

Kendali Logika Fuzzy pada *Car Like Mobile Robot* (CLMR) Penjejak Garis

FAHMIZAL¹, BUDI BAYU MURTI², DONNY BUDI PRATAMA³,
AFRIZAL MAYUB⁴

^{1,2,3} Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada

⁴ Pascasarjana S2 Pendidikan IPA FKIP, Universitas Bengkulu

Email: fahmizal@ugm.ac.id

Received 3 September 2018 | *Revised* 17 September 2018 | *Accepted* 30 September 2018

ABSTRAK

Makalah ini memaparkan perancangan sistem kendali logika fuzzy untuk mengatur kecepatan dan arah sudut steering pada car like mobile robot (CLMR) dengan menggunakan metode Ackermann steering. CLMR penjejak garis dirancang menggunakan 16 buah photodiode, dan terdapat 7 buah membership fuzzifikasi dari pembacaan error dan last error sehingga terbentuk 49 aturan. Untuk menguji performa kendali fuzzy pada sistem CLMR dalam mengikuti lintasan garis maka dilakukan pengujian dengan bentuk lintasan berupa garis lurus dan berbelok serta zig-zag dalam satu lintasan putar. Proses variasi nilai keanggotaan fuzzifikasi masukan dan defuzzifikasi keluaran dilakukan sebanyak lima kali. Dari hasil pengujian diperoleh bahwa kendali logika fuzzy yang diaplikasikan pada sistem mampu membuat pergerakan CLMR sukses mengikuti lintasan uji selama 9,38 detik lebih baik 0,53 detik dari kendali PID. Selanjutnya, hasil rancangan sistem CLMR ini merupakan sebuah prototipe self-driving car.

Kata kunci: *car like mobile robot, robot penjejak garis, fuzzy, self-driving car*

ABSTRACT

This paper describes the design of a fuzzy logic control system to adjust the speed and direction of the angle of the steering on the car like mobile robot (CLMR) using the Ackermann steering method. CLMR line tracking is designed using 16 photodiode pieces, and there are 7 fuzzification membership from reading error and last error so that 49 rules are formed. To test the fuzzy control performance on the CLMR system in following the line trajectory, it was tested with the form of a straight line and a turn and a zigzag in a rotary track. The process of varying input membership fuzzification values and output defuzzification is done five times. From the test results, it was found that the fuzzy logic control applied to the system was able to make CLMR movement successfully followed the test path for 9.38 seconds better than 0.53 seconds of PID control. Furthermore, the results of the CLMR system design are a prototype self-driving car.

Keywords: *car like mobile robot, line tracking robot, fuzzy, self-driving car*

1. PENDAHULUAN

Perkembangan teknologi yang semakin maju pada bidang robotika telah menjadikan bidang ini sebagai salah satu bidang yang sangat populer dan mulai banyak dikembangkan untuk berbagai aspek kehidupan manusia. Robotika merupakan perpaduan antara mekanika dan elektronika yang disebut dengan mekatronika **(Angeles, 2002)**. Sistem mekatronika ini telah banyak dikembangkan pada bidang industri, otomotif, pertanian, dan masih banyak lagi bidang lainnya. Bukan itu saja, bidang robotika memang sangat bermanfaat dan dapat memudahkan serta memperlancarkan kegiatan manusia, selain itu banyak juga yang mengaplikasikan robotika sebagai hobi, ajang kompetisi, dan penelitian.

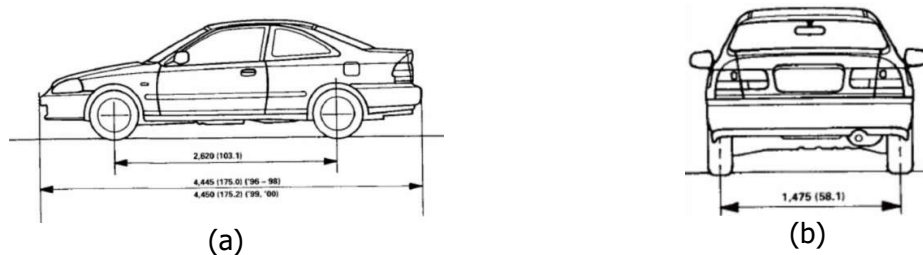
Salah satu bidang robotika yang mengalami perkembangan pesat akhir ini adalah bidang otomotif lebih spesifik lagi ialah *self-driving car*. NVIDIA adalah salah satu pihak yang telah mengembangkan sistem *self-driving car* **(Bojarski, 2016)**. Semua kemajuan teknologi tersebut berfokus pada keamanan, kenyamanan dan efisiensi **(Maslak & Butkiewicz, 2013)**. Indonesia merupakan suatu negara dengan jumlah kendaraan mobil terbanyak di Asia Tenggara. Kedepan akan menjadi suatu pangsa pasar terbesar bagi perusahaan *autonomous self-driving car* untuk memasarkan kendaraan tersebut. Oleh karena itu, hal ini mendorong penulis untuk melakukan pembuatan prototipe *self-driving car* yang nantinya akan menjadi cikal bakal sistem *self-driving car* di Indonesia. Sistem prototipe *self-driving car* yang dirancang sedemikian rupa seperti pada sistem *real car*. Rancangan prototipe *self-driving car* menggunakan sistem *Ackerman steering* **(Hunt, 1978)**. Prototipe ini akan dikategorikan kedalam *Car Like Mobile Robot* (CLMR). Dengan adanya prototipe robot mobil ini dapat digunakan sebagai pengganti *real car* dalam proses penerapan *self-driving car*.

Penelitian menggunakan CLMR telah banyak dilakukan oleh beberapa peneliti seperti pada **(Baturone et al. 2004; Chiu et al. 2005; El-Khatib & Hamilton, 2006; Guccro et al. 2014; Liu et al. 2011; Ouadah et al. 2006; Scicluna et al. 2012; Abatari et al. 2013)**. Pada makalah ini memaparkan suatu implementasi CLMR dengan *Ackerman steering* pada sistem robot penjejak garis. Robot ini bergerak mengikuti lintasan berupa garis lurus atau belok **(Hasan & Mamun, 2012)**. Salah satu kegunaan robot penjejak garis dapat diaplikasikan sebagai pengangkut barang, agar barang dapat ditunjukkan ke tempat tujuannya dengan melintasi garis lintasan. Akan tetapi dari pengendalian robot tersebut pasti memiliki kendala yaitu pergerakan robot yang tidak stabil dalam menjejak lintasan yang ada, sehingga suatu sistem pasti memerlukan pengendali agar pergerakan robot saat dioperasikan dapat sesuai dengan yang diharapkan.

Umumnya jenis kendali seperti *Proporsional-Integratif-Derivatif* (PID) lebih sering digunakan pada robot penjejak garis seperti pada **(Nath et al. 2013; Normey-Rico, 2001)** namun pada penelitian ini menerapkan kendali cerdas seperti logika fuzzy untuk diterapkan pada CLMR. Dengan menggunakan metode *Ackerman steering*, desain *steering* CLMR akan dipaparkan pada makalah ini. Selanjutnya, kendali logika fuzzy pada penelitian ini digunakan pada pengaturan kendali kecepatan putar roda belakang dari CLMR dan kendali *steering* untuk roda bagian depan CLMR. Sebagai masukan untuk proses fuzzifikasi diperoleh dari hasil pembacaan sensor garis dengan menggunakan *photodiode* sebanyak 16 *channel* menggunakan sistem multiplexer. Terdapat 49 aturan yang digunakan pada proses *inference rules* logika fuzzy pada sistem ini. Selanjutnya, dua buah keluaran dari proses defuzzifikasi kecepatan putar roda belakang dan sudut derajat *steering* dari roda bagian depan diselesaikan dengan metode *weight average* (WA).

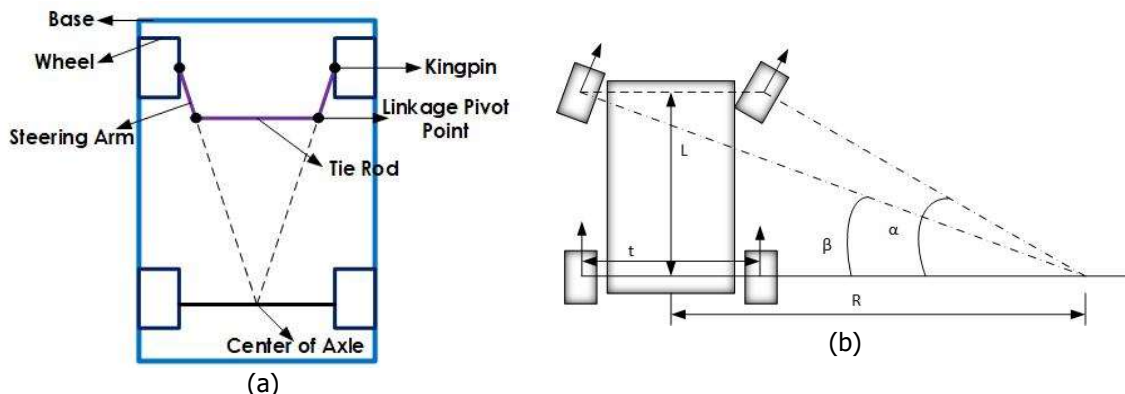
2. CAR LIKE MOBILE ROBOT (CLMR)

Definisi CLMR adalah suatu desain robot yang sistem kemudinya mengikuti sistem pada *car* (kendaraan mobil). Umumnya desain kendaraan mobil yang dimaksud disini ialah memiliki empat buah roda yang diposisikan di depan dua buah roda dan dua buah roda di belakang. Roda bagian depan dan roda bagian belakang memiliki jarak tertentu begitu juga dengan roda bagian kanan dan roda bagian kiri. Roda mobil bagian depan memiliki fungsi sebagai penentu arah gerