

Kontrol Keseimbangan Robot *Hexapod* EILERO menggunakan *Fuzzy Logic*

IWAN KURNIANTO WIBOWO, DANY PREISTIAN, FERNANDO ARDILLA

Departemen Informatika dan Komputer, Politeknik Elektronika Negeri Surabaya,
Indonesia
Email: eone@pens.ac.id

Received 19 Januari 2021 | *Revised* 8 Februari 2021 | *Accepted* 15 Maret 2021

ABSTRAK

Penelitian dengan topik robot hexapod telah banyak dikembangkan, namun sampai saat ini masih sedikit yang mengulas tentang kontrol keseimbangannya. Permasalahan yang kerap muncul adalah ketika robot berada dalam bidang miring, robot dapat terjatuh jika robot tidak dapat menyeimbangkan badan. Begitu pula dengan robot hexapod EILERO yang telah kami bangun. Untuk mengatasi permasalahan itu, selain pemodelan kinematik dan kinematika terbalik yang tepat, juga diperlukan suatu sistem keseimbangan yang baik. Dalam penelitian ini, kami menggunakan fuzzy logic untuk mengontrol keseimbangan robot EILERO dengan umpan balik data kemiringan dari sebuah sensor IMU. Setelah melalui beberapa pengujian yang komprehensif, didapatkan hasil bahwa robot dapat menyeimbangkan diri pada kondisi kemiringan papan pijakan antara -15° dan 15° pada orientasi kemiringan roll dan pitch. Robot mampu merespon dengan capaian steady state di bawah 3000 ms. Dengan demikian, robot EILERO semakin stabil dalam melintasi bidang yang tidak datar.

Kata kunci: *hexapod, EILERO, kinematika terbalik, fuzzy logic*

ABSTRACT

Research on the topic of the hexapod robot has been developed a lot, but until now there is little that has been discussed about balance control. The problem that often arises is that when the robot is on an inclined plane, the robot can fall if the robot cannot balance its body. Likewise with the EILERO hexapod robot that we have built. To solve this problem, besides proper kinematic modeling and inverse kinematic modeling, a good balance system is also needed. In this study, we used fuzzy logic to control the balance of the EILERO robot, with tilt data feedback from an IMU sensor. After going through several comprehensive tests, the results show that the robot can balance itself on the slope of the stepboards between -15° and 15° in the orientation of roll and pitch tilt. The robot is able to respond with steady state achievements below 3000 ms. Thus, the EILERO robot is increasingly stable in traversing uneven planes.

Keywords: *hexapod, EILERO, inverse kinematic, fuzzy logic*

1. PENDAHULUAN

Banyak penelitian tentang sistem keseimbangan (*balancing*) robot. Sistem keseimbangan robot yang sering diulas adalah keseimbangan robot humanoid dan robot-robot beroda, padahal untuk menghadapi bidang alas yang memiliki kemiringan, robot *hexapod* akan jauh lebih baik dalam menyeimbangkan tubuhnya (**Zak & Rozman, 2015**). Seperti pada penelitian (**Huang, dkk, 2017**), dibangun robot humanoid kecil dan sepeda dengan ukuran yang sebanding. Seperti pengendara manusia, robot ini dirancang untuk mengayuh, menyeimbangkan, dan mengarahkan sepeda (**Suebsomran, 2012**). Penelitian ini secara khusus fokus pada desain sistem kontrol untuk robot untuk menyeimbangkan dan mengarahkan sepeda menggunakan stang. Dijelaskan dalam sebuah studi kasus (**Tedeschi & Carbone, 2014**) untuk menunjukkan keefektifan dan kelayakan prosedur desain robot berkaki yang diusulkan secara khusus mempertimbangkan fitur-fitur utama, seperti struktur mekanis dan konfigurasi kaki, sistem penggerak, muatan, kondisi gerakan, dan gaya berjalan.

Penelitian (**Nasution, dkk, 2019**) melakukan penelitian tentang kontrol keseimbangan menggunakan robot EILERO versi empat kaki (*quadruped*). Robot *quadruped* memiliki kelincahan yang lebih baik daripada robot *hexapod*. Namun kurang baik dalam menghadapi kemiringan lantai jika dibandingkan dengan robot *hexapod*. (**De Silva & Sitte, 2012**) merancang simulasi dinamis dari robot *hexapod* 18 derajat kebebasan dengan tujuan mengembangkan algoritma kontrol untuk berjalan mulus, efisien dan kuat di medan yang tidak beraturan. Mereka menggunakan sensor gaya sebagai tambahan dari sensor sudut sambungan konvensional sebagai proprioceptor. Gaya reaksi di kaki robot memberikan informasi yang diperlukan tentang interaksi robot dengan medan. Studi (**Zhang, dkk, 2017**) memperkenalkan sistem penginderaan gaya kaki yang rumit untuk robot *hexapod* yang sama untuk keenam kakinya. Pertama, tata letak dan konfigurasi sistem penginderaan dirancang sesuai dengan struktur dan ukuran kaki. Kedua, sensor torsi sendi, sensor gaya ujung kaki 3-DOF dan modul pemrosesan informasi gaya, dan parameter kinerja sensor gaya diuji dengan simulasi dan eksperimen.

Pada penelitian (**Sartika, dkk, 2020**) menggunakan sensor *Inertial Measurement Unit (IMU)* untuk mengetahui kemiringan suatu bidang dengan perbaikan data menggunakan metode *least square*. (**Charel, dkk, 2017**) mengusulkan estimasi posisi kemiringan pada sebuah robot menggunakan Kalman Filter berdasarkan dua pembacaan sensor IMU untuk mempercepat respons sensor dan mendapatkan data yang akurat selama periode yang lebih singkat. Sensor IMU dapat digabungkan dengan sensor-sensor lain untuk mendapatkan performa yang lebih baik seperti pada penelitian (**Romadon, dkk, 2019**).

Menurut (**Syamsudin, dkk, 2018**), jika dibandingkan dengan sistem konvensional, sistem *fuzzy* memiliki beberapa kelebihan, karena banyaknya aturan yang dapat digunakan. Sistem *fuzzy* mengadopsi kemampuan penalaran dimiliki oleh manusia (**Zadeh, 2012**). Hal ini dikarenakan sistem *fuzzy* memiliki kemampuan untuk memberikan respon berdasarkan informasi yang bersifat kualitatif, tidak akurat, dan ambigu. (**Juang, dkk, 2015**) menerapkan *fuzzy logic* untuk mengontrol robot *hexapod* dalam mengikuti dinding. Sensor yang digunakan untuk mendeteksi dinding adalah sensor *infrared*. Performa belok dan jalan lurus sangat bagus. (**Hidayati, dkk, 2017**) merancang sebuah robot pemadam api *hexapod* berdasarkan *Fuzzy Mamdani Logic*. Navigasi robot *hexapod* pemadam api berbasis deteksi jarak dengan sensor ultrasonik yang menentukan pergerakan kaki robot menggunakan *fuzzy logic*. Robot *hexapod* pemadam api akan menjelajahi arena terowongan yang memiliki beberapa ruangan.

Oleh karena itu pada penelitian ini berfokus untuk mewujudkan suatu robot *hexapod* yang memiliki sistem keseimbangan pada segala sudut kemiringan bidang alas dengan cara

mengontrol semua kaki dan badan robot *hexapod* untuk menyesuaikan robot tetap tegak. Peneliti menggunakan model persamaan kinematika pada robot *hexapod* seperti yang dilakukan oleh (Roy, dkk, 2020), dengan mengacu pada suatu parameter Denavit-Hartenberg. Selanjutnya ditanamkan kontrol *fuzzy logic* untuk memaksimalkan respon robot untuk menyesuaikan diri terhadap bidang alas. Pemodelan kinematika terbalik dan *fuzzy logic* dapat diolah pada sebuah mikrokontroler. Untuk mengetahui respon robot dalam menyeimbangkan diri, data kemiringan badan robot akan ditampilkan pada komputer, sehingga akan mudah untuk dianalisis

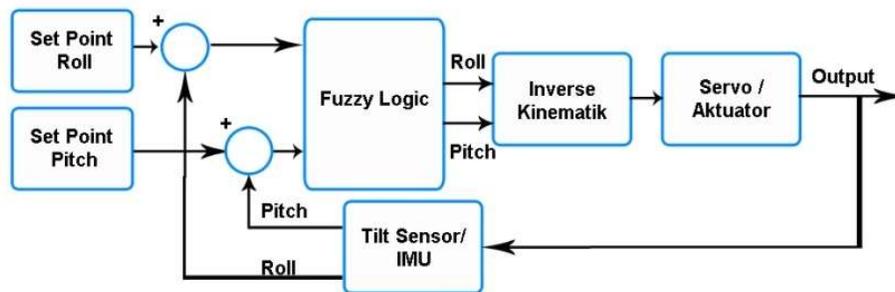
2. METODE PENELITIAN

Robot yang digunakan pada penelitian ini adalah robot *hexapod* EILERO dengan 3-DOF setiap kakinya, sehingga menggunakan 18 motor *servo* sebagai *joint* kaki robot. Desain robot EILERO ditunjukkan pada Gambar 1.



Gambar 1. Desain Robot EILERO

Untuk mewujudkan sistem keseimbangan robot *hexapod*, robot dilengkapi dengan dua sensor yang terdapat pada IMU CHR UM6 yaitu *accelerometer* dan *gyroscope*. Kedua sensor tersebut akan dikombinasikan dengan metode *fuzzy logic* untuk mengontrol pergerakan robot. *Output* dari *fuzzy logic* akan diolah dengan perhitungan matematika dari permodelan kinematika terbalik agar diperoleh nilai PWM untuk masing-masing DOF. Nilai-nilai PWM tersebut yang akan digunakan untuk menggerakkan *actuator* berupa motor *servo* yang terdapat pada tiap kaki.

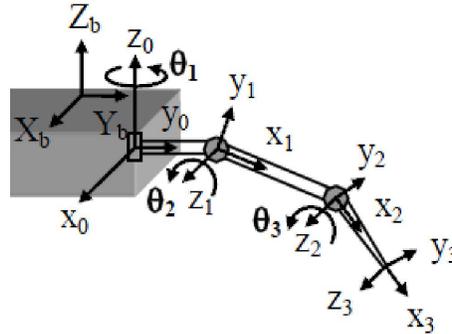


Gambar 2. Blok Diagram Sistem

Gambar 2 menunjukkan blok diagram sistem yang telah dirancang. *Set point* berupa sudut data *roll* dan *pitch*. Sistem kontrol yang digunakan adalah *fuzzy logic*, dimana *input fuzzy* adalah *error roll*, *last error roll*, *error pitch*, dan *last error pitch*. Untuk *plan* pada penelitian ini adalah kinematika terbalik yang menghasilkan data sudut untuk diteruskan ke berupa motor *servo*. Diberikan dua *feedback* dari sebuah sensor kemiringan (IMU) berupa data sudut *pitch* dan *roll*.

2.1 Model Kinematika Maju (Forward Kinematic)

Gambar 3 merupakan model geometri dari kaki robot. $Z_0, Y_0,$ dan X_0 adalah *base frame* dari kaki, sedangkan $Z_b, Y_b,$ dan X_b adalah titik pusat robot yang berada tepat di tengah badan robot yang merupakan referensi global. Pada titik global inilah sumbu putar dan titik 0 translasi dari robot yang akan di beri input berupa translasi dan rotasi pada tubuh robot nantinya.



Gambar 3. Representasi D-H dari Kaki Robot

Parameter D-H dari kaki robot ditunjukkan pada **Tabel 1**.

Tabel 1. Parameter D-H Kaki Robot

Link	α_i	l_i	θ_i	d_i
1	-90	l_1	θ_1	0
2	0	l_2	θ_2	0
3	0	l_3	θ_3	0

Hasil dari transformasi matrik antara ujung kaki dengan *base frame* Z_0 adalah sebagai berikut:

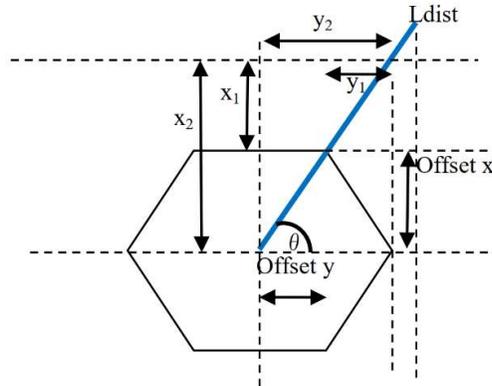
$$T_3^0 = \begin{pmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1(l_1 + l_2 + l_3 C_{23}) \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & S_1(l_1 + l_2 + l_3 C_{23}) \\ S_{23} & C_{23} & 0 & l_3 S_{23} + l_2 S_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

Dari matrik T di atas maka persamaan kinematika maju dari kaki 3-DOF ini adalah:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} C_1(l_1 + l_2 + l_3 C_{23}) \\ S_1(l_1 + l_2 + l_3 C_{23}) \\ l_3 S_{23} + l_2 S_2 \end{pmatrix} \tag{2}$$

2.2 Model Kinematika Terbalik Badan Robot (Body Inverse Kinematic)

Posisi ujung kaki pada awalnya harus kita inialisasi koordinat lokalnya. Untuk melakukan perhitungan *body inverse kinematic* koordinat lokal tersebut harus ditransformasikan ke koordinat global robot. Seperti yang telah dibahas pada permodelan D-H parameter robot. Koordinat lokal yang mempunyai referensi pada Z_0 harus ditransformasikan ke koordinat global yang mempunyai referensi pada titik tengah robot Z_b . Sehingga nilai x dan y global tiap kaki dapat diketahui dengan menggunakan Persamaan (3) dan (4).



Gambar 4. Pemodelan Geometri Badan Robot

$$x_2 = x_1 + offset_x \quad (3)$$

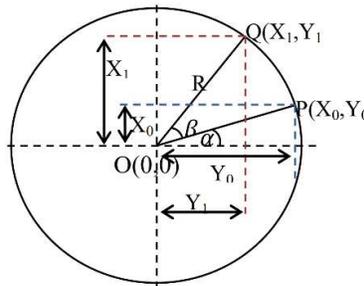
$$y_2 = y_1 + offset_y \quad (4)$$

Jarak antara ujung kaki tiap robot (*end of effector*) dengan titik tengah robot dapat dihitung menggunakan Persamaan (5) dan nilai sudut θ menggunakan Persamaan (6):

$$Ldist = \sqrt{x^2 + y^2} \quad (5)$$

$$\theta = atan\left(\frac{x_2}{y_2}\right) \quad (6)$$

Dengan mengetahui posisi awal badan robot, maka koordinat akhir yang harus dituju oleh setiap kaki dapat dihitung dengan Persamaan (7) – (12).



Gambar 5. Pemodelan Geometri Badan Robot Saat Terkena Rotasi di Bidang Z

$$\tan(\alpha) = \frac{x_0}{y_0} \quad (7)$$

$$\tan(\alpha + \beta) = \frac{x_1}{y_1} \quad (8)$$

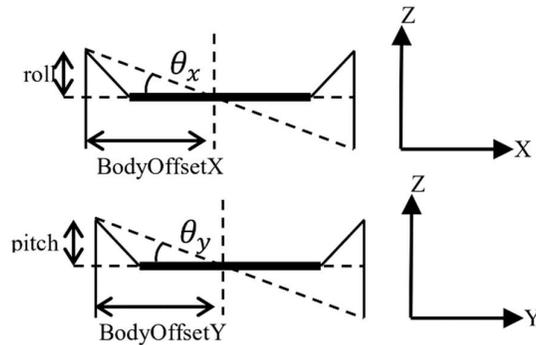
$$x_1 = R \cos(\alpha + \beta) \quad (9)$$

$$y_1 = R \sin(\alpha + \beta) \quad (10)$$

$$BodyIK_x = x_1 - x_0 \quad (11)$$

$$BodyIK_y = y_1 - y_0 \quad (12)$$

Rotasi pada bidang x dan y (*roll* dan *pitch*) menyebabkan nilai koordinat z pada tiap kaki berubah. Gambar 6 menunjukkan pemodelan geometri badan robot Ketika terjadi rotasi di bidang X dan Y.



Gambar 6. Pemodelan Geometri Badan Robot Saat Terkena Rotasi di Bidang X Dan Y

Nilai *roll* dan *pitch* dapat diperoleh dengan Persamaan (13) dan (14). Sedangkan nilai koordinat z dapat diketahui melalui Persamaan (15).

$$roll = BodyOffsetY \tan \theta_y \quad (13)$$

$$pitch = BodyOffsetX \tan \theta_x \quad (14)$$

$$BodyIK_z = roll + pitch \quad (15)$$

2.3 Model Kinematika Terbalik Kaki Robot (*Leg Inverse Kinematic*)

Dengan mengetahui nilai koordinat akhir yang harus dituju setiap kaki, maka *leg inverse kinematic* digunakan untuk membangkitkan sudut-sudut yang harus dibentuk oleh tiap sendi kaki robot agar ujung kaki dapat menuju ke koordinat yang telah diinginkan. Analisis dibagi menjadi 2, yaitu pada bidang x,y dan x,z. **Gambar 6** merupakan permodelan geometri kaki robot. Struktur kaki robot terdiri dari tiga bagian utama yakni *coxa*, *femur*, dan *tibia*. Analisis bidang x dan y digunakan untuk mencari nilai. *Active leg* merupakan bagian kaki yang aktif bergerak melangkah sedangkan *coxa* adalah bagian kaki yang pasif yang merupakan penyambung antara poros dengan *active leg*.

$$\theta_1 = \text{atan} \left(\frac{x_0}{y_0} \right) \quad (16)$$

Analisis bidang z dan *LegLength* digunakan untuk mencari nilai θ_2 dan θ_3 . Melalui penurunan trigonometri segitiga femur, tibia, dan garis bayangan swe. Nilai θ_2 dan θ_3 dapat dihitung dengan Persamaan (17)-(20).

$$swe = \sqrt{z^2 + (LegLength - coxa)^2} \quad (17)$$

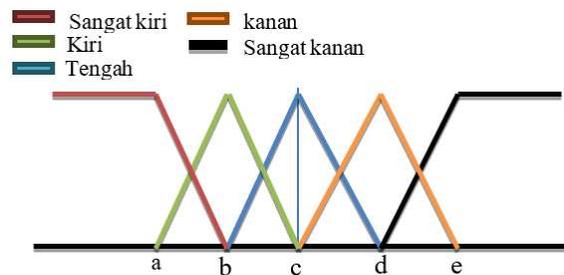
$$\beta = \text{atan} \left(\frac{z}{LegLength - coxa} \right) \quad (18)$$

$$\theta_2 = \text{acos} \left(\frac{tibia^2 - femur^2 - swe^2}{2 femur swe} \right) - \beta \quad (19)$$

$$\theta_3 = \text{acos} \left(\frac{swe^2 - femur^2 + tibia^2}{2 femur tibia} \right) \quad (20)$$

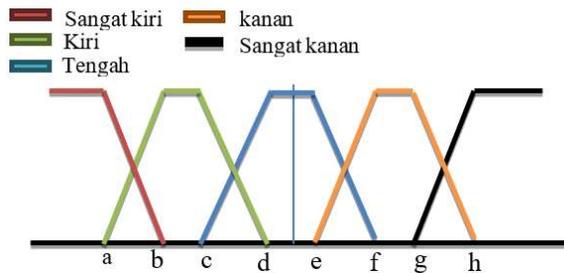
2.4 Sistem Kontrol *Fuzzy*

Sistem keseimbangan robot *hexapod* dikontrol menggunakan *fuzzy logic* dengan parameter berupa data dari sensor *accelerometer* dan *gyroscope* yang menghasilkan nilai *roll* dan *pitch*. *Fuzzy* yang digunakan antara *input roll* dan *pitch* dibedakan sehingga akan menghasilkan dua *output fuzzy* yang nilainya jika ditambahkan menjadi nilai kemiringan robot terhadap *rol* dan *pitch*. Setiap *fuzzy* akan memiliki masukan berupa data sekarang dan sebelumnya. Parameter-parameter ini yang diubah menjadi *fuzzy input* dengan proses *fuzzifikasi*. Kemudian *input-input* tersebut digunakan dalam proses *inference* dengan metode Sugeno untuk selanjutnya dilakukan proses *defuzzifikasi* untuk mendapatkan *output* berupa nilai sudut *roll* dan *pitch*. Fungsi keanggotaan yang digunakan adalah dengan model segitiga yang nilai batas dari masing-masing fungsi keanggotaan akan diubah-ubah untuk mendapatkan sistem kontrol yang terbaik dengan respon yang cepat. Fungsi keanggotaan yang dirancang adalah seperti pada Gambar 7.



Gambar 7. Fungsi Keanggotaan *Fuzzy* Model Segitiga

Fungsi keanggotaan *fuzzy* yang digambarkan pada Gambar 8 menunjukkan bahwa peneliti menggunakan fungsi keanggotaan berbentuk segitiga dan membuat fungsi keanggotaan dengan bentuk trapezium sebagai pembanding.



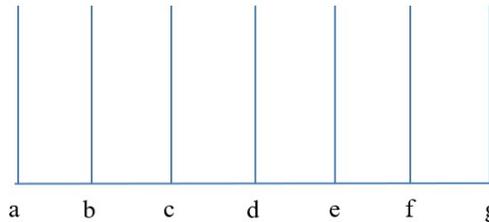
Gambar 8. Fungsi Keanggotaan *Fuzzy* Model Trapezium

Setelah melakukan *fuzzifikasi*, hasil dari proses *fuzzifikasi* akan diproses lagi ke dalam *rule fuzzy*. *Rule fuzzy* akan terbentuk sebanyak 25 kali karena *output* dari *fuzzifikasi* menghasilkan 5 data *error* dan 5 data *last error*. Dalam proses *rule fuzzy*, setiap data pada *fuzzifikasi error* akan dibandingkan dengan setiap data dari *fuzzifikasi last error*. Data dari *rule* dimasukkan ke dalam sebuah tabel untuk kemudian akan diproses ke proses *defuzzifikasi*.

Tabel 2. *Fuzzy Rule Base*

	LS_Kiri	LKiri	LTengah	LKanan	LS_Kanan
S_Kiri	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Kiri	Rule 6	Rule 7	Rule 8	Rule 9	Rule 10
Tengah	Rule 11	Rule 12	Rule 13	Rule 14	Rule 15
Kanan	Rule 16	Rule 17	Rule 18	Rule 19	Rule 20
S_Kanan	Rule 21	Rule 22	Rule 23	Rule 24	Rule 25

Nilai *output fuzzy* akan dihasilkan melalui *defuzzifikasi COA (Center Of Area)*. Untuk itu, peneliti membuat sebuah fungsi keanggotaan baru yang berbentuk garis tegak lurus keatas, seperti yang digambarkan pada **Gambar 9**. Nilai-nilai dari fungsi keanggotaan akan diubah-ubah untuk menguji, nilai mana yang paling cocok untuk kontrol keseimbangan robot *hexapod*.

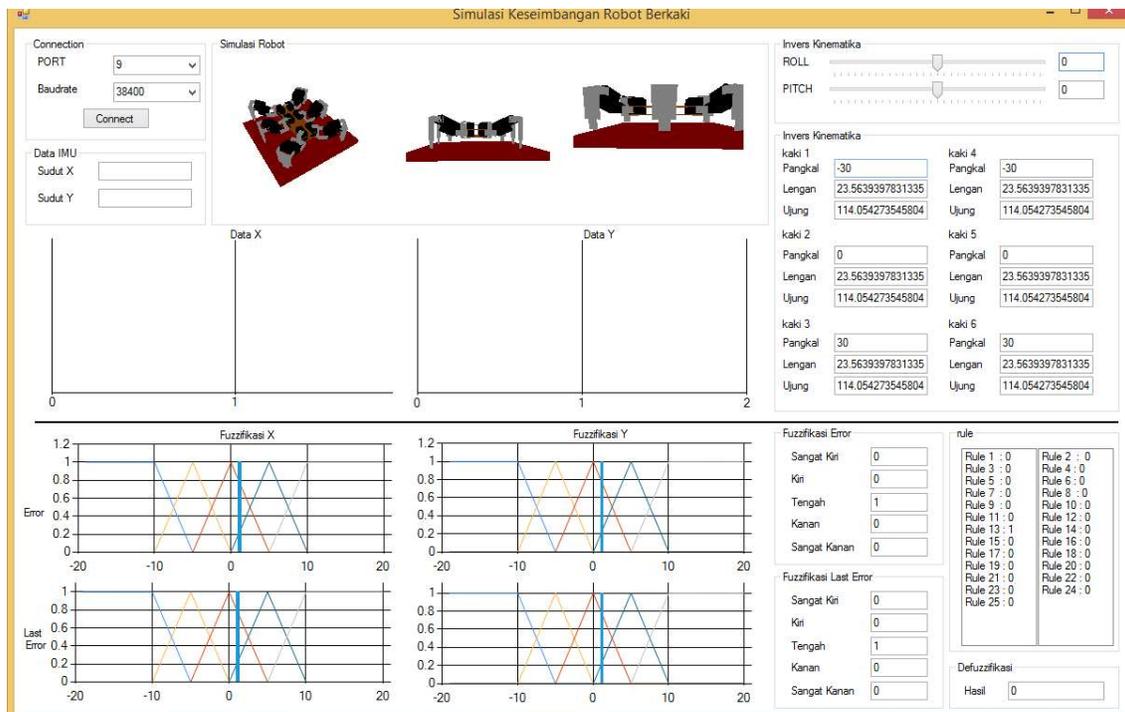


Gambar 9. Fungsi Keanggotaan Fuzzy Output

Keterangan:

- a = pergerakan robot kekiri dengan cepat
- b = pergerakan robot kekiri dengan sedang
- c = pergerakan robot kekiri dengan lambat
- d = tegak, atau tidak bergerak
- e = pergerakan robot kekanan dengan lambat
- f = pergerakan robot kekanan dengan sedang
- g = pergerakan robot kekanan dengan cepat

Gambar 10 merupakan desain yang dirancang untuk memenuhi semua kebutuhan analisis dalam keseimbangan robot *hexapod*. Didesain hanya menggunakan 1 *form* untuk memudahkan pengguna dalam mengoperasikan aplikasi.



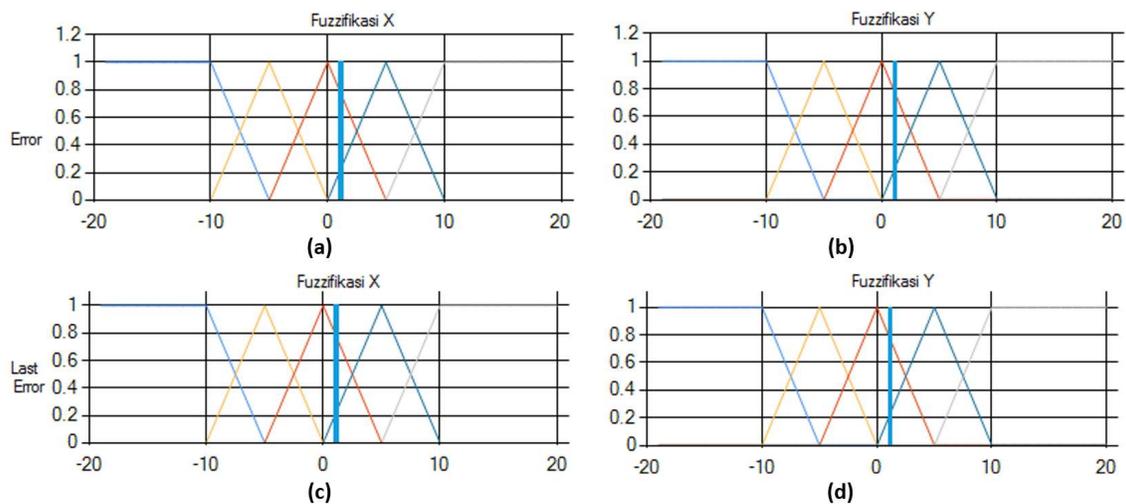
Gambar 10. Rancangan Aplikasi Simulasi Keseimbangan Robot Hexapod EILERO

Sistem kontrol keseimbangan *fuzzy* dibangun pada sebuah PC atau dapat disebut sebagai simulator dan hasil *output* dikirimkan ke *hardware embedded* robot berkaki EILERO menggunakan komunikasi serial melalui perangkat Bluetooth. Beberapa hal penting dapat diterapkan dalam keseimbangan robot *hexapod* yaitu animasi kinematika terbalik robot, nilai-nilai sudut tiap kaki yang dibentuk oleh robot ketika berada dalam posisi sudut kemiringan tertentu, grafik data yang menampilkan data yang diterima komputer berupa data *roll* dan *pitch*, kontrol manual yang menggantikan komunikasi dengan robot, serta fungsi-fungsi yang mencakup sistem kontrol *fuzzy*.

Invers Kinematika					
kaki 1	Pangkal	-30	kaki 4	Pangkal	-30
	Lengan	23.5639397831335		Lengan	23.5639397831335
	Ujung	114.054273545804		Ujung	114.054273545804
kaki 2	Pangkal	0	kaki 5	Pangkal	0
	Lengan	23.5639397831335		Lengan	23.5639397831335
	Ujung	114.054273545804		Ujung	114.054273545804
kaki 3	Pangkal	30	kaki 6	Pangkal	30
	Lengan	23.5639397831335		Lengan	23.5639397831335
	Ujung	114.054273545804		Ujung	114.054273545804

Gambar 11. Data Sudut Tiap *Joint* Kaki pada Aplikasi

Gambar 11 merupakan fungsi yang menampilkan data-data tiap *joint* yang dibutuhkan robot untuk memenuhi sudut kemiringan yang dibentuk dari data sudut *roll* dan *pitch* yang diterima komputer, maupun dari inputan manual *trackbar*. Di dalam fungsi ini sudah terdapat rumus perhitungan kinematika terbalik dengan pengukuran yang sama persis dengan robot, sehingga diharapkan dapat membantu pengguna aplikasi dalam menganalisis robot *hexapod*.



Gambar 12. Grafik Fuzzifikasi Data
(a) Error Roll , (b) Error Pitch, (c) Last Error Roll , dan (d) Last Error Pitch

Gambar 12 adalah grafik data fuzzifikasi *roll* dan *pitch*. Grafik fuzzifikasi *roll* dan *pitch* masing-masing dibagi kedalam 2 grafik yaitu grafik fuzzifikasi *error* dan grafik fuzzifikasi *last error*. Pada Gambar 12, (a) adalah fuzzifikasi *error roll*, (b) adalah fuzzifikasi *error pitch*, (c) adalah fuzzifikasi *last error roll*, dan (d) adalah fuzzifikasi *last error pitch*. Grafik akan menggambarkan data yang masuk, sehingga pengguna akan mudah mengetahui keberadaan input sudut dalam fungsi keanggotaan.

Fuzzifikasi Error		rule	
Sangat Kiri	<input type="text" value="0"/>	Rule 1 : 0	Rule 2 : 0
Kiri	<input type="text" value="0"/>	Rule 3 : 0	Rule 4 : 0
Tengah	<input type="text" value="0"/>	Rule 5 : 0	Rule 6 : 0
Kanan	<input type="text" value="1"/>	Rule 7 : 0	Rule 8 : 0
Sangat Kanan	<input type="text" value="0"/>	Rule 9 : 0	Rule 10 : 0
Fuzzifikasi Last Error		Rule 11 : 0	Rule 12 : 0
Sangat Kiri	<input type="text" value="0"/>	Rule 13 : 0	Rule 14 : 0
Kiri	<input type="text" value="0"/>	Rule 15 : 0	Rule 16 : 0
Tengah	<input type="text" value="0"/>	Rule 17 : 0	Rule 18 : 0
Kanan	<input type="text" value="1"/>	Rule 19 : 1	Rule 20 : 0
Sangat Kanan	<input type="text" value="0"/>	Rule 21 : 0	Rule 22 : 0
		Rule 23 : 0	Rule 24 : 0
		Rule 25 : 0	
		Defuzzifikasi	
		Hasil <input type="text" value="-0.25"/>	

Gambar 13. Proses Perhitungan Fuzzy

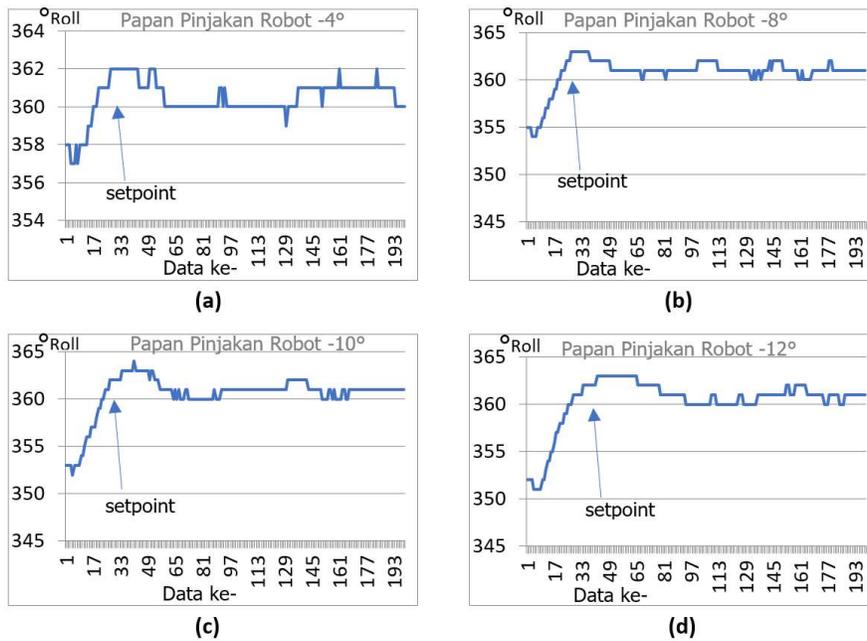
Gambar 13 menunjukkan fungsi yang digunakan untuk menganalisis data proses *fuzzy* mulai dari proses fuzzifikasi, *rule evaluation*, maupun hasil dari defuzzifikasi.

