

RANCANG BANGUN SISTEM PENGENDALIAN *SELF BALANCING* PADA *DUAL MOTOR PROPELLER* MENGGUNAKAN KONTROLER PID

Mohamad Samsul Hudha

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

Muhamad Syariffuddin Zuhrie

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

Abstrak

Salah satu jenis dari Pesawat tanpa awak yaitu *Multirotor* yang menggunakan sistem *vertical take off landing* dalam sikap terbangnya. Seiring dengan perkembangan zaman, kebutuhan akan pesawat tanpa awak (*Unmanned Aerial Vehicle, UAV*) semakin diminati untuk menunjang berbagai aspek seperti pertahanan, pemetaan lahan perkebunan, sipil dan dokumentasi video. Permasalahan dalam *multirotor* yaitu dalam stabilitas terbang pada saat beban di *multirotor* tidak seimbang akan mempengaruhi stabilitas sikap terbang *multirotor* yang bisa saja menimbulkan jatuhnya *multirotor* saat di udara. Tujuan penelitian ini adalah untuk merancang *multirotor (dual motor propeller)* skala kecil dan mengontrol kecepatan motor *propeller* menggunakan kontroler PID untuk deteksi kecepatan sudut *frame* mendapatkan stabilitas walaupun mendapat beban tidak seimbang. Hasil penelitian menunjukkan bahwa *Dual Motor Propeller* dengan pengendali PID berhasil dibuat dan bekerja dengan baik. Alat dapat melakukan *control self balancing* sesuai *set point* (0 derajat) titik seimbang. Implementasi kontroler PID pada sistem pengendali *self balancing* pada *Dual Motor Propeller* dengan nilai respon dinamis terbaik pada saat parameter PID dengan nilai yang didapat dari simulasi tuning *Ziegler-Nichols* yaitu $K_p = 1.2$, $K_i = 0.045$ dan $K_d = 0.89$ dapat memperbaiki respon sistem dengan $Ess = 1.23\%$ Maksimal *overshoot* = 3.16% $t_d = 1.04$ s, $t_r(5\% - 95\%) = 0.836$ s, $t_s(5\%) = 1.73$ s.

Kata Kunci: UAV, *Multirotor*, Stabilitas, Kontroler PID.

Abstract

One type of unmanned aircraft is the *Multirotor* that uses the *vertical take-off* system in its flying attitude. Along with the times, the need for *Unmanned Aerial Vehicle (UAV)* is increasingly in demand to support various aspects such as defense, mapping plantations, civil and video documentation. Problems in *multirotor* that is in flight stability when the load in *multirotor* is not balanced will affect the stability of the *multirotor* flight attitude which could lead to the fall of *multirotor* when in air. The purpose of this study is to design a small scale *multirotor (dual motor propeller)* and controlling the speed of the propeller motor using the PID controller for detection of frame angular speed gets stability even though it has an unbalanced load. The results of the study show that *Dual Propeller Motor* with PID controller was successfully made and works well. The tool can control *self balancing* according to the *set point* (0 degree) of the balance. Implementation of the PID controller on the *self balancing* control system at *Dual Motor Propeller* with the best dynamic response value when the PID parameter with the value obtained from *Ziegler-Nichols* tuning simulation is $K_p = 1.2$, $K_i = 0.045$ and $K_d = 0.89$ can improve the system response with $Ess = 1.23\%$ Maximum *overshoot* = 3.16% $t_d = 1.04$ s, $t_r(5\% - 95\%) = 0.836$ s, $t_s(5\%) = 1.73$ s.

Keywords: UAV, *Multirotor*, Stability, PID Controller.

PENDAHULUAN

Pesawat Tanpa Awak (*Unmanned Aerial Vehicle, UAV*) adalah wahana terbang tanpa awak yang dalam satu dasawarsa terakhir ini berkembang kian pesat. Bukan

hanya mereka yang berada di ranah departemen pertahanan atau badan-badan riset, termasuk di perguruan tinggi, yang meneliti, mengkaji dan mengembangkan, tapi dunia industri dan bidang sipil pun telah mulai banyak

memanfaatkan teknologi *unmanned system* ini dalam mendukung kegiatan keseharian manusia. (I. Suroso,2016).

Salah satu jenis dari Pesawat tanpa awak yaitu *Multirotor* yang menggunakan sistem *vertical take off landing* dalam sikap terbangnya, yang menjadi permasalahan dalam *multirotor* yaitu dalam keseimbangan terbang pada saat beban *multirotor* tidak seimbang. (Masahiko Mizui,2012) .

Seiring dengan perkembangan zaman, kebutuhan akan pesawat tanpak awak (*Unmanned Aerial Vehicle, UAV*) semakin diminati untuk menunjang berbagai aspek seperti pertahanan, pemetaan lahan perkebunan, sipil dan dokumentasi video. Berbagai jenis pesawat tanpa awak yang telah dikembangkan salah satunya yaitu *multirotor* yang memanfaatkan motor dan *propperler* sebagai daya angkat wahana menggunakan sistem *Vertical Take Off Landing (VTOL)*.(Yusuf Rahmat,2016).

UAV (Unmanned Aerial Vehicle) dimungkinkan dikontrol dari jarak jauh melalui radio. Kontrol penerbangan menggunakan model dinamika dari helikopter untuk optimalisasi sikap pada kontrol dan pelayaran otonom oleh mikrokontroler dan sensor yang mengukur parameter penerbangan. *multirotor* mengontrol lebih dari satu rotor,bagian rotor motor, proppeler, dll. (Muhammad Zaki Mustapa,2015).

