Panjang lengan Beban | $l_b$    | 0,212       | M      |
| 2.  | Panjang beban        | $l_{cb}$ | 0,195       | M      |
| 3.  | Massa beban          | $m_b$    | 0,085       | Kg     |
| 4.  | Massa lengan beban   | $m_{cb}$ | 0,02        | Kg     |
| 5.  | Panjang lengan Tail  | $l_t$    | 0.31        | M      |
| 6.  | Panjang lengan Main  | $l_m$    | 0,3         | M      |
| 7.  | Massa Main           | $m_m$    | 0,052       | Kg     |
| 8.  | Massa motor Main     | $m_{mr}$ | 0,06        | Kg     |
| 9.  | Massa Tail           | $m_t$    | 0.0537      | Kg     |
| 10. | Massa motor Tail     | $m_{tr}$ | 0.06        | Kg     |

Setelah diketahui spesifikasi hardware alat maka selanjutnya adalah mendesain pemodelan sistem berdasarkan rumus persamaan state space yaitu:

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & a_{25} & 0 \\ 0 & 0 & a_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & a_{54} & a_{55} & a_{56} \\ 0 & 0 & 0 & 0 & 0 & a_{66} \end{pmatrix} \begin{pmatrix} \alpha_v \\ \Omega_v \\ \varpi_m \\ \alpha_h \\ \Omega_v \\ \varpi_t \end{pmatrix} +$$

$$\begin{pmatrix} 0 & 0 \\ 0 & b_{22} \\ b_{31} & 0 \\ 0 & 0 \\ b_{51} & 0 \\ 0 & b_{62} \end{pmatrix} \begin{pmatrix} u_v \\ u_h \end{pmatrix} \tag{3}$$

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} x \tag{4}$$

dimana:

$$a_{21} = \frac{1}{J_v} ((A_{lt} - B_{lm}) \cos(\alpha_v) - C \cos(\alpha_v)) g \tag{5}$$

$$a_{22} = \frac{1}{J_v} (-k_v) \tag{6}$$

$$a_{23} = \frac{1}{J_v} (F_v (W_m (\varpi_m))) \tag{7}$$

$$a_{25} = \frac{1}{J_v} (-0.5 \omega_h^2 \sin(2\alpha_v) H) \tag{8}$$

$$a_{33} = -\frac{1}{T_m} \quad (9)$$

$$a_{54} = \frac{-k_c}{D\cos(u)^2 + E\sin(u)^2 + F} \quad (10)$$

$$a_{55} = \frac{-k_h}{D\cos(u)^2 + E\sin(u)^2 + F} \quad (11)$$

$$a_{56} = \frac{F_h(W_t(\omega_t))\cos(\alpha_v)}{D\cos(u)^2 + E\sin(u)^2 + F} \quad (12)$$

$$a_{66} = -\frac{1}{T_t} \quad (13)$$

$$b_{22} = \frac{k_{mc}}{J_v} \quad (14)$$

$$b_{31} = \frac{1}{T_m} \quad (15)$$

$$b_{51} = \frac{k_{cm}\cos(\alpha_v)}{D\cos(u)^2 + E\sin(u)^2 + F} \quad (16)$$

$$b_{52} = \frac{1}{T_t} \quad (17)$$

$$A_{lt} = \left(\frac{m_t}{2} + m_{tr} + m_{ts}\right) l_t \quad (18)$$

$$B_{lt} = \left(\frac{m_m}{2} + m_{mr} + m_{ms}\right) l_m \quad (20)$$

$$C_{lt} = \frac{m_b}{2} l_b + m_{cb} l_{cb} \quad (21)$$

$$D_{lt} = m_{cb} l_{cb}^2 + m_b \frac{l_b^2}{3} \quad (22)$$

$$E = \left(m_{mr} + m_{ms} + \frac{M_m}{3}\right) l_m^2 + \left(m_{mts} + m_{ts} + \frac{m_t}{3}\right) l_t^2 \quad (23)$$

$$F = m_{ms} r_{ms}^2 + \frac{m_{ts}}{2} r_{ts}^2 \quad (24)$$

$$J_v = J_{v1} + J_{v2} + J_{v3} + J_{v4} + J_{v5} + J_{v6} \quad (25)$$

$$H = (A_{lt} + B_{lt} + \frac{m_b}{2} l_b^2 + m_{cb} l_{cb}^2) \quad (26)$$

Keterangan:

- $x$  = Matriks keluaran
- $y$  = Matriks keluaran
- $a_{xy}$  = Posisi matriks input terhadap kolom dan baris
- $b_{xy}$  = Posisi matriks output terhadap kolom dan baris
- $u$  = Keluaran (derajat)
- $\alpha_v$  = Posisi vertikal (m)
- $\alpha_h$  = Posisi horizontal (m)
- $\Omega_v$  = Kecepatan vertikal (m/s)
- $\Omega_h$  = Kecepatan horizontal (m/s)
- $\omega_v$  = Kecepatan rotasi vertikal (rad/s)
- $\omega_h$  = Kecepatan rotasi horizontal (rad/s)
- $A_{lt}$  = Persamaan *main rotor*
- $B_{lt}$  = Persamaan *tail rotor*
- $C_{lt}$  = Persamaan lengan beban
- $D_{lt}$  = Persamaan beban
- $E$  = Persamaan kopleng
- $F$  = Persamaan kopleng
- $H$  = Penjumlahan antar persamaan
- $J_v$  = Momen inersia

Parameter dari *Twin Rotor MIMO* telah dicantumkan pada Tabel 3. dimasukkan ke persamaan 3 dan persamaan 4 melalui *Mathscript Node*, sehingga akan dihasilkan keluaran *state space Twin Rotor MIMO* yaitu:

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -7.151 & -0.072 & 9.857 & 0 & 0 & 0 \\ 0 & 0 & -0.752 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1.046 & -1.582 & 10.141 \\ 0 & 0 & 0 & 0 & 0 & -2.326 \end{pmatrix} x$$

$$\begin{pmatrix} \alpha_v \\ \Omega_v \\ \omega_m \\ \alpha_h \\ \Omega_h \\ \omega_t \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0.813 \\ 0.752 & 0 \\ 0 & 0 \\ 1.333 & 0 \\ 0 & 2.326 \end{pmatrix} \begin{pmatrix} u_v \\ u_h \end{pmatrix}$$

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} x$$

Setelah didapatkan hasil *state space*, kemudian dirubah menjadi persamaan *transfer function*, menggunakan persamaan:

$$\frac{U(s)}{E(s)} = C(sI - A)^{-1}B \quad (27)$$

Keterangan:

- $U(s)$  : Keluaran sistem
- $E(s)$  : *Error* sistem
- $A$  : Matriks A
- $B$  : Matriks B
- $C$  : Matriks C
- $sI$  : Matriks identitas dalam variabel  $s$
- $s$  : identitas dari transformasi *laplace*

Setelah dihitung menggunakan persamaan 27 akan didapatkan persamaan *transfer function* dengan menggunakan karakteristik orde 2:

$$\frac{U(s)}{E(s)} = \frac{b_0 s}{f_0 s^n + f_1 s^{n-1} + f_2 s^{n-2} + f_3 s^{n-3} + f_4 s^{n-4}} \quad (28)$$

Keterangan:

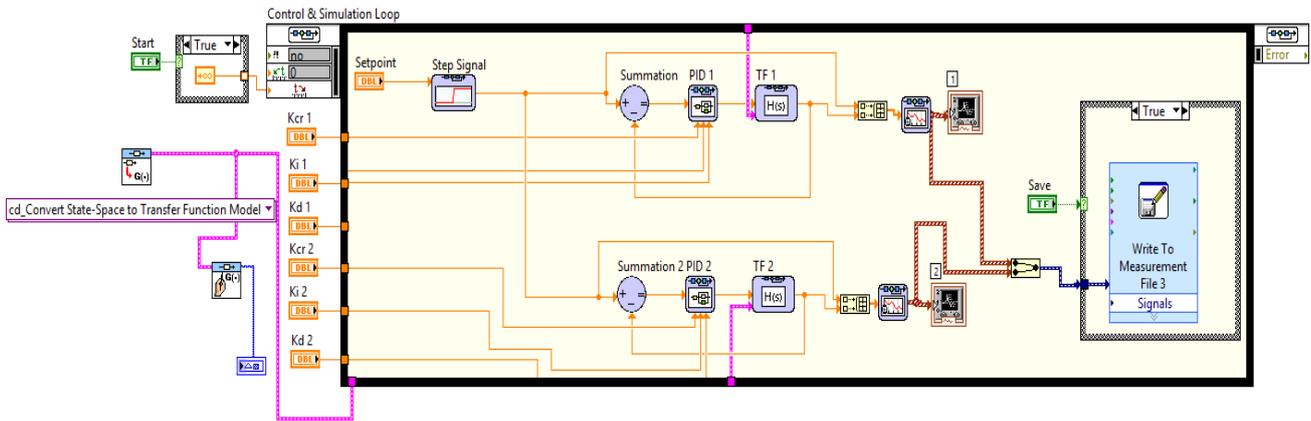
- $f$  = Nilai Input
- $b_0$  = Nilai Output
- $n$  = Nilai ke-

Sehingga didapatkan *transfer function* dengan karakteristik seperti pada persamaan 28 menggunakan simulasi di labVIEW2014 yaitu:

$$TF_{main} = \frac{7.41272}{s^3 + 0.824815s^2 + 67.20671s + 5.37827}$$

$$TF_{tail} = \frac{0.813102s^2 + 1.28678s + 0.850587}{s^4 + 1.65538s^3 + 8.31329s^2 + 11.3946s + 7.48167}$$

Diagram blok sistem *Twin Rotor MIMO* digunakan untuk melihat respon sistem dari alat. Berikut adalah blok Vi diagram simulasinya:



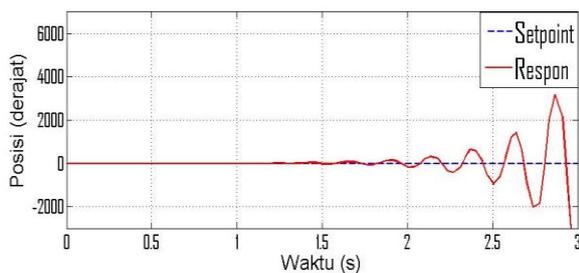
Gambar 8. Vi Diagram Loop Tertutup *Twin Rotor MIMO*

Gambar 8. adalah Vi diagram untuk simulasi dari *Twin Rotor MIMO*. Model *state space* yang telah dirubah menjadi *transfer function* akan masuk ke dalam *block control and simulation loop*. Blok atur waktu yang ada diluar *control simulation loop* yang berfungsi untuk memberikan pilihan *loop program*. Ketika blok atur waktu bernilai *true* maka *final time* pada *control and simulation loop* sama dengan "*Inf*" atau *loop* akan terus berjalan selama simulasi dijalankan. Namun ketika blok atur waktu bernilai *false* maka program hanya dijalankan selama 10 detik setelah itu program akan otomatis berhenti.

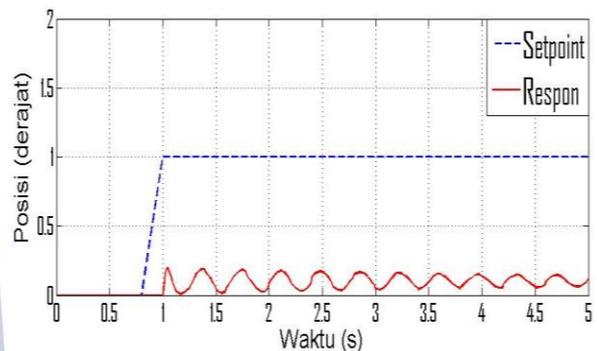
Pada diagram blok sistem tersebut *setpoint* adalah *step signal* yang dapat kita ubah-ubah nilainya. Untuk merubah nilai *setpoint* digunakan sebuah knop. Untuk *tuning* digunakan blok kontroler PID dengan nilai masukannya adalah  $K_p$ ,  $K_i$  dan  $K_d$ . Kemudian fungsi alih atau *transfer function* "TF" dari sistem *Twin Rotor MIMO* akan masuk ke blok *transfer function* yang terdapat didalam *control and simulation loop*. Kemudian untuk keluaran respon sistem akan ditampilkan menggunakan *waveform chart* dalam bentuk grafik.

Data hasil simulasi sistem selain ditampilkan dalam grafik juga akan disimpan dalam bentuk file dengan blok *write to measurement file*. File respon sistem tersebut bisa dibuka menggunakan *microsoft excel* dan memuat respon pengontrolan, *setpoint* dan waktu dari respon sistemnya.

Pertama dilakukan pengujian untuk respon sistem sebelum diberikan kontroler.



Gambar 9. Respon Sistem *Main* tanpa Kontroler

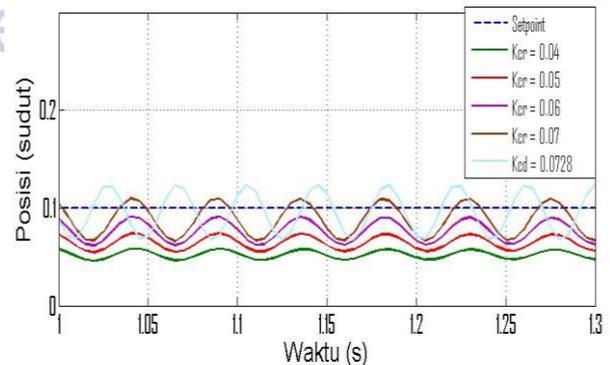


Gambar 10. Respon Sistem *Tail* tanpa Kontroler

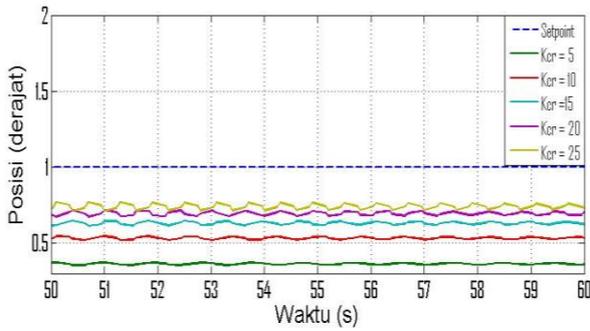
Pada Gambar 9. dan Gambar 10. terlihat bahwa respon sistem dari *Twin Rotor MIMO* tidak dapat mengikuti *setpoint* yang diberikan. Respon sangat tidak stabil dan ini membuktikan bahwa *Twin Rotor MIMO* tidak dapat mempertahankan keseimbangan posisinya sehingga perlu dirancang sebuah kontroler.

### Merancang Kontroler PID

Kontroler PID dirancang menggunakan *tuning* metode kedua *Ziegler-Nichols*. Pertama, mencari nilai  $K_{cr}$  yaitu dengan variasi nilai  $K_{cr}=0.04$ ,  $K_{cr}=0.05$ ,  $K_{cr}=0.06$ ,  $K_{cr}=0.07$ ,  $K_{cr}=0.0728$  untuk *Main*, dan nilai  $K_{cr}=5$ ,  $K_{cr}=10$ ,  $K_{cr}=15$ ,  $K_{cr}=20$ ,  $K_{cr}=25$  untuk *Tail*. Berikut hasil *tuning* yang didapat:



Gambar 11. Hasil respon sistem nilai  $K_{cr}$  *Main*

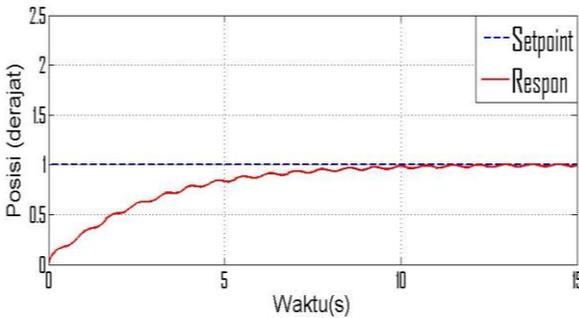


Gambar 12. Hasil respon sistem nilai Kcr Tail

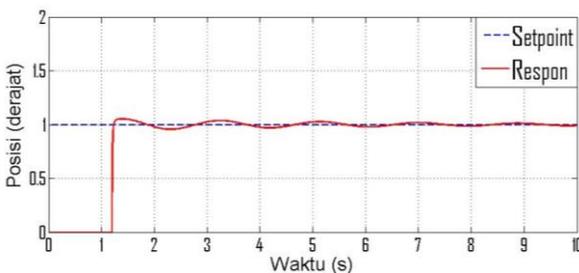
Berdasarkan Gambar 11. dan Gambar 12. Kcr yang paling sesuai adalah ketika respon mengalami osilasi yang berkelanjutan, sesuai dengan aturan metode kedua *Ziegler-Nichols*, maka dapat disimpulkan bahwa nilai Kcr yang paling sesuai adalah 0.0728 untuk *Main* dan 25 untuk *Tail*. Kemudian dihitung nilai parameter PID berdasarkan Tabel 1. Aturan *tuning Ziegler-Nichols* Metode Kedua yaitu:

|     |            |     |        |
|-----|------------|-----|--------|
| Kpx | : 0.04368  | Kpz | : 15   |
| Kix | : 1.7472   | Kiz | : 25   |
| Kdx | : 0.000273 | Kdz | : 2.25 |

Setelah disimulasikan didapatkan respon keluaran sistem seperti pada Gambar 16. dan Gambar 17.



Gambar 13. Respon Sistem *Main* dengan Kontroler PID dengan *Tuning* Metode Kedua *Ziegler-Nichols*

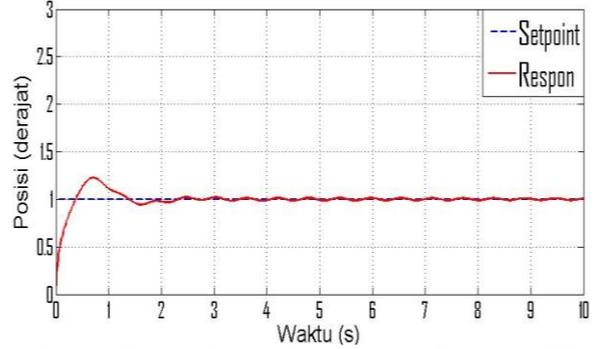


Gambar 14. Respon Sistem *Tail* dengan Kontroler PID dengan *Tuning* Metode Kedua *Ziegler-Nichols*

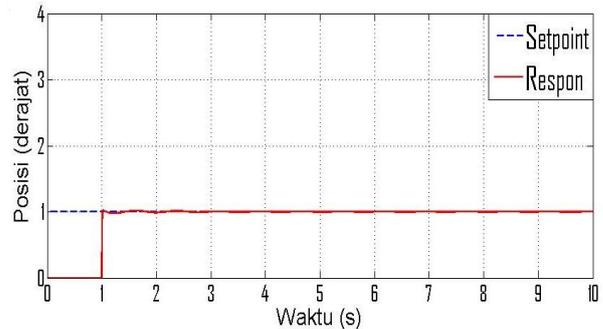
Dikarenakan respon sistem masih terbilang lambat dan tidak stabil maka parameter PID yang didapatkan dari metode *Ziegler-Nichols* diperbaiki sehingga nilai PID menjadi sebagai berikut.

|     |            |     |          |
|-----|------------|-----|----------|
| Kpx | : 0.04368  | Kpz | : 8      |
| Kix | : 6.5397   | Kiz | : 16.55  |
| Kdx | : 1.000273 | Kdz | : 2.2025 |

Sehingga didapatkan respon keluaran sistem seperti pada Gambar 13. dan Gambar 14.



Gambar 15. Respon Sistem *Main* dengan Kontroler PID

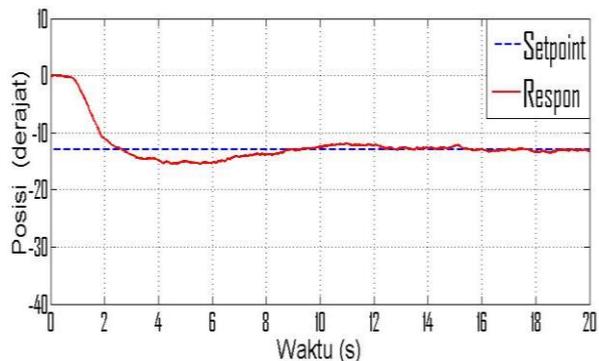


Gambar 16. Respon Sistem *Tail* dengan Kontroler PID

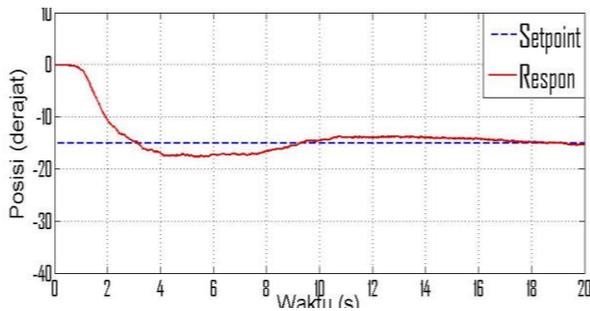
Pada Gambar 15. dan Gambar 16. terlihat sistem lebih stabil dengan *overshoot* dan osilasi yang kecil. Respon sistem sesuai dengan karakteristik dari kontroler PID (*proportional-integral-derivative*) yaitu mempercepat reaksi sebuah sistem mencapai *setpoint*-nya, menghilangkan *offset*, dan menghasilkan perubahan awal yang besar serta mengurangi *overshoot*.

