

Kesetimbangan Robot Beroda Dua Menggunakan Metode Fuzzy Logic

Handry Khoswanto¹, Djoko Purwanto²

¹ Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Kristen Petra, Surabaya, Indonesia

handry@petra.ac.id; handry@gmail.com

² Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

djoko@ee.its.ac.id

Abstrak

Bahan bakar minyak memiliki nilai yang melambung cukup tinggi. Apabila ditinjau dari perhitungan, penggunaan energi listrik masih lebih murah dibandingkan energi minyak bumi. Dalam penelitian ini mengangkat topik *Electric self balancing robot*. *Self Balancing Robot* yang memiliki roda dua dan mampu berdiri stabil dengan kedua rodanya. Diharapkan penelitian Balancing Robot dapat digunakan sebagai prototipe awal *human transporter* yang handal dan murah.

Dengan menggabungkan *Intelligent Control System* (ICS), sensor dan *hardware motor* diharapkan *Self Balancing Robot* (SBR) dapat direalisasikan. Dalam penelitian ini ICS yang digunakan adalah metode *Fuzzy Logic*, sedangkan sensor yang akan digunakan adalah *accelerometer sensor* yang merepresentasikan sudut elevasi terhadap bumi. ICS akan diimplementasikan ke dalam *microcontroller* kelas AVR. Dengan input berupa Error sudut dan Del Error sudut terhadap bumi maka didapatkan hasil defuzzifikasi berupa *angular rate* yang digunakan untuk mengontrol kedua rodanya.

SBR memiliki *rise time/fall time* maksimum sebesar 1000ms dan *settling time* maksimum sebesar 9000ms dan SBR mampu mencapai kesetimbangannya kembali (*steady state*) setelah mendapatkan gangguan dari luar.

Kata kunci: *Self balancing robot, accelerometer, fuzzy logic*

1. Pendahuluan/Pengantar

Saat ini di Indonesia telah terjadi krisis energi terutama bahan bakar. Bahan bakar minyak memiliki nilai yang melambung cukup tinggi. Apabila ditinjau dari perhitungan, penggunaan energi listrik masih lebih murah dibandingkan energi minyak bumi. Oleh karena itu dalam penelitian tesis ini mengangkat topik *electric balancing robot* yang merupakan starting awal alat transportasi berenergi listrik yang murah dari segi bahan bakar dan handal. Penelitian dan pengembangan *Self Balancing Robot* pun meningkat pada akhir-akhir ini. Sejak Dean Kamen, 2001 telah mendeklarasikan SEGWAY sebagai "The first Self-balancing, electric powered transportation device" Kemudian diikuti oleh penelitian-penelitian sejenis di berbagai negara. Hal ini tentunya dipicu oleh perkembangan teknologi elektronika, robotika serta IT yang cukup pesat di seluruh dunia. *Self balancing robot* merupakan robot yang memiliki prinsip kerja yang hampir mirip dengan sistem pendulum terbalik (Geoffrey D, Benneth, 2007).

Dalam kajian pustaka yang telah dilakukan ada beberapa penelitian serupa dengan beberapa perbedaan metode kontrol dan penggunaan mikrokontroler dan komponen pendukungnya. Penelitian Legway *Balancing Robot* menggunakan LEGO Mindstorms RCX. Legway menggunakan dua buah EOPDs (*Electro-Optical Proximity Detector*) oleh HiTechnic untuk membuat robot menjadi seimbang dan dapat mengikuti garis. EOPDs memiliki fungsi yang sama dengan IRPD (*Infrared Proximity Detector*). Perbedaannya hanya terletak pada penggunaan cahaya tampak atau tidak. Metode yang digunakan untuk kontrol keseimbangannya menggunakan BrickOS (LegOS), pemrograman C/C++ untuk LEGO Mindstorms. EOPDs digunakan untuk mendeteksi jarak antara sistem dan permukaan lantai. Apabila terjadi perubahan jarak maka program akan merespon maju atau mundur tergantung seberapa besar perubahannya. Setiap 50 ms, Legway akan menghitung *balance point* dengan mempertimbangkan jarak dan kecepatan motor. Beberapa keunggulan dan kelemahan dapat ditemukan pada sistem Legway. Keunggulannya terletak pada kesederhanaan sistem sehingga mudah dibuat. Sedangkan kelemahan yang nampak pada sistem ini adalah sistem mudah terkena pengaruh cahaya nampak luar (*external interference*). Selain itu kelemahan lain yang nampak adalah penempatan posisi EOPDs sangat mempengaruhi kinerja robot. Perbedaan kemiringan robot akan membuat sensor tidak tepat pada garis normalnya terhadap permukaan. Hal inilah yang tidak menguntungkan Legway *Balancing Robot*.

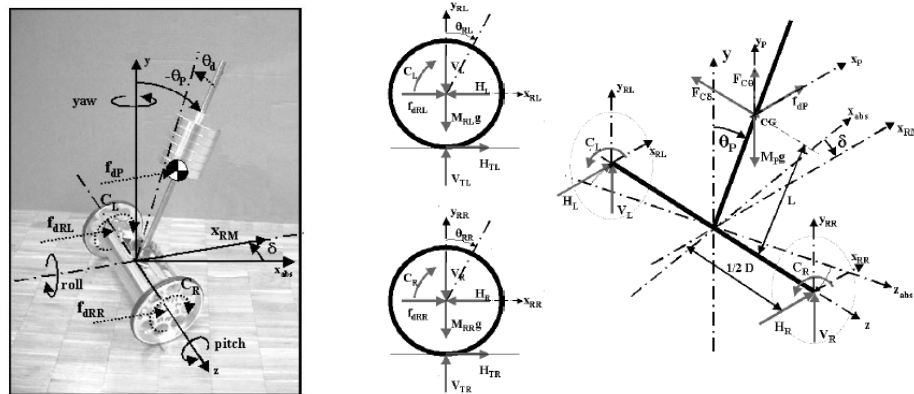
Pada penelitian *balancing robot* lain yaitu n-Bot, *two wheeled balancing robot* mirip dengan karya JOE. n-Bot diciptakan oleh David. P Anderson. Sistem ini menggunakan *inertial sensor* dan sensor posisi yang menggunakan *encoder*. Kesetimbangan yang diciptakan dalam n-Bot sangat baik.

Setelah mengulas beberapa penelitian sebelumnya maka dalam penelitian ini akan mengulas kembali *Self Balancing Robot* yang memiliki roda dua dan mampu berdiri stabil dengan kedua rodanya. Diharapkan penelitian Balancing Robot dalam tesis ini dapat digunakan sebagai prototipe awal alat transportasi yang handal dan murah.

2. Kajian pustaka dan teori

Ide dasar untuk membuat *robot* beroda dua dapat setimbang adalah sangat mudah yaitu dengan cara mengendalikan roda searah dengan arah jatuhnya bagian atas sebuah *robot*. Apabila proses tersebut dapat terlaksana maka *robot* tersebut dapat setimbang. Secara praktis ini membutuhkan dua *sensor* sebagai umpan baliknya yaitu sensor kemiringan atau sudut terhadap gaya gravitasi dan *sensor encoder* untuk mengukur posisi robot. Berikut ini akan diberikan ulasan literatur dua makalah sebelumnya yang menggunakan dua buah sensor yang berbeda dan menggunakan metode kontrol yang berbeda pula.

Dalam penelitian JOE: *A Mobile, Inverted Pendulum* menggunakan 3 *Degree of Freedom* (DoF) (Grasser dkk, 2001). Sistem dapat berputar terhadap sumbu z (*pitch*), pergerakannya dideskripsikan sebagai posisi P (θ_P) dan ω_P (kecepatan sudut *angular velocity*). Pergerakan linier dideskripsikan dalam translasi (x_{RM}) dan kecepatan linier (v_{RM}). Perputaran sumbu vertikal (*yaw*) dikarenakan putaran roda. Putaran tersebut dapat dideskripsikan sebagai δ dan kecepatan sudut $d(\delta)$. *Modeling* sistem JOE dapat digambarkan sebagai berikut.



Gambar 1. Definisi state variable JOE: *Inverted Pendulum* dan diagram model

Berikut ini 6 *state space* variabel yang dapat diubah antara lain (Grasser dkk, 2001):

x_{RM}	<i>straight line position</i>	[m]
v_{RM}	<i>straight line speed</i>	[m/s]
θ_P	<i>pitch angle</i>	[rad]
ω_P	<i>pitch rate</i>	[rad/s]
δ	<i>yaw angle</i>	[rad]
$d(\delta)$	<i>yaw rate</i>	[rad/s]

Persamaan untuk masing-masing roda antara lain (Grasser dkk, 2001):

$$\ddot{x}_{RL} M_{RL} = f_{dRL} - H_L + H_{TL} \dots\dots\dots (1)$$

$$\ddot{y}_{RL} M_{RL} = V_{TL} - M_{RL} g - V_L \dots\dots\dots (2)$$

$$\ddot{\theta}_{RL} J_{RL} = C_L - H_{TL} R \dots\dots\dots (3)$$

Sedangkan persamaan untuk *chassis*-nya antara lain (Grasser dkk, 2001):

