

## DESAIN SISTEM PENGATURAN POSISI SUDUT AERO PENDULUM MENGUNAKAN PENGENDALI PID BERBASIS LABVIEW

**Aris Dwi Romadhon**

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

**Endryansyah**

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

### Abstrak

*Aero Pendulum* adalah sejenis pendulum yang dirakit dengan motor baling-baling di ujung pendulum. Dengan demikian, hardware berputar disekitar titik tetap dengan menggunakan gaya dorong. Permasalahan yang timbul yaitu dalam menstabilkan posisi sudut pendulum saat keadaan tidak seimbang dan saat diberi beban. Hasil penelitian menunjukkan bahwa *Aero Pendulum* dengan pengendali PID berhasil dibuat dan bekerja dengan baik. *Aero Pendulum* dapat melakukan *self stabilizing* sesuai dengan *setpoint* yang ditentukan. Selain itu, respon yang dihasilkan dapat dipantau secara *real-time* melalui GUI LabVIEW 2014. Implementasi kontroler PID pada sistem pengendali *self stabilizing* pada *Aero Pendulum* dengan nilai respon dinamis terbaik adalah disaat parameter PID dengan nilai yang didapat dari simulasi tuning *Ziegler-Nichols* yaitu  $K_p=0.06$ ,  $K_i=5$  dan  $K_d=0.05$ . Respon terbaik dihasilkan pada data dengan *setpoint*  $45^\circ$ , yang mana mempunyai waktu tunda ( $t_d$ ) sebesar 1,71 detik, waktu naik ( $t_r$ ) sebesar 5,421 detik, waktu tunak ( $t_s$ ) sebesar 7,403 detik, dan error steady state ( $E_{ss}$ ) sebesar 0.017%. Pada pengujian beban respon memiliki error steady state ( $E_{ss}$ ) sebesar 0.069%.

**Kata Kunci:** *Aero Pendulum*, Motor Brushless DC, PID, LabVIEW, *Ziegler-Nichols*.

### Abstract

*Aero Pendulum* is a type of pendulum that is assembled with a propeller motor at the end of the pendulum. Thus, the hardware rotates around a fixed point using thrust. The problem that arises is in stabilizing the position of the pendulum angle when the condition is not balanced and when given a load. The results showed that *Aero Pendulum* with PID controller was successfully created and worked well. *Aero Pendulum* can do *self-stabilizing* according to the specified *setpoint*. Besides, the response generated can be monitored in *real-time* through the 2014 LabVIEW GUI. Implementation of the PID controller on the *self-stabilizing* control system on *Aero Pendulum* with the best dynamic response value is the PID parameter with the value obtained from the *Ziegler-Nichols* tuning simulation,  $K_p=0.06$ ,  $K_i=5$  and  $K_d=0.05$ . The best response is generated on data with a  $45^\circ$  *setpoint*, which has a delay time ( $t_d$ ) of 1,71 seconds, rise time ( $t_r$ ) of 5,421 seconds, settling time ( $t_s$ ) of 7,403 seconds, and error steady-state ( $E_{ss}$ ) of 0,017%. In response, load testing has a error steady-state ( $E_{ss}$ ) of 0,069%.

**Keywords:** *Aero Pendulum*, Brushless DC Motor, PID, LabVIEW, *Ziegler-Nichols*.

### PENDAHULUAN

*Unmanned Aerial Vehicle* (UAV) atau kendaraan udara tanpa awak adalah salah satu teknologi yang sedang mengalami perkembangan yang pesat, Contoh aplikasi yang dapat diimplementasikan pada UAV adalah untuk kebutuhan survei, patroli, deteksi tambang mineral, riset, fotografi, dan keperluan lainnya. Permasalahan dalam *singlerotor/multirotor* yaitu dalam stabilitas terbang, pada saat beban tidak seimbang maka akan mempengaruhi sikap terbangnya, yang bisa saja menimbulkan jatuhnya beban saat di udara. Maka dari itu

dibutuhkannya tenaga dari motor untuk mengatasi gaya gravitasi (SA,Inkyu, 2011).

*Aero Pendulum* adalah sejenis pendulum yang dirakit dengan motor baling-baling di ujung pendulum. Dengan demikian, hardware berputar disekitar titik tetap dengan menggunakan gaya dorong. Gaya dorong dihasilkan oleh baling-baling efek aerodinamik. Umumnya, digunakan untuk mengatur jam bandul, dalam instrumen ilmiah seperti akselerometer, seismometer dan secara luas dalam teknologi *aerospace*, robotika, wahana dan lain-lain (R.Ghasemi, 2014).

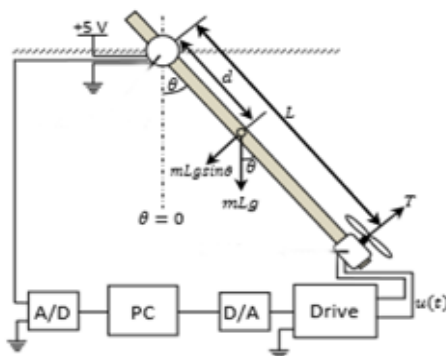
Pada tahun 2011, Amin Mohammadbagheri dan Mahdi Yaghoobi membuat penelitian yang berjudul “A New Approach to Control A Driven Pendulum with PID Method”. Makalah ini mengusulkan metode PID untuk mengontrol pendulum yang digerakkan. Driven pendulum adalah pendulum yang ditangguhkan, yang memiliki baling-baling bermotor di ujung tongkat. Metode yang dibahas dapat meningkatkan kinerja domain waktu dari sistem linierisasi. Hasil simulasi membuktikan bahwa metode ini meningkatkan stabilitas serta kemudahan penyetelan. Kontroler yang disajikan dirancang dan dievaluasi menggunakan MATLAB / Simulink. Dapat disimpulkan bahwa data respon dari penelitian sebelumnya tidak dapat dilihat secara *real-time* karena harus dipindahkan dulu data dari serial monitor ke MATLAB untuk bisa dilihat datanya.

Tujuan penelitian ini adalah membuat dan merancang pengendalian *self stabilizing* pada *Aero Pendulum* menggunakan *PID controller* untuk menstabilkan posisi sudut *Aero Pendulum*. Penelitian ini menggunakan *hardware* MPU-6050 sebagai sensor, arduino sebagai mikrokontroler, motor *brushless dc* sebagai actuator dan juga *software* LabVIEW yang digunakan untuk mengamati respon sistem secara *real-time*. Berdasarkan latar belakang masalah dan hasil studi literatur dari penelitian yang sebelumnya maka akan dilakukan penelitian dengan judul “Desain Sistem Pengaturan Posisi Sudut Aero Pendulum Menggunakan Pengendali PID Berbasis LabVIEW”.

## KAJIAN PUSTAKA

### Aero Pendulum

Pada sistem pendulum yang diberikan pada Gambar1 mempunyai motor dengan baling-baling pada ujung tongkat yang ditangguhkan. Di sini,  $u(t)$  adalah input kontrol dan sudut yang berada di antara lengan bandul dan sumbu vertikal adalah variabel kontrol. Pendulum digerakkan oleh motor. Setelah tegangan diterapkan, baling-baling berputar dan menghasilkan torsi  $T$  untuk menarik pendulum (Kizmaz dkk, 2010).



**Gambar 1.** Diagram Sistematis dari Sistem Pendulum  
(Sumber : Kizmaz dkk, 2010)

*Aero Pendulum* bertujuan untuk memerintahkan pendulum ke sudut yang ditentukan. Oleh karena itu, variabel yang dikontrol untuk sistem ini, adalah sudut pendulum yang menetap dan variabel yang dimanipulasi adalah tegangan yang diberikan kepada baling-baling bermotor (Raju dkk, 2012).

Menurut hukum Newton dan momentum sudut, persamaan gerak pendulum yang digerakkan dapat diturunkan menjadi.

$$J \cdot \ddot{\theta} + c \cdot \dot{\theta} + m \cdot g \cdot d \cdot \theta = T \quad (1)$$

Hasil persamaan tersebut, kemudian di *Laplace* kan menggunakan domain  $s$  menjadi :

$$J \cdot s^2 \theta(s) + c \cdot s \theta(s) + m \cdot g \cdot d \theta(s) = T(s) \quad (2)$$

Sehingga didapatkan fungsi alih

$$\frac{\theta(s)}{T(s)} = \frac{1}{J \cdot s^2 + c \cdot s + m \cdot g \cdot d} \quad (3)$$

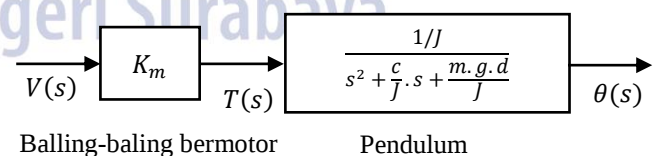
Persamaan 3 dapat disederhanakan hingga menjadi

$$\frac{\theta(s)}{T(s)} = \frac{1/J}{s^2 + \frac{c}{J} \cdot s + \frac{m \cdot g \cdot d}{J}} \quad (4)$$

Gaya dorong yang dihasilkan dalam persamaan di atas bukan merupakan variabel yang dimanipulasi untuk sistem kontrol karena pendulum disesuaikan oleh tegangan yang diberikan. Jadi persamaan rasional antara Tegangan  $V$  yang diterapkan pada motor *brushless DC* dan daya dorong  $T$ , dapat ditulis sebagai berikut:

$$T(s) = K_m \cdot V(s) \quad (5)$$

Akhirnya dari persamaan (4) dan (5). Diagram blok dari sistem *aero pendulum* dapat ditunjukkan pada Gambar 2.



**Gambar 2** Open-Loop Diagram Blok Sistem dari *Aero Pendulum*

(Sumber : Kizmaz dkk, 2010)

Sekarang, dari Gambar 2. *transfer function aero pendulum* dapat dituliskan

$$\frac{\theta(s)}{V(s)} = \frac{K_m/J}{s^2 + \frac{c}{J} \cdot s + \frac{m \cdot g \cdot d}{J}} \quad (6)$$

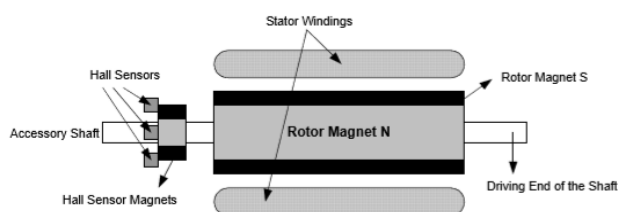
**Keterangan**

- $s$  = hasil dari *Laplace* dalam domain waktu  
 $T$  = gaya dorong yang disediakan oleh motor *brushless* DC (Nm/kg)  
 $T(s)$  = gaya dorong dalam domain waktu (*Laplace*)  
 $V$  = tegangan yang diterapkan pada motor *brushless* DC (Volt)  
 $V(s)$  = tegangan dalam domain waktu (*Laplace*)  
 $\theta$  = sudut (derajat)  
 $\theta(s)$  = sudut dalam domain waktu (*Laplace*)  
 $\dot{\theta}$  = kecepatan sudut (rad/s)  
 $\ddot{\theta}$  = percepatan sudut (rad/s<sup>2</sup>)  
 $K_m$  = parameter motor *propeller*  
 $m$  = berat bandul (kg)  
 $d$  = jarak dari titik gantung ke pusat massa (m)  
 $J$  = momen inersia (kg.m<sup>2</sup>)  
 $g$  = percepatan gravitasi (m/s<sup>2</sup>)  
 $c$  = koefisien redaman kental (kg.m<sup>2</sup>/s)

**Motor Brushless DC**

Motor *brushless* DC adalah motor sinkron dengan magnet permanen dan sebuah rangkaian elektronika (*Inverter*) untuk mengendalikan arus *armature* yang ada. Motor *brushless* DC kadangkala dimodelkan sebagai suatu motor DC 2 arah karena *armature*-nya ada di stator dan magnet berada pada rotor dan karakteristik operasinya mirip dengan motor DC. Daripada menggunakan suatu komutator mekanik seperti yang ada pada motor DC konvensional, motor *brushless* DC menggunakan komutasi elektronik yang menjadikannya motor yang minim perawatan. (Azzumar, 2012 : 5)

Perbedaan utama antara motor DC magnet permanen (DC-MP) dengan motor *brushless* DC adalah terletak pada pembangkitan medan magnet untuk menghasilkan gaya gerak. Jika pada motor DC-MP medan magnet yang dikontrol berada di rotor dan medan magnet tetap berada di stator. Sebaliknya, motor *brushless* menggunakan pembangkitan medan magnet stator untuk mengontrol gerakanya sedang medan magnet tetap berada di rotor. Struktur dari Motor *Brushless* DC ditunjukkan pada Gambar 3.



**Gambar 3.** Struktur Motor *Brushless* DC  
(Sumber : Stefán Baldursson, 2005)

**IMU (Inertial Measurement Unit)**

Alat ukur inersia atau *Inertial Measurement Unit* (IMU) adalah alat yang memanfaatkan sistem pengukuran seperti gyroskop dan akselerometer untuk memperkirakan posisi relatif, kecepatan, dan akselerasi dari gerakan motor (Raranda, 2017). Dan juga merupakan bagian dari navigasi sistem yang dikenal sebagai *Inertial Navigation System* atau INS. Pada Aero Pendulum yang dibuat, penggunaan sensor *accelerometer* dan *gyroscope* akan digantikan dengan sensor MPU-6050 module.

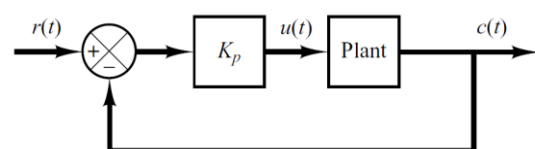
**Kontroler PID**

Kontroler *Proportional-Integral-Derivative* (PID) merupakan kontroler yang berfungsi untuk sistem pengendali yang bekerja dengan sistem umpan balik untuk mendeteksi nilai *error* sehingga sistem yang menggunakan PID dengan nilai masing-masing P-I-D yang tepat akan memperoleh hasil kendali yang stabil meskipun dalam kondisi dinamis (Gunnel, 2017).

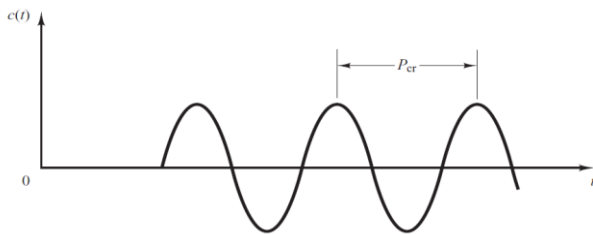
**Tuning Ziegler-Nichols Metode Kedua**

Proses pemilihan parameter kendali untuk memenuhi spesifikasi kinerja yang diberikan dikenal sebagai *tuning* kontroler salah satunya adalah metode Ziegler-Nichols. Ziegler dan Nichols menyarankan peraturan untuk menyetel kendali PID (yang berarti menetapkan nilai parameter  $K_p$ ,  $T_i$  dan  $T_d$ ) berdasarkan respon langkah eksperimental atau berdasarkan nilai  $K_p$  yang menghasilkan stabilitas yang baik bila hanya tindakan kontroler proporsional yang digunakan.

Pada metode kedua, pertama-tama kita mengatur nilai parameter  $T_i = \infty$  dan  $T_d = 0$ . Menggunakan tindakan kontroler proporsional saja, tingkatkan nilai  $K_p$  dari 0 ke nilai kritis  $K_{cr}$  dimana output pertama menunjukkan osilasi berkelanjutan atau sering disebut *sustained oscillations* (Ogata, 1985). Jika respon keluaran tidak menunjukkan osilasi berkelanjutan untuk nilai  $K_p$  apa pun yang diperlukan, metode ini tidak berlaku. Dengan demikian, gain kritis  $K_{cr}$  dan periode yang sesuai  $P_{cr}$  ditentukan secara eksperimental (Ogata, 1985).



**Gambar 4.** Sistem Loop Tertutup  
(Sumber : Ogata, 1985)



**Gambar 5.** Osilasi Berkelanjutan dengan Periode  
(Sumber : Ogata, 1985)

Ziegler dan Nichols menyarankan agar kita menetapkan nilai parameter  $K_p$ ,  $T_i$ , dan  $T_d$  sesuai dengan rumus yang ditunjukkan pada Tabel 1.

**Tabel 1.** Aturan tuning Ziegler-Nichols Metode 2

| Tipe Kendali | $K_p$         | $T_i$         | $T_d$          |
|--------------|---------------|---------------|----------------|
| <b>P</b>     | 0,5 $K_{cr}$  | $\infty$      | 0              |
| <b>PI</b>    | 0,45 $K_{cr}$ | 0,83 $P_{cr}$ | 0              |
| <b>PID</b>   | 0,6 $K_{cr}$  | 0,5 $P_{cr}$  | 0,125 $P_{cr}$ |

(Sumber: Ogata, 1985)

#### Keterangan

- P : Proporsional  
 PI : Proporsional-Integral  
 PID : Proporsional-Integral-Derivatif  
 $K_p$  : Parameter untuk Proporsional  
 $T_i$  : Parameter untuk Integral  
 $T_d$  : Parameter untuk Derivatif  
 $K_{cr}$  : Konstanta Kritis  
 $P_{cr}$  : Waktu Kritis

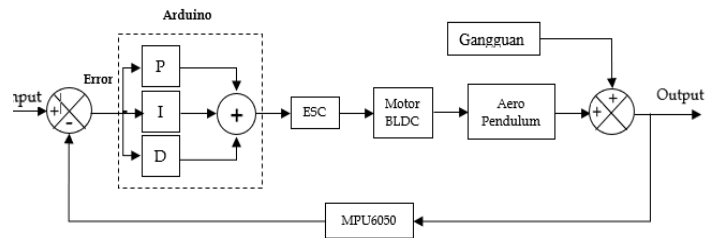
## METODE PENELITIAN

### Pendekatan Penelitian

Pada penelitian ini, pendekatan penelitian yang digunakan adalah penelitian kuantitatif. (Kasiram, 2008) dalam bukunya Metodologi Penelitian Kualitatif dan Kuantitatif, mendefinisikan penelitian kuantitatif adalah suatu proses menemukan pengetahuan yang menggunakan data berupa angka sebagai alat menganalisis keterangan mengenai apa yang ingin diketahui.