## 2. Pengujian *Hardware Twin Rotor MIMO*

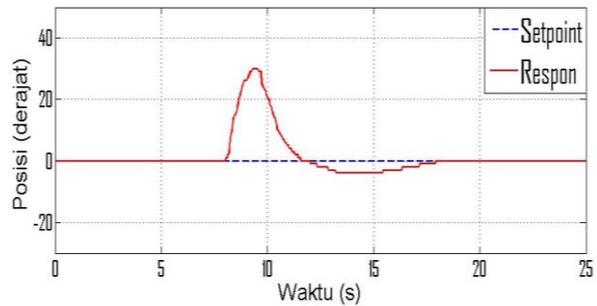
Pada pengujian *hardware Twin Rotor* dilakukan dengan 2 cara, yaitu tanpa gangguan dan dengan gangguan, dengan cara diberikan beberapa *setpoint* kemudian dilihat respon dari *hardware* dapat mencapai *setpoint* dan menjaga keseimbangan tersebut. *Setpoint* yang diberikan meliputi -12, -13, -14 derajat untuk *Main* dan 60, 70, 80 derajat untuk *Tail*. Berikut adalah grafik respon dari *hardware twin Rotor*.



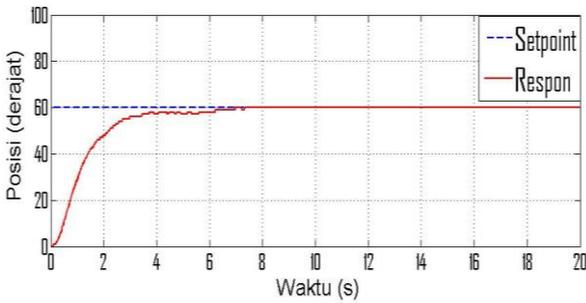
Gambar 17. Respon *Main* *Setpoint* -13 Derajat



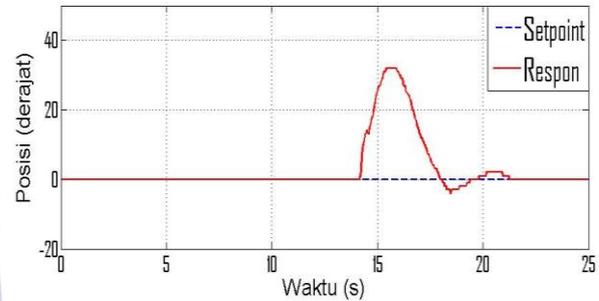
Gambar 18. Respon *Main* Setpoint -15 Derajat



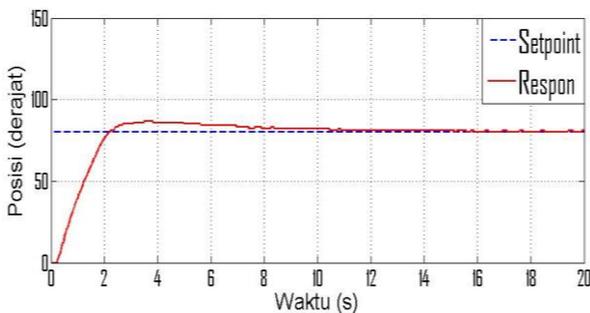
Gambar 23. Respon *Tail* dengan Gangguan 30 Derajat