3. HASIL DAN PEMBAHASAN

3.1 Respon Fuzzy Dalam Mengontrol Kesimbangan Axis Roll

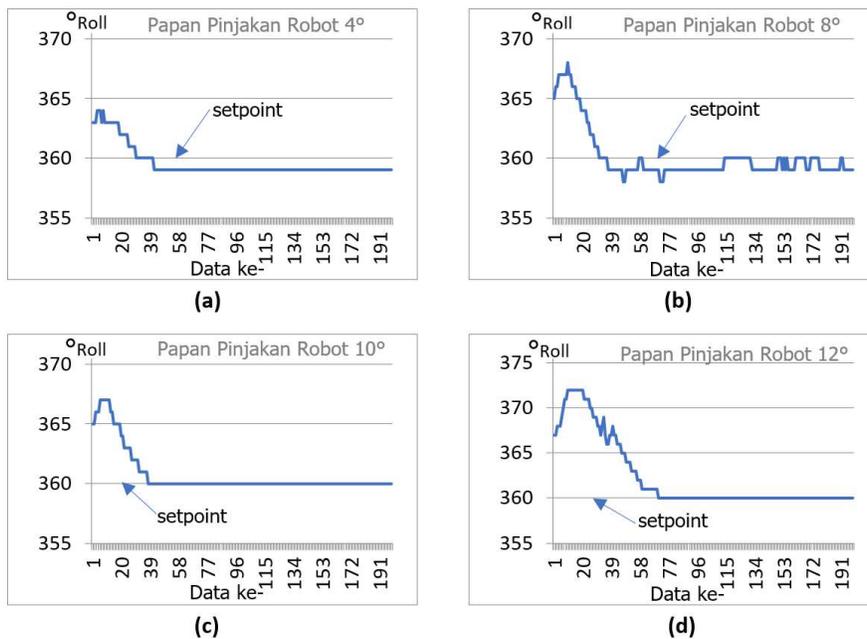
Pengujian respon kontrol *fuzzy logic* terhadap sudut *roll* bertujuan untuk mengetahui data keseimbangan dan respon robot setiap kemiringan 2° . Peneliti mengambil data sebanyak 200 data dalam waktu 10 detik untuk mengetahui respon kontrol *fuzzy* yang ditanamkan pada robot. Pengujian pertama dilakukan dengan mengubah kemiringan papan pijakan robot mulai dari 0° sampai -12° kemudian dilanjutkan dengan sudut 0° sampai 12° . Kemiringan -12° dan 12° adalah batas pergerakan kaki robot yang bisa dilakukan. Robot dianggap seimbang jika kemiringan tubuh robot tegak lurus terhadap arah gaya gravitasi atau dalam penelitian ini mempunyai sudut 0° atau 360° dengan toleransi tertentu.

Perlu diketahui bahwa sensor IMU yang digunakan mempunyai persentase kesalahan rata-rata sudut *roll* 0.46%, dan *pitch* 0.63%. Berdasarkan grafik pada Gambar 14 (a)-(d), kontrol *fuzzy* yang dibangun dapat menyeimbangkan robot ketika diberikan papan pijakan dengan kemiringan *roll* -4° , -8° , -10° , dan -12° . Untuk kemiringan sudut papan pijakan sebesar -4° , robot mampu mencapai *steady state* pada data ke-56 atau sama dengan 2800 ms sejak robot dijalankan. Sedangkan dengan kemiringan papan pijakan sebesar -8° , robot mampu mencapai *steady state* pada data ke-50 atau sama dengan 2500 ms sejak robot dijalankan. Kemiringan papan pijakan ditingkatkan dengan nilai -10° , menghasilkan waktu yang lebih lama untuk mencapai *steady state*, yaitu pada data ke-65 atau sama dengan 3250 ms sejak robot dijalankan. Begitu pula dengan kemiringan papan pijakan -12° , waktu yang dibutuhkan untuk mencapai *steady state* adalah sebesar 4500 ms atau data ke-90. Secara keseluruhan setelah mencapai *steady state*, sistem sedikit berosilasi berkisar 1° - 2° .



Gambar 14. Grafik Respon Keseimbangan Robot terhadap Kemiringan *Roll*/Papan Pijakan (a)- 4° , (b) -8° , (c) -10° , dan (d) -12°

Grafik pada **Gambar 15** (a)-(d) adalah hasil pengujian respon robot ketika diberikan papan pijakan positif sebesar 4° , 8° , 10° , dan 12° . Kontrol *fuzzy* yang dibangun dapat menyeimbangkan robot ketika diberikan papan pijakan dengan kemiringan *roll* positif hingga 12° .