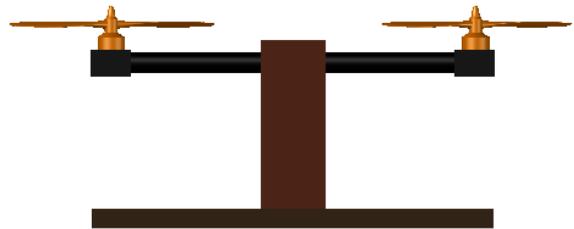
Twinrotor memiliki dua buah baling-baling sebagai penggerakannya. Balingbaling tersebut terletak disamping kanan dan kiri dari badan pesawat dan baling-balingUAV jenis ini memiliki Sistem VTOL (*Vertical Take Off and Landing*) sehingga membuatnya tidak membutuhkan landasan pacu yang panjang untuk dapat mengudara dan karena baling-baling dapat digerakkan secara longitudinal, maka *twinrotor* sewaktu-waktu dapat melakukan *hovering* (Shlok Agarwal, 2013).

Permasalahan dalam *multirotor* yaitu dalam stabilitas terbang pada saat beban di *multirotor* tidak seimbang akan mempengaruhi sikap terbang *multirotor* yang bisa saja menimbulkan jatuhnya *multirotor* saat di udara. Maka dilakukan penelitian rancang bangun sistem pengendalian *self balancing* pada *dual motor proppeler* menggunakan kontroler PID .

Pemodelan Dinamik Dual Motor Propeller

UAV Dual motor proppeler atau *Twinrotor* adalah jenis kopter yang digerakkan oleh dua *rotor*. Pisau-pisau berputar dalam arah yang berlawanan dan *rotor* ekor tidak diperlukan untuk melawan bertindak *angular momentum* baling-baling. Sebagai sistem dinamik yang digabungkan, dengan mengubah kecepatan motor, posisi juga berubah. Sistem ini digerakkan dan sangat dinamis tidak stabil. Di banyak situasi yang diinginkan adalah sistem harus sekecil mungkin untuk mencapai gerakan besar, bisa bergerak baik

secara vertikal. karakteristik khusus, seperti penerbangan vertikal kemampuan dan terbang dengan kecepatan rendah, memungkinkan model untuk melakukan tugas yang sulit diterapkan melalui mekanisme dan struktur lain. (Shlok Agarwal,2013).



Gambar 1. Dual Motor Propeller (Shlok Agarwal,2013).

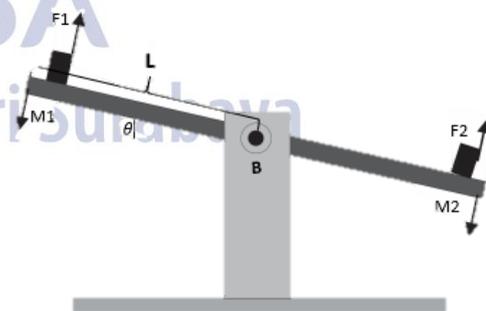
Model Plant

Untuk pemodelan sistem ini, kami menggunakan metode Newton. Dalam metode ini kita menulis persamaan gaya dan torsi yang diterapkan pada sistem. sistem telah dilihat sebagai benda tegar dengan distribusi massa seragam. Penulis berasumsi bahwa massa total balok terkonsentrasi di masing-masing sisi balok. M_1 dan M_2 adalah jumlah motor massa dan massa terkonsentrasi balok di masing-masing sisi. B , j dan L adalah koefisien redaman, momen inersia dan setengah panjang balok. Dari gambar 2 dapat kita tuliskan persamaan :

$$j\ddot{\theta} = \sum T = F_1L - F_2L - M_1Lg\cos\theta + M_2Lg\cos\theta - B\dot{\theta} \quad (1)$$

Untuk persamaan sistem linierisasi, penulis menggunakan 2 variabel keadaan :

$$\begin{aligned} x_1 &= \theta \\ x_2 &= \omega = \dot{\theta} \end{aligned} \quad (2)$$



Gambar 2. Skematik Gaya pada Sistem (Shlok Agarwal,2013)

Sekarang persamaan sistem nonlinier adalah :

$$\begin{aligned} \dot{x} &= x_2 = f_1(t) \\ \dot{x}_2 &= \ddot{\theta} = \frac{1}{j}(F_1L - F_2L - m_1Lg\cos\theta + m_2Lg\cos\theta - B\dot{\theta}) \end{aligned} \quad (3)$$

Dengan menggunakan metode jacobian, kita dapat membuat melinearkan persamaan (3) menjadi

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = J_x(0) = \begin{bmatrix} 0 & 1 \\ \frac{Lg\sin(0)}{j} & \frac{-B}{j} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial f_1}{\partial F} \\ \frac{\partial f_2}{\partial F} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{L}{j} \end{bmatrix} \quad (5)$$

Dan sekarang persamaan linear *state space* menjadi :

$$\dot{x} = Ax + Bu$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ \frac{Lg\sin(0)}{j} & \frac{-B}{j} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{L}{j} \end{bmatrix} F$$

$$y = Cx + Du$$

$$y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (6)$$

Fungsi transfer sistem adalah:

$$G(s) = [1 \ 0] \begin{bmatrix} s & -1 \\ \frac{Lg\sin(0)(m_1-m_2)}{j} & s + \frac{B}{j} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \frac{L}{j} \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} s & -1 \\ \frac{Lg\sin(0)(m_1-m_2)}{j} & s + \frac{B}{j} \end{bmatrix}^{-1}$$

$$= \frac{1}{s(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}} \begin{bmatrix} s + \frac{B}{j} & 1 \\ \frac{Lg\sin(0)(m_1-m_2)}{j} & s \end{bmatrix}$$

$$= \begin{bmatrix} \frac{s + \frac{B}{j}}{s(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}} & \frac{1}{s(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}} \\ \frac{\frac{Lg\sin(0)(m_1-m_2)}{j}}{s(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}} & \frac{s}{s(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}} \end{bmatrix}$$

$$[1 \ 0] \begin{bmatrix} \frac{s + \frac{B}{j}}{s(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}} & \frac{1}{s(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}} \\ \frac{\frac{Lg\sin(0)(m_1-m_2)}{j}}{s(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}} & \frac{s}{s(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{L}{j} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{s + \frac{B}{j}}{s(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}} & \frac{1}{s(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{L}{j} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\frac{L}{j}}{s^2 + \frac{B}{j}s + \frac{Lg\sin(0)(m_1-m_2)}{j}} \end{bmatrix}$$

$$= \frac{\frac{L}{j}}{\frac{s}{j}(s + \frac{B}{j}) + \frac{Lg\sin(0)(m_1-m_2)}{j}}$$

$$= \frac{L}{js^2 + Bs + Lg\sin(0)(m_1-m_2)}$$

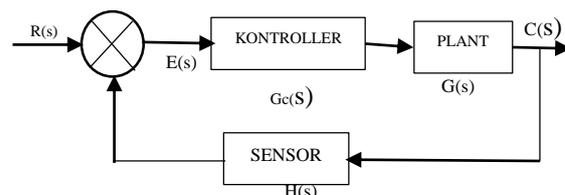
$$G(s) = \frac{L}{js^2 + Bs} = \frac{\frac{L}{j}}{s^2 + \frac{Bs}{j}} \quad (8)$$

Keterangan

- $x_1 = \theta =$ Posisi Sudut (rad)
- $x_2 = \omega = \dot{\theta} =$ Kecepatan Sudut (rad/s)
- $\ddot{\theta} =$ Percepatan (rad/s²)
- $F_1 =$ Gaya Dorong Motor 1 /Trust (kg)
- $F_2 =$ Gaya Dorong Motor 2/Trust (kg)
- $\partial f_1 =$ Deferensial gaya dorong motor 1
- $\partial f_2 =$ Deferensial gaya dorong motor 2
- $\partial x_1 =$ Deferensial posisi sudut 1
- $\partial x_2 =$ Deferensial posisi sudut 2
- $\partial F =$ Deferensial percepatan
- $Ax =$ formula variabel state
- $Bu =$ formula variabel state
- $Cx =$ formula variabel state
- $Du =$ formula variabel state
- $y =$ Hasil variabel state
- $g =$ nilai gravitasi
- $L =$ Panjang Lengan (Cm)
- $M1 =$ massa ujung kiri (kg)
- $M2 =$ massa ujung kanan (kg)
- $B =$ Koefisien Redaman (g/mL)
- $J =$ Moment Inersia (Kg/m²)

Kontroller PID Tuning Ziegler-Nichols

Metode umum untuk mendapatkan nilai awal kontrol PID adalah untuk merancang sistem kontrol PID. Hal ini disebabkan karena parameter Kp, Ki dan Kd tidak *independent*. Untuk mendapatkan aksi kontrol yang baik diperlukan langkah mencoba kombinasi antara P, I dan D sampai ditemukan nilai Kp, Ki dan Kd seperti yang diinginkan. Alat pengendali PID sangat sering digunakan dalam pengendali industri.

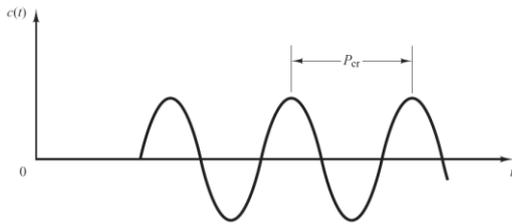


Gambar 3. Diagram Closeloop (Sumber : Ogata, 1985)

Metode Kedua

Pada metode kedua, pertama-tama kita mengatur $T_i = \infty$ dan $T_d = 0$. Menggunakan tindakan kontrol

proporsional saja, tingkatkan K_p dari 0 ke nilai kritis K_{cr} dimana output pertama menunjukkan osilasi berkelanjutan. Jika output 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. Osilasi berkelanjutan dengan periode (Sumber : Ogata, 1985)

METODE PENELITIAN

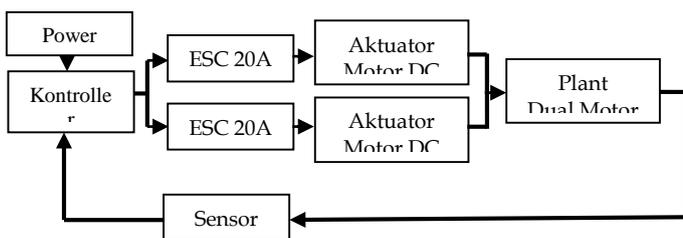
Pendekatan Penelitian

Pada penelitian ini, pendekatan penelitian yang digunakan adalah penelitian kuantitatif. Kasiram (2008:149) 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.

Pada penelitian ini menggunakan *software* LabVIEW 2014 dan Arduino IDE 1.8.5. *Software* LabVIEW akan digunakan untuk mencari fungsi alih dari *Dual Motor Proppeler*, mendesain kontrolernya, yaitu kontroler PID (*Proporsional-Integral-Derivative*) dengan *zsecond method* dan *tuning Ziegler-Nichols* serta menampilkan respon sistem dari *hardware* plant yang sedang menerapkan *balancing control* ketika diberikan beban disalah satu sisi. Sedangkan *software* Arduino IDE digunakan untuk memprogram *hardware* plant.

Rancang Bangun Hardware

Desain sistem Pengendali *Self Balancing* pada *Dual Motor Proppeler*. Ditunjukkan pada Gambar 5.

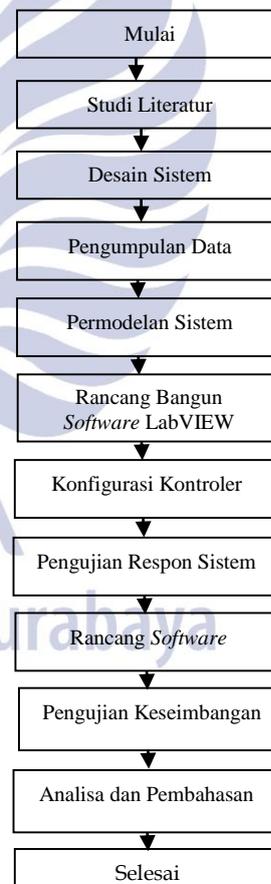


Gambar 5. Desain Sistem Self Balancing pada Dual Motor Proppeler (Sumber : Dokumen Pribadi, 2019)

Blok sistem kendali dari perangkat *hardware balancing dual motor proppeler*. 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 balancing control*. Hasil dari keluaran kontroler berupa tegangan yang akan masuk ke *Electrical Speed Control (ESC) 20A* yang dikonversi ke nilai PWM dan 2 (dua) buah *Motor DC Brushles, Electrical Speed Control (ESC) 20A* akan mengatur arah putar dan kecepatan motor *DC Brushless*. Sensor *accelerometer* dan *gyroscope* akan bekerja untuk mengukur nilai dari derajat kemiringan plant setelah adanya gerakan akibat perputaran motor dan proppeler dan kemudian menjadikan nilai pembacaan sudut sebagai nilai umpan balik.

Rancangan Penelitian

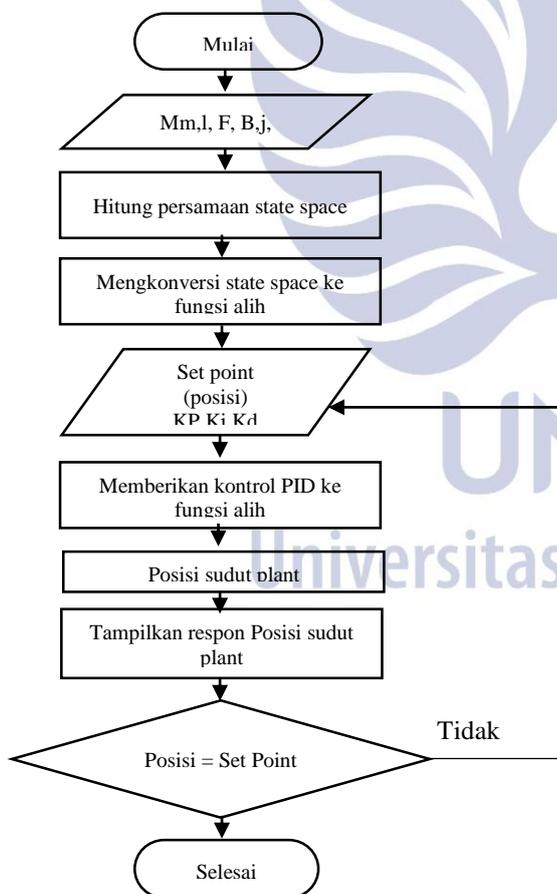
Urutan rancangan penelitian ditunjukkan pada Gambar 6.



Gambar 6. Tahapan rancangan penelitian (Sumber : Dokumen Pribadi, 2019)

Rancang Bangun Software

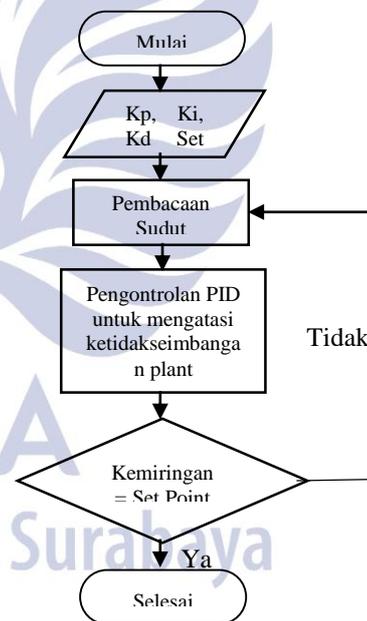
Sebelum merancang bangun *software* LabVIEW maka harus membuat desain algoritma dari sistem. Desain algoritma tersebut selanjutnya akan digunakan untuk membuat diagram alir dari kerja *software* LabVIEW 2014. Berikut adalah desain algoritma untuk simulasi dengan *software* LabVIEW 2014 : (1) Input spesifikasi *hardware* dari *Dual Motor Propeller* yaitu massa motor, daya dorong motor propeller . jarak antara titik tengah ke ujung lengan,(2) Input *Transfer function* (3) Input *setpoint* (posisi), K_p, K_i dan K_d . (4)Memberikan kontrol PID pada fungsi alih *self balancing Dual Motor Propeller* dengan *control and simulation loop* LabVIEW 2014. (5) Output dari sistem adalah posisi plant *Dual Motor Propeller*. (6) Menampilkan respon posisi dari pengendali PID menggunakan *Waveform Chart* pada LabVIEW. (7) Apabila plant *Dual Motor Propeller* mampu menyeimbangkan diri antara motor kanan dan kiri atau mampu mempertahankan posisinya sesuai *set point* maka program selesai. Namun jika tidak mampu menyetabilkan diri seimbang maka harus mengatur nilai konstanta *proporsional*, konstanta *integral* dan konstanta *derivative*.