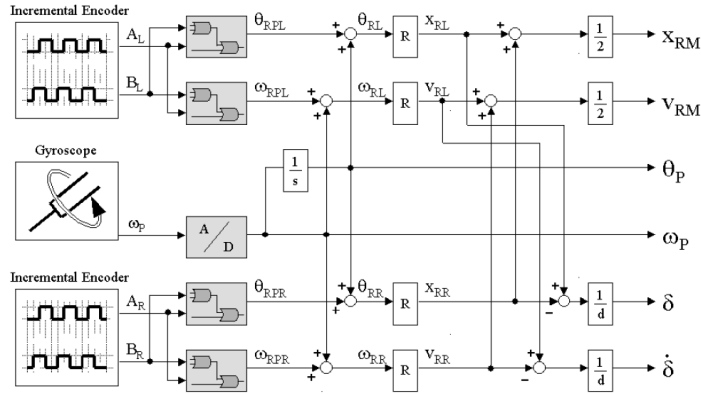
$$\ddot{x}_{RL} M_P = f_{dP} + H_R + H_L \dots\dots\dots (4)$$

$$\ddot{y}_P M_P = V_R - V_L - M_{PB} + F_{CG} \dots\dots\dots (5)$$

$$\theta_P J_{P\theta} = (V_R + V_L) L \sin \theta_P \dots\dots\dots (6)$$

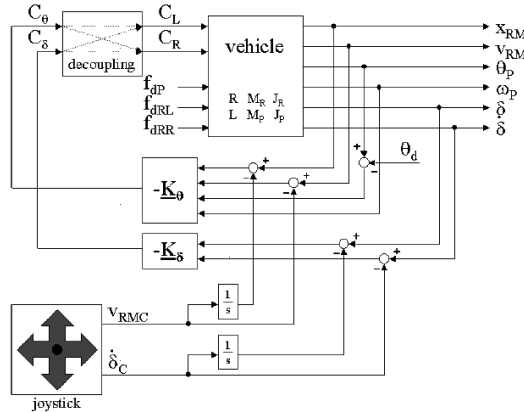
$$\ddot{\delta} J_{P\delta} = (H_L - H_R) \frac{D}{2} \dots\dots\dots (7)$$

JOE menggunakan *state space* untuk menghitung sistem di atas. Selain itu JOE menggunakan beberapa sensor yaitu *tilt sensor*, *Tilt* dan *gyroscope*. *Tilt sensor* digunakan untuk mengukur *pitch angle* (sudut anggukan) terhadap gaya gravitasi; mengukur perubahan *pitch angle*. Accelerometer digunakan untuk mengukur static dan dynamic acceleration. Sedangkan *gyroscope* digunakan untuk mengukur *pitch rate*. Berikut ini diagram dari sistem JOE.



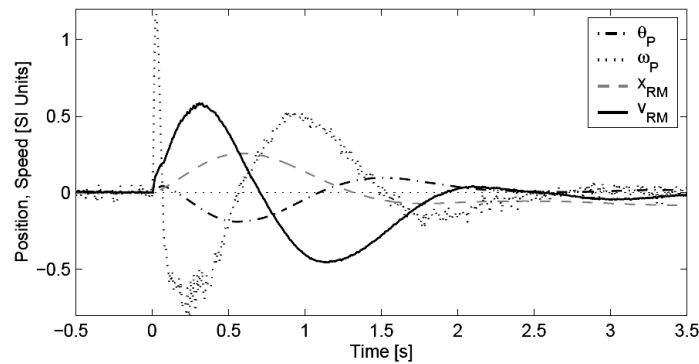
Gambar 2. Diagram sensor JOE (Grasser dkk, 2001)

Untuk blok penggeraknya dapat dijelaskan melalui gambar di bawah ini.



Gambar 3. Diagram sistem kontrol (Grasser dkk, 2001)

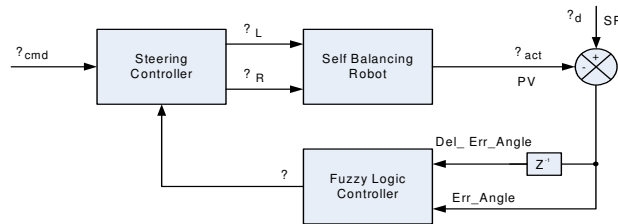
Dalam pengujian yang telah dilakukan JOE, maka dapat ditampilkan seperti gambar di bawah ini (Grasser dkk, 2001).



Gambar 4. Respon hasil pengujian JOE (Grasser dkk, 2001)

3. Metode Penelitian

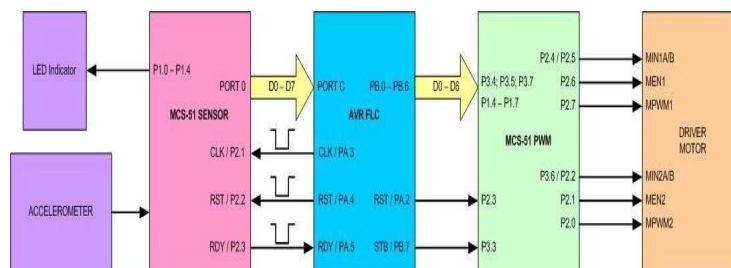
Untuk mencapai kesetimbangan tersebut digunakan metode fuzzy logic yang cenderung cocok diberikan pada sistem *non linier* dan penggerak yang tidak diketahui pemodelan matematisnya. Berikut ini diagram blok sistem kontrol yang direalisasikan.



Gambar 3.6. Diagram blok sistem kontrol

Ada tiga blok penting dalam SBR, blok *Steering Kontroller*, blok *Self Balancing Robot*, dan blok *Fuzzy Logic Kontroller*. Sistem kontrol tersebut menggunakan sensor accelerometer untuk mendapatkan data yang dibutuhkan. Sensor accelerometer sangat *sensitive* terhadap perubahan gerakan sehingga dapat diaplikasikan sebagai *tilt sensor*. Pada perencanaan ini menggunakan *sensor Memsic 2125 Accelerometer sensor*.

Gerakan SBR akan menghasilkan output θ_{act} . Pembacaan ini dimodifikasi menjadi *Err_Angle* dan *Del_Err_Angle*. Kedua hasil tersebut digunakan sebagai *crisp input* dari blok *Fuzzy Logic Kontroller* (FLC). Melalui proses fuzzifikasi dan defuzzifikasi akan menghasilkan *crisp output* berupa ω (kecepatan sudut). Nilai ini akan diterima oleh *Steering Kontroller* sebagai nilai *duty cycle PWM* untuk kedua penggerak SBR. Apabila diinginkan SBR berbelok dengan sudut tertentu maka sistem akan mendapatkan data Input θ_{cmd} . Untuk langkah awal penelitian ini, θ_{cmd} dibuat sama dengan nol (maju lurus). Sensor potentiometer akan memberikan data tersebut sehingga *Steering Kontroller* akan memberikan penambahan atau pengurangan nilai *duty cycle PWM* pada DC motor kiri maupun kanan.



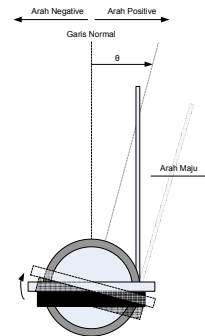
Gambar 7. Diagram blok hardware SBR

3.1. Perancangan *Fuzzy Logic Controller*

Salah satu kontrol yang sering digunakan untuk mengontrol motor DC selain kontrol *on/off* ialah kontrol *fuzzy*. Kontrol *fuzzy* biasanya digunakan untuk sistem yang cukup kompleks dan membutuhkan ketelitian tinggi. Dalam *fuzzy* terdapat tiga proses utama yaitu *fuzzification*, *rule evaluation* dan *defuzzification*. Dalam merancang *input membership function*, *rules* dan *output membership function* dibutuhkan sebuah software PetraFuz.

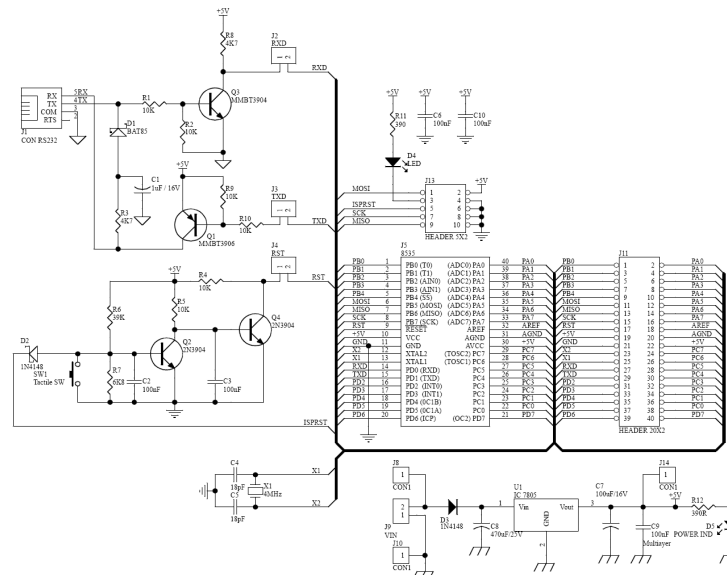
PetraFuz merupakan suatu software yang didalamnya berisi *kernel fuzzy* dengan bahasa C-Code Vision AVR. Hal tersebut memudahkan kita dalam merancang sistem *fuzzy* berbasis AVR. Jumlah maksimal *crisp input* adalah 5 dan jumlah maksimal *crisp output* adalah 3. Jumlah maksimal *membership fuction input* dan *output* adalah 8. Untuk *membership function input* digunakan 4 point karena bentuknya adalah segitiga dan trapesium. Jumlah *rule* maksimum adalah 1024.

Desain FLC harus memperhatikan pergerakan body dan platform dari balancing robot. Berikut ini ilustrasi untuk gerakan dan perubahan titik berat balancing robot.



Gambar 8. Sketsa SBR

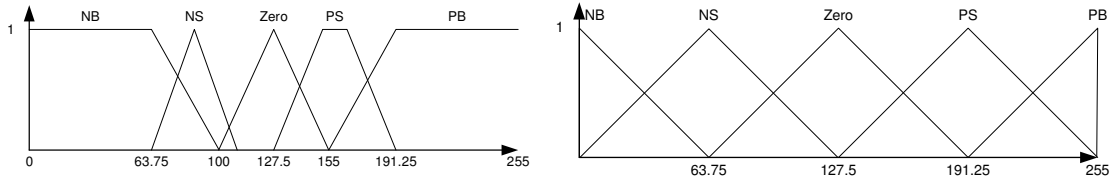
Ada tiga buah parameter fuzzy yang harus kita desain untuk merancang suatu plant fuzzy yaitu membership function input, membership function output dan rule. Nilai patokan yang kita berikan untuk nilai *universe of discourse* adalah 0 sampai 255. Nilai ini diberikan karena nilai 0 sampai 255 merupakan *range* yang umumnya mampu diolah oleh mikrokontroler. Mikrokontroler FLC yang digunakan adalah produk Innovative Electronics tipe DT Proto 40 AVR 8535 yang tertera pada gambar di bawah ini. Dalam set tersebut dapat dilakukan komunikasi serial.



Gambar 9. Mikrokontroler FLC DT Proto40 (Innovative Electronics 2007)

3.2. Pemodelan Data Fuzzy

Input yang kita berikan untuk perancangan *fuzzy* yang pertama adalah sebuah *input* yang diberi nama *Angle_Err* dan *Del_Angle_Err*. Nilai *Angle_Err* didapat dari *setting point* (SP) dikurangi dengan *present value* (PV). Nilai *Del_Angle_Err* adalah nilai Err sekarang dikurangi Err terdahulu. Kedua nilai yang digunakan untuk *input error* dalam aplikasi pengaturan kecepatan motor DC pada implementasi SBR dapat dilihat pada tabel di bawah ini.



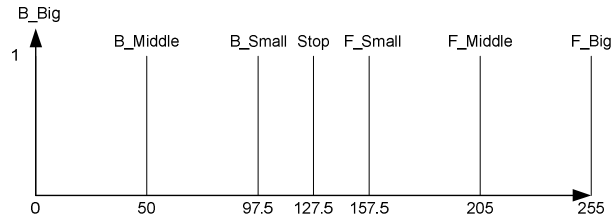
Gambar 10. Membership function input: Err_Angle dan Del_Err

Gambar membership function di atas tidak simetris. Hal ini dikarenakan perbedaan respon kecepatan pada motor DC ketika maju dan mundur. Perhatikan gambar di bawah ini menunjukkan ilustrasi arah maju dan mundur sebuah *balancing robot*.

Apabila platform dari balancing robot bergerak arah maju maka platform miring dengan sudut tertentu. Kemiringan ini dapat diketahui dari sensor accelerometer.

Ada 5 buah keanggotaan membership function yaitu: Negative Big (NB), Negative Small (NS), Zero (Z), Positive Small (PS), dan Positive Big (PB). Sedangkan untuk *Del_Angle_Err* dapat ditunjukkan pada tabel dan gambar di bawah ini.

Output dari pemodelan *fuzzy* adalah nilai PWM untuk pengaturan kecepatan motor. Nilai-nilai yang dimasukkan ke dalam perancangan *membership function output* untuk aplikasi pengaturan kecepatan motor dapat dilihat pada tabel dan gambar di bawah ini.



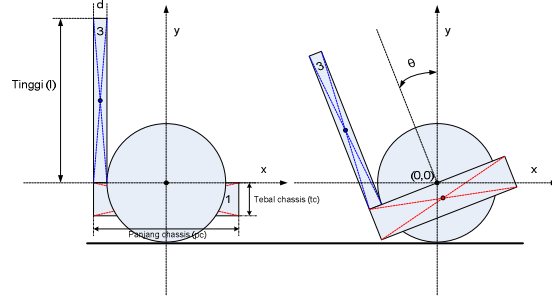
Gambar 11. Membership function output

Membership function output tersebut dibagi menjadi 2 sisi yaitu sisi *backward* dan sisi *forward*, dan memiliki 7 Label Membership Function yaitu B_Big, B_Middle, B_Small, Stop, F_Small, F_Middle dan F_Big. Berikut ini tabel rules dari fuzzy logic yang di desain.