### Desain Sistem

Desain diagram blok sistem pengendalian posisi sudut pada *aero pendulum* menggunakan sensor mpu6050 untuk mengatur posisi sudut dengan kendali PID ditunjukkan pada Gambar 6.



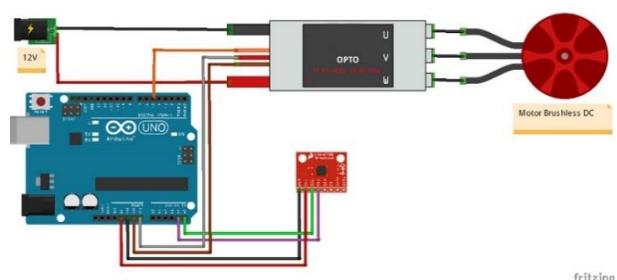
**Gambar 6.** Diagram Blok Sistem

Gambar 6. adalah diagram blok sistem kendali dari perangkat *hardware Aero Pendulum*. Nilai masukan (*Setpoint*) sistem berupa posisi sudut. Kontroler *Proporsional-Integral-Derivative* menggunakan *tuning Ziegler Nichols*, akan diproses pada Arduino yang nantinya akan membantu proses *self stabilizing control*. Hasil dari keluaran kontroler berupa tegangan yang akan masuk ke *Electrical Speed Control (ESC) 30A* yang dikonversi ke nilai PWM dan *Electrical Speed Control (ESC) 30A* akan mengatur arah putar dan kecepatan motor *brushless DC*. Sensor MPU6050 akan bekerja untuk mengukur nilai dari derajat kemiringan plant setelah adanya gerakan akibat perputaran motor dan propeller dan kemudian menjadikan nilai pembacaan sudut sebagai nilai umpan balik.

Pada penelitian ini menggunakan *software LabVIEW 2014* dan *Arduino IDE 1.8.5*. *Software LabVIEW* akan digunakan untuk mencari fungsi alih dari *Aero Pendulum*, mendesain kontrolernya, yaitu kontroler PID (*Proporsional-Integral Derivative*) serta menampilkan respon sistem dari *hardware Aero Pendulum* yang sedang menerapkan *self stabilizing control*. Sedangkan *software Arduino IDE* digunakan untuk memprogram *hardware Aero Pendulum*.

### Rancang bangun Hardware

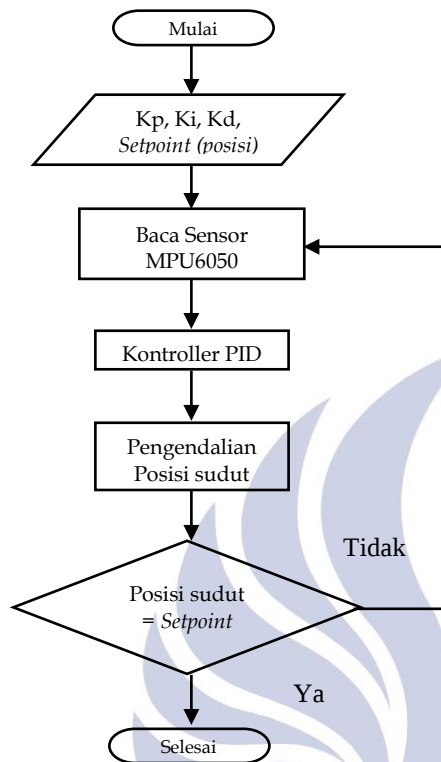
Perancangan perangkat keras (*Hardware*) merupakan proses desain bagian-bagian penyusun dari sistem Aero Pendulum. Bagian tersebut meliputi komponen yang digunakan dalam hal ini adalah motor *brushless DC*, MPU6050, arduino dan ESC. Perancangan perangkat keras sistem dapat dilihat pada Gambar 7.



**Gambar 7.** Skema Rangkaian Hardware Aero Pendulum

### Rancang bangun Software

Berikut adalah rancangan diagram alir dari perancangan *software* Arduino IDE : yang ditunjukkan pada Gambar 8.



Gambar 8. Flowchart Software

Program dimulai dengan memberi nilai  $K_p$ ,  $K_i$ ,  $K_d$  dan *setpoint*. Data tersebut akan dikirim ke arduino untuk diolah, kemudian sensor MPU6050 akan menghitung posisi sudut pada lengan aero pendulum. Setelah itu dari hasil pembacaan, akan dilakukan pengontrolan PID untuk mengatasi posisi sudut sehingga memiliki sudut yang sesuai dengan *Setpoint* yang sudah ditentukan. Data posisi sudut akan dikirim ke laptop dengan mengintegrasikan Arduino dan *software* LabVIEW, agar bisa dilihat hasil responnya. Jika lengan aero pendulum sudah memiliki sudut yang sesuai dengan *setpoint* maka *self stabilizing control* telah berhasil namun jika tidak maka akan terus distabilkan dengan menghitung *error* sudut yang terjadi.

Selain digunakan sebagai simulasi sistem dari *aero pendulum*, *software* LabVIEW 2014 digunakan untuk menampilkan data pada saat *plant* sedang melakukan *stabilizer control*. Ketika program pada *software* LabVIEW berjalan maka harus diatur dahulu COM Port dan *baudrate* yang sesuai dengan *serial monitor* dan *sketch* dari *software* Arduino IDE. Kemudian LabVIEW akan mengambil data yang terbaca oleh *hardware* Arduino UNO R3. Data yang masuk tersebut akan ditampilkan dalam bentuk angka, dan juga grafik.

### HASIL DAN PEMBAHASAN

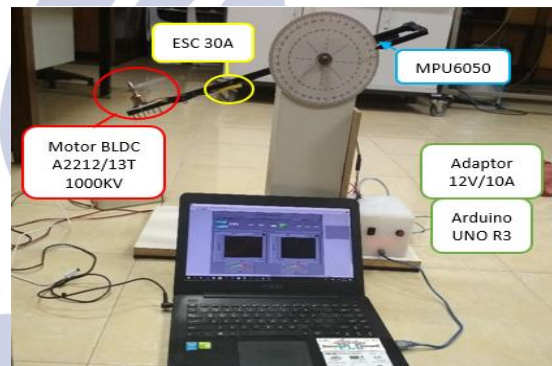
Pada bagian ini akan dijelaskan serta ditampilkan *hardware system* dan hasil pengujian diagram Vi pada *software* LabVIEW 2014. Pengujian ini dilakukan dengan dua tahap yaitu adalah pengujian simulasi dari fungsi alih *Aero Pendulum* dan pengujian terhadap GUI (*Graphic User Interface*) hasil dari pembacaan sensor MPU6050 pada *hardware Aero Pendulum*.

#### 1. Pengujian Simulasi Sistem

Pada pengujian simulasi sistem akan dicari pemodelan sistem dari *hardware aero pendulum*. Kemudian akan dirancang kontroler PID agar dapat mengeluarkan respon dalam bentuk grafik.

#### Pemodelan Sistem Aero Pendulum

Sebelum mencari pemodelan sistemnya, yang harus dilakukan pertama adalah mencari spesifikasi *hardware* untuk dimasukkan ke fungsi alih yang sudah diketahui. Berikut adalah Gambar 9. dan Tabel 2. spesifikasi *hardware Aero Pendulum*.



Gambar 9. Aero Pendulum

Tabel 2. Spesifikasi Aero Pendulum

| No | Nama Alat            | Simbol | Spesifikasi | Satuan                   |
|----|----------------------|--------|-------------|--------------------------|
| 1. | Panjang lengan       | $L$    | 0,4         | m                        |
| 2. | Panjang pusat massa  | $d$    | 0,2         | m                        |
| 3. | Massa                | $m$    | 0,32        | Kg                       |
| 4. | Momen inersia        | $J$    | 0,0512      | $\text{Kg.m}^2$          |
| 5. | Percepatan gravitasi | $g$    | 9,8         | $\text{m/s}^2$           |
| 6  | Koefisien redaman    | $c$    | 0,0001      | $\text{kg.m}^2/\text{s}$ |

Setelah diketahui spesifikasi dari *hardware Aero Pendulum*, maka langkah selanjutnya adalah memasukkan nilai yang sudah didapatkan dari Tabel 2. ke dalam persamaan 9 dan didapatkan hasil fungsi alih pemodelan sistem sebagai berikut.

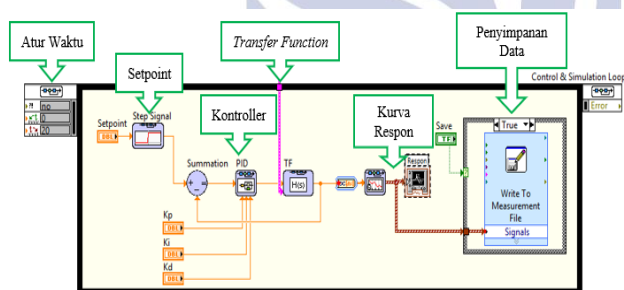
$$G(s) = \frac{13.6719}{s^2 + 0.00195s + 12.25}$$

Keterangan:

$G(s)$  = Fungsi alih pemodelan sistem

### Blok Diagram LabVIEW untuk Simulasi Sistem

Program pada LabVIEW dikenal dengan sebutan Vi-Diagram atau *Virtual Instruments Diagram* karena bahasa pemrograman pada LabVIEW berbasis grafis atau blok diagram yang dapat meniru sebuah *instrument*. Vi-diagram pada penelitian ini digunakan untuk mensimulasikan respon sistem dari *Aero Pendulum* dan mencari nilai kontroler PID (*Proportional-Integral-Derivative*) dengan tuning *Ziegler-Nichols* metode kedua yang digunakan untuk mengontrol posisi agar sesuai *setpoint* yang diinginkan.

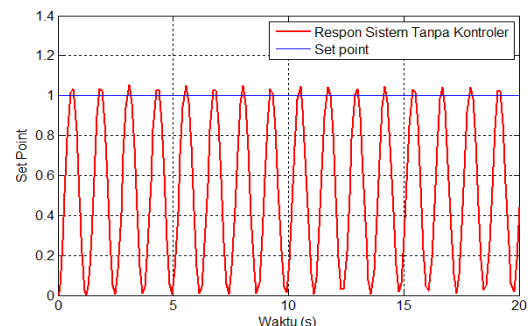


Gambar 10. Diagram Vi Loop Tertutup dengan Kontrol PID

Blok atur waktu yang ada diluar *control simulation loop* berfungsi untuk memberikan pilihan loop program. Blok atur waktu akan diprogram agar berjalan selama 20 detik setelah itu 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 *Aero Pendulum* akan masuk ke blok *transfer function* yang terdapat di dalam *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. Dari fungsi alih yang didapatkan maka diketahui respon sistem dari *Aero Pendulum* yang ditunjukkan pada Gambar 11.

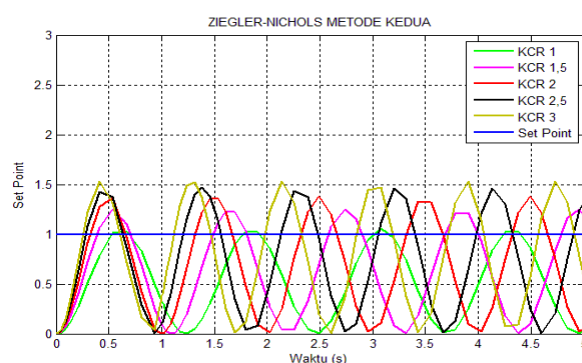


Gambar 11. Simulasi Respon Tanpa Kontroler

Pada Gambar 11. terlihat bahwa respon sistem dari *Aero Pendulum* tidak bisa mengikuti *setpoint* yang diberikan. Respon sistem tidak stabil (selalu berosilasi), ini membuktikan bahwa *Aero Pendulum* tidak bisa mempertahankan posisinya. Dari respon sistem yang tidak bisa mempertahankan posisinya sesuai *setpoint* tersebut maka perlu dirancang suatu kontroler yang dapat membuat sistem mempertahankan posisinya sesuai *setpoint*.