Gambar 7. Diagram Alir Simulasi Dual motor Propeller dengan *software* LabVIEW 2014 (Sumber : Dokumen Pribadi, 2019)

Rancang bangun *software* Arduino IDE yang dimaksud adalah sebuah program yang dimasukkan ke dalam mikrokontroler Arduino. Bahasa pemrograman yang digunakan di *software* Arduino IDE adalah Bahasa C++, Arduino IDE bersifat *opensource* sehingga banyak *library* yang dikembangkan secara perorangan atau komunitas diluarr *developer* Arduino IDE sendiri. Berikut adalah penjelasan desain algoritma dari diagram alir *software* Arduino IDE yang akan dibuat: (1) Input nilai kontroler PID yaitu K_p (konstanta proporsional), K_i (konstanta integral) dan K_d (konstanta derivative) serta pemberian nilai *set point*. (2) Melakukan pembacaan nilai dari sensor MPU6050 yaitu berupa sudut posisi kemiringan dari lengan *self balancing dual motor propeller*. (3) Melakukan pengontrolan PID untuk menyeimbangkan lengan *self balancing dual motor propeller* agar sesuai *set point* dengan cara menggerakkan motor DC yang terpasang *propeller*. (4) Apabila *self balancing dual motor propeller* dapat seimbang atau kemiringan sudah sesuai *set point* maka *self balancing control* berhasil diterapkan. Namun jika kemiringan tidak sesuai *set point* maka akan dilakukan pembacaan lagi sampai didapatkan nilai kemiringan sama dengan *set point*.

Berikut adalah rancangan diagram alir dari perancangan *software* Arduino IDE :



Gambar 8. Diagram alir *software* Arduino IDE (Sumber : Dokumen Pribadi, 2019)

Rancangan Plant Dual Motor Propeller

Pembuatan *plant* terdiri dari rangka penopang. Rangka penopang terbuat dari Papan ketebalan 10 cm dan Plat Holo 3x4 dengan dimensi $P = 40$ cm, $l = 40$ cm, $t = 17$ cm. Gambar 9. adalah desain rancangan *plant Self Balancing Dual Motor Propeller* :

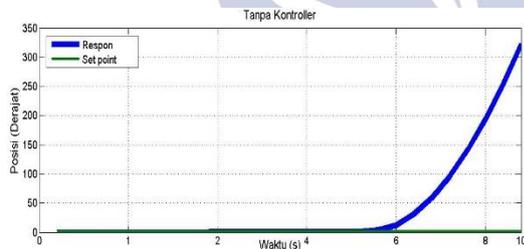


Gambar 9. Desain Mekanik Self Balancing Dual Motor Propeller yang digunakan untuk penelitian (Sumber : Dokumen Pribadi, 2019)

HASIL DAN PEMBAHASAN

Pengujian dan analisis meliputi ; 1) Rancangan rangkaian *hardware*, 2) Pengujian dan analisis *plant* respon lup tertutup tanpa kendali, 3) Pengujian dan analisis *plant* dengan kontroler PID lalu pemberian gangguan.

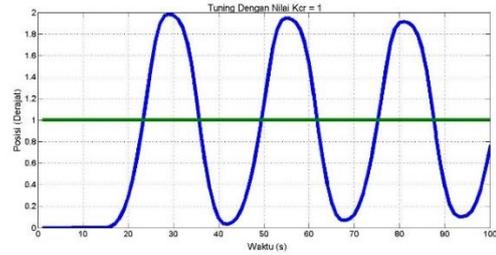
Rancangan Rangkaian Hardware Respon Simulasi lup tertutup tanpa kendali



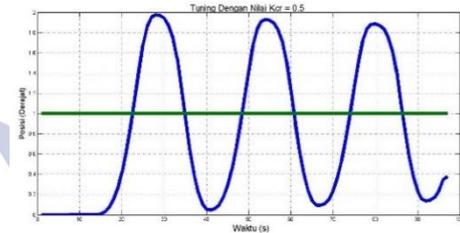
Gambar 10. Respon Tanpa Kontroler. (Sumber : Dokumen Pribadi, 2019)

Pada Gambar 10. terlihat bahwa respon sistem dari *Dual Motor Propeller* tidak bisa mengikuti *set point* yang diberikan. Respon sistem *Dual Motor Propeller* langsung menuju nilai tak hingga dan ini membuktikan bahwa *Dual Motor Propeller* tidak bisa mempertahankan posisinya seimbang. Dari respon sistem *Dual Motor Propeller* yang tidak bisa mempertahankan posisinya sesuai *set point* tersebut maka perlu dirancang suatu kontroler yang dapat membuat *Dual Motor Propeller* mempertahankan posisinya sesuai *set point*.

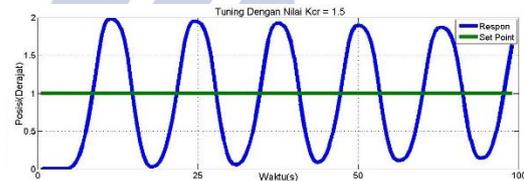
Pada penelitian ini digunakan kontroler PID (*proportional-integral-Derivative*). Kemudian untuk *tuning* kontrol PID digunakan metode *tuning second method Ziegler-Nichols*. Untuk metode *tuning Ziegler-Nichols* pertama-tama memberikan nilai *Kcr* sampai output pertama menunjukkan osilasi berkelanjutan. Berikut adalah hasil *tuning* nilai *Kcr* dengan *set point* = 1 :



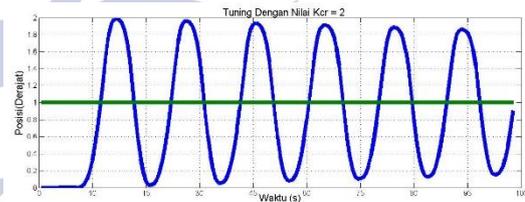
Gambar 11. Hasil respon sistem dengan nilai $K_{cr}=0.5$ (Sumber : Dokumen Pribadi, 2019)



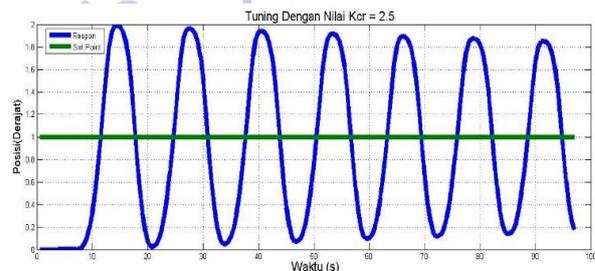
Gambar 12. Hasil respon sistem dengan nilai $K_{cr} = 1$ (Sumber : Dokumen Pribadi, 2019)



Gambar 13. Hasil respon sistem dengan nilai $K_{cr} = 1.5$ (Sumber : Dokumen Pribadi, 2019)



Gambar 14. Hasil respon sistem dengan nilai $K_{cr} = 2$ (Sumber : Dokumen Pribadi, 2019)



Gambar 15. Hasil respon sistem dengan nilai $K_{cr} = 2.5$ (Sumber : Dokumen Pribadi, 2019)

Dalam uji respon sistem dari *transfer function Dual Motor Propeller* dengan variasi nilai *Kcr* yang diberikan yaitu dari Gambar 11-15 , menunjukkan respon sistem

berosilasi. Maka sesuai dengan *second method Ziegler-Nichols* yaitu ketika meningkatkan nilai K_p dari 0 ke nilai kritis K_{cr} dimana output pertama menunjukkan osilasi berkelanjutan maka metode ini berlaku untuk fungsi alih tersebut, namun jika output tidak menunjukkan osilasi berkelanjutan untuk nilai K_{cr} apa pun yang diperlukan, metode ini tidak berlaku. Karena respon sistem berosilasi maka model fungsi alih *Dual Motor Propeller* bisa menggunakan *tuning second method Ziegler-Nichols*.

Dari data yang telah didapatkan dari hasil simulasi maka dapat dianalisis respon sistem dari *Dual Motor Propeller* yang menunjukkan respon osilasi. Berikut adalah Tabel 2 perbandingan respon sistem *Dual Motor Propeller* saat menunjukkan respon osilasi.

Tabel 2. Analisis Respon Sistem Hasil Simulasi

No	Kcr	Pcr	Mp	tr	Tp
1.	0.5	3.1	1.975	1.65	2.47
2.	1	1.6	1.982	1.55	2.06
3.	1.5	2.7	1.985	1.44	1.75
4.	2	5.1	1.986	1.42	1.57
5.	2.5	6.8	1.988	1.38	1.44

(Sumber : Dokumen Pribadi, 2019)

Nilai K_{cr} yang paling sesuai adalah ketika respon sistem memiliki nilai *Maximum peak* (M_p) yang paling kecil dan *rise time* (t_r) dan *peak time* (t_p) yang paling cepat. Dari beberapa variasi *tuning* nilai K_{cr} maka dapat disimpulkan bahwa nilai K_{cr} yang paling sesuai adalah 2. Karena pada saat nilai $K_{cr}=2$, respon sistem memiliki nilai *Maximum peak* (M_p) yang paling kecil, kemudian waktu naik ketika $t=0$ sampai ke *steady state* (*rise time*) dan waktu ketika $t=0$ hingga mencapai puncak pertama *overshoot* memiliki waktu yang paling kecil.

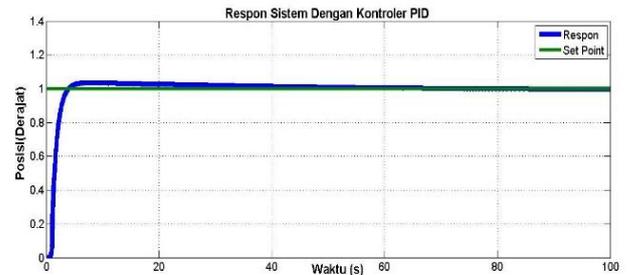
Setelah didapatkan nilai K_{cr} maka langkah selanjutnya adalah mencari nilai K_p (*konstanta proporsional*), T_i (*Integral Time*) dan T_d (*Derivative Time*) dengan rumus *second method Ziegler-Nichols*, berikut adalah Tabel 3. aturan *tuning Ziegler-Nichols* :

Tabel 3. Aturan tuning kontrollor PID Ziegler-Nichols

Tipe Kendali	K_p	T_i	T_d
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

(Sumber : Dokumen Pribadi, 2019)

Dari rumus tersebut maka akan dapat dihitung nilai kontroler PID yaitu K_p , K_i dan K_d yaitu $K_p=1,2$, $K_i=0.045$ dan $K_d=0.89$.



Gambar 16. Hasil grafik respon sistem simulasi (Sumber : Dokumen Pribadi, 2019)

Pada Gambar 10. ditampilkan respon sistem dengan kontroler PID (*Proportional-Integral-Integratif*) dengan *tuning second method Ziegler-Nichole*. Terlihat sistem lebih stabil dengan osilasi yang kecil. 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*. Berikut adalah Tabel 4. respon sistem dengan kontroler PID :

Tabel 4. Respon Sistem dengan Kontroler PID

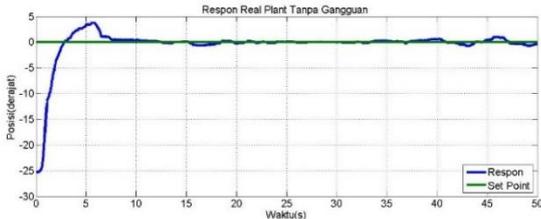
Data ke	Waktu	Respon	Set Point
1	1	0.000162	1
2	1.1	0.837	1
3	1.2	1.001	1
4	1.3	1.029	1
5	1.4	1.033	1
6	1.5	1.033	1
7	1.6	1.032	1
8	1.7	1.031	1
9	1.8	1.03	1
10	1.9	1.029	1
11	2	1.028	1
12	2.1	1.027	1
13	2.2	1.026	1
14	2.3	1.025	1
15	2.4	1.024	1

(Sumber : Dokumen Pribadi,2019)

Nilai dari Tabel 4. diambil pada rentang waktu 0,1 detik sekali dalam rentang waktu 5 detik. Dari Tabel 4. tersebut dapat diketahui bahwa nilai *Maximum peak* (M_p) adalah 1,033. Dari nilai *maximum peak* dapat diketahui nilai dari *overshoot* sistem adalah 0,033. Sistem mulai masuk ke keadaan *steady state* pada saat detik ke 2,7. Kemudian nilai *steady state error* didapatkan dari nilai rata-rata ketika keadaan sudah *steady state*. Setelah

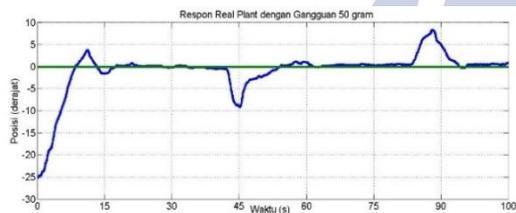
dilakukan perhitungan maka didapatkan nilai *steady state error* yaitu 1.23.

Pengujian Respon *Hardware Dual Motor Propeller* Tanpa Gangguan

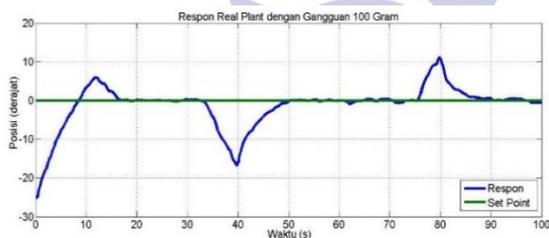


Gambar 11. Respon Real Plant Tanpa Gangguan. (Sumber : Dokumen Pribadi, 2019)

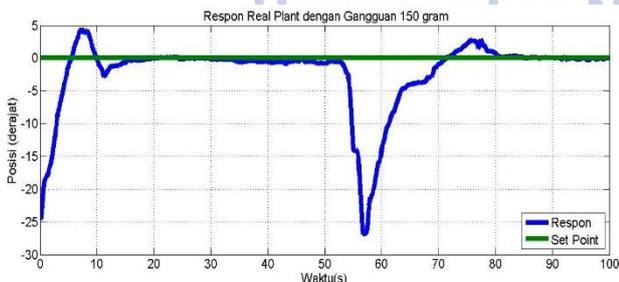
Dengan Gangguan



Gambar 12. Respon Real Plant Mendapat Gangguan Beban 50 gram. (Sumber : Dokumen Pribadi, 2019)



Gambar 13. Respon Real Plant Mendapat Gangguan Beban 100 gram. (Sumber : Dokumen Pribadi, 2019)



Gambar 14. Respon Real Plant Mendapat Gangguan Beban 150 gram. (Sumber : Dokumen Pribadi, 2019)

Dari percobaan dengan beberapa kondisi tersebut dapat diketahui bahwa *Dual Motor Propeller* melakukan

self balancing control dengan baik. Untuk lebih lengkapnya tentang hasil pengujian *hardware balancing Dual Motor Propeller*. ketika mengatasi kemiringan lengan akan dijelaskan pada Tabel 5.

