

## Two Wheels Balancing Line Tracer Robot Using Fuzzy Logic Control

Indra Dwisaputra<sup>1,2</sup>, Indra Adji Sulistijono<sup>1</sup>, M. Iqbal Nugraha<sup>2</sup>

<sup>1</sup>Jurusan Teknik Mekatronika, Politeknik Elektronika Negeri Surabaya  
Institut Teknologi Sepuluh Nopember  
Jln raya ITS Sukolilo, surabaya 60111,  
Tel: (031) 5947280, Fax: (031)594 6114

<sup>2</sup>Jurusan Teknik Elektronika, Politeknik Manufaktur Negeri Bangka Belitung  
Jln. Timah Raya Kawasan Industri Air KantungSungailiat-Bangka 33211,  
Tel: 0717-93586, 95252, Fax. 0717-93585  
indrads@yahoo.com, indra.adji.s@gmail.com, iqbal\_nugrah@yahoo.com

### Abstract

Sekarang ini sudah ada alat transportasi yang menggunakan dua roda yang di kendalikan oleh satu orang (*segway*). Proyek akhir ini juga akan membuat alat yang sama dalam hal prinsip kerjanya, namun dengan ukuran yang berbeda. Alat ini juga nantinya dibuat agar dapat mengikuti garis sambil menyeimbangkan bodi robot agar tidak terjatuh. Kedua roda robot di hubungkan dengan motor DC sebagai pemberi gaya. Gaya yang diberikan motor sesuai dengan arah titik jatuhnya bodi robot. Sistem kontrol yang diterapkan pada tugas akhir ini adalah fuzzy logic control. Tujuan pembuatan proyek akhir ini selain untuk pembelajaran tentang keseimbangan, juga merupakan penerapan metode fuzzy logic untuk sistem yang labil. Permasalahan yang ada disini adalah bagaimana membuat robot tetap stabil pada titik 90°. Permasalahan juga muncul ketika robot dapat mengikuti garis sambil mengatur keseimbangan. Sensor yang digunakan untuk mendeteksi kemiringan robot terhadap bumi adalah *accelerometer Hitachi H48C 3-Axis Accelerometer Module*. Dengan sistem kontrol fuzzy logic robot mampu menjaga keseimbangan sambil mengikuti garis. Kemampuan robot mengikuti garis hanya pada garis lurus dan belokan yang berupa radius.

**Kata kunci :** robot, keseimbangan, *accelerometer*, sensor garis, *fuzzy logic control*

### 1. Pendahuluan

Perkembangan dunia robotika semakin meningkat dari hari ke hari. Sebagian besar robot dibuat untuk membantu melaksanakan aktifitas manusia. Namun sebagian juga ada yang dibuat untuk keperluan hobi dan

pembelajaran, yang salah satunya adalah robot keseimbangan.

Penelitian pernah dilakukan oleh Handry Khoswanto, dalam pembuatan robot beroda dua dengan metode *fuzzy logic control*. Robot itu dinamakan SBR (*Self Balancing Robot*). Penelitian Handry dengan menggunakan sensor *accelerometer* sebagai sensor pendeteksi kemiringan robot terhadap bumi. 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<sup>[1]</sup>.

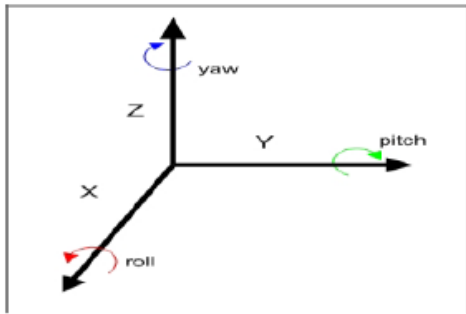
Pada penelitian ini akan dibuat robot yang sama dalam prinsip kerjanya, namun dengan fungsi yang berbeda. Selain untuk mengatur keseimbangan robot ini juga nantinya dibuat agar dapat *tracking line* (mengikuti garis). Digunakan dua motor DC yang difungsikan untuk memberikan gaya searah dengan titik jatuhnya robot. Dalam mendeteksi kemiringan robot terhadap bumi menggunakan sensor *accelerometer* tipe *Hitachi H48C 3-Axis Accelerometer Module*. Hal tersebut tentu membutuhkan kendali yang bisa membuat robot menyeimbangkan bodi sekaligus mengikuti garis dengan baik. Untuk menyelesaikan permasalahan ini digunakan lah metode *fuzzy logic control*.

### 2. Tinjauan Pustaka

#### 2.1. Accelerometer

*Accelerometer* adalah perangkat yang digunakan untuk mengukur akselerasi dan dampak gravitasi pada percepatan. Beberapa sifat tersebut adalah *roll*, *pitch*

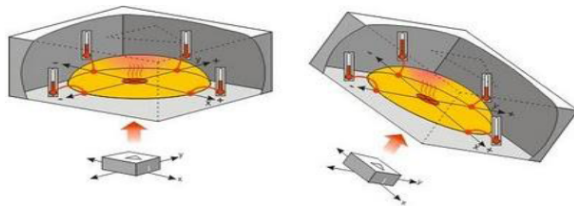
dan *yaw*. Sifat ini merupakan sifat pergerakan terhadap sumbu X, Y dan Z yang telah disesuaikan dengan titik gravitasi bumi. Sumbu X merepresentasikan gerakan perputaran. Sumbu Y merepresentasikan gerakan terhadap kemiringan pada posisi bumi. Sumbu Z merepresentasikan gerakan menoleh atau berbelokan terhadap bidang. Dalam sifat sistem dari pergerakan tersebut dapat dipresentasikan dari gambar 1.



**Gambar 1.** Analogi sumbu pada *accelerometer*

(sumber: <http://www.digitdude.com/2010/01/rangkaian-parallax-hitachi-3g-tri-axis.html>)

Sensor menghasilkan panas yang mana jika dimiringkan maka panas tersebut akan memanasi bagian pinggir sensor yang merupakan bahan seperti *thermocoupler*. Sehingga jika dipanaskan akan menghasilkan perubahan dan perbedaan panas yang mengakibatkan terjadinya perubahan tegangan yang nantinya dimasukkan kedalam ADC. Proses pembacaan nilai ADC pada *accelerometer* dapat dilihat dari gambar 2.



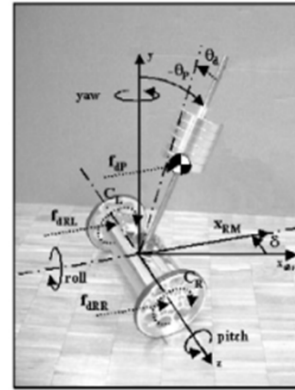
**Gambar 2.** Analogi sumbu pada *accelerometer*

(sumber: <http://www.digitdude.com/2010/01/rangkaian-parallax-hitachi-3g-tri-axis.html>)

## 2.2. Inverted Pendulum

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. Dalam penelitian JOE: *A Mobile, Inverted Pendulum*

menggunakan 3 *Degree of Freedom* (DoF). Sistem dapat berputar terhadap sumbu z (*pitch*). Perputaran sumbu vertikal (*yaw*) dikarenakan putaran roda. *Modeling* sistem JOE dapat dilihat dari gambar di bawah ini.



**Gambar 3.** Definisi state variable JOE: *Inverted Pendulum*

(sumber: [http://leiwwww.epfl.ch/publications/grasser\\_darrigo\\_colombi\\_rufer\\_ieee\\_02.pdf](http://leiwwww.epfl.ch/publications/grasser_darrigo_colombi_rufer_ieee_02.pdf))

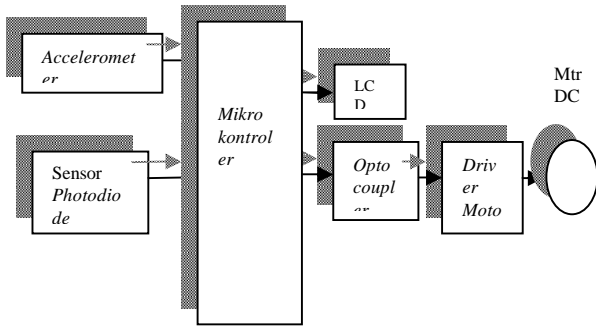
Dalam penelitian lainnya *Self Balancing Robot* (SBR) penelitian yang dilakukan oleh Handry Khoswanto dalam kesetimbangan robot beroda dua menggunakan metode *fuzzy logic*. Menghasilkan output  $\Theta_{act}$ , pembacaan ini dimodifikasi menjadi *Err\_Angle* dan *Del\_Err\_Angle*. Kedu hasil tersebut digunakan sebagai *crisp input* dari blok *fuzzy Logic Controller*. Melalui proses fuzzifikasi dan defuzzifikasi akan menghasilkan menghasilkan *crisp output* berupa  $\omega$  (kecepatan sudut)<sup>[1]</sup>.

Dari penelitian dengan 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<sup>[1]</sup>.

## 3. Perancangan Sistem

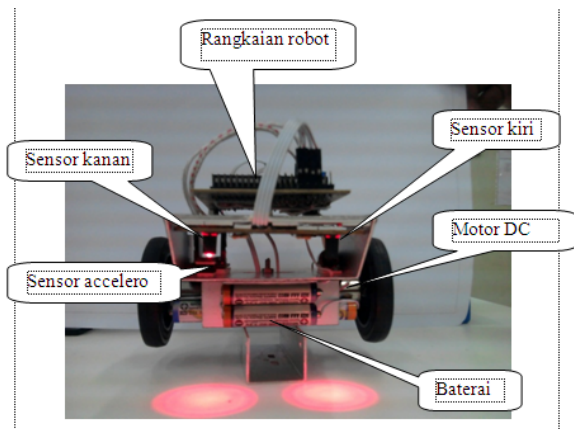
### 3.1. Konfigurasi Sistem

Dalam pembuatan proyek akhir ini secara umum terbagi dalam pembuatan *hardware* dan *software* yang digunakan. Perancangan *hardware* dibuat seminimum mungkin agar tidak menambah beban robot. Perencanaan sistem kontrol *fuzzy* yang tepat juga menentukan kestabilan robot dalam mengatur keseimbangan. Untuk blok diagram sistem yang digunakan terlihat pada gambar 4.



Gambar 4. Blok diagram sistem

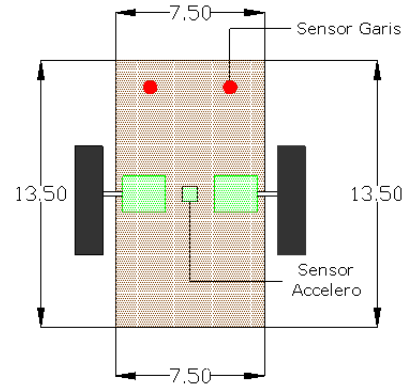
Robot ini dilengkapi dengan 2 buah sensor yaitu: *accelerometer* yang digunakan untuk mendeteksi kemiringan robot terhadap gaya gravitasi dan sensor *photodiode* yang digunakan untuk mendeteksi garis. LCD hanya difungsikan untuk menampilkan data sensor, sedangkan pada saat robot diaktifkan LCD dilepas karena mengingat secara fungsi dalam proses keseimbangan dan mengikuti garis tidak dibutuhkan menampilkan data. Peletakkan sensor dan hardware dapat dilihat pada gambar 5.



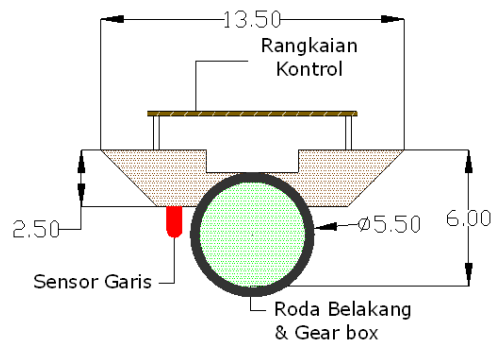
Gambar 5. Peletakkan hardware pada robot

### 3.2. Perancangan dan pembuatan konstruksi mekanik robot

Pembuatan mekanik robot ini menggunakan pelat aluminium. Pemilihan pelat aluminium dikarenakan bahan ini mudah didapat dan ringan. Dibawah ini terdapat gambar robot yang sebenarnya dari berbagai perspektif.



a. Ukuran robot tampak atas

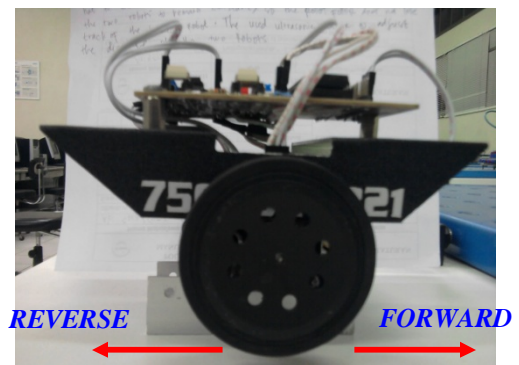


b. Robot tampak samping kiri

Gambar 6. Ukuran robot tampak atas dan samping kiri

### 3.3. Sistem Keseimbangan

Yang harus diperhatikan dalam pembuatan robot ini ialah menjaga robot agar tetap seimbang pada posisi 90°. Caranya adalah dengan memberikan respon gaya sesuai dengan titik jatuh bodi robot. Prinsip dari sistem keseimbangan yang akan digunakan terlihat dari gambar 6.

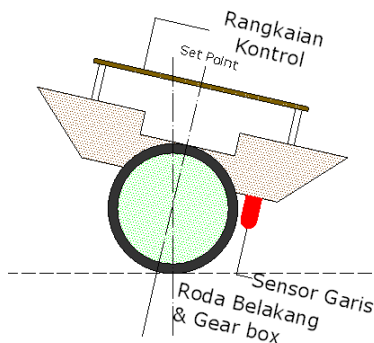


Gambar 6. Sistem keseimbangan

Pergerakan motor roda dipengaruhi oleh titik jatuhnya *body robot*. Seperti pada gambar 6 dapat dilihat bahwa pergerakan roda sesuai dengan titik jatuhnya *body robot*. Jika robot cenderung jatuh kedepan ( $err > setpoint$ ) maka motor diberi respon *forward*. Jika robot cenderung jatuh ke belakang ( $err < setpoint$ ) maka motor diberi respon *reverse*. Semakin besarnya selisih *err* terhadap *setpoint* maka akan semakin besar pula respon yang diberikan (diatur sesuai dengan *membership function output* pada *fuzzy*).

### 3.4. Sistem Berjalan Robot

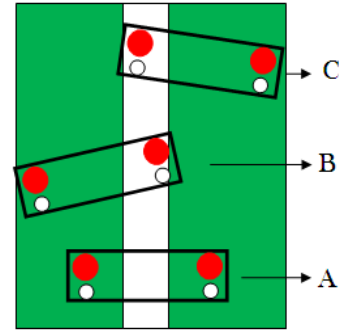
Sebelum robot mengikuti garis yang harus diperhatikan ialah bagaimana robot dapat berjalan, sekaligus mengatur kecepatan motor agar robot tidak terjatuh kedepan maupun kebelakang. Sistem berjalan robot adalah dengan cara mengatur nilai dari *setpoint* lebih besar dari nol. Dengan proses demikian maka robot akan cenderung jatuh kedepan sambil mengatur keseimbangan. Semakin besar selisih *setpoint* terhadap titik  $90^\circ$  maka robot akan berjalan semakin cepat. Dalam proyek akhir ini nilai dari *setpoint* pada saat maju disesuaikan dengan kemampuan robot dalam bernavigasi mengikuti garis sekaligus mengatur keseimbangan. Sistem jalannya robot dapat dilihat