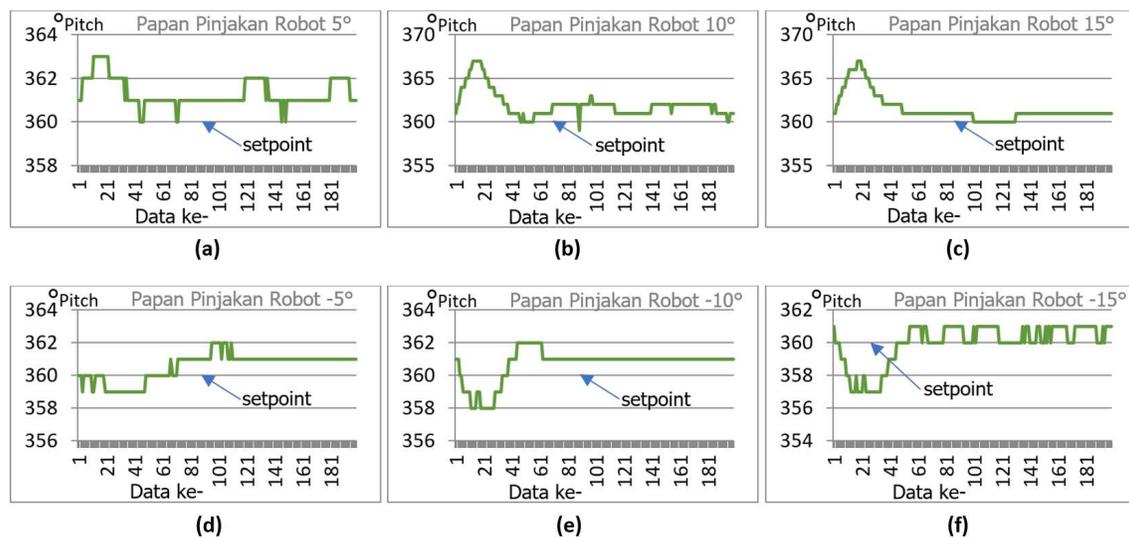


Gambar 15. Grafik Respon Keseimbangan Robot terhadap Kemiringan *Roll*/Papan Pijakan (a) 4° , (b) 8° , (c) 10° , dan (d) 12°

Untuk kemiringan sudut papan pijakan sebesar 4° , robot mampu mencapai *steady state* pada data ke-45 atau sama dengan 2250 ms sejak robot dijalankan. Tidak terjadi osilasi setelahnya. Sedangkan dengan kemiringan papan pijakan sebesar 8° , robot mampu mencapai *steady state* pada data ke-35 atau sama dengan 1750 ms sejak robot dijalankan. Kemiringan papan pijakan ditingkatkan dengan nilai 10° , menghasilkan waktu yang sama untuk mencapai *steady state*, yaitu pada data ke-35 atau sama dengan 1750 ms sejak robot dijalankan. Sedangkan dengan kemiringan papan pijakan -12° , waktu yang dibutuhkan untuk mencapai *steady state* menjadi lebih lama yaitu sebesar 3500 ms atau data ke-70. Sudut kemiringan papan pijakan di atas 4° menghasilkan osilasi sebesar 1° - 2° setelah mencapai *steady state*.

3.2 Respon *Fuzzy* dalam Mengontrol Keseimbangan Axis Pitch

Pengujian respon kontrol *fuzzy logic* terhadap sudut *pitch* bertujuan untuk mengetahui data keseimbangan dan respon robot jika dihadapkan dengan papan pijakan miring pada bagian *pitch*. Peneliti mengambil data sebanyak 200 data untuk mengetahui respon kontrol *fuzzy* yang ditanamkan pada robot.

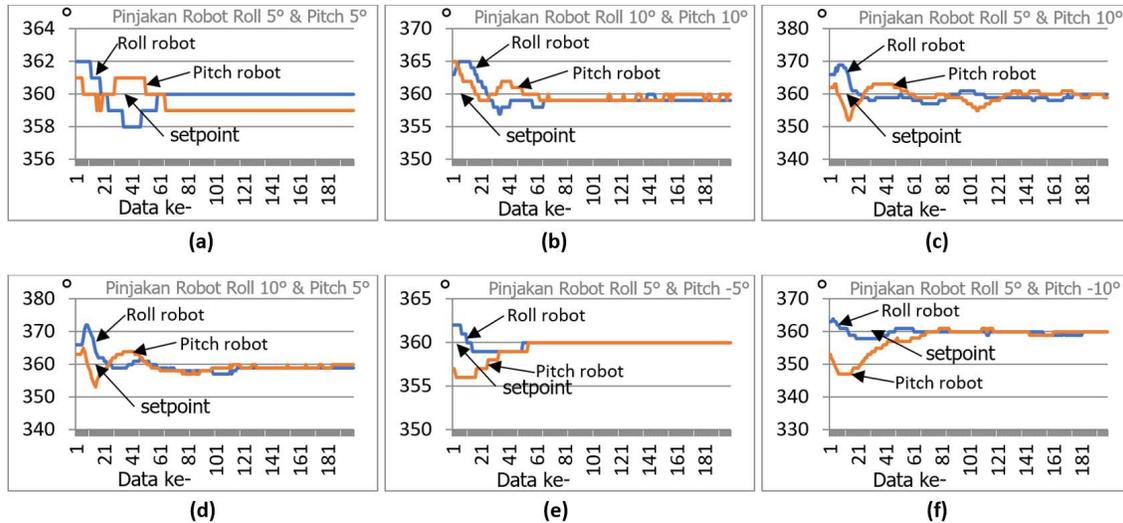


Gambar 16. Grafik Respon Keseimbangan Robot terhadap Kemiringan *Pitch* Papan Pijakan (a) 5° , (b) 10° , (c) 15° , (d) -5° , (e) -10° , dan (f) -15°

Hasil pengujian dapat dilihat pada **Gambar 16** (a)-(f). Pada kemiringan papan pijakan sebesar 5° , robot dapat menyeimbangkan posisi badan atau dalam kondisi *steady state* pada data ke-30 atau setelah 1500 ms. Jika papan pijakan berada pada kemiringan 10° , robot baru dapat mencapai *steady state* pada data ke-40 atau setelah 2000 ms. Sedangkan pada kemiringan papan pijakan sebesar 15° , robot dapat menyeimbangkan posisi badan pada data ke-40 atau setelah 2000 ms. Masih terdapat sedikit osilasi pada kemiringan papan pijakan robot dengan sudut positif, yaitu sebesar maksimal 2° . Robot mampu memberikan hasil yang sangat baik pada kemiringan papan pijakan sebesar -5° . Robot dapat menyeimbangkan posisi badan sesaat setelah robot pertama kali dinyalakan. Namun Ketika diberikan kemiringan -10° pada papan pijakan, robot baru dapat menyeimbangkan posisi badan pada data ke-60 atau setelah 3000 ms. Dengan kemiringan papan pijakan sebesar -15° , robot mampu menyeimbangkan badan pada data ke-50 atau pada 2500 ms.

3.3 Respon *Fuzzy* dalam Mengontrol Keseimbangan Axis *Pitch* dan *Roll*

Sebanyak 200 data diambil untuk mengetahui respon kontrol *fuzzy* terhadap kemiringan papan pijakan pada bidang *pitch* dan *roll*. Hasil pengujian yang telah dilakukan dapat dilihat pada grafik Gambar 17 (a)-(f).