Tabel 1. Rules FLC

Del_Angle_Err	Angle_Err					
	MOTOR	NB	NS	Zero	PS	PB
	NB	B_Big	B_Big	B_Middle	B_Small	Stop
	NS	B_Big	B_Middle	B_Small	Stop	F_Small
	Zero	B_Middle	B_Small	Stop	F_Small	F_Middle
	PS	B_Small	Stop	F_Small	F_Middle	F_Big
	PB	Stop	F_Small	F_Middle	F_Big	F_Big

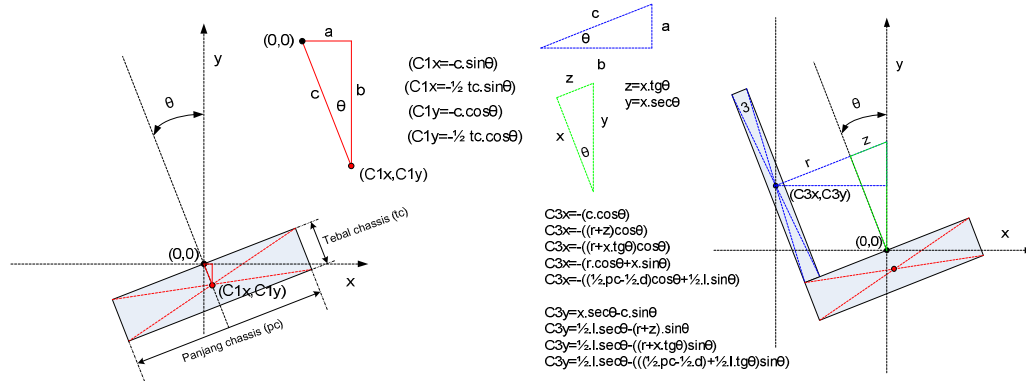
Untuk merealisasikan balancing robot tersebut maka dirancang sebuah mekanik yang memiliki dua buah roda yang terletak pada kanan dan kiri robot. Perlu dipertimbangkan juga posisi titik berat dari SBR yang akan dirancang. Dalam pembuatan SBR ini ada dua perancangan kestabilan, yaitu kestabilan metode kontrol dan kestabilan mekanika. Kestabilan metode telah dijelaskan pada sub bab terdahulu dan kestabilan mekanika akan dirancang di sub bab ini. Penentuan posisi titik berat (*Center of Mass*) juga diperlukan dalam desain ini. Berikut ini sketsa rancangan tersebut.



Gambar 12. Prediksi posisi titik berat SBR

Ada tiga bagian dalam SBR yaitu: *chassis*, *roda* dan *stick*. Ketiga bagian tersebut memiliki titik beratnya masing-masing. Asumsikan ketiga benda tersebut homogeny dan memiliki massa M_1 , M_2 , dan M_3 . Berikut ini modeling untuk masing-masing benda tersebut.

1. Bagian *chassis*



Gambar 13. Titik berat chassis dan stick

$$COM_x = \frac{C1x.M1 + C2x.M2 + C3x.M3}{M1 + M2 + M3}$$

$$COM_x = \frac{\left(-\frac{1}{2} \cdot tc \cdot \sin\theta\right) M1 + \left(\left(\frac{1}{2} \cdot pc - \frac{1}{2} \cdot d\right) \cos\theta + \frac{1}{2} \cdot l \cdot \sin\theta\right) M3}{M1 + M2 + M3}$$

$$COM_y = \frac{C1y.M1 + C2y.M2 + C3y.M3}{M1 + M2 + M3}$$

$$COM_y = \frac{\left(-\frac{1}{2} \cdot tc \cdot \cos\theta\right) M1 + \left(\frac{1}{2} \cdot l \cdot \sec\theta - \left(\left(\frac{1}{2} \cdot pc - \frac{1}{2} \cdot d\right) + \frac{1}{2} \cdot l \cdot \tan\theta\right) \cdot \sin\theta\right) M3}{M1 + M2 + M3}$$

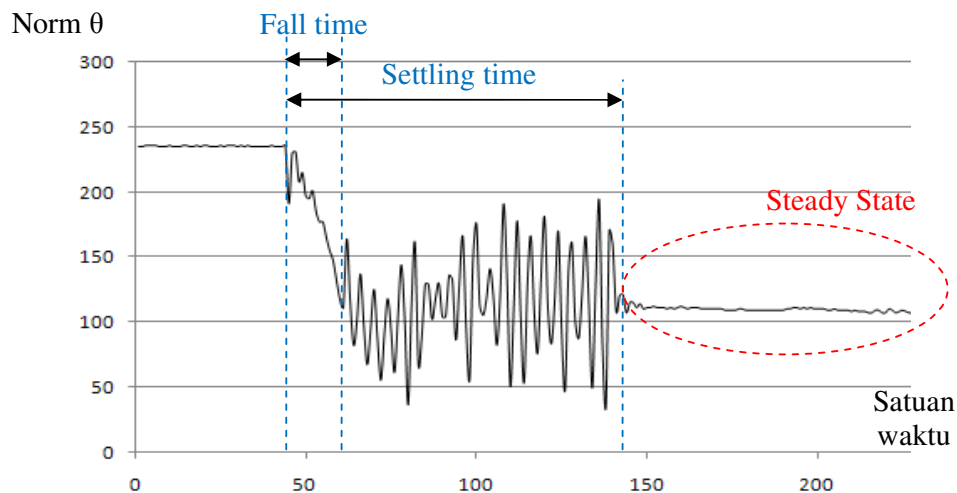
Secara matematis, posisi titik berat dapat ditemukan apabila kemiringan SBR diketahui. Hal ini dapat membantu untuk memprediksikan berapa kecepatan yang harus diberikan kepada motor agar dapat mempertahankan keseimbangannya. Berikut ini hasil realisasi SBR.



Gambar 14. Hasil konstruksi mekanik SBR

4. Pengujian

Pengujian dilakukan untuk mengetahui seberapa handal SBR dalam melakukan proses kesetimbangan. Pengujian pertama akan diambil 244 data yang bergerak terhadap waktu. Kondisi pengujian dilakukan perubahan nilai normalisasi input sudut yang masuk ke dalam FLC. Nilai representasi θ tersebut berkisar dari 3750 μs sampai dengan 6250 μs , Nilai tersebut merepresentasikan nilai sudut -90° samapi dengan 90° . Berikut ini grafik pengujian.



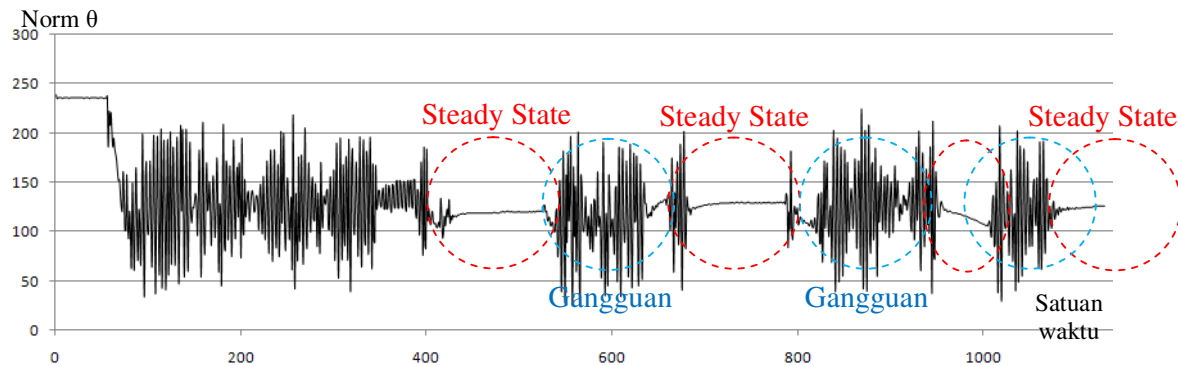
Gambar 15. Normalisasi θ VS waktu

Nilai Norm θ merupakan data 8 bit berkisar mulai 0 – 255. Data 0 menyatakan 90° dan 255 menyatakan -90° . Perubahan pembatasan nilai sudut yang masuk merubah performa dari metode fuzzy yang telah diimplementasikan. SBR mampu berhenti pada saat mencapai nilai SP yang diinginkan. Karena

sistem dirancang *free running* dan masing-masing iterasi membutuhkan waktu ± 45 ms sehingga SBR memiliki spesifikasi:

fall time/rise time : 25 iterasi * 45 ms = 1125ms
Settling time : 100 iterasi * 45 ms = 4500ms

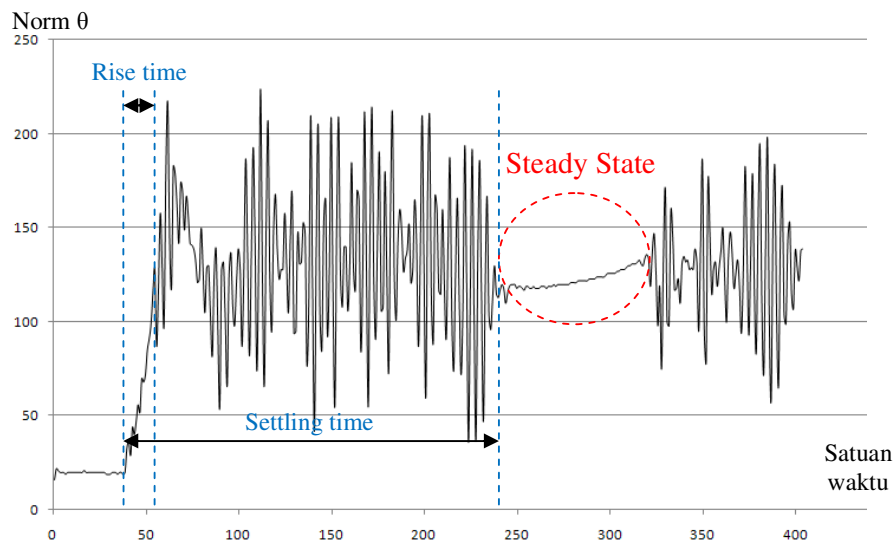
Pengujian berikutnya SBR diberikan gangguan berupa dorongan maju dan mundur secara bergantian. Pengujian ini bertujuan untuk mendapatkan hasil apakah SBR mampu mencapai *steady state* nya kembali setelah mendapatkan gangguan tersebut. Metode pengujian ini membuat kondisi sebelum mikrokontroler SBR dinyalakan, SBR dalam kondisi jatuh menghadap ke depan. Ketika SBR dinyalakan penyimpanan data dilakukan. Pengujian ini menggunakan 1130 data yang diambil dari SBR *monitoring software*.



Gambar 16 Normalisasi θ VS waktu dengan gangguan

Berdasarkan hasil pengujian tersebut di atas, SBR mampu mengembalikan posisi *steady state* nya dengan baik meskipun diberikan gangguan berupa dorongan maupun tarikan dari luar. Gangguan tersebut memiliki simpangan yang cukup besar dibandingkan osilasi keseimbangan metode kontrol.

Pengujian selanjutnya adalah membuat SBR dalam kondisi jatuh posisi ke belakang dan kemudian mengaktifkan mikrokontroler. Dalam pengujian ini diambil 404 data percobaan dan di gambarkan dalam bentuk grafik.



Gambar 17. Normalisasi θ VS waktu tanpa gangguan

Berdasarkan grafik pengujian tersebut di atas dapat disimpulkan bahwa SBR dapat menyeimbangkan diri dengan baik, dalam kondisi tanpa beban. Berikut ini spesifikasi SBR pada pengujian kedua:

fall time/rise time : 20 iterasi * 45 ms = 900ms
Settling time : 200 iterasi * 45 ms = 9000ms

Waktu yang dibutuhkan SBR untuk mencapai ke *setting point* pertama kali (*rise time/fall time*) maksimum sebesar 1000ms dan *settling time* maksimum sebesar 9000ms dan SBR mampu mencapai kesetimbangannya kembali (*steady state*) setelah mendapatkan gangguan dari luar.

5. Kesimpulan

Dari hasil yang didapatkan, maka dapat diambil kesimpulan bahwa :

1. Dengan menggunakan 2 buah input pada proses fuzzifikasi dan menggunakan 25 rules, SBR memiliki *rise time/fall time* maksimum sebesar 1000ms dan *settling time* maksimum sebesar 9000ms dan SBR mampu mencapai kesetimbangannya kembali (*steady state*) setelah mendapatkan gangguan dari luar.
2. Motor memiliki respon yang kurang cepat dan memiliki konsumsi daya puncak yang cukup tinggi, yaitu 3120W pada tegangan 48V. Apabila dilakukan penurunan tegangan eksitasi menjadi 12 V maka konsumsi daya juga berkurang tetapi torsi juga berkurang cukup besar. Hal ini membuat respon balancing robot tidak maksimal.
3. Dengan menggunakan metode fuzzy masih diperlukan tuning yang banyak agar mendapatkan hasil yang maksimum.

Daftar pustaka

- [1] Grasser, Felix; D'arrigo, Aldo; Colombi, Silvio; Rufer, Alfred (2001), "JOE: A Mobile, Inverted Pendulum", *Laboratory of Industrial Electronics Swiss Federal Institute of Technology Lausanne*.
- [2] Grasser, Felix; D'arrigo, Aldo; Colombi, Silvio; Rufer, Alfred (2002), "JOE: A Mobile, Inverted Pendulum", *IEEE Transactions on Industrial Electronics*, Vol. 49, No. 1, pp. 107 - 114, 2002.
- [3] S.W. Nawawi, M.N. Ahmad and J.H.S. Osman; "Real Time Control of Two Wheeled Inverted Pendulum Mobile Robot", *International Journal of Computer, Information, and Systems Science, and Engineering*, Volume 2 Number
- [4] Michael Baloh, Michael Parent; "Modeling and Model Verification of an Intelligent Self-Balancing Two Wheeled Vehicle for an Autonomous Urban Transportation System", *The Conference on Computational Intelligence, Robotics and Autonomous Systems*, December, 15 2003 at Singapore