

RANCANG BANGUN SISTEM PENGENDALIAN SELF BALANCING PADA DUAL MOTOR PROPELLER BERBASIS FUZZY LOGIC CONTROLLER

Armanda Hari Setiyawan

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

Muhammad Syariffuddin Zuhrie

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

Abstrak

Unmanned Aerial Vehicle (UAV) merupakan pesawat tanpa awak yang dapat dikendalikan dari jarak jauh sehingga tidak membutuhkan pilot untuk menerbangkannya. *Twincopter/twinrotor* merupakan salah satu jenis UAV yang sedang dikembangkan akhir-akhir ini yang memiliki sistem penggerak dua buah motor beserta baling baling. Pada penerapannya, sistem kendali *on/off* masih belum mampu menyeimbangkan twinrotor dengan baik jika memiliki beban yang tidak seimbang maupun seimbang. Oleh karena itu dibutuhkan metode kontrol lain untuk menyeimbangkan sistem dari *self balancing* pada *dual motor propeller* secara otomatis. Tujuan penelitian ini adalah membuat dan merancang pengendalian *self balancing* pada *dual motor propeller* berbasis *fuzzy logic controller* untuk menyeimbangkan disaat tidak memiliki beban yang sama imbang. Penelitian ini menggunakan *hardware* MPU-6050 sebagai sensor, arduino sebagai mikrokontroler dan dua motor *brushless dc* sebagai aktuator. Hasil penelitian ini menunjukkan bahwa sistem pengendalian *self balancing* pada *dual motor propeller* berbasis *fuzzy logic controller* dapat menyeimbangkan disaat tidak memiliki beban yang tidak seimbang. Dengan mengimplementasikan kendali *fuzzy* pada *plant* dapat menghasilkan analisis respon dinamik $E_{ss} = 0.362$, simpangan maksimum = 24 %, $t_d = 1,013$ detik, $t_p = 1,501$ detik, $t_s (5\%) = 0,954$ detik, dan $t_r = 0,954$ detik.
Kata Kunci : *Unmanned Aerial Vehicle (UAV)*, *Self Balancing Dual Motor Propeller* , *Fuzzy Logic Controller*

Abstract

Unmanned Aerial Vehicle (UAV) is a unmanned aircraft that can be controlled remotely so it does not need a pilot to fly it. Twincopter / twinrotor is one type of UAV that is being developed lately which has a two motor drive system with propellers. In its application, the on / off system is still not able to balance the twin motor properly if it has an unbalanced and balanced load. Therefore another control method is needed to balance the system from self balancing on dual motor propellers automatically. The purpose of this study was to create and design a self balancing control on a dual motor propeller based fuzzy logic controller to balance when not having the same load of draw. This study uses MPU-6050 hardware as a sensor, Arduino as a microcontroller and two dc brushless motors as actuators. The results of this study indicate that the self balancing control system on a dual motor propeller based on fuzzy logic controller can balance when not having an unbalanced load. By implementing fuzzy control on the plant can produce E_{ss} dynamic response analysis = 0.362, maximum deviation = 24%, $t_d = 1.013$ seconds, $t_p = 1.501$ seconds, $t_s (5\%) = 0.954$ seconds, and $t_r = 0.954$ seconds.

Keywords: Unmanned Aerial Vehicle (UAV), Self Balancing In Dual Motor Propeller, Fuzzy Logic Controller.

PENDAHULUAN

Pesawat Tanpa Awak (*Unmanned Aerial Vehicle, UAV*) merupakan pesawat tanpa awak yang dapat dikendalikan dari jarak jauh sehingga tidak membutuhkan pilot untuk menerbangkannya. Akhir-akhir ini UAV menjadi sebuah tren baru pada bidang militer maupun non militer seperti, pencarian dan penyelamatan korban bencana alam, mendeteksi aktivitas gunung berapi, dan masih banyak penggunaan UAV yang lainnya. (Alvarado, 2014). *Twincopter/twinrotor* merupakan salah satu jenis *multicopter* yang sedang dikembangkan akhir-akhir ini

serta memiliki sistem penggerak dua buah motor beserta baling baling yang terletak disamping kanan dan kiri dari badan pesawat.

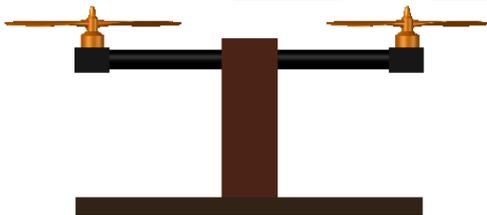
Twincopter/twinrotor memiliki sistem *Vertical Take Of and Landing (VTOL)* dalam sikap terbangnya, sehingga tidak memerlukan landasan pacu yang panjang untuk mengudara (Agarwal, Shlok dkk 2013). Terdapat berbagai macam sistem kendali yang dapat digunakan pada *self balancing* pada *dual motor propeller* salah satunya sistem kendali *on/off*. Pada penerapannya, sistem kendali *on/off*

masih belum mampu menyeimbangkan twinrotor dengan baik jika memiliki beban yang tidak seimbang maupun seimbang (Masahiko dkk : 2012).

Oleh karena itu dibutuhkan metode kontrol lain untuk menyeimbangkan sistem dari *self balancing* pada *dual motor propeller*. Pada penelitian yang penulis buat ini menggunakan sistem kendali *fuzzy* untuk menyeimbangkan *self balancing* pada *dual motor propeller*. Tujuan penelitian ini adalah membuat dan merancang pengendalian *self balancing* pada *dual motor propeller* menggunakan *fuzzy logic controller* untuk menyeimbangkan *dual motor propeller* disaat tidak memiliki beban yang sama imbang. Penelitian ini menggunakan *hardware* MPU-6050 sebagai sensor, arduino sebagai mikrokontroler dan dua motor *brushless dc* sebagai aktuator. Berdasarkan permasalahan tersebut penulis mengambil judul "Rancang bangun sistem pengendali *self balancing* pada *dual motor propeller* berbasis *fuzzy logic controller*".

Pemodelan Dinamik Dual Motor Propeller

UAV *Dual motor propeller* 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 terapkan melalui mekanisme dan struktur lain. (Agarwal,Shlok dkk 2013).



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

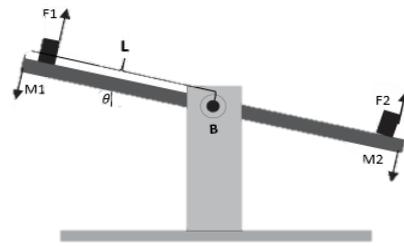
Model Plant

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. M1 dan M2 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 (Hossealizde,2016)

Sekarang persamaan sistem nonlinier adalah :

$$\begin{aligned} \dot{x}_1 &= x_2 = f_1(t) \\ \dot{x}_2 &= \ddot{\theta} = \frac{1}{j}(F_1L - F_2L - m_1Lg\cos\theta + m_2Lg\cos\theta - B\dot{\theta}) = f_2(t) \end{aligned} \quad (3)$$

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

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

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} = J_x(0) = \begin{bmatrix} 0 & 1 \\ \frac{Lg\sin(0)}{j} & \frac{-B}{j} \end{bmatrix} \quad (5)$$

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

Sehingga dari transfer fungsi sistem diatas didapatkan :

$$G(s) = \frac{L}{Js^2 + Bs} = \frac{\frac{L}{j}}{s^2 + \frac{Bs}{j}} \quad (6)$$

Keterangan

$G(s)$ = Gain Sistem (°)

$x_1 = \theta$ = 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)

$\frac{\partial f_1}{\partial F}$ = Deferensial gaya dorong motor 1

$\frac{\partial f_2}{\partial F}$ = Deferensial gaya dorong motor 2

$\frac{\partial x_1}{\partial \theta}$ = Deferensial posisi sudut 1

$\frac{\partial x_2}{\partial \omega}$ = Deferensial posisi sudut 2

$\frac{\partial F}{\partial \theta}$ = Deferensial percepatan

L = Panjang Lengan (Cm)

$M1$ = massa ujung kiri (kg)

$M2$ = massa ujung kanan (kg)

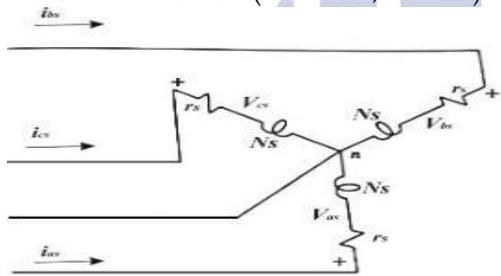
B = Koefisien Redaman (g/mL)

J = Moment Inersia (Kg/m²)

Motor Brushless DC

Motor *brushless* DC telah digunakan secara luas untuk kebutuhan rumah tangga, otomotif, medis, maupun industri. Berbeda dengan motor DC, motor *brushless* DC tidak menggunakan brush (sikat) sebagai media komutasinya melainkan terkompresi secara elektrik. Motor *brushless* DC memiliki beberapa keuntungan dibandingkan dengan motor DC atau motor induksi diantaranya memiliki keandalan dan efisiensi yang tinggi, noise akustik yang rendah, dan memiliki range kecepatan yang tergolong tinggi. (Jian : 2014)

Terdapat dua tipe dari motor *brushless* DC: tipe *trapezoidal* dan tipe *sinusoidal*. Pada *trapezoidal*, back-emf yang diinduksikan pada belitan stator mempunyai bentuk trapesium dan tiap fasanya disuplai oleh arus dengan bentuk quasi-square untuk operasi torsi yang bebas *ripple*. Bentuk dari back-emf ditentukan oleh bentuk magnet rotor dan distribusi belitan stator. (Azzumar, 2012 : 6)



Gambar 3 Rangkaian Ekuivalen *Brushless* DC Motor
(Sumber : Azzumar, 2012)

Pengendali Fuzzy

Untuk merancang pengendali *fuzzy* seperti kendali P, PD, PI atau PID, maka kita harus memilih keadaan proses dan variabel keluaran proses. Menentukan pula isi dari aturan anteseden dan aturan konsekuen. Variabel keadaan proses mewakili isi aturan anteseden yaitu sinyal kesalahan atau *error*, dilambangkan *e*. Perubahan *error*, dilambangkan dengan Δe . Jumlah kesalahan atau integral *error*, dilambangkan Σe . Variabel keluaran kendali mewakili isi aturan konsekuen yaitu perubahan keluaran kendali, dilambangkan Δu . keluaran kendali, dilambangkan *u*. Jika dianalogikan dengan kendali konvensional variabel tersebut menjadi persamaan :

$$e(t) = y_{sp} - y(t) \tag{7}$$

$$\Delta e(t) = e(t) - e(t - 1) \tag{8}$$

$$\Delta u(t) = u(t) - u(t - 1) \tag{9}$$

Pada persamaan diatas, y_{sp} menyatakan nilai keluaran yang diinginkan yaitu sudut atau masukan, *y* adalah nilai variabel keluaran proses. *k* menunjukkan waktu saat ini (Kusumadewi dan Purnomo, 2010).