Gambar 17. Grafik Respon Keseimbangan Robot terhadap Kemiringan *Pitch* dan *Roll* Papan Pijakan (a) 5° dan 5°, (b) 10° dan 10°, (c) 5° dan 10°, (d) 10° dan 5°, (e) 5° dan -5°, dan (f) 5° dan -10°

Dengan sudut kemiringan papan pijakan *roll* sebesar 5° dan *pitch* 5°, robot dapat mencapai kondisi seimbang atau *steady state* pada data ke-60 atau 3000 ms sejak robot dijalankan. Ketika diberikan papan pijakan *roll* sebesar 10° dan sudut *pitch* 10°, robot baru mampu stabil pada data ke-70 atau setelah 3500 ms robot dijalankan. Selanjutnya diikuti osilasi data yang tidak begitu berarti. Untuk papan pijakan dengan *roll* sebesar 10° dan sudut *pitch* 5°, robot mampu berada dalam kondisi *steady state* setelah data ke-50 atau 2500 ms. Osilasi yang dihasilkan lebih halus daripada ketika diletakkan pada papan pijakan *roll* sebesar 10° dan sudut *pitch* 10°. Dengan sudut papan pijakan *roll* sebesar 5° sudut *pitch* -5°, kontrol keseimbangan robot dapat menyeimbangkan robot pada data ke-45 atau 2250 ms sejak robot dijalankan. Kemudian robot stabil pada *steady state* dan terdapat *noise* dibeberapa data. Sedangkan ketika dihadapkan pada papan pijakan dengan sudut *roll* sebesar 5° sudut *pitch* -10°, robot seimbang pada data ke-60 atau 3000 ms sejak robot dijalankan.

4. KESIMPULAN

Kontrol *fuzzy logic* yang dibangun untuk keseimbangan robot *hexapod* EILERO adalah kontrol *fuzzy logic* dengan fungsi keanggotaan berbentuk segitiga dengan parameter data *error*, dan *last error* sama yaitu sangat kiri < 10, kiri = 3, tengah = 0, kanan = 3, dan sangat kanan > 10. Dengan parameter ini, respon waktu robot untuk mencapai *steady state* adalah maksimal 3000 ms dimana kondisi kemiringan papan pijakan antara -15° dan 15°. Data *roll* dan *pitch* dari sensor *IMU* mempunyai persentase kesalahan rata-rata sudut *roll* 0.46%, dan *pitch* 0.63%, sehingga dapat dikatakan bahwa sistem yang diterapkan robot sudah cukup baik. Dengan demikian, sistem kontrol *fuzzy* yang dirancang memberikan kontribusi yang cukup bagus untuk menyeimbangkan robot *hexapod* EILERO Ketika dihadapkan pada bidang atau lantai yang tidak datar.

UCAPAN TERIMA KASIH

Kami ucapkan terima kasih kepada para rekan-rekan yang meneliti topik ini selama kurang lebih 3 tahun. Kemudian terima kasih kepada jajaran laboratorium robotika PENS terkait atas semua fasilitas yang diberikan.

DAFTAR RUJUKAN

- Charel, S. E. R., Binugroho, E. H., Rosyidi, M. A., Dewanto, R. S., & Pramadihanto, D. (2016). Kalman filter for angle estimation using dual inertial measurement units on unicycle robot. *2016 International Electronics Symposium (IES)*, (pp. 256–261). <https://doi.org/10.1109/ELECSYM.2016.7861013>
- De Silva, S., & Sitte, J. (2012). Force Controlled Hexapod Walking. In *Advances in Autonomous Mini Robots*, (pp. 265–277). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-27482-4_25
- Hidayati, Q., Rachman, F. Z., & Yanti, N. (2017). Intelligent Control System of Fire-Extinguishing and Obstacle-Avoiding Hexapod Robot. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 1–10. <https://doi.org/10.22219/kinetik.v3i1.470>
- Huang, C.-F., Tung, Y.-C., & Yeh, T.-J. (2017). Balancing control of a robot bicycle with uncertain center of gravity. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 5858–5863). <https://doi.org/10.1109/ICRA.2017.7989689>
- Juang, C.-F., Chen, Y.-H., & Jhan, Y.-H. (2015). Wall-Following Control of a Hexapod Robot Using a Data-Driven Fuzzy Controller Learned Through Differential Evolution. *IEEE Transactions on Industrial Electronics*, 62(1), 611–619. <https://doi.org/10.1109/TIE.2014.2319213>
- Nasution, Z., Ilham Suparman, A. F., Prasetyo, G. A., Alasiry, A. H., Binugroho, E. H., & Darmawan, A. (2019). Body Balancing Control for EILERO Quadruped Robot while Walking on Slope. *2019 International Electronics Symposium (IES)*, (pp. 364–369). <https://doi.org/10.1109/ELECSYM.2019.8901598>
- Romadon, Z. T., Oktavianto, H., Wibowo, I. K., Sena Bayu Dewantara, B., Nurrohmah, E. A., & Adryantoro Priambudi, R. (2019). Pose Estimation on Soccer Robot using Data Fusion from Encoders, Inertial Sensor, and Image Data. *IES 2019 - International Electronics Symposium: The Role of Techno-Intelligence in Creating an Open Energy System Towards Energy Democracy, Proceedings*, (pp. 454–459). <https://doi.org/10.1109/ELECSYM.2019.8901578>
- Roy, S. S., Singh, A. K., & Pratihari, D. K. (2020). Analysis of six-legged walking robots. *14th ELKOMIKA* – 546

- National Conference on Machines and Mechanisms, NaCoMM 2009*, (pp. 259–265).
- Sartika, E. M., Gany, A., & Yuvens, V. (2020). Implementasi Sensor IMU untuk mengetahui Sudut Elevasi Kendaraan menggunakan Metode Least Square. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 8(2), 301. <https://doi.org/10.26760/elkomika.v8i2.301>
- Suebsomran, A. (2012). Balancing control of bicycle robot. *2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, (pp. 69–73). <https://doi.org/10.1109/CYBER.2012.6392529>
- Syamsudin, A., Wibowo, I. K., & Mobed Bachtiar, M. (2018). Improving the Accuracy and Controlling the Shooting Power in a Wheeled Soccer Robot. *2018 International Electronics Symposium on Engineering Technology and Applications, IES-ETA 2018 - Proceedings*, (pp. 100–106). <https://doi.org/10.1109/ELECSYM.2018.8615513>
- Tedeschi, F., & Carbone, G. (2014). Design Issues for Hexapod Walking Robots. *Robotics*, 3(2), 181–206. <https://doi.org/10.3390/robotics3020181>
- Zadeh, L. A. (2012). Fuzzy Logic. In *Fuzzy Logic*. Springer New York. https://doi.org/10.1007/978-1-4614-1800-9_73
- Zak, M., & Rozman, J. (2015). Design, construction and control of hexapod walking robot. *2015 IEEE 13th International Scientific Conference on Informatics*, (pp. 302–307). <https://doi.org/10.1109/Informatics.2015.7377851>
- Zhang, H., Wu, R., Li, C., Zang, X., Zhang, X., Jin, H., & Zhao, J. (2017). A Force-Sensing System on Legs for Biomimetic Hexapod Robots Interacting with Unstructured Terrain. *Sensors*, 17(7), 1514. <https://doi.org/10.3390/s17071514>