Gambar 19. Respon *Tail* Setpoint 60 Derajat

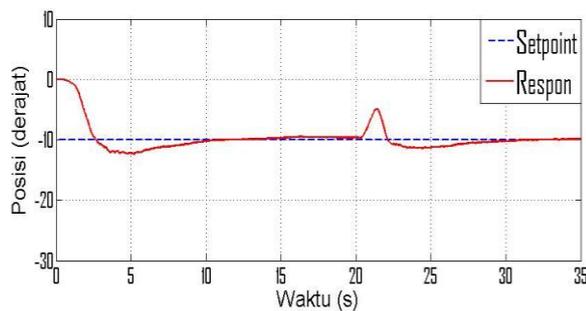


Gambar 24. Respon *Tail* dengan Gangguan 32 Derajat



Gambar 20. Respon *Tail* Setpoint 80 Derajat

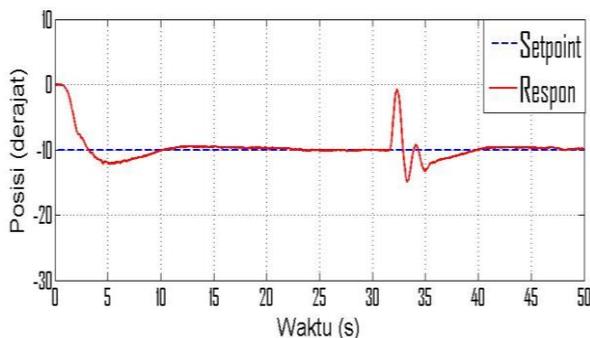
Dari gambar tersebut terlihat twin *Rotor* dapat mencapai dan mempertahankan setpoint, namun jika diberikan setpoint dan gangguan yang terlalu besar maka twin *Rotor* tidak dapat mempertahankan setpointnya. Untuk lebih lengkap untuk hasil pengujian *hardware* twin *Rotor* akan dijelaskan pada Tabel 3.



Gambar 21. Respon *Main* dengan Gangguan 5 Derajat

Tabel 4. Hasil pengujian *hardware* Twin Rotor MIMO

| Type              | Setpoint | Waktu Tunda ( <i>td</i> ) | Waktu Tunak ( <i>ts</i> ) | Overshoot Maksimum ( <i>Mp</i> ) | Error Steady State |
|-------------------|----------|---------------------------|---------------------------|----------------------------------|--------------------|
| <i>Main Rotor</i> | -13      | 1.475                     | 9.233                     | -14.9                            | 0.120              |
|                   | -14      | 1.795                     | 10.649                    | -16                              | 0.163              |
|                   | -15      | 1.88                      | 15.10                     | -17                              | 0.388              |
| <i>Tail Rotor</i> | 60       | 0.625                     | 2.317                     | 60                               | 0.176              |
|                   | 70       | 0.786                     | 2.421                     | 72                               | 0.265              |
|                   | 80       | 1.006                     | 2.527                     | 87                               | 0.305              |



Gambar 22. Respon *Main* dengan Gangguan 10 Derajat

Berdasarkan Tabel 4. diketahui bahwa nilai *error steady state* terkecil terdapat pada setpoint -13 untuk *Main* dan 60 untuk *Tail Rotor*, sedangkan *error steady state* terbesar terdapat pada setpoint -15 untuk *Main Rotor* dan 80 untuk *Tail Rotor*. *Time respon* tercepat ada pada setpoint -13 untuk *Main Rotor* dan 60 untuk *Tail Rotor*, sedangkan terlama ada pada setpoint -15 untuk *Main Rotor* dan 80 untuk *Tail Rotor*.

## PENUTUP Simpulan

Berdasarkan analisis dari hasil simulasi *Twin Rotor MIMO* dan pengujian *hardware* yang telah dilakukan maka dapat diambil kesimpulan bahwa dalam penelitian

ini didapatkan pemodelan matematika dari *Twin Rotor MIMO* sesuai spesifikasi *hardware* yang digunakan yaitu pada persamaan 8 dan 9. Model matematika tersebut digunakan untuk mendesain kontroler PID dengan metode kedua *Ziegler-Nichols* sehingga didapatkan nilai  $K_{px}=0.04368$ ,  $K_{ix}=6.5397$ , dan  $K_{dx}=1.000273$ , serta  $K_{pz}=8$ ,  $K_{iz}=16.55$ ,  $K_{dz}=2.2025$ . Pada pengujian *hardware Twin Rotor MIMO*, nilai  $K_p$ ,  $K_i$ ,  $K_d$  yang didapatkan dari simulasi akan diterapkan pada *hardware twin Rotor* dengan memasukkannya ke script Arduino IDE. Dari hasil uji *hardware* bisa mencapai dan menjaga keseimbangan pada setpoint -13, -14, dan -15 untuk *Main Rotor*, sedangkan untuk *Tail Rotor* dengan setpoint 60, 70, dan 80 seperti yang dijelaskan pada Tabel 4.

Pada hasil pengujian *hardware* yang merujuk pada Tabel 4. Hasil pengujian *hardware Twin Rotor MIMO* yaitu Nilai *Error Steady State* terkecil terdapat pada setpoint -13 untuk *Main* dan 60 untuk *Tail Rotor*, sedangkan *Error Steady State* terbesar terdapat pada setpoint -15 untuk *Main Rotor* dan 80 untuk *Tail Rotor*. *Time respon* tercepat ada pada setpoint -13 untuk *Main Rotor* dan 60 untuk *Tail Rotor*, sedangkan terlama ada pada setpoint -15 untuk *Main Rotor* dan 80 untuk *Tail Rotor*.

#### Saran

Berdasarkan penelitian yang telah dilakukan, ada beberapa saran yang dapat dilakukan untuk pengembangan sistem *Twin Rotor MIMO* selanjutnya yaitu model atau fungsi alih yang didapatkan masih kurang baik sehingga dapat dilakukan metode pendekatan yang lain agar mendapatkan fungsi alih yang lebih baik dari penelitian ini. Pada penelitian ini simulasi sistem yang digunakan adalah *tuning* metode kedua *Ziegler-Nichols*, untuk selanjutnya bisa menggunakan metode *tuning* PID lainnya misalnya metode *Cohen-Coon* atau PSO. Untuk selanjutnya dapat menggunakan kontroler adaptif atau *fuzzy* karena pada penelitian ini masih memiliki *overshoot* yang cukup tinggi dan masih terdapat ketidakstabilan.

#### DAFTAR PUSTAKA

- Akash A. Patel, Prakash M. Pithadiya, Hardik V. Kannad, Control of Twin Rotor MIMO System (TRMS) Using PID Controller, National Conference on Emerging Trends in Computer, Electrical & Electronics (ETCEE-2015) International Journal of Advance Engineering and Research Development (IJAERD)e-ISSN: 2348 - 4470, print-ISSN:2348-6406.
- A. Rahideh, M. H. Shaheed. 2006. "Mathematical Dynamic Modelling Of a Twin-Rotor Multiple Input - Multiple Output System". Department of Engineering, Queen Mary, University of London. London: UK.
- Autonics. "Rotary Encoder". [autonics.se/sensors/rotary-encoders/](http://autonics.se/sensors/rotary-encoders/) (diakses pada 26 Juni 2019, 11.44).
- Bobby, Grace, Susanto, E., & Suratman, F.Y. (2015). "Implementasi Robot Keseimbangan Beroda Dua Berbasis Mikrokontroler". Jurnal Elkomika. Vol. 3(2): hal 142-160.
- D. Xue, Y. Chen and D. P. Atherton, Linear Feedback control, Society of Industrial and Applied Mathematics, 2007, ch. 6.
- Invensense. MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTracking Devices. [www.invensense.com/products/motion-tracking/6-axis/mpu-6050/](http://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/). (diakses pada 26 Juni 2019, 11.42).
- K. Astrom and T. Hagglund, PID Controller: Theory, Design and Tuning, 2nd ed, Library of Congress Cataloging-in-Publication Data, 1994,pp.120-134.
- Marie, Mehdi J., Ghaida A. AL-Suhail, & Wisam A. Latif. 2015. "PSO-based Optimal PID Controller for Twin Rotor MIMO System. Dept. of Electrical & Computer Engineering, University of Basrah. Basrah, Iraq.
- Ogata, Katsuhiko. 1985. Teknik Kontrol Automatik jilid 1. Terjemahan Edi Laksono. Jakarta: Erlangga.
- Ogata, Katsuhiko. 2010. "Modern Control Engineering fifth edition". New York: Prentice-Hall, Inc.
- Ramalakhshmi, A.P.S, & P.S. Manoharan. 2012. "Non-linear Modeling and PID Control of Twin Rotor MIMO System". Department of Electrical & Electronics Engineering, Thiagarajar College of Engineering, Madurai.
- Raranda. 2017. "Implementasi Kontroler PID Pada Two Wheels Self Balancing Robot Berbasis Arduino UNO". Jurnal Teknik Elektro. Vol. 06: hal. 89-96.
- Sarwono, Jonathan. 2006. Metode Penelitian Kuantitatif dan Kualitatif. Yogyakarta: Graha Ilmu.
- Shah, Dipesh. 2011. "Design and Implementation of PID Controller Using Relay Feedback on TRMS (*Twin Rotor MIMO* System)". Vasad: Sardar Vallabhbhai Patel Institute of Technology.
- Wardana, I Nyoman Kusuma. 2015. Teknik Antarmuka MATLAB dan Arduino. Denpasar: Vaikutha International Publication.