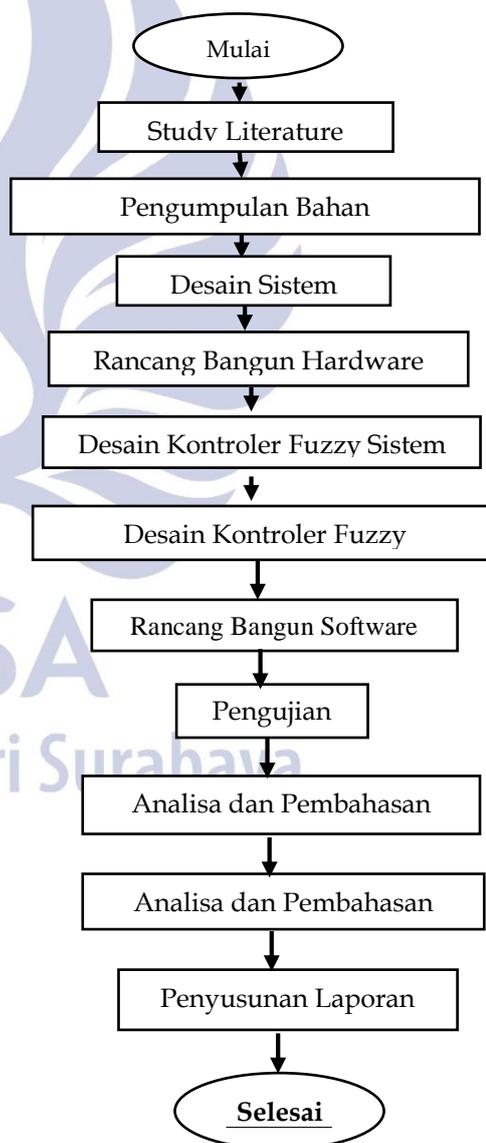
Pendekatan Penelitian

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* MATLAB 2009b dan Arduino IDE 1.8.5. *Software* MATLAB 2009b digunakan untuk menjalankan simulasi kendali Fuzzy yang akan diterapkan pada *plant*. Sedangkan *software* Arduino IDE digunakan untuk memprogram *hardware plant*.

Rancangan Penelitian

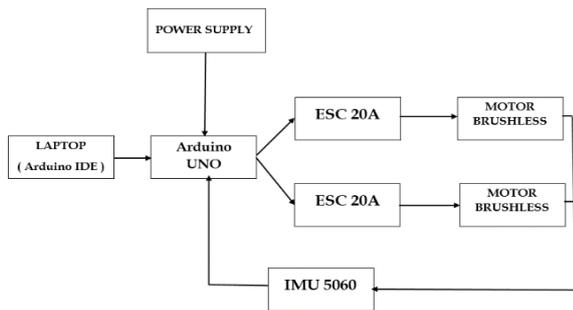
Urutan rancangan penelitian ditunjukkan pada Gambar 4



Gambar 4 Tahapan rancangan penelitian
(Sumber : Dokumen Pribadi, 2019)

Desain Sistem

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

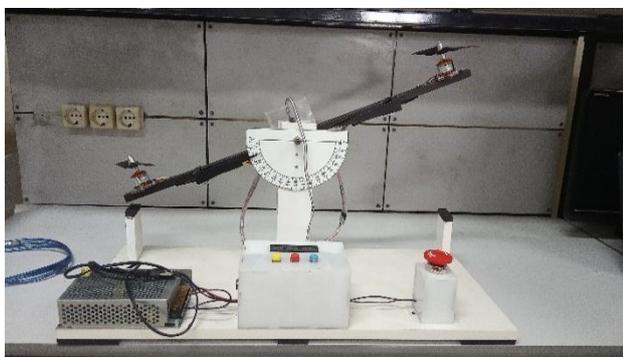


Gambar 5 Desain Sistem Self Balancing pada Dual Motor Proppeler (Sumber : Dokumen Pribadi, 2019)

Blok sistem kendali dari perangkat *hardware balancing dual motor proppeler*. Nilai masukan (*Setpoint*) sistem berupa posisi sudut. *Fuzzy Logic Controller*, 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 IMU-6050 akan bekerja untuk mengukur nilai dari derajat kemiringan *plant* setelah adanya gerakan akibat perputaran motor dan proppeler kemudian menjadikan nilai pembacaan sudut sebagai nilai umpan balik.

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 6. adalah desain rancangan *plant Self Balancing Dual Motor Proppeler* :



Gambar 6 Desain Mekanik *Self Balancing Dual Motor Proppeler* yang digunakan untuk penelitian (Sumber : Dokumen Pribadi, 2019)

Desain Kontroler Fuzzy

Untuk merancang konfigurasi pengendali *fuzzy* yang sesuai dengan *plant* maka diimplementasikan *fuzzy logic controller Self Balancing Pada Dual Motor Propeller* dengan metode Mamdani.

Fuzzification

Sisitem kendali dirancang menggunakan dua masukan. Masukan yang pertama berupa nilai *Error* dimana nilai tersebut merupakan selisih antara nilai pembacaan dari sensor IMU 6050 dengan nilai keluaran. sedangkan *Delta error* yang merupakan perubahan *error* yang didapat dari turunan pertama besaran *error*. Kedua masukan akan diproses oleh kendali *fuzzy logic controller*.

Fungsi keanggotaan dari metode *fuzzy* menggunakan representasi kurva segitiga dan trapesium terbagi atas 5 himpunan *fuzzy* yaitu *Large Negative (LN)*, *Small Negative (SN)*, *Zero (Z)*, *Small Positive (SP)* dan *Large Positive (LP)* merupakan *variable* masukan *error* dan *delta error*. Fungsi keanggotaan untuk variabel masukan yaitu *error* dan *delta error* memiliki bentuk yang sama. Variabel keluaran memiliki 5 himpunan *fuzzy* memiliki fungsi keanggotaan segitiga dan trapesium dengan nama domain yang sama seperti masukan. Semesta pembicaraan adalah nilai *PWM*.

Rule Base

Rule base atau basis aturan yang digunakan adalah fungsi implikasi “*IF THEN*” dengan metode implikasi “*MIN*”. Jumlah aturan yang dihasilkan sesuai dengan jumlah himpunan *fuzzy* yang digunakan pada setiap variabel kemudian hasil dari setiap aturan diintegrasikan menggunakan metode *Max*.

Defuzzification

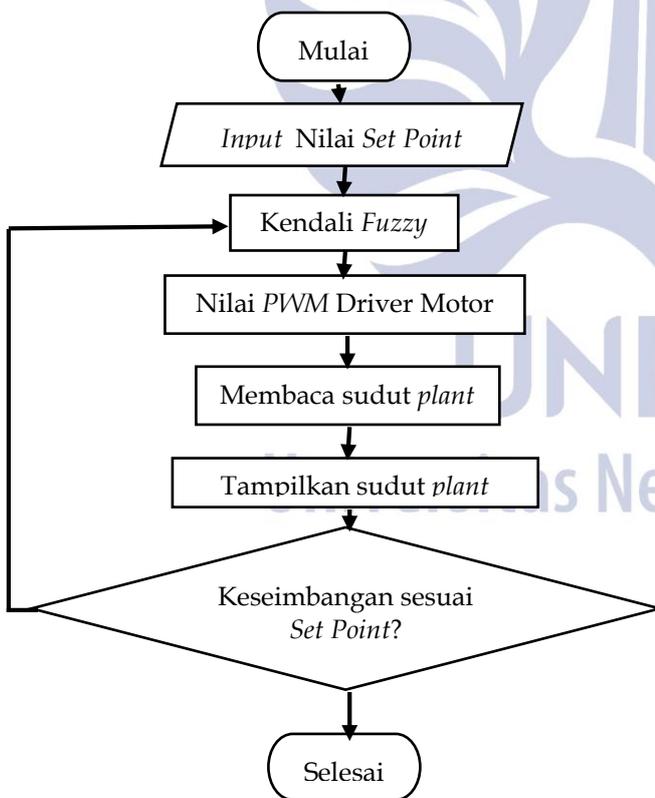
Suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan *fuzzy*, sedangkan keluaran yang dihasilkan berupa bilangan pada domain himpunan *fuzzy* merupakan masukan dari proses *defuzzification*. Metode *defuzzification* yang digunakan adalah metode *Centroid* yaitu mengambil titik pusat dari daerah *Fuzzy* yang telah diinferensi.

Rancang Bangun Software

Sebelum merancang bangun *software* MATLAB 2009b maka harus membuat desain algoritma dari sistem. Desain algoritma tersebut selanjutnya akan digunakan untuk membuat diagram alir dari kerja *software* MATLAB 2009b. Berikut adalah desain algoritma untuk simulasi dengan *software* MATLAB 2009b : 1) *Input* spesifikasi *hardware* dari *Dual Motor Propeller* yaitu daya dorong motor propeller, massa motor dan jarak antara titik tengah ke ujung lengan. 2) *Input Transfer function*. 3) *Input set point* (posisi). 3) *Input* parameter *fuzzy* pada fungsi alih *self balancing Dual Motor Proppeler* dengan simulink pada Matlab 2009b. 4) *Output* dari sistem adalah posisi pada *plant Dual Motor proppeler*. 5) Menampilkan respon berupa posisi dari pengendali *Fuzzy* menggunakan *Scope* pada Matlab 2009b. Apabila *plant Dual Motor Proppeler* mampu menyeimbangkan diri antara motor kanan dan kiri atau mampu mempertahankan posisinya sesuai *set point* 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 ulang parameter parameter dari kendali *Fuzzy Logic Controller* yaitu semesta pembicara, variabel *fuzzy* dan himpunan *Fuzzy*.

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 diluar *developer* Arduino IDE sendiri.

Berikut adalah rancangan diagram alir dari perancangan *software* Arduino IDE. Perangkat lunak arduino akan memproses data masukan yang telah didapat dengan metode *fuzzy logic controller*. Pembacaan sudut kemiringan *plant* oleh sensor MPU 6050 serta data dari sensor akan dikirim ke kontroler. Kemudian arduino akan mengirimkan nilai berupa PWM pada driver motor untuk mengatur kecepatan motor *brushless* dengan *propeller* supaya sudut kemiringan *plant* sesuai dengan set point yang telah di tentukan menggunakan kaidah kaidah pengendali *fuzzy* mulai dari *fuzzification*, *rule based*, dan *defuzzification*. Apabila sudut kemiringan *plant* sesuai dengan nilai set point, maka kecepatan motor akan tetap serta mempertahankan sudut kemiringan tersebut, akan tetapi jika tidak sesuai, maka akan terbaca sebagai nilai *error* dan *delta error* yang akan menjadi acuan dari pengendali *Fuzzy* sehingga mengubah kembali kecepatan motor *brushless* dengan *propeller*. Proses tersebut akan di ulang ulang hingga sudut kemiringan *plant* sesuai dengan set point yang telah ditentukan.