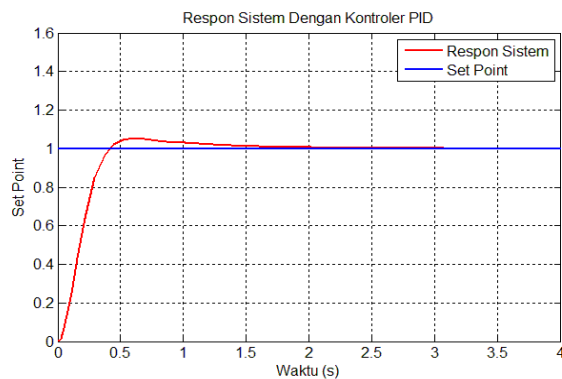
### Merancang Kontroler PID

Pada penelitian ini digunakan kontroler PID (*proportional-integral-derivative*). Kemudian untuk *tuning* kontrol PID digunakan metode *tuning second method Ziegler-Nichols*. Untuk metode *tuning Ziegler-Nichols* pertama-tama memberikan nilai  $K_{cr}$  sampai output pertama menunjukkan osilasi berkelanjutan. Berikut adalah hasil *tuning* nilai  $K_{cr}=1$ ,  $K_{cr}=1.5$ ,  $K_{cr}=2$ ,  $K_{cr}=2.5$  dan  $K_{cr}=3$  dengan *setpoint* =1 yang ditunjukkan pada Gambar 12.



Gambar 12. Hasil Respon Sistem

Setelah didapatkan nilai  $K_{cr}$  maka langkah selanjutnya adalah mencari nilai  $K_p$  (konstanta proportional),  $T_i$  (Integral Time) dan  $T_d$  (Derivative Time) dengan rumus second method Ziegler-Nichols. Dari rumus tersebut maka akan dapat dihitung nilai kontroler PID yaitu  $K_p$ ,  $K_i$  dan  $K_d$  yaitu  $K_p = 1,5$ ,  $K_i = 3.158$  dan  $K_d = 0.179$ . Kemudian dimasukkan pada diagram Vi simulasi dan didapatkan respon sistem yang ditunjukkan pada Gambar 13.

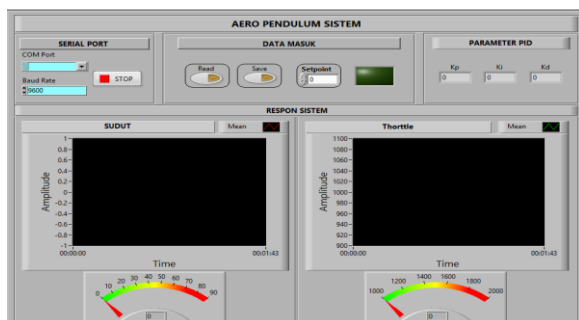


**Gambar 13.** Respon Sistem dengan Kontroler PID

Pada Gambar 13, ditampilkan respon sistem dengan kontroler PID (*proportional-integral-derivative*) dengan *tuning second method Ziegler-Nichole*. Terlihat sistem lebih stabil dengan osilasi yang kecil dan sesuai setpoint. Respon sistem sesuai dengan karakteristik dari kontroler PID (*proportional-integral-derivative*) yaitu sistem memiliki respon atau tanggapan yang cepat dan menghilangkan respon *steady state error*.

## 2. Pengujian Hardware Aero Pendulum

Setelah diketahui nilai kontroler PID (*proporsional-integral-derivative*) dari hasil simulasi, kemudian nilai tersebut dimasukkan kedalam script pemrograman Arduino. Respon real-time Aero Pendulum dengan kendali PID akan disimpan dalam file dan ditampilkan pada GUI Labview 2014. Berikut adalah tampilan GUI untuk menampilkan data dari Aero Pendulum yang ditunjukkan pada Gambar 14.

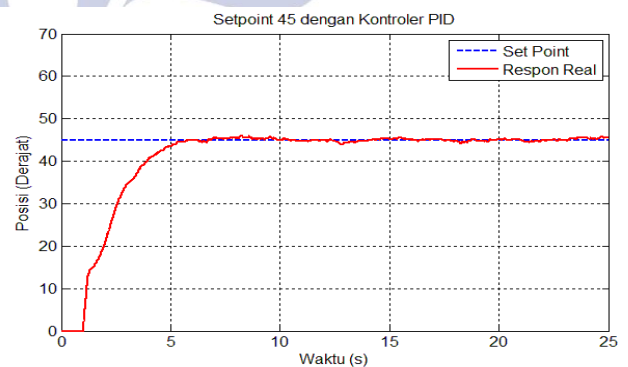


**Gambar 14.** Respon Sistem pada Setpoint 60°

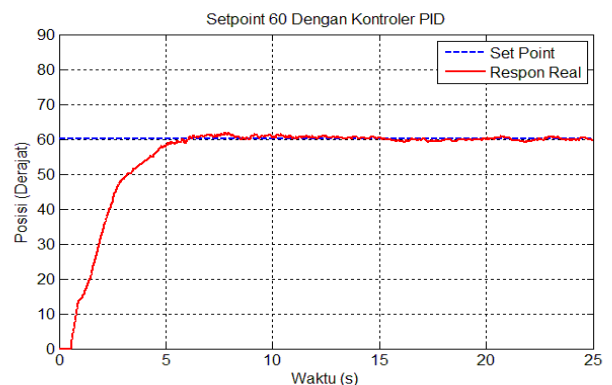
Pada Gambar 14, terlihat ada 4 sub tampilan untuk GUI Aero Pendulum sistem. Pertama adalah "SERIAL PORT", dalam kotak menu ini pengguna dapat mengatur Port COM Arduino yang akan digunakan, kemudian mengatur Baudrate serta juga dapat menghentikan program yang sedang berjalan dengan tombol STOP. Kedua adalah "DATA MASUK", dalam kotak menu ini pengguna bisa mengatur pembacaan dengan tombol Read selain itu juga dapat diatur penyimpanan data yang masuk dengan tombol Save. Selain itu juga dapat mengatur setpoint yang diinginkan. Ketiga adalah "PARAMETER PID", pada kotak menu ini ditampilkan nilai dari parameter PID yang sebelumnya didapatkan. Dan yang Keempat adalah "RESPON SISTEM", dalam kotak menu ini ditampilkan hasil respon posisi sudut dan thorttle dalam bentuk grafik dan Vi-diagram Gauge.