Gambar 7. Sistem jalan robot

### 3.5. Sistem Pembacaan Garis

Proses pembacaan garis pada robot ini dilakukan dengan 2 buah sensor garis, yaitu sensor yang dapat membaca perubahan warna gelap dan terang. Dalam pembacaannya digunakan LED dan photodiode sebagai sensor. Kedua sensor diletakkan diluar garis.



Gambar 8. sistem pembacaan garis

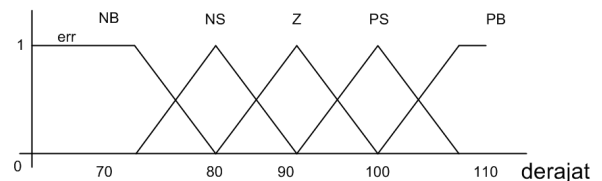
Peletakan sensor berada didepan robot (gambar 8), hal tersebut dimaksudkan agar lebih mempermudah robot dalam mengikuti garis. Dari gambar 8 dapat dilihat ada 3 kondisi robot ketika mengikuti garis, yaitu:

- A : kedua sensor berada di luar garis, sehingga yang harus dilakukan ialah kedua motor (motor kanan & motor kiri) maju dengan kecepatan yang sama.
- B : sensor kiri berada pada luar garis dan sensor kanan berada didalam garis, sehingga yang harus dilakukan ialah motor kanan diperlambat.
- C : sensor kanan berada diluar garis dan sensor kiri berada di dalam garis, sehingga yang harus dilakukan ialah motor kiri diperlambat.

Dengan demikian maka robot akan terus berjalan mengikuti garis. Pergerakan robot ketika mundur hanya digunakan pada saat memnyeimbangkan bodi robot agar selalu tegak lurus. Dibawah ini adalah *flowchart* sistem robot dalam mengikuti garis sekaligus mengatur keseimbangan.

### 3.6. Pemodelan Fuzzy

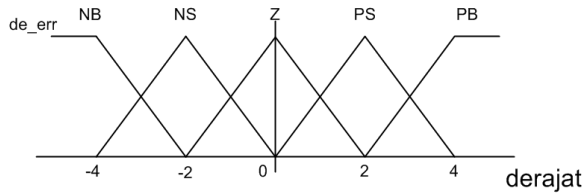
Input yang kita berikan untuk perancangan *fuzzy* yang pertama adalah sebuah *input* yang diberi nama *err*. Nilai *err* didapat dari nilai ADC sensor *accelerometer* dikurangi dengan *setpoint* yang kemudian dikonversi menjadi sudut.



Gambar 9 Membership function input *err*

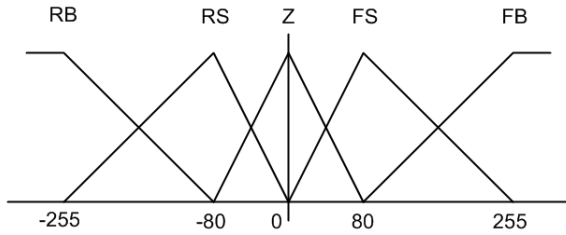
Input yang kedua adalah berupa nilai yang di beri nama *de\_err*. Nilai dari *de\_err* ini merupakan nilai dari error sekarang dikurangi dengan error yang lama. Nilai keanggotaan untuk *de\_err* terlihat lebih linier. Bentuk

nilai juga merupakan perubahan nilai ADC yang diambil kemudian dikonversi menjadi derajat. Untuk *membership function* input *de\_err* dapat dilihat pada gambar 10.



