Kendali Kecepatan dan Posisi *Automated Guide Vehicle* Menggunakan *Fuzzy Logic* dan *PID Control*

Speed And Position Control *Of Automated Guide Vehicle Using Fuzzy Logic And Pid Control*

M Darfyma Putra[1], Dudi Darmawan[2] , Angga Rusdinar[3]

[1,2,3]Prodi S1 Teknik Fisika, Fakultas Teknik Elektro, Universitas Telkom

[1]darfymadani@gmail.com, [2]dudiddw@gmail.co, [3]angga22002@yahoo.com,

Abstrak

Secara umum Automated Guided Vehicle (AGV) adalah sebuah *mobile* robot pengangkut barang yang mengikuti petunjuk atau jalur berupa garis. Saat ini AGV terus dikembangkan dan disempurnakan, dan masih perlu banyak pengembangan yang lebih mendalam, untuk beberapa kasus AGV mengalami kesulitan pada kendali kecepatanya dikarenakan load yang berubah, maka diperlukan sebuah sistem yang dapat mempertahankan kecepatan AGV meski beban yang dibawa mengalami perubahan. Hasil pengujian tenyata sistem kontrol Fuzzy dan PID masih belum optimal dalam perancangannya dikarenakan tidak bisa semua beban membuat AGV dibawa dengan kecepatan konstan, hanya beban yang berada di rentang nilai 0-2,5 kg.

Kata Kunci : Automated Guide Vehicle, Fuzzy Logic, *PID Control*

Abstract

Generally Automated Guided Vehicle (AGV) is a mobile robot trolley who followed the instructions or track by line. AGV is currently being developed and refined, and still needs a lot more in-depth development, for some cases AGV is difficult to control the velocity due to load changes, it needs a system that can maintain velocity AGV although the load carried changes. After the test Fuzzy and PID control system is still not optimal in its design because not all the load of making AGV can move constantly, just load that is in the range of values 0-2,5Kg Only.

Keywords : Automated Guide Vehicle, Fuzzy Logic, PID Control

1. Pendahuluan

Secara umum *Automatic Guided Vehicle* (AGV) adalah sebuah robot yang bergerak dengan roda penggerak, sebagai pengangkut barang dan bergerak mengikuti petunjuk garis yang berada di lantai, dan dapat menuju tempat yang dituju secara otomatis, AGV juga baik dalam pengangkutan reguler dari beban yang stabil. Dari sedikit gambaran pengertian ini, timbulah pertanyaan bagaimana dengan proses produksi yang bebannya tidak stabil (berubah-ubah), bisa bertambah sewaktu waktu, atau berkurang. Jika sebuah AGV memiliki beban yang terus berubah, tentunya terjadi perubahan kecepatan, AGV akan mengalami perlambatan, hal ini tentu akan menggangu proses produksi. Oleh karena itu dibutuhkannya AGV yang memiliki kecepatan konstan meskipun *load* yang diterima terus berubah-ubah, *load* sebagai gangguan sistem robot. Robot untuk memiliki kecerdasan tertentu maka mebutuhkan kontrol yang baik, kontrol yang sudah banyak kita kenal antara lain adalah *PID Control*, *Fuzzy Logic*.[24]Untuk saling mengoptimalkan antara dua janis kontrol tersebut, maka ditawarkan adanya penggabungan kontrol yang menggunakan gabungan antara *PID Control* dan *Fuzzy Logic* yang diterapkan dalam sistem AGV.

2. Dasar Teori

2.1 Konsep Dasar Robot

Pada hakikatnya robot mengandung beberapa unsur, yaitu: *Programmable*, *Automatic* (otomatis), *Manipulator*(perangkat manipulasi), *Humanoid* (mempunyai kemiripan dengan manusia). Dari unsur-unsur di atas jelaslah bahwa robot bukan hanya sekedar perkakas biasa. Namun merupakan mesin khusus yang dapat dikontrol oleh manusia lewat suatu *processor* atau kontroler.[1] Terdapat dua definisi yang dapat diterima dikalangan industri mengenai robot, yaitu:

1. Menurut *Robotic Institute of America* (RIA), robot adalah "*Manipulator* yang berfungsi jamak dan dapat diprogram ulang dan dirancang untuk mengangkut material, *part*, peralatan atau perangkat khusus melalui peubah Posisi terprogram untuk melakukan tugas bervariasi".[1]
2. Robot merupakan peralatan yang melakukan fungsi-fungsi yang biasa dilakukan oleh manusia, atau peralatan yang bekerja dengan kecerdasan yang mirip dengan kecerdasan manusia.[1]

2.2 Mikrokontroller Atmega128

ATmega128A termasuk dalam keluarga Atmel, merupakan kelas 8-bit mikrokontroler. Memiliki kemampuan mengeksekusi instruksi dalam satu siklus clock tunggal, dan dapat ditambahakan clock eksternal sesaui dengan kebutuhan, Berikut adalah fitur yang terdapat pada ATMega 128A :

1. Low Power AVR 8-bit mikrokontroller
2. 128Kbytes Flash Program Memory
3. 4 Kbytes EEPROM
4. 4 Kbytes SRAM
5. Two 8-bit Timer/Counters
6. Real Time Counter with Separate Oscillator
7. Two 8-bit PWM Channels
8. 6 PWM Channels with Programmable Resolutionfrom 2 to 16 Bits
9. 8-channel, 10-bit ADC
10. 8 Single-ended Channels
11. 7 Differential Channels
12. 53 Programmable I/O Lines
13. Operating Voltages 2.7 - 5.5V
14. Speed Grades 16MHz [12]

2.3 Mikrokontroller ATmega8

ATmega8 dengan 8-bit mikrokontroler berbasis AVR. Dengan arsitektur RISC (*Reduced Instruction Set Computing*). Berikut adalah fitur yang terdapat pada ATMega 8:

9. 512Bytes EEPROM
10. 1Kbytes Internal SRAM
11. Two 8-bit Timer/Counters
12. One 16-bit Timer/Counter
13. Three PWM Channels
14. 8-channel, 10-bit ADC
15. 23 Programmable I/O Lines
16. Operating Voltages 2.7 - 5.5V
17. Speed Grades 0 - 16MHz [15]

2.4 *Encoder*

Encoder adalah sensor yang menghasilkan sinyal digital dalam menanggapi gerakan. Kedua poros *encoder*, yang menanggapi rotasi, dan *linier encoder*, yang menanggapi gerak dalam garis, yang tersedia[8]. *Encoder* sangat berguna untuk mengukur kecepatan motor maupun jarak yang dilalui oleh motor.

2.5 *Kontrol Fuzzy*

Suatu sistem berbasis aturan fuzzy yang lengkap terdiri dari tiga komponen utama: ***Fuzzyfication,Inference dan Defuzzification***. *Fuzzyfication* mengubah masukan-masukan yang nilai kebenarannya bersifat pasti (*crispinput*) ke dalam bentuk *fuzzy input*, yang berupa nilai*linguistic* yang semantiknya ditentukan berdasarkan fungsi keanggotaan tertentu. *Inference* melakuakn penalaran menggunakan *fuzzy input* dan *fuzzy rules* yang telah ditentukan sehingga menghasilkan *fuzzy output*. Sedangkan *deffuzyfication* mengubah *fuzzy output* menjadi *crisp rule* berdasarkan fungsi keanggotaan yang telah ditentukan.[17]

2.6 Kontrol PID

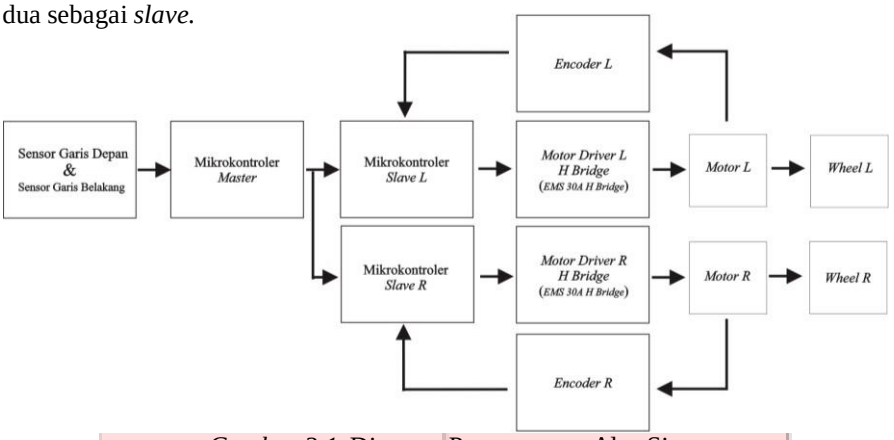
Kontrol PID adalah nama umum yang diberikan untuk tiga-istilah kontrol. PID *mnemonic* mengacu ke huruf pertama dari nama-nama istilah individu yang membentuk standar tiga-istilah kontrol. Ini adalah P untuk istilah proportional, I untuk istilah integral dan D untuk istilah derivative di kontrol. Interpretasi pengendali operator s-domain dan hubungan antara domain waktu dan domain frekuensi memasuki pembahasan PID kinerja pengendali.

Dari tiga istilah P, I dan D memiliki perannya masing-masing dalam kontrol *PID*. Kendali *Proporsional* akan menurunkan *Rise Time*, Meningkatkan *Overshoot* dan Menurunkan atau mengurangi *Steady-State Error*. Untuk Kendali *Integral* akan Menurunkan *Rise Time*, Meningkatkan *Overshoot*, Meningkatkan *Settling Time* dan Mengurangi *Steady-State Error*. Sedangkan Kendali *Derivatif* akan menurunkan *Overshoot* dan Menurunkan *Settling Time*. Dengan merubah variabel kendali P, I, dan D maka akan merubah hasil akhir dari kendali PID tersebut sesuai dengan perannya masing-masing.

2. Pembahasan

3.1 Perancangan Alur Sistem

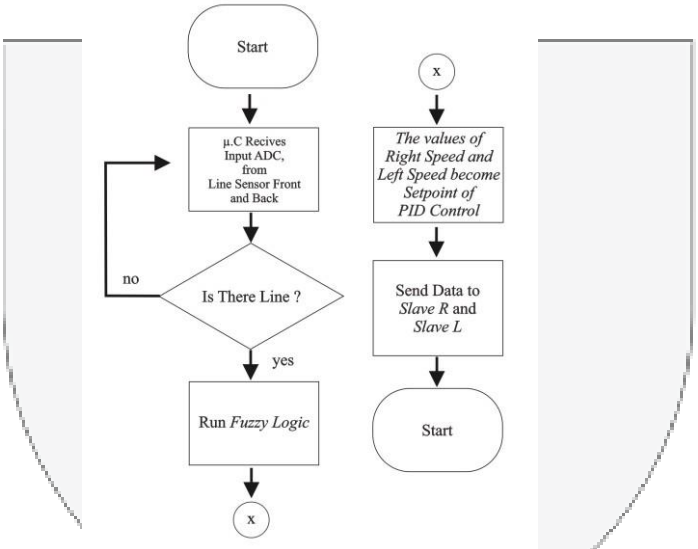
Berikut adalah perancangan sistem secara keseluruhan, menggunakan tiga mikrokontroler, satu sabagi *master* dan dua sebagai *slave.*



Gambar.3.1. Diagram Perancangan Alur Sistem

3.2 Perancangan Sistem Minimum Master

Sistem minimum adalah rangkaian yang digunakan untuk mendukung kerja suatu mikrokontroller sesuai dengan kebutuhan yang kita inginkan. Komponen pendukung tersebut berupa rangkaian pembangkit kristal (clock), rangkaian reset, sumber tegangan 5V, Input/Output, dan ISP (*In-System Programmer*) downloader.

