

BALANCING ROBOT BERBASIS FUZZY LOGIC

Sumantri K Risandriya, ST, MT⁽¹⁾, Rifqi Amalya Fatekha, S.ST⁽²⁾, Irda
Zusmaniar⁽³⁾

Mechatronics Engineering, Batam Polytechnics
Parkway Street, Batam Centre, Batam 29461, Indonesia
E-mail: sumantri@polibatam.ac.id

Abstrak

Balancing robot adalah berupa robot beroda yang memiliki 2 roda pada sisi kanan dan kiri, robot ini harus dapat menjaga keseimbangan. Untuk dapat menjaga keseimbangan secara maksimal terdapat banyak metode yang akan digunakan untuk menentukan optimasi kendali yang baik sehingga didapatkan nilai respon yang baik sensor dengan reaksi motor. Pada *balancing robot* ini menggunakan metode *Fuzzylogic* dengan *input accelerometer* dan *gyroscope*. Untuk mendapatkan respon *accelerometer* yang bagus tanpa noise menggunakan *KalmanFilter*. Hasil yang didapat Robot dapat menyeimbangkan sendiri dengan tingkat keberhasilan 90% dan robot dapat menyeimbangkan sendiri meskipun mendapat gangguan dengan didorong sekalipun.

Kata kunci : *Fuzzy logic, Kalman Filter, Balancing robot dan Accelerometer*

Abstract

Balancing robot is in the form of a wheeled robot that has two wheels on the right and left, the robot must be able to maintain a balance. In order to maintain the maximum balance there are many methods that will be used to determine good control optimization to obtain the value of a good response sensor with the motor reactions. At this balancing robot using the method Fuzzylogic with input accelerometer and gyroscope. To get a good response without noise accelerometer use KalmanFilter. The results obtained robot can balance itself with a success rate of 90% and a robot can balance itself despite being disturbed by even encouraged.

Keyword : *Fuzzy logic, Kalman Filter, Balancing robot and Accelerometer*

1. Pendahuluan

Balancing Robot adalah Robot yang memiliki 2 Roda yang terdapat pada sisi kanan dan sisi kiri. Robot ini dituntut untuk dapat menyeimbangkan dirinya sendiri tegak lurus dan tidak jatuh saat diberi beban, didukung dengan menggunakan metode *fuzzy logic* untuk dapat menentukan titik keseimbangan pada Robot saat berdiri.

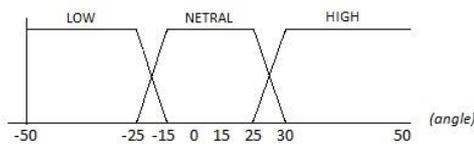
Pada penelitian ini akan dibuat *Balancing Robot* berbasis *Fuzzy logic* dengan pergerakan yang diharapkan akan halus dibandingkan dengan menggunakan metode *PID*. Hal ini dikarenakan pada output *Accelerometer* yang masih terdapat *noises*. *Noise* tersebut akan di *filter* dengan menggunkannya metode *Kalman Filter* sehingga *noise* dari keluaran *Accelerometer* akan hilang.

2. Dasar Teori

2.1 Fuzzy Logic

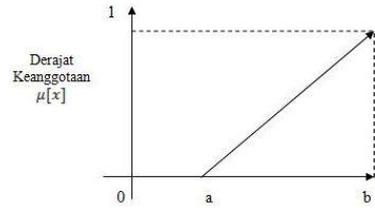
2.2.1 Fuzzifikasi

Fuzzifikasi yaitu suatu proses untuk mengubah suatu masukan dari bentuk tegas (*crisp*) menjadi *fuzzy* (variabel linguistik). Biasanya disajikan dalam bentuk himpunan-himpunan *fuzzy* dengan suatu fungsi keanggotaannya masing-masing, seperti pada gambar 2.1.



Gambar 2.1 Fungsi keanggotaan

Fuzzy *membership* digunakan untuk menghitung derajat keanggotaan suatu himpunan fuzzy. Yang pertama ada representasi linier naik digunakan untuk menghitung pemetaan kenaikan dari 0 (nol) ke 1 (satu), berikut rumus fungsi keanggotaan linier naik : Gambar 2.2 menunjukkan representasi linier naik dari a yang bernilai 0 (nol) menuju b yang bernilai 1 (satu).



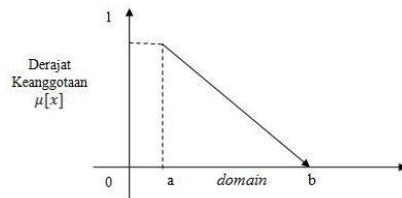
Gambar 2.2 Grafik keanggotaan kurva linear naik

Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x - a)/(b - a); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2.1)$$

Yang kedua ada representasi linier turun digunakan untuk menghitung pemetaan penurunan dari 1 (satu) ke 0 (nol). Berikut rumus fungsi keanggotaan linier turun :

Gambar 2.3 menunjukkan representasi linier turun dari a yang bernilai 1 (satu) menuju b yang bernilai 0 (nol).



Gambar 2.3 Grafik keanggotaan kurva linear turun Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} (b - x)/(b - a); & a \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (2.2)$$

Yang ketiga ada representasi kurva segitiga adalah gabungan dari 2 garis linier (naik dan turun). Berikut rumus fungsi keanggotaan kurva segitiga :

Gambar 2.4 menunjukkan representasi kurva segitiga dari 2 garis linier naik dan linier turun.

2

Gambar 2.4 Grafik keanggotaan kurva segitiga

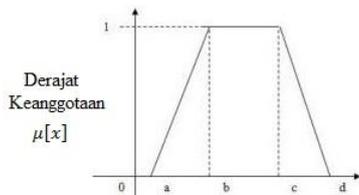
Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x - a)/(b - a); & a \leq x \leq b \\ (c - x)/(c - b); & b \leq x \leq c \end{cases} \quad (2.3)$$

Yang keempat ada representasi kurva trapesium

adalah gabungan dari 2 garis linier (naik dan turun) dan terdapat beberapa titik yang memiliki nilai keanggotaan 1 (satu). Berikut rumus fungsi keanggotaan kurva trapesium :

Gambar 2.4 menunjukkan representasi kurva trapesium dari 2 garis linier naik dan linier turun serta ada beberapa titik yang memiliki nilai keanggotaan 1.



Gambar 2.4 Grafik keanggotaan kurva trapesium

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x - a)/(b - a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d - x)/(d - c); & x \geq d \end{cases} \quad (2.4)$$

Yang kelima ada representasi kurva bentuk bahu adalah daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Himpunan fuzzy bahu, bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah fuzzy, bahu kiri bergerak dari benar ke salah, sedangkan bahu kanan bergerak dari benar ke salah [3]

2.2 Sensor Accelerometer

Accelerometer adalah alat yang digunakan untuk mengukur percepatan, mendeteksi dan mengukur getaran, dan mengukur percepatan akibat gravitasi. *Accelerometer* mengukur percepatan akibat gerakan benda yang melekat padanya. dengan satuan g yaitu unit percepatan yang nilainya sebanding dengan gravitasi bumi diatas permukaan air laut yaitu $9,8\text{m/s}^2$. *Accelerometer* menggunakan hukum Newton kedua tentang pergerakan yang terdapat dalam persamaan 2.5.

$$\vec{F} = M \cdot \vec{a} \quad (2.5)$$

Accelerometer digunakan untuk mengukur getaran pada mobil, mesin, bangunan, dan instalasi pengamanan.

Gravitasi bumi merupakan percepatan yang konstan selalu mengarah lurus menuju inti bumi. Ketika *Accelerometer* sejajar dengan gravitasi pada keadaan diam maka percepatan yang terukur adalah $1g$, ketika *Accelerometer* tegak lurus dengan gravitasi pada keadaan diam maka percepatan sebesar $0g$. Sehingga jika *Accelerometer* pada keadaan diam dapat diukur pada *Accelerometer* 2 sumbu berlaku persamaan 2.6 dan 2.7

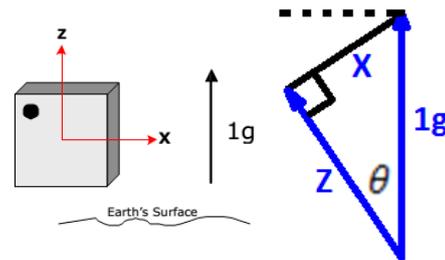
$$\theta = \cos^{-1}\left(\frac{z}{\text{gravitasi}}\right) \quad (2.6)$$

$$\theta = \sin^{-1}\left(\frac{x}{\text{gravitasi}}\right) \quad (2.7)$$

Dimana:

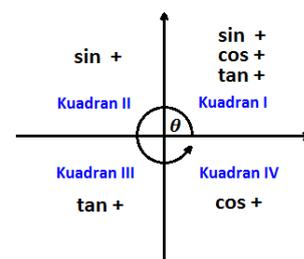
x = nilai percepatan terukur pada sumbu yang tegak lurus dengan *gravitasi*

z = nilai percepatan terukur pada sumbu yang sejajar dengan *gravitasi*



Gambar 2.5 Penempatan sumbu x dan y pada

Accelerometer



Gambar 2.6 Kuadran pada persamaan *trigonometri*

Kelemahan dari persamaan 2.5 dan 2.6 adalah nilai keluaran hanya efektif ketika kemiringan berada pada kuadran ke 1. sehingga untuk menutupi kelemahan perumusan ini adalah dengan menggunakan persamaan 2.8. Dengan keluaran sebesar -180 hingga $+180$. [1]

$$\theta = \text{atan2}^{-1}(z, x) \quad (2.8)$$

Kelemahan teori ini adalah hanya dapat diaplikasikan ketika *Accelerometer* pada keadaan diam, jika *Accelerometer* bergerak maka fungsi asli sebagai pengukur percepatan akan bekerja dan mempengaruhi nilai akhir. Solusi dari permasalahan ini adalah dengan menggunakan *kalman filter*. [4]

Kalman filter adalah teknik perhitungan matematika (algoritma) yang memberikan perhitungan efisien dalam mengestimasi *state* proses. Yaitu dengan cara meminimalkan rata-rata kuadran galat (*Mean Squared Error / MSE*).

Pada kalman filter berfungsi untuk memisahkan sinyal dari sinyal yang lain yang tidak dikehendaki, karena pada faktaknya sinyal yang dikeluarkan *Accelerometer* adalah tidak akurat sehingga dengan digunakannya filter terhadapnya, maka hasil pengukuran akan mendekati hasil sebenarnya (hampir tidak ada *noise*). [5]

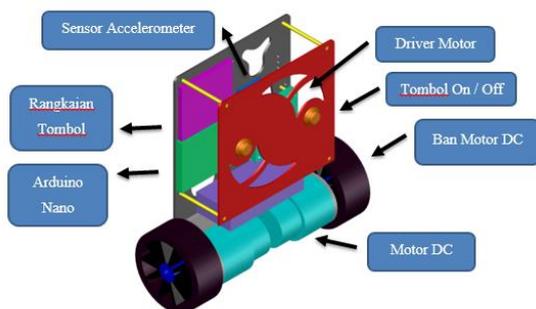
$$\hat{X}_k = K_k \cdot Z_k + (1 - K_k) \hat{X}_{k-1} \quad (2.9)$$

Keterangan :

- \hat{X}_k = estimasi saat ini
- K_k = kalman gain
- Z_k = nilai yang diukur
- \hat{X}_{k-1} = estimasi sebelumnya

3. Perancangan Sistem

3.1 Perancangan Perangkat Keras

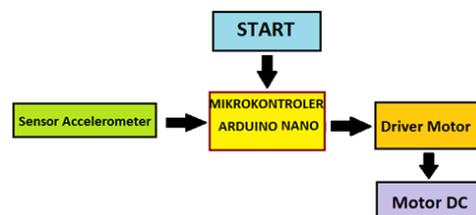


Gambar 3.1. Perencanaan Konstruksi Robot Balancing.



Gambar 3.2 Bentuk fisik *balancing* robot

Gambar 3.1 dan gambar 3.2. menjelaskan konstruksi *balancing* robot. Robot ini terdiri dari 2 motor DC sebagai actuator dan sensor *accelerometer* sebagai acuan robot agar bisa seimbang di tambah dengan kalman *filter* sebagai penghilang *noise* yang dihasilkan pada sensor *accelerometer*.

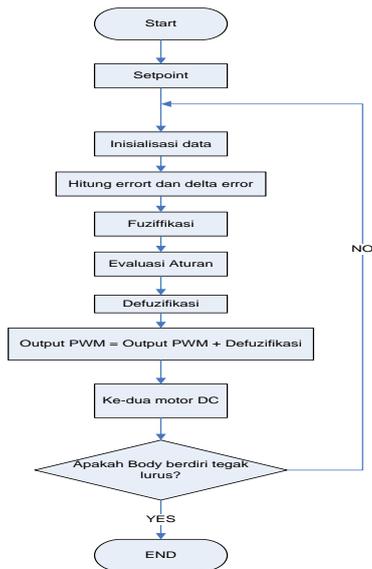


Gambar 3.3 Diagram Blok Robot

Gambar 3.3 menjelaskan system *Balancing* Robot, dimulai dari input pada *switich* *Start* yang digunakan sebagai *switich* yang akan mengaktifkan semua komponen pada robot. Dan ada juga *input* pada *Sensor Accelerometer* yang digunakan sebagai sensor keseimbangan pada robot. Semua *input* ini akan dikirimkan ke *Mikrokontroler* yaitu *Arduino nano*. Lalu *Mikrokontroler* mengirimkan data pada *Driver Motor* untuk menggerakkan *actuator* yaitu Motor DC.

3.2. Perancangan Perangkat Lunak

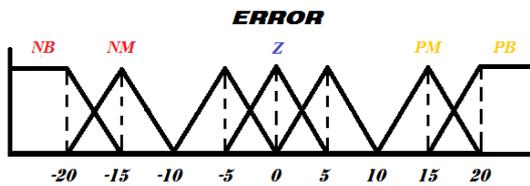
Setelah perancangan perangkat keras telah di selesaikan setelah itu akan dilanjutkan pada pembuata perancangan perangkat lunak untuk mengatur proses kerja alat atau kendali yang akan di lakukan. Perancangan lunak sebagai berikut.



Gambar 3.4 Diagram Alur (*Flow Chart*) prinsip dasar sistem

➤ Rancangan Membership

- Input Membership Function Error

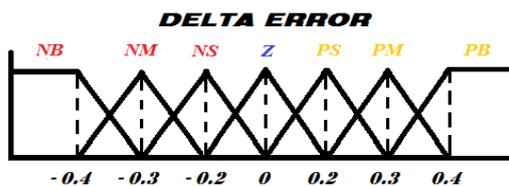


Gambar 3.6 Input Membership Function Error

Keterangan :

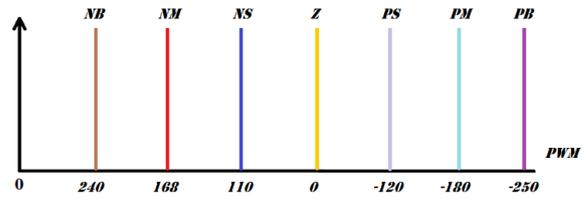
- NB : Negative besar
- NM : Negative besar
- NS : Negative besar
- Z : Zero
- PB : Positive besar
- PM : Positive besar
- PS : Positive besar

- Input keanggotaan *delta error*



Gambar 3.7 Input Membership Functional Delta Error

- Output fungsi keanggotaan *PWM*



Gambar 3.8 Output Membership Fungsional *PWM*

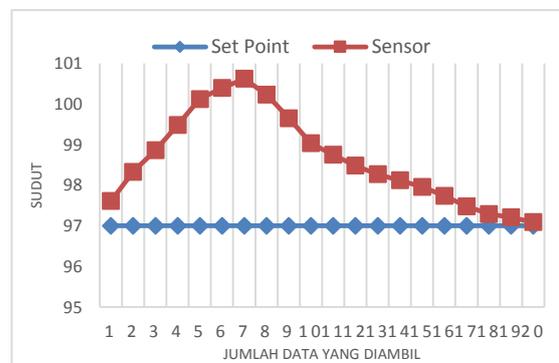
- *Fuzzy Rule*

Tabel 3.1 Rule

ERROR \ DELTA ERROR	NB	NM	NS	Z	PS	PM	PB
NB	PB	PB	PB	PB	PM	PS	Z
NM	PB	PB	PM	PM	PS	Z	Z
NS	PB	PS	PS	PS	PS	Z	Z
Z	PB	PM	PS	Z	NS	NM	NB
PS	Z	Z	NS	NS	NS	NS	NB
PM	Z	Z	NS	NM	NM	NB	NB
PB	Z	NS	NM	NB	NB	NB	NB

4. Hasil dan Analisa

4.1 Pengambilan data pengujian pertama

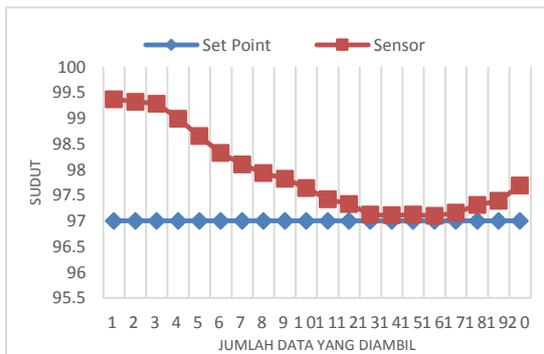


Gambar 4.1 Data Perubahan Sensor pengujian pertama

pada gambar 4.1 ketika *balancing* robot dijalankan didapat perubahan data sensor pada data 100.62 dengan detik ke 7 μ s perubahan sensornya menjauh dari data *set point* dan ketika detik 20 μ s perubahan sensor mendekati data *set point*, pada start awal *balancing* robot memiliki *error* yang besar dan saat mencapai detik 8 μ s *balancing* robot semakin mendekati *set point*. Dari hasil yang di peroleh dilihat pada perbedaan antara setpoint, sensor, *error*, *delta*

error dan *output fuzzy* telah didapat rata – rata sebesar 1.69 %.

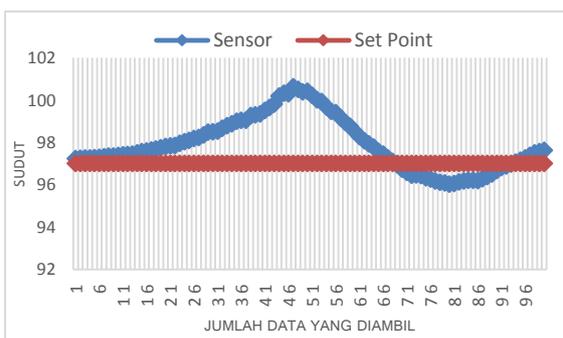
4.2 Pengambilan data pengujian ke-dua



Gambar 4.2 Data Perubahan Sensor pengujian ke-dua

Gambar 4.2. pengujian gerak *balancing* robot dan dapat dilihat pada gambar 4.2 ketika *balancing* robot dijalankan didapat perubahan data sensor pada data 99.37 Dari hasil yang di peroleh dilihat pada perbedaan antara *set point*, *sensor*, *error* , *deltaerror* dan *output fuzzy* telah didapat rata – rata sebesar 0.94 %.

4.3 Pengambilan data pengujian ketika diberi gangguan kearah kiri

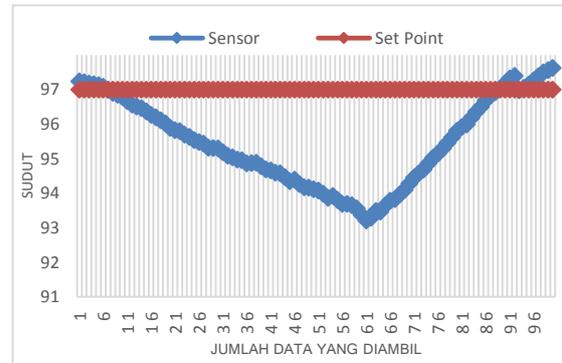


Gambar 4.3 Data perubahan sensor ketika diberi gangguan kearah kiri

Gambar 4.3 merupakan Grafik pengujian gerak *balancing* robot yang diberi gangguan dengan mendorong ke arah kiri , grafik yang condong ke nilai 97.23 sampai 100.51 yang terdapat pada detik 1us sampai detik ke 48us . Dari hasil yang di peroleh dilihat pada perbedaan antara *set point*, *sensor*, *error* ,

delta error dan *output fuzzy* telah didapat rata – rata sebesar 1.23 %.

4.4 Pengambilan data pengujian ketika diberi gangguan kearah kanan



Gambar 4.4 Data perubahan sensor ketika diberi gangguan kearah kanan

Gambar 4.4 merupakan Grafik pengujian gerak *balancing* robot yang diberi gangguan dengan mendorong ke arah kanan yang menghasilkan grafik yang condong ke nilai 96.91 sampai 93.38 yang terdapat pada detik 8us sampai detik ke 60us. Dari hasil yang di peroleh dilihat pada perbedaan antara *setpoint*, *sensor*, *error* , *delta error* dan *output fuzzy* telah didapat rata–rata sebesar 1.67 %.

4. Penutup

5.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Robot dapat berfungsi menyeimbangkan dengan baik, dengan persen keberhasilan 90%
2. Rata-rata response time menuju ke steady state sampai dengan 800ms, dengan osilasi sudut rata-rata 10⁰
3. Tegangan dan arus *battery* mempengaruhi pergerakan *balancing* robot
4. Robot masih dalam kondisi stabil walupun didorong ke kiri ataupun ke kanan.

5.2 Saran

1. Bisa di kembangkan dengan menambahkan jenis lintasan yang menaik dan menurun.

Daftar Pustaka

- [1] Toar Hendri, “Sistem Kendali Kinematic Pergerakan Robot Manipulator 3DOF”, Buku Tugas Akhir Diploma VI, Jurusan Teknik Mekatronik Politeknik Negeri Batam, Batam, 2013.
- [2] Alvianto Ekky, “Implementasi Point To Point (P2P) Pada *Mobile Robot*”, Buku Tugas Akhir Diploma III, Jurusan Teknik Mekatronik Politeknik Negeri Batam, Batam, 2015.
- [3] T.Sutojo,Ssi, M.kom, Edy Mulyanto, Ssi, Mkom and DR. Vincent Suhartono “**Kecerdasan Buatan**” Semarang, 2010, pp 219-227.
- [4] Mubarok Asep, “ Pendeteksi Rotasi Menggunakan *Gyroscope* Berbasis Mikrokontoler AT8535”, Buku Tugas Akhir S1, Jurusan Teknik Elektro Universitas Diponegoro, Semarang.
- [5] http://a-research.upi.edu/operator/upload/s_m at-0700767_chapter3.pdf