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

HASIL DAN PEMBAHASAN

Pengujian dan analisis meliputi ; 1) Rancangan rangkaian *hardware*, 2) pengujian pada pengimplementasian konfigurasi pengendali *fuzzy* dilakukan dengan memberikan gangguan pada salah satu lengan *plant*, 3) menganalisa hasil pengujian pengendali *fuzzy* apakah mampu mempertahankan keseimbangan pada sistem.

1. Pengujian Sensor MPU 6050

Sensor MPU 6050 dihubungkan ke arduino. Sensor MPU 6050. Sensor MPU 6050 memiliki rentang pembacaan *gyroscope* +250 500 1000 2000°/s dan rentang *accelerometer* ± 2 ± 4 ± 8 ± 16 g (*datasheet*). Sensor dikalibrasi dengan membandingkan sudut pembacaan yang terdeteksi dengan busur derajat.

Busur yang digunakan adalah busur yang memiliki rentang 0° - 180°. Pada pengujian ini peneliti mengambil 5 sampel sudut yang akan dibandingkan yaitu -20°, -10°, 0°, 10° dan 20°.

Tabel 1 Hasil Uji Sensor MPU 6050

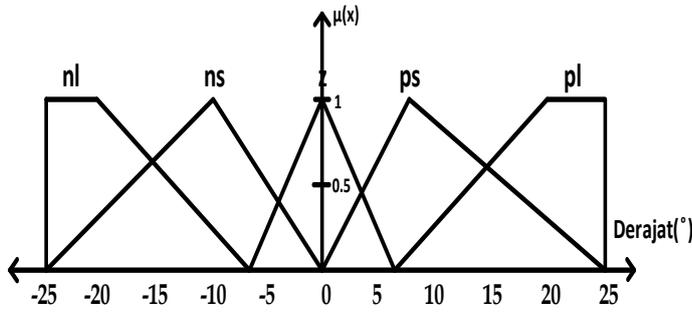
No	Busur Derajat (°)	Sensor MPU 6050 (°)	Error (%)
1	-20	-20,20	0,20
2	-10	-10,25	0,25
3	0	-0,10	0,10
4	10	10,09	0,09
5	20	20,59	0,59
Rata-rata			0,24

(Sumber : Dokumen pribadi, 2019)

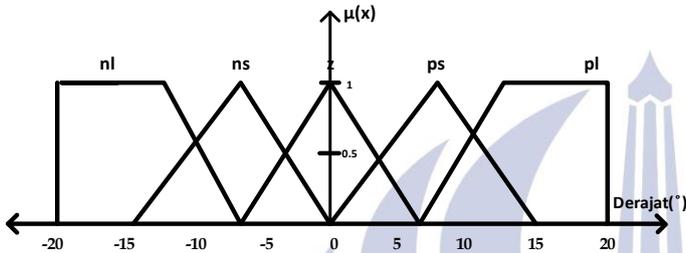
Perancangan Pengendali Logika Fuzzy

Perancangan pengendali logika *fuzzy* didasarkan pada pengetahuan dan pengalaman peneliti terhadap sistem. Untuk dapat merancang pengendali logika *fuzzy*, hal-hal yang harus diperhatikan adalah karakteristik sistem, masukan sistem, dan keluaran sistem.

Terdapat 3 variabel, yaitu 2 variabel *input error* dan *delta_error*, sedangkan untuk output terdapat 1 variabel yaitu variabel *pwm*. Variabel *error*, *delta_error* dan *pwm* memiliki 5 nilai linguistik, yaitu *nl(negative long)*, *ns(negative small)*, *z(zero)*, *ps(positive small)*, *pl(positive long)*. memiliki nilai rentang semesta pembicaraan pada masukan sistem yang digunakan adalah berupa sudut -25° sampai 25° untuk *Error* dan -20° hingga 20° untuk *Delta Error*. Digunakan representasi segitiga, dan trapesium untuk memperoleh derajat keanggotaan kedua masukan tersebut.



Gambar 9 Fungsi Keanggotaan Masukan Error (Sumber : Dokumen Pribadi, 2019)



Gambar 10 Fungsi Keanggotaan Masukan Delta Error (Sumber : Dokumen Pribadi, 2019)

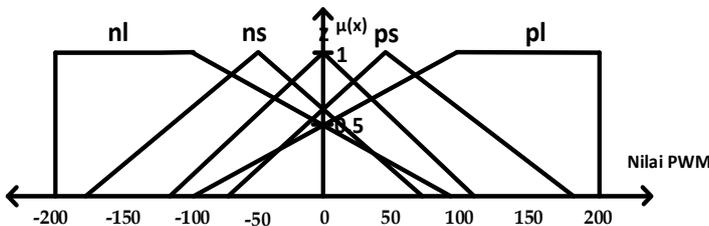
Dengan menggunakan *rule base* atau basis aturan *IF-THEN* metode implikasi *MIN* yang kemudian hasil dari setiap aturan *fuzzy* diintegrasikan menggunakan metode *MAX*, maka matrik *rule base* yang didapatkan adalah seperti yang ditunjukkan pada Tabel 2

Tabel 2 Matrik Rule Base

e/de	nl	ns	z	ps	pl
nl	nl	nl	ns	ns	z
ns	nl	nl	ns	z	z
z	ns	ns	z	ps	ps
ps	z	z	ps	pl	pl
pl	z	ps	ps	pl	pl

(Sumber : Dokumen Pribadi, 2019)

Keluaran dari sistem *Self Balancing* pada *Dual Motor Propeller* ini berupa nilai *pwm* untuk memutar kedua motor dengan rentang nilai semesta pembicaraannya -200 hingga 200.