Tabel 5. Respon Sistem dengan Kontroler PID

Pengujian Respon	Mp	t_p	t_r	E_{ss}	t_s	t_d
Tanpa Gangguan	3.16	1.44	0.836	1.23	1.73	1.04
Beban 50 g	8.19	1.57	0.873	2.43	2.65	1.12
Beban 100 g	12.2	1.58	0.922	2.69	2.69	1.23
Beban 150 g	17.2	1.63	0.955	2.89	3.12	1.29
	%	detik	detik	Dera jat	detik	Deti k

(Sumber : Dokumen Pribadi, 2019)

Pada pengujian *hardware Dual Motor Propeller* dilakukan 4 kali yaitu pengujian respon *Dual Motor Propeller* tanpa diberikan gangguan dan diberikan 3 (tiga) variabel gangguan yaitu beban 50 g, 100 g, 150 g. Gangguan yang dimaksud adalah *Dual Motor Propeller* ketika plant sudah dalam keadaan seimbang baru setelah itu diberikan beban pada salah satu lengan *Dual Motor Propeller*. Pengujian ini dilakukan dengan kontroler PID dengan nilai $K_p=1,2$, $K_i=0,045$ dan $K_d=0,85$.

Hasil pengujian *hardware Dual Motor Propeller* disajikan pada Tabel 5 yang dapat diketahui bahwa *Dual Motor Propeller steady state error* paling kecil terjadi ketika tidak diberikan gangguan yaitu *steady state error* sebesar 1,23. Kemudian nilai *steady state error* paling besar terjadi ketika diberikan gangguan 150 gram yaitu *steady state error* sama dengan 2,89.

Time response didapatkan dari waktu yang dibutuhkan *Dual Motor Propeller* untuk mengatasi gangguan yang diberikan ketika dalam keadaan seimbang sampai menuju *steady state*. Uji terhadap gangguan ini dilakukan sebanyak 3 kali. Untuk gangguan pertama (beban 50 gram) *time response* sebesar 0.873 detik, untuk gangguan yang kedua (beban 100 gram) *time response* sebesar 0.922 detik. Dan untuk gangguan yang ketiga (beban 150 gram) *time response* sebesar 0.955 detik

PENUTUP

Simpulan

Berdasarkan data hasil penelitian yang telah dilakukan, maka didapatkan kesimpulan yaitu *Dual Motor Propeller* dengan pengendali PID berhasil dibuat dan bekerja dengan baik. Alat dapat melakukan *control self balancing* sesuai

set point (0 derajat) titik seimbang dan Implementasi kontroler PID pada sistem pengendali *self balancing* pada *Dual Motor Propeller* dengan nilai respon dinamis terbaik pada saat parameter PID yang merujuk pada Tabel 5 dari simulasi tuning Ziegler-Nichols yaitu $K_p = 1.2$, $K_i = 0.045$ dan $K_d = 0.89$ dapat memperbaiki respon sistem dengan $E_{ss} = 1.23\%$ Maksimal *overshoot* = 3.16% $t_d = 1.04$ s, $t_r(5\% - 95\%) = 0.836$ s, $t_s(5\%) = 1.73$ s.

Saran

Berdasarkan penelitian yang telah dilakukan, ada beberapa saran yang dapat dilakukan untuk pengembangan sistem pengendali *self balancing* pada *dual motor propeller* selanjutnya yaitu pada penelitian ini simulasi sistem *balancing robot* dilakukan dengan *tuning second method Ziegler-Nichols*, untuk selanjutnya bisa menggunakan metode *tuning* PID lainnya misalnya metode *Cohen-Coon*, *Tyru-Luyben*, *Minimum Integral Error* dan Pada penelitian ini masih belum memiliki GUI, untuk penelitian selanjutnya dapat ditambahkan tampilan GUI LabVIEW yang digunakan untuk menampilkan data secara *real time* dan mengontrol memberikan gangguan secara program.

DAFTAR PUSTAKA

Agarwal, Shlok, Apoorva Mohan dan Kamlesh Kumar. 2013. “*Design And Fabrication Of Twinrotor UAV*”. India. *Department of Mechatronics, Manipal University*.

Gemilang, Yusuf Rahmat, Bambang Suprianto, 2016. “*Kendali Jarak Jauh UAV (Unmanned Aerial Vehicle) Tipe Quadcopter Menggunakan Transceiver Nrf24l01+ Beserta Job Sheet Uji Coba*”. Pendidikan Teknik Elektro, Universitas Negeri Surabaya.

Kasiram, Moh. 2008. *Metodologi Penelitian Kualitatif dan Kuantitatif*. Malang : UIN Maliki Press.

Mustapa, Muhammad Zaki. 2015. “*Kontrol Kecepatan Motor DC Dengan Metode PID Menggunakan Visual Basic 6.0 Dan Mikrokontroler ATmega 16*” *Jurnal EECCIS* Vol. 6, No. 2.

Mizui, Masahiko, Ikuo Yamamoto, dan Ryouga Ohsawa. 2012. “*Resonance Analysis of the UAV Rotor-arm part*”. Kyushu Kyouritu University. *IOSR Journal of Engineering (IOSRJEN)*.

Ogata, Katsuhiko, Edi Leksono. 1995. “*Teknik Kontrol Automatik Jilid 1*”. Bandung. Erlangga,

Suroso, Indreswari. 2016. “*Peran Drone/Unmanned Aerial Vehicle (UAV) Buatan STTKD Dalam Dunia Penerbangan*”. *Jurnal Teknik Aeronautika Sekolah Tinggi Teknologi Kedirgantaraan*.