Pengujian respon *hardware Aero Pendulum* dilakukan dengan beberapa cara yaitu: pertama, pengujian dengan setpoint tetap. Kedua, pengujian dengan setpoint berubah. Dan ketiga, pengujian dengan beban. Pengujian ini dilakukan dengan kontroler PID dengan nilai parameter  $K_p = 0.06$ ,  $K_i = 5$  dan  $K_d = 0.05$ . Berikut adalah beberapa pengujian untuk *hardware Aero Pendulum*:

### Pengujian dengan Setpoint Tetap



**Gambar 15.** Respon Sistem pada Setpoint 45°



**Gambar 16.** Respon Sistem pada Setpoint 60°

Dari 2 percobaan terhadap pengujian dengan setpoint tetap dapat diasumsikan bahwa sistem bekerja dengan cukup baik. Namun responnya masih lama untuk mencapai setpoint. Kemudian digunakanlah rumus analitik untuk perhitungan parameter respon yang akan dijelaskan pada Tabel 3, sebagai berikut :

1. Waktu tunda ( $t_d$ )  

$$t_d = \tau \ln 2 \quad (7)$$

2. Waktu naik ( $t_r$ ) (10%-90%)  

$$t_r = \tau \ln 9 \quad (8)$$

3. Waktu Tunak ( $t_s$ ) (5%)  

$$t_s = 3\tau \quad (9)$$

4. Persentasi *Error Steady State* ( $E_{ss}$ )

$$E_{ss} = \left| \frac{Y_{ss} - X_{ss}}{X_{ss}} \right| \times 100\% \quad (10)$$

Keterangan

$\tau$  : Waktu saat respon mencapai 63.2%

$X_{ss}$  : Setpoint

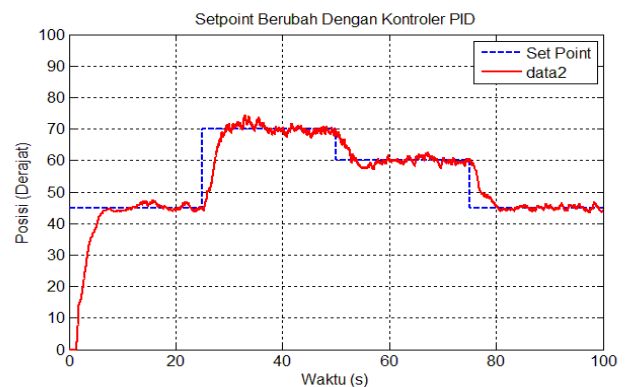
$y_{ss}$  : Rata-rata respon saat steady-state

**Tabel 3.** Hasil Pengujian Hardware Aero Pendulum

| Setpoint                     | 45°   | 60°   |         |
|------------------------------|-------|-------|---------|
| Overshoot Maksimum ( $M_p$ ) | 46,1  | 61,96 | Derajat |
| Waktu Puncak ( $t_p$ )       | 8,264 | 7,81  | Detik   |
| Waktu Tunda ( $t_d$ )        | 1,71  | 1,574 | Detik   |
| Waktu Naik ( $t_r$ )         | 5,421 | 4,991 | Detik   |
| Waktu Tunak ( $t_s$ )        | 7,403 | 6,815 | Detik   |
| Error Steady State           | 0,017 | 0.196 | %       |

Berdasarkan Tabel 3. maka dapat diketahui parameter-parameter dari hasil respon berupa waktu puncak ( $t_p$ ), waktu tunda ( $t_d$ ) saat 10%-90%, waktu naik ( $t_r$ ), waktu tunak ( $t_s$ ) saat 5% dan Error Steady State. Pada sistem dengan *setpoint* 45°, nilai  $t_p$  yang dihasilkan yaitu 8,264 detik, nilai  $t_d$  sebesar 1,71 detik, nilai  $t_r$  sebesar 5,421 detik, nilai  $t_s$  sebesar 7,403 detik dan mempunyai *error steady state* sebesar 0.017%. Sedangkan pada sistem dengan *setpoint* 60°, nilai  $t_p$  yang dihasilkan yaitu 7,81 detik, nilai  $t_d$  sebesar 1,574 detik, nilai  $t_r$  sebesar 4,991 detik, nilai  $t_s$  sebesar 6,815 detik dan mempunyai *error steady state* sebesar 0.196%.

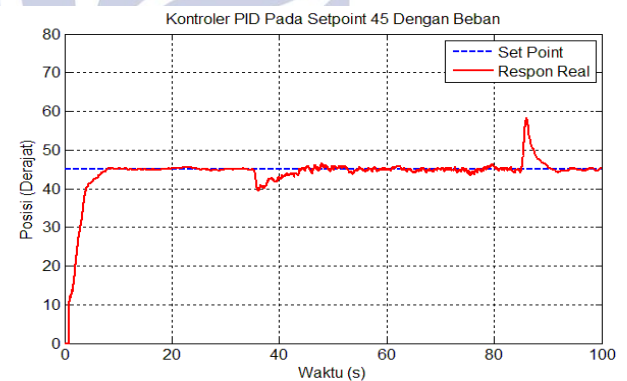
### Pengujian dengan Setpoint Berubah



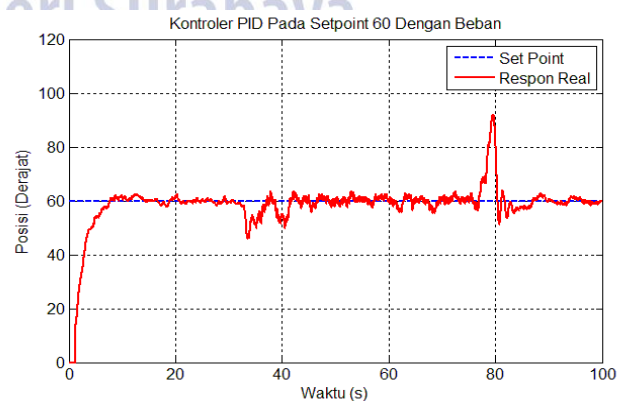
**Gambar 17.** Respon Sistem dengan Setpoint Berubah