**Gambar 10** Membership function input *de\_err*

Ada 5 buah *membership function* input baik untuk *err* maupun *de\_err*. *Negative Big (NB)*, *Negative Small (NS)*, *Zero (Z)*, *Positive Small (PS)* dan *Positive Big (PB)*. Sedangkan untuk *membership function* output dapat dilihat dari gambar dibawah ini:



**Gambar 11** Membership function output

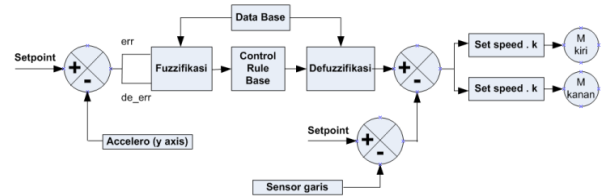
*Membership function* output tersebut terbagi menjadi dua bagian sisi *Forward* dan sisi *Reverse*, dan memiliki 5 buah *membership function* output yaitu, *Reverse Big (RB)*, *Reverse Small (RS)*, *Zero (Z)*, *Forward Small (FS)* dan *Forward Big (FB)*. Proses penentuan *membership function* output dilakukan dengan cara *try & error*. Tanda minus (-) pada *membership function* output merupakan penanda bahwa motor diberi respon yang berlawanan arah. Berikut ini tabel *rules FLC* yang di terapkan pada robot.

**Tabel 1** Rules FLC

<i>err</i> / <i>de_err</i>	N B	NS	Z	PS	PB
NB	FB	FB	FS	Z	RS
NS	FB	FS	Z	Z	RS
Z	FB	FS	Z	RS	RB
PS	FS	Z	Z	RS	RB
PB	FS	Z	RS	RB	RB

### 3.7. Sistem keseimbangan sambil mengikuti garis

Sistem perancangan yang terakhir untuk pencapaian hasil adalah proses bagaimana robot berjalan sambil mengikuti garis. Sistem kontrol *fuzzy logic* hanya diterapkan pada saat pengaturan keseimbangan. Pada saat robot mendeteksi garis maka output *fuzzy logic* tidak langsung diterapkan ke *speed* motor kanan dan kiri. Namun digunakanlah proses perbandingan, yaitu ketika robot ingin berbelok kekiri maka *speed kanan* dipercepat dan *speed* kiri diperlambat. Demikian pula untuk proses sebaliknya ketika robot ingin berbelok kanan maka *speed* motor kiri dipercepat dan *speed* motor kanan diperlambat. Digunakan perbandingan kecepatan motor kanan dan kiri agar robot dapat berbelok kekanan dan kekiri. Proses beloknya robot harus juga dengan memberikan respon terhadap gaya grafitasi kedepan bodi robot. Untuk bolg diagram sistem kontrol keseluruhan dapat dilihat pada gambar 3.14.



**Gambar 12.** Blok diagram sistem kontrol keseimbangan dengan mengikuti garis

Blok diagram sistem kontrol tersebut menerangkan bahwa sensor garis tidak difungsikan dalam sistem kontrol *fuzzy logic* namun hanya diletakkan di luar dari sistem kontrol. Dalam hal ini sensor garis hanya digunakan sebagai pengganggu dari sistem keseimbangan yang menyebabkan robot tidak bisa diam setimbang tanpa ada gerakan selama proses mengikuti garis. Robot akan bergerak 2 axis yaitu terhadap sumbu Y (*yaw*) ketika robot berbelok kekanan atau kekiri dan sumbu Z (*pitch*) ketika robot bergerak maju atau mundur.

## 4. Pengujian Alat

Bab ini berisi proses pengujian dan analisa terhadap alat sistem yang telah dirancang pada bab sebelumnya. Juga ditampilkan *snapshoot* dari proses mengikuti garis oleh robot.

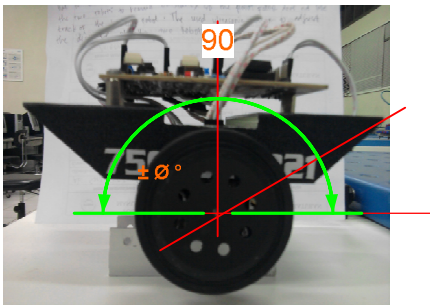
### 4.1. Pengujian dan Analisa sensor *accelerometer*

Sensor yang digunakan adalah *Hitachi H48C 3-Axis Accelerometer Module*. Data nilai yang diambil dari

sensor ini adalah hasil konversi tegangan analog menjadi digital. Pengambilan data dengan cara menampilkan nilai data pada LCD. Proses pengukuran dilakukan dengan dua cara yaitu pengujian terhadap sudut dan pengujian terhadap jarak.

4.1.1 Prosedur Pengujian

1. Tujuan  
Untuk mengetahui sensitivitas sensor terhadap kemiringan.
2. Peralatan
  - Board mikrokontroler + LCD
  - Sensor *hitachi H48C 3-Axis Accelerometer Module*.
  - Busur derajat
3. Prosedur  
Adapun langkah yang dilakukan adalah sebagai berikut:
  - Mempersiapkan peralatan berupa robot dengan dilengkapi sistem minimum, LCD dan sensor *accelerometer*.
  - Mempersiapkan perlengkapan berupa busur derajat.
  - Ukur posisi robot terhadap bumi lihat gambar 4.1
  - Proses pengambilan data saat robot dalam keadaan diam

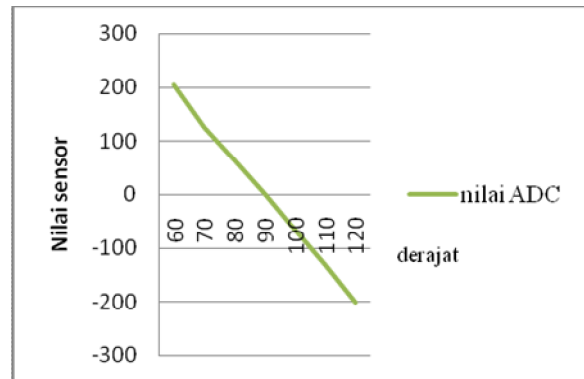


**Gambar 13** Proses pengambilan data *accelerometer* berdasarkan sudut

**Tabel 4.1** data hasil pengukuran berdasarkan sudut

Sudut (derajat)	60	70	80	90	100	110	120
Nilai	204	126	64	0	-64	-130	-200

Dari data yang ada maka dapat dibuat grafik, terlihat pada gambar 14.



**Gambar 14** Grafik data pengukuran berdasarkan sudut

4. Analisa data  
Dari data yang didapat ternyata perubahan nilai ADC pada sensor *accelerometer* hampir linier terhadap perubahan sudut. Hal ini bisa terjadi karena proses pengambilan data hanya menggunakan busur derajat proses menentukan sudut juga dilakukan secara visual.

4.2. Hasil pengujian robot mengikuti garis

Proses robot dalam mengikuti garis diuji coba dengan lintasan yang berupa garis lurus dan belokan. Lebar garis adalah 3 cm. Ukuran garis disesuaikan dengan konstruksi robot yang kecil. Arena robot berupa lantai datar yang terbuat dari karpet. Pengambilan gambar dilakukan dengan *cut video* berikut ditampilkan 15 gambar yang mewakili proses robot mengikuti garis.

5. Penutup

5.1. Kesimpulan

Setelah melakukan tahap perancangan dan pembuatan sistem yang kemudian dilanjutkan dengan tahap pengujian dan analisa maka dapat diambil kesimpulan sebagai berikut:

- Penerapan metode *fuzzy logic control* telah digunakan pada robot dan mampu mengendalikan robot agar selalu berada dalam kondisi setimbang.
- Terjadinya penurunan sumber tegangan akan berakibat respon motor menjadi lambat sehingga memungkinkan robot tidak dapat menjaga keseimbangannya.
- Perlu perlakuan khusus untuk photodiode karena setiap photodiode memiliki sensitivitas yang

berbeda yaitu dengan diberi cover penutup untuk menghindari cahaya dari luar.

- Semakin tinggi kecepatan robot dalam mengikuti garis maka robot menjadi mudah untuk keluar dari garis.

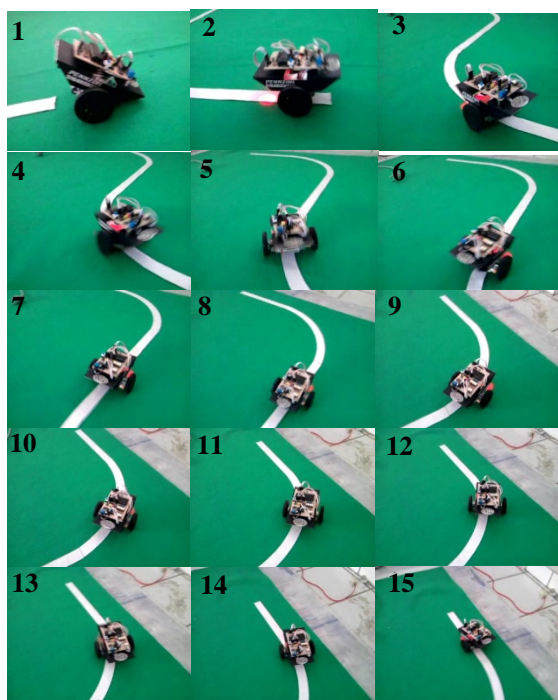
## 5.2. Saran

Dari hasil proyek akhir ini masih terdapat beberapa kekurangan dan dimungkinkan untuk melakukan pengembangan lebih lanjut. Beberapa saran yang perlu penulis sampaikan adalah sebagai berikut:

- Keadaan robot mengikuti garis hanya bisa dengan kondisi tikungan yang tidak tajam. Bila terdapat tikungan yang tajam maka memungkinkan robot untuk keluar line.
- Kemampuan robot menjaga keseimbangan hanya untuk bidang yang rata. Untuk penyempurnaan dapat dibuat arena yang mempunyai tanjakan.

## DAFTAR PUSTAKA

- [1] Khuswanto, Hendry. Kesetimbangan Robot Beroda Dua Menggunakan Metode Fuzzy Logic.
- [2] Heryanto, M. Ary. Adi P, Wisnu. Pemrograman Bahasa C untuk Mikrokontroler ATMEGA8535.
- [3] <http://www.digitdude.com/2010/01/rangkaian-parallax-hitachi-3g-tri-axis.html> (diakses 10 januari 2011)
- [4] [http://staff.ui.ac.id/internal/040603019/material/DCMotor PaperandQA.pdf](http://staff.ui.ac.id/internal/040603019/material/DCMotor%20PaperandQA.pdf) (diakses 20 juni 2011)
- [5] [http://leiwwww.epfl.ch/publications/grasser\\_darrigo\\_colombi\\_rufer\\_ieee\\_02.pdf](http://leiwwww.epfl.ch/publications/grasser_darrigo_colombi_rufer_ieee_02.pdf) (di akses desember 2010)
- [6] [http://www.parallax.com/dl/docs/prod/acc/HitachiH48C3Axis Accelerometer.pdf](http://www.parallax.com/dl/docs/prod/acc/HitachiH48C3Axis%20Accelerometer.pdf) (di akses desember 2010)



**Gambar 15** Snapshot robot mengikuti garis