Gambar 11 Fungsi Keanggotaan Keluaran (Sumber : Dokumen Pribadi, 2019)

Permodelan Dinamik Plant

Identifikasi *plant* digunakan untuk mencari fungsi alih sehingga dapat diketahui respon dari *plant* tanpa menjalankan alatnya. Identifikasi *plant* menggunakan pemodelan dinamik dengan mencari nilai dari parameter fisik *plant*. Parameter fisik diketahui berdasarkan hukum Newton dengan menggunakan persamaan gaya dan torsi yang diterapkan pada sistem. Didapatkan parameter fisik yang ditunjukkan Tabel 3

Tabel 3 Parameter fisik plant

Parameter	Simbol	Nilai
Kefisien Redaman	B	1
Momen inersia	j	0.33
Setengah panjang lengan	L	3.15

(Sumber : Dokumen Pribadi, 2019)

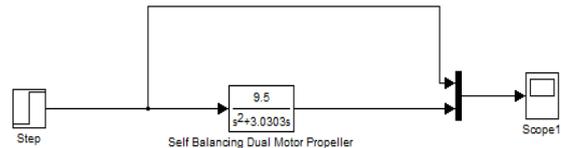
Berdasarkan Tabel 3 dapat dicari fungsi alih dari sistem *Self Balancing* pada *Dual Motor Propeller* dengan mensubstitusikan nilai yang telah didapat dengan persamaan (5) kedalam persamaan (10)

$$G(s) = \frac{\frac{L}{j}}{s^2 + \frac{Bs}{j}} = \frac{\frac{3.15}{0.33}}{s^2 + \frac{1s}{0.33}} = \frac{9.5}{s^2 + 3.0303s} \quad (10)$$

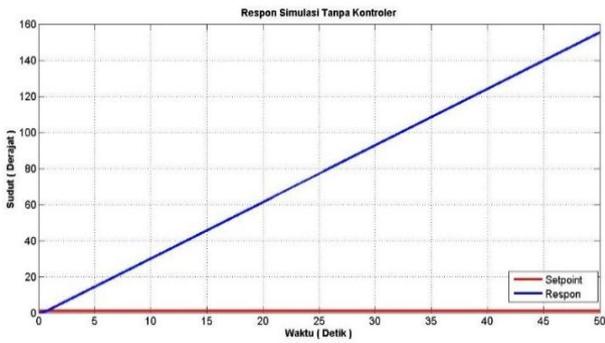
Dari persamaan (10), telah didapatkan fungsi alih *plant*. Kemudian permodelan fisik yang telah didapatkan disimulasikan menggunakan *software* SIMULINK /MATLAB untuk melihat respon dari *plant* tanpa kontroler dan kontroler pengendali logika *fuzzy*.

Pengujian Respon Sistem dengan Pengendali Logika Fuzzy

1. Simulasi Tanpa Kontroler



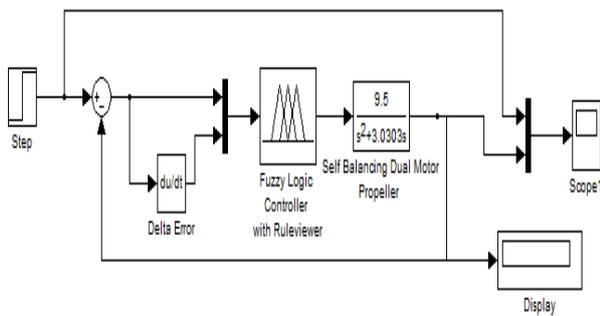
Gambar 12 Diagram blok simulasi sistem lup terbuka tanpa kontroler (Sumber : Dokumen Pribadi, 2019)



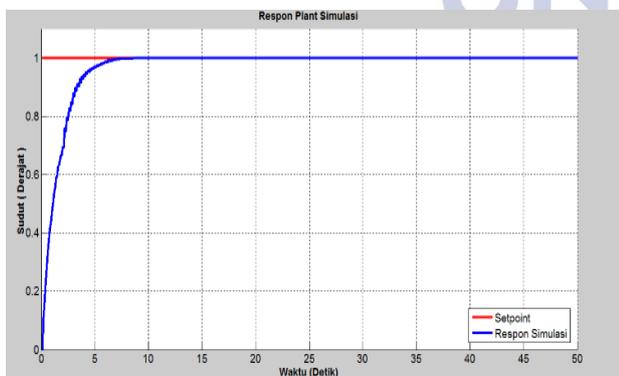
Gambar 13 Hasil respon simulasi tanpa kontroler (Sumber : Dokumen Pribadi, 2019)

Pada Gambar 13 terlihat bahwa respon sistem dari *self balancing* pada *dual motor propeller* tidak bisa mengikuti set point yang diberikan. Respon sistem *self balancing* pada *dual motor propeller* langsung menuju nilai tak hingga dan ini membuktikan bahwa sistem dari *self balancing* pada *dual motor propeller* tidak bisa mempertahankan posisinya agar seimbang. Dari respon sistem yang tidak bisa mempertahankan posisinya sesuai set point.

2. Simulasi Kontroler Pengendali Logika Fuzzy



Gambar 14 Diagram blok sistem lup tertutup dengan logika fuzzy (Sumber : Dokumen Pribadi, 2019)



Gambar 15 Hasil respon simulasi dengan logika fuzzy (Sumber : Dokumen Pribadi, 2019)

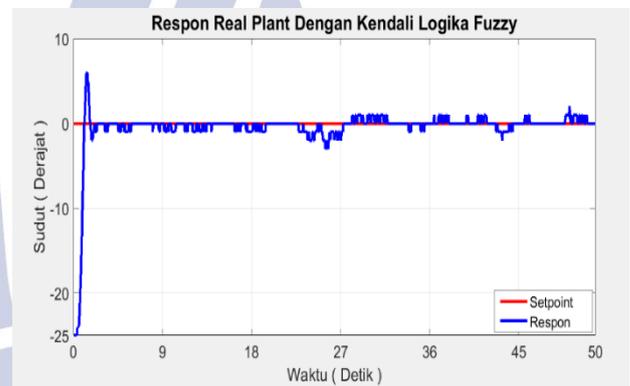
Tabel 4 Respon dinamik simulasi dengan kontroler logika fuzzy

Parameter	Simulasi
M_p (simpangan maksimum)	-
E_{ss} (error steady state)	0.000121 s
t_p (waktu simpangan maksimum)	-
t_s (waktu tunak) 5 %	4.631 s
t_r (waktu naik)	4.545 s
t_d (waktu tunda)	1.070 s

(Sumber : Dokumen Pribadi, 2019)

Pengujian Respon Sistem Pada Plant Nyata dengan Pengendali Logika Fuzzy

1. Pengujian Real Plant Sistem Self Balancing Pada Dual Motor Propeller Berbasis Kendali Logika Fuzzy



Gambar 16 Grafik Respon Real Plant Sistem Self Balancing Pada Dual Motor Propeller Berbasis Kendali Logika Fuzzy (Sumber : Dokumen Pribadi, 2019)

Tabel 5 Respon dinamik sistem pada pengujian real plant sistem self balancing pada dual motor propeller berbasis kendali logika fuzzy

Parameter	Real Plant	Satuan
M_p (Simpangan Maksimum)	24	%
E_{ss} (Error Steady State)	0.362	Derajat
t_p (waktu M_p)	1.501	Detik
t_s (waktu tunak) 5%	3.155	Detik
t_r (waktu naik)	0.954	Detik
t_d (waktu tunda)	1.013	Detik

(Sumber : Dokumen Pribadi, 2019)

2. Pengujian Real Plant Sistem Self Balancing Pada Dual Motor Propeller Berbasis Kendali Logika Fuzzy Dengan Gangguan

1. Model 1

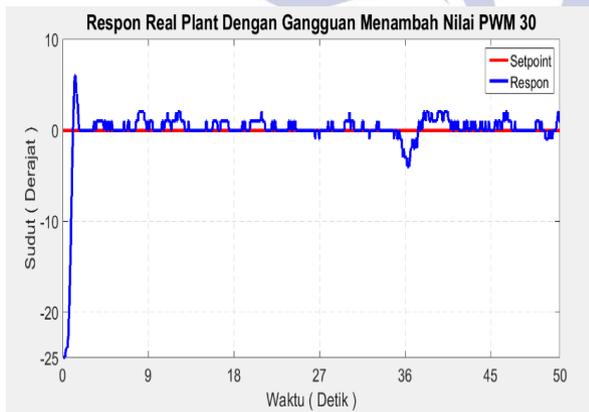
Respon pengujian real plant sistem *self balancing* pada *dual motor propeller* berbasis kendali logika fuzzy pada model pertama dengan memberikan gangguan mengurangi nilai pwm 10



Gambar 17 Grafik Respon *Real Plant* Sistem *Self Balancing* Pada *Dual Motor Propeller* Berbasis Pengendali Logika Fuzzy Dengan Gangguan Mengurangi Nilai Pwm 10 Pada Salah Satu Motor (Sumber : Dokumen Pribadi, 2019)

2. Model 2

Respon pengujian *real plant* sistem *self balancing* pada *dual motor propeller* berbasis kendali logika fuzzy pada model kedua dengan memberikan gangguan menambah nilai pwm 30



Gambar 18 Grafik Respon *Real Plant* Sistem *Self Balancing* Pada *Dual Motor Propeller* Berbasis Pengendali Logika Fuzzy Dengan Gangguan Menambah Nilai Pwm 30 Pada Salah Satu Motor (Sumber : Dokumen Pribadi, 2019)

3. Model 3

Respon pengujian real plant sistem *self balancing* pada *dual motor propeller* berbasis kendali logika fuzzy pada model kedua dengan memberikan gangguan menambah nilai pwm 50



Gambar 19 Grafik Respon *Real Plant* Sistem *Self Balancing* Pada *Dual Motor Propeller* Berbasis Pengendali Logika Fuzzy Dengan Gangguan Menambah Nilai Pwm 50 Pada Salah Satu Motor (Sumber : Dokumen Pribadi, 2019)

Tabel 6 Respon dinamik sistem pada pengujian *real plant* sistem *self balancing* pada dual motor propeller berbasis *fuzzy logic controller* dengan gangguan

Parameter	Model 1	Model 2	Model 3	Satuan
M_p (Simpangan Maksimum)	24.44	26	36,4	%
E_{ss} (Error Steady State)	0.433	0.503	0.892	Derajat
t_p (waktu M_p)	1.43	1.417	1.447	Detik
t_s (waktu tunak 5%)	2.962	3.15	4.291	Detik
t_r (waktu naik)	0.906	0.891	0.866	Detik
t_d (waktu tunda)	1.006	0.964	1.039	Detik

(Sumber : Dokumen Pribadi, 2019)

Berdasarkan Tabel 6 hasil analisis respon dinamik sistem pada pengujian *real plant* sistem *self balancing* pada *dual motor propeller* dapat diketahui bahwa sistem dapat kembali mengikuti nilai masukan dengan cepat setelah diberi gangguan. Selain itu, Tabel 6 menyajikan analisis pada model pertama yaitu memberi gangguan dengan mengurangi nilai pwm 10 pada salah satu motor memiliki waktu tunda (t_d) 1.006 detik, waktu naik (t_r) 0.906 detik, waktu tunak (t_s) 5% 2.962 detik, simpangan maximum sebesar 24.44 %, nilai *error steady state* (E_{ss}) sebesar 0,433 derajat serta memiliki respon yang stabil dan *error steady state* (E_{ss}) paling kecil dari pada model kedua dan ketiga. Model ke kedua dengan memberi gangguan menambah nilai pwm 30 pada salah satu motor membutuhkan waktu tunda (t_d) 0.964 detik, waktu naik (t_r) 0.891 detik, waktu tunak (t_s) 5% 3.15 detik, simpangan maximum sebesar 26 %, nilai eror steady state (E_{ss}) sebesar 0.503 derajat dan pada pengujian model kedua memiliki respon yang paling cepat daripada model pertama

dan ketiga. Sedangkan model ketiga dengan gangguan yaitu menambah nilai pwm 50 pada salah satu motor memiliki waktu tunda (t_d) 1.039 detik, waktu naik (t_r) 0.866 detik, waktu tunak (t_s) 5% 4.291 detik, simpangan maximum sebesar 36.4 % akan tetapi pada pengujian model ketiga memiliki nilai eror steady state (Ess) yang sangat besar yaitu sebesar 0.892 derajat.

PENUTUP

Simpulan

Sistem *self balancing* pada *dual motor propeller* berbasis *fuzzy logic controller* berhasil dibuat dan bekerja dengan baik. Penerapan pengendali logika *fuzzy* yang telah dirancang memiliki aturan basis 5x5 menghasilkan respon yang mampu menyeimbangkan sistem *self balancing* pada *dual motor propeller*. Merujuk pada tabel 5 dan tabel 6 nilai kesalahan untuk mencapai keadaan tunak (Ess) paling kecil pengujian *real plant* sistem *self balancing* pada *dual motor propeller* berbasis kendali logika *fuzzy* yaitu 0.365 sedangkan nilai kesalahan untuk mencapai keadaan tunak (Ess) pada pengujian *real plant* saat diberi gangguan paling besar pada saat merubah nilai pwm pada salah satu motor pada model ketiga yaitu sebesar 0.892, nilai kesalahan untuk mencapai keadaan tunak (Ess) paling kecil pada model pertama yaitu 0.433 serta pada model pertama juga memiliki respon yang paling stabil saat pengujian dengan gangguan.

Saran

Menggunakan metode pengendalian yang lain seperti PID, PID-Fuzzy, LQR dan lain sebagainya untuk mendapatkan kesalahan *error* yang lebih rendah. Menggunakan jenis sensor *gyroscope* dan *accelerometer* yang lebih akurat agar mendapatkan pembacaan yang lebih baik sehingga *error* yang dihasilkan lebih rendah seperti DT-Sense 3 Axis *Gyroscope*, kamera, dan lain lain. Dapat menambahkan pengendalian posisi elevasi pada *plant*. *Plant* dapat dikembangkan lebih lanjut dengan menambah jumlah motor. *Plant* dapat dikembangkan lebih lanjut agar dapat terbang. Untuk menggunakan kendali *fuzzy*, peneliti menyarankan untuk menggunakan mikrokontroler yang memiliki spesifikasi *clock* yang lebih tinggi dari arduino karena untuk mengimplementasikan algoritma *fuzzy* membutuhkan banyak perhitungan sehingga proses eksekusi program lebih lama.

DAFTAR PUSTAKA

- Agarwal, Shlok dkk. 2013. *Design and fabrication of twinmotor UAV*. DOI : 10.5121/csit.2013.3830
- Alvarado, Martínez, dkk. 2014. "Performance Evaluation Of Propeller-Engine Systems For Unmanned Aerial Vehicles In Hovering Mode". *Instituto Politécnico Nacional Grupo de Robótica y Manufactura Avanzada*.

Azzumar, Muhammad.2012. " *Permodelan dan Simulasi BLDC Motor Kecil untuk Aplikasi Aktuator Sirip Roket* " Depok: Universitas Indonesia

Hosseaialixzade, dkk. 2016. *Design and Implementation Classical, State Feedback and Fuzzy Controller on Twin Rotor System. International Conference on Control, Instrumentation, and Automation. Qazvin Islamic Azad University, Qazvin, Iran*

Jian, Z., Yangwei, Y. 2014, "Brushless DC Motor Fundamental Application Node" MPS Module.

Kusumadewi, Sri, dan Hari Purnomo. 2010. "Aplikasi Logika Fuzzy untuk Pendukung Keputusan Edisi 2" Yogyakarta: Graha Ilmu.

Masahiko dkk, 2012, "Resonance Analysis of the UAV Rotor-arm part", The University of Kitakyushu, Dept. of Mechanical Systems Engineering, IOSR Journal of Engineering volume 2 issue 8