Pengujian kedua dilakukan dengan mengubah *setpoint* yang telah ditentukan dalam satu percobaan untuk melihat kestabilan respon yang dimiliki *hardware Aero Pendulum*. *Setpoint* yang diberikan adalah 45° kemudian diubah menjadi 70°, 60° dan akhirnya kembali ke 45°. *Setpoint* diubah-ubah dalam kurun waktu 25 detik.

### Pengujian dengan Menggunakan Beban



**Gambar 18.** Respon Sistem pada Setpoint 45° dengan Beban



**Gambar 19.** Respon Sistem pada Setpoint 60° dengan Beban

**Tabel 4.** Hasil Pengujian Hardware Aero Pendulum Dengan Beban

| Setpoint | Error Steady State |
|----------|--------------------|
| 45°      | 0.069 %            |
| 60°      | 0.299 %            |

Berdasarkan Tabel 4. maka dapat disimpulkan dari pengujian ketiga yang dilakukan dengan memberi beban sebesar 50 gram pada *setpoint* tertentu yaitu. Pada *setpoint* 45° mempunyai *Error Steady State* 0.069%. Sedangkan pada *setpoint* 60° mempunyai *Error Steady State* 0.299%.

## PENUTUP

### Simpulan

Berdasarkan data hasil penelitian yang dilakukan, maka didapatkan simpulan bahwa sistem Aero Pendulum dengan pengendali PID berhasil dibuat dan dapat bekerja dengan baik. Pengujian juga dilakukan dengan menggunakan *software* LabView 2014 yang mampu mengolah data respon sistem riil dengan *real time*. Alat dapat melakukan *self stabilizing control* sesuai dengan *setpoint* yang sudah ditentukan. Dari hasil tuning Ziegler-Nichols metode kedua didapatkan parameter PID dengan respon dinamis terbaik yaitu  $K_p=0.06$ ,  $K_i=5$  dan  $K_d=0.05$ . Dan didapatkan hasil respon yang merujuk pada Tabel 3. berupa waktu puncak (*tp*), waktu tunda (*td*) saat 10%-90%, waktu naik (*tr*), waktu tunak (*ts*) saat 5% dan Error Steady State. Pada sistem dengan *setpoint* 45° didapatkan respon terbaik dengan nilai *tp* yang dihasilkan yaitu 8,264 detik, nilai *td* sebesar 1,71 detik, nilai *tr* sebesar 5,421 detik, nilai *ts* sebesar 7,403 detik dan mempunyai *error steady state* sebesar 0.017%. Sedangkan pada sistem dengan *setpoint* 60°, nilai *tp* yang dihasilkan yaitu 7,81 detik, nilai *td* sebesar 1,574 detik, nilai *tr* sebesar 4,991 detik, nilai *ts* sebesar 6,815 detik dan mempunyai *error steady state* sebesar 0.196%.

### Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dilakukan untuk pengembangan sistem pengendali *self stabilizing* pada Aero Pendulum selanjutnya yaitu, pada penelitian ini, simulasi sistem yang digunakan adalah *tuning second method* Ziegler-Nichols, untuk selanjutnya bisa menggunakan metode *tuning* PID lainnya misalnya metode Cohen-Coon atau PSO. Sistem ini juga masih dapat dikembangkan dengan menggunakan metode pengendali yang lain seperti *Fuzzy Logic Controller* (FLC), *Sliding Mode Control* (SMC), *LQR* dan lain-lain untuk mendapatkan hasil respon yang lebih baik dan stabil.

## DAFTAR PUSTAKA

- Azzumar, Muhammad. 2012. "Pemodelan dan Desain Kendali Sistem Aktuator Kendali Sirip Berbasis Brushless DC Motor". Depok: Universitas Indonesia, hal.5.
- Baldursson, Stefán. 2005. "BLDC Motor Modelling and Control-A Matlab®/ Simulink® Implementation". Gothenburg, Sweden: Chalmers University of Technology.
- Ghasemi R. dan M.R.Rahimi Khoiyani. 2014. "Designing Intelligent Adaptive Controller for Nonlinear Pendulum Dynamical System". International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:8, No:11, 2014.
- Gunnel, Onur dan Arif Ankarali. 2017. "Tuning Pid Controller Using Genetic algorithm And particle Swarm optimization Algorithm For Single propeller Pendulum". ICAME2017.
- Kasiram, Moh. 2008. Metodologi Penelitian Kualitatif dan Kuantitatif. Malang : UIN Maliki Press.
- Kizmaz, Hakan, dkk. 2010. "Sliding Mode Control of Suspended Pendulum". Modern Electric Power Systems, Wroclaw, Poland. MEPS'10 - paper P14.
- Mohammadbagheri, Amin dan Mahdi Yaghoobi, 2011. "A new approach to control a driven pendulum with PID method". IEEE, UKSim 2011, pp. 207–211.
- Ogata, Katsuhiko. 1985. Teknik Kontrol Automatik jilid 1. Terjemahan Edi Laksono. Jakarta: Erlangga.
- Raju, S.Srinivasulu, dkk. 2012. "Design of Quadratic Dynamic Matrix Control for Driven Pendulum System". International Journal of Electronics and Communication Engineering, Vol.5, pp.363-370.
- Raranda. 2017. "Implementasi Kontroler PID Pada Two Wheels Self Balancing Robot Berbasis Arduino UNO". Jurnal Teknik Elektro. Vol. 06: hal. 89-96.
- SA,Inkyu. 2011. "Indoor visual SLAM with a Quadrotor". University Queensland, Juli 2011.