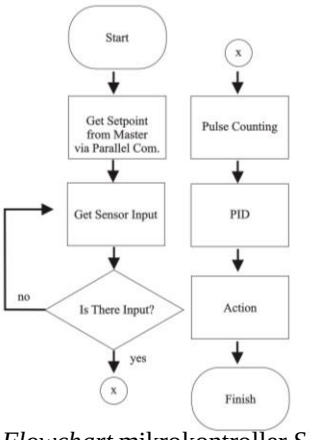


Gambar 3.2.Flowchart mikrokontroller master

Mikrokontroller menggunakan chip Atmega 128A, dengan total 53 pin input/output, 4 pin timer/counter, 8 pin PWM (Pulse With Modulation) Channel, 8 pin ADC channel, 2 Serial USARTs, watchdog timer, dan analog comparator. Namun dari banyaknya pin input output tidak semua digunakan melainkan hanya beberapa, diantaranya beberapa pin yang digunakan adalah 6 pin untuk akses motor driver (PB.1, PB.2, PB.3, PB.4, PB.5, PB.6), 3 pin adc (PF.0, PF.1, PF.2), 3 pin selector mux (PA.4, PA.5, PA.6), 4 pin serial (PE.0, PE.1, PD.2, P.D3). Skema perancangan dilampirkan secara lengkap pada bagian lampiran.

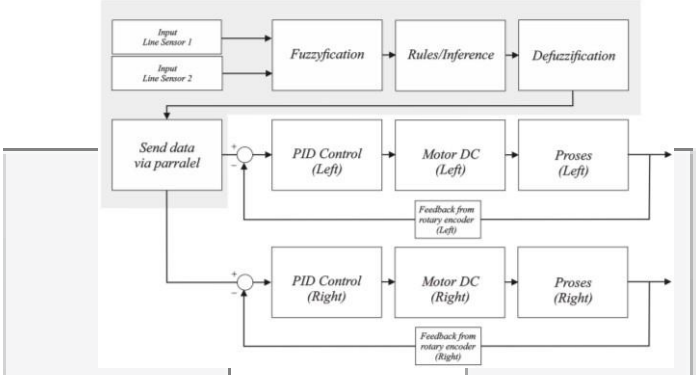
3.3 Perancangan Sistem Minimum *Slave & Slave II*

Pada sistem minimum *slave dan slave II* ada banyak kesamaan penggunaan pin, adapun seperti yang telah kita ketahui pada landasan teori Atmega 8 yang memiliki 23 pin input/output dan beberapa pin yang digunakan adalah 2 pin serial (PD.0, & PD.1), 1 pin interupt (PD.2), 1 pin output dan satu pin input (PD.3 & PD.4) sebagai komunikasi antara slave yang biasa kita sebut sebagai komunikasi handsake untuk antrian data yang dikirim. Fitur yang digunakan pada slave adalah interupt timer, untuk counting jumlah pulsa yang dibaca oleh encoder,



Gambar 3.3.Flowchart mikrokontroller *Slave L* dan *Slave R*

Slave L dan *Slave R* memiliki program yang identik, mulai dari konstanta *gain*nya meliputi nilai Kp=2, Ki=0.05, Kd=0.6, PWM awal diset 23 (dari 255), dan waktu *interupt* timer setiap satu detik, proses mendapatkan nilai *gain* dari cara tuning dan melihat respon secara langsung pada robot. Dengan menggunakan keadaan PWM awal sebesar itu maka robot mampu bergerak dengan kecepatan normal (tanpa beban) sekitar 0,364 m/s, dan beban maksimum dengan PWM awalan sebesar itu adalah 20 kg.



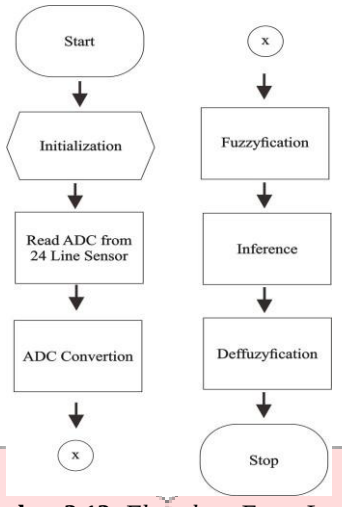
*Gambar 3.4.*Diagram blok hubungan antara *Fuzzy Logic* dan *PID Control*

3.4. Perancangan Sensor Garis

Sensor garis adalah sensor yang dapat membaca perbedaan gelap terang pada suatu bidang. Photodioda dan LED merupakan komponen utama dari sensor garis ini, prinsip kerjanya adalah memanfaatkan pantulan sinar dari LED, pantulan tersebut mengenai bidang pantul, dan terpantul menuju photodioda hal ini lah yang akan memberikan efek nilai resistansi yang berbeda-beda pada photodioda, pantulan pada bidang terang akan sangat berbeda dengan pantulan di bidang gelap.Sensor garis ini menggunakan 24 buah photodioda dan LED sebagai input sensor ke mikrokontroller master, menggunakan port ADC (Analog Digital Converter) pada mikrokontroller master atau Atmega 128. Namun dikarenakan mikrokontroller master hanya memiliki 8 port ADC maka digunakan *multiplexer* sebanyak tiga buah, dan masuk ke dalam tiga port ADC saja.Bentuk sensor dibuat memanjang terbagi menjadi 2 baris sensor, depan dan belakang, dan diletakan pada bagian bawah robot agar garis dapat terbaca pada lantai. Pemilihan bentuk ini untuk mengetahui tingkat kemiringan garis yang dilewati robot, memeperbaiki sudut kemiringan robot dan dapat dengan cepat menyesuaikan robot berada ditengah garis, sensor inilah yang nantinya akan menjadi input kontrol fuzzy.

3.5. Perancangan Logika Fuzzy

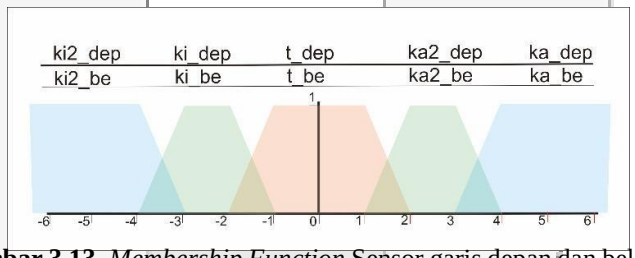
Pada sistem ini logika fuzzy menerima *input* bagian depan dan belakang sensor garis, yaitu 12 depan dan 12 belakang, dengan keluaran berupa kecepatan/ pulsa yang harus dicapai PID motor kiri dan PID motor kanan nantinya, atau bisa kita sebut sebagai set point dari masing masing motor. Logika fuzzy secara umum digunakan sebegai kontrol posisi dari robot agar tetap berada pada jalurnya. Berikut adalah *flowchart* yang menjadi dasar pembuatan algoritma dalam mikrokontroller:



Gambar 3.13. *Flowchart* FuzzyLogic.

3.5.1 *Fuzzyfication*

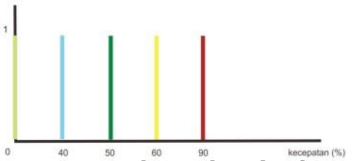
Sebelum melewati tahapan ini yang harus dilakukan adalah mikrokontroller master harus melakukan inisialisasi berupa, proses pembacaan 24 ADC dengan menggunakan *multyplexer*. Memanfaatkan 3 buah pin *selector*, maka mulltyplexer akan membuka tutup gerbang dan melakukan proses pembacaan pada tiap ADC secara bergantian dan kemudian ditampung dalam satu variabel penampung sesuai dengan jumlah sensornya.Seperti yang telah kita ketahui, proses *fuzzyfication* atau fuzifikasi akan mengubah besaran numerik menjadi besaran linguistik, dengan cara memasukan nilai sensor ke dalam suatu *membership function*, di dalam membership function ini setiap data sensor akan diklasifikasikan batas-batasnya, nantinya akan menghasilkan beberapa variabel linguistik dan fungsi.Kemudian proses yang dilakukan adalah, dengan menetukan berapa banyaknya *variabel linguistik* yang kita inginkan, setelah itu didapat nilai derajat keanggotaan dari masing masing variabel. Untuk mendapatkan nilai derajat keanggotaan dengan menggunakan fungsi persamaan garis yang berupa trapesium.Dari sensor garis yang dibuat, dibentuk beberapa nilai linguistik, yaitu *ki2_dep, ki_dep, t_dep, ka_dep, dan ka2_dep,* untuk nilai linguistik sensor depan, dan belakang disamakan jumlah variabel linguistiknya sehingga didapatkan *ki2_be, ki_be, t_be, ka_be, dan ka2_be*



Gambar 3.13. *Membership Function* Sensor garis depan dan belakang.

Untuk keluaran sistem menggunakan model Sugeno, dikarenakan model sugeno lebih sederhana dan mudah dalam komputasi dalam mikrokontrollernya. Sugeno menggunakan bentuk Singletone. Singletone sendiri adalah nilai 1 pada keluaran yang diinginkan dan 0 pada nilai yang tidak diinginkan.

Untuk keluaran yang saya gunakan adalah nilai setpoint, yang nantinya akan digunakan sebagai faktor koreksi dengan bacaan nilai sensor encoder, nilai setpoint PID kiri dan PID kanan dengan bigitu kecepatan yang diinginkan oleh fuzzy tiap rodanya akan lebih sesuai dengan kenyataan. Pada **Gambar 3.14** dibawah menggambarkan output dari fuzzy memiliki 5 nilai keluaran linguistik, yaitu Break, Slow, Medium, Fast, VeryFast untuk masing masing PID kiri dan kanan.



Gambar 3.14. *Output untuk PID kiri dan kanan dari Fuzzy*

3.5.2 *Inference*

Dalam proses *inference* data olah dari input *fuzzification* dengan hasil keluar yang dikehendaki dengan aturan aturan yang diinginkan, aturan tersebut menentukan pergerakan atau respon robot terhadap bacaan sensor garis yang digunakan. Berikut adalah aturan aturan yang telah dibentuk dalam mikrokontroller master.

Tabel 3.1. Rule Fuzzy Logic

		Depan				
		ki2_dep	ki_dep	t_dep	ka_dep	ka2_dep
Belakang	ki2_be	s/v.f	b/f	v.f/m	m/s	v.f/b
	ki_be	s/v.f	b/m	f/m	m/s	v.f/b
	t_be	b/v.f	b/m	m/m	m/b	v.f/b
	ka_be	b/v.f	s/m	m/f	m/b	v.f/s
	ka2_be	b/v.f	s/m	m/v.f	f/b	v.f/s

3.5.3 *Deffuzyfication*

Pada proses ini, merupakan proses pemngembalian nilai linguistik yang berasal dari rule menuju besaran crisp, agar bisa dipahami oleh sistem kontrol PID. Metode *Weight Average* adalah metoda yang digunakan dalam penelitian ini, dan output berupa jumlah pulsa yang harus dicapai motor DC.

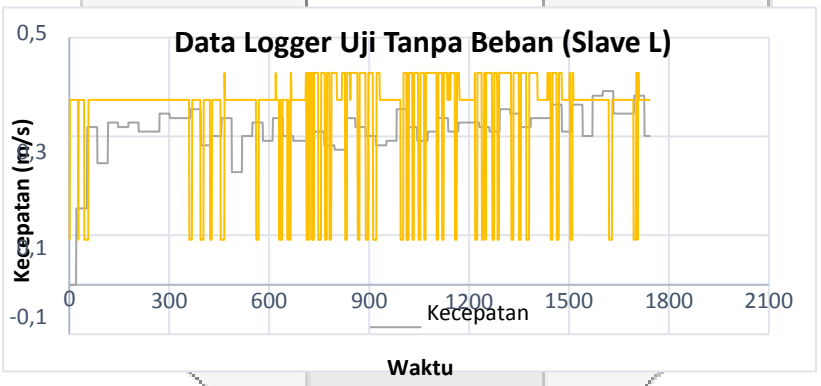
2.6 Pengujian

3.6.1 Pengujian Fuzzy Logic dan PID Control

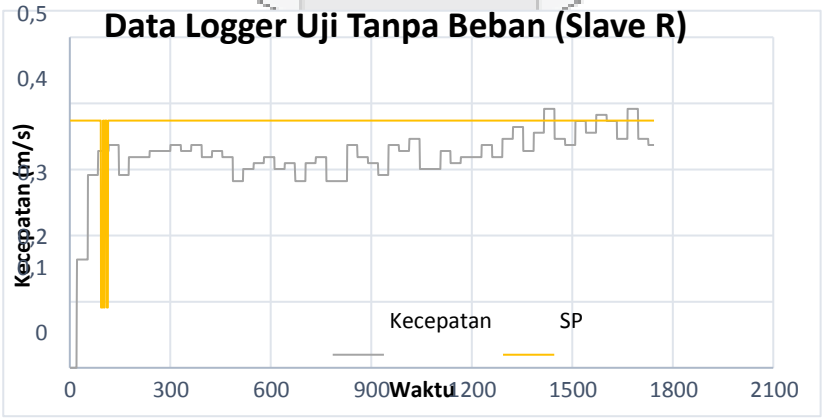
Pengujian tanpa *load* adalah pengujian yang diharapkan dengan laju paling maksismum, berikut ini data yang kita dapatkan dalam pengujian tanpa *load* :

Tabel 4. 1. Hasil Pengujian Tanpa *Load*

Jarak	12	Meter	Sumber : Video VID_20160913_174036	
V sistem	11,9	Volt	Finish	Start
V motor	24,2	Volt	0:01:05	0:00:18
T	0:00:47	Second		



Gambar 4. 1. Hasil yang didapatkan dari mikrokontroler *Slave L*



Gambar 4. 2. Hasil yang didapatkan dari mikrokontroler *Slave R*

3. Kesimpulan

1. Sistem *Fuzzy Logic* telah berhasil digunakan dalam sensor garis dua baris dan telah berjalan dengan baik untuk mengikuti garis pada pengujian tanpa beban, terlihat proses susur garis paling cepat pada keadaan tidak ada beban, dengan respon perubahan *setpoint* selalu berubah ubah menyesuaikan pembacaan sensor garis dengan waktu tempuh yang dicapai 57 detik dijarak 12m dan dengan tegangan batrai 11,9v (sistem) dan 24,2v (motor).
2. Sensor *encoder* FC-03 Memiliki ketelitian yang berbeda antara pembacaan kecepatan rendah hingga tinggi dengan pembacaan kecepatan tinggi hingga rendah. Untuk pembacaan dengan kecepatan rendah hingga tinggi, Ketelitian *Encoder* kiri mencapai 73,82 %, dan kanan lebih baik 96,84 %. Kemudian pembacaan dengan kecepatan tinggi hingga rendah *encoder* kiri mencapai ketelitian 97,77 % dan yang kanan mencapai 97,29 %, dari segi ketelitian memang terlihat counting turun lebih teliti, namun secara respon terlihat kurang berdasarkan dari gradien grafik.
3. Metoda *Fuzzy Logic* dan *PID Control* telah bekerja dengan baik dapat dibuktikan dengan pengujian sistem *Fuzzy Logic* dan *PID Control*. Dengan PWM awal=40, Nilai Kp=1, Ki=0.01, dan Kd=0.3 dan dengan *load* hanya direntang 0 sampai 3 *Load* (0-2.5 kg) berdasarkan percobaan.

Daftar Pustaka :

[1] Wikipediawan. (2013), Robot, http://id.wikipedia.org/wiki/Robot

[2] Comparison between PID and fuzzy control, Internal Report R93.36I

[3] Priyono, Agung. 2014. Perancangan dan Implementasi *One Steered Traction Wheel Robot Circular Line Sensor* menggunakan Kontrol Logika Fuzzy. Bandung: Universitas Telkom

[4] Bagus, Ivan.(2014),Generator DC dan Motor DC, https://kaptenivan.wordpress.com/2014/11/05/generator-dc-dan-motor-dc-2/

[5] Electronics, Inovatice.(2007), EMS 30 A H-Bridge, http://www.innovativeelectronics.com/innovative_electronics/download_files/manual/EMS_30A_HBridge_manual.pdf

[6] Al Williams.(2002). *Microcontroller projects using the Basic Stamp* (2nd ed.). Focal Press. p. 344. ISBN 978-1-57820-101-3.

[7] Tantos, Andras.(2012), H-Bridges-The basics, http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/

[8] Fahmizal.(2010), Merancang Rangkaian Sensor Garis, https://fahmizaleeits.wordpress.com/tag/cara-kerja-sensor-garis/

[9] Mech, Hades.(2010), Rotary Encoder, http://hades.mech.northwestern.edu/index.php/Rotary_Encoder

[10] Eitel, Elisabeth.(2014), Basic Rotary Encoder: Overview and New Technologies, http://machinedesign.com/sensors/basics-rotary-encoders-overview-and-new-technologies-0

[11] Mechatronicscrew.(2014), Pengantar Mikrokontroler, https://mechatronicscrew.wordpress.com/praktikum/praktikum-mekatronika/mikrokontroler/

[12] Atmel Datasheet. Microkontroller ATmega128A

[13] Image: http://www.bdspeedytek.com/product/atmega128a/

[14] Atmel Datasheet. Microkontroller Atmega8

[15] Image: http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8_l_datasheet.pdf

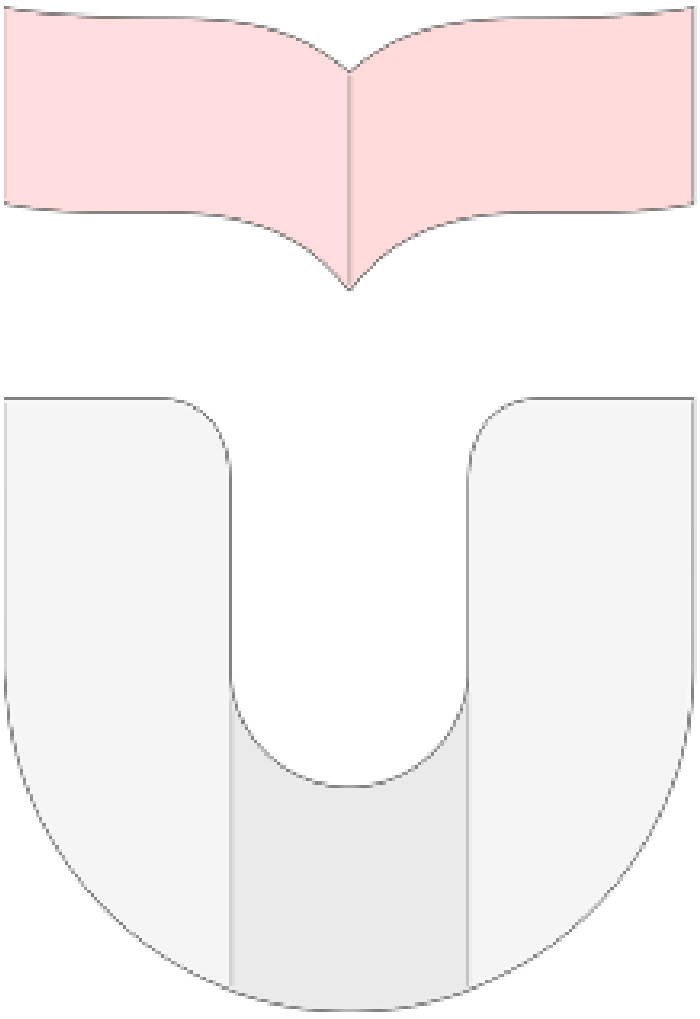
[16] Rusiana, Endang. (2011), Fuzzy Logic, http://wangready.files.wordpress.com

[17] Suyanto. 2007. Artificial Intelegent. Bandung: Informatika Bandung

[18] K. J. Astrom and R. M. Murray, "Control PID," in *Feedback Systems: An Introduction for Scientists and Engineers*, 2012, pp. 293-314.

[19] Jurnal Ilmiah Ali Muhammad "Pembelajaran Perancangan Sistem Kontrol PID dengan Software MATLAB"

[20] Ilman Arzak, Mochamad : "Disain dan Implementasi Pengendali Kursi Roda Menggunakan Sinyal EEG berbasis Mikrokontroler". Bandung : Universitas Telkom, 2012.

[21] Amanda, Iyodha : "Kendali Kecepatan Mobil Listrik Menggunakan Dua Motor Listrik dengan Fuzzy-PID". Bandung : Universitas Telkom, 2012.

Widayanto, Dian Wahyu. 2012. Desain dan Implementasi Robot Mobil Pengikut Benda Bergerak Dengan Kontrol Logika Fuzzy Berbasis Mikrokontroler ATMega128. Bandung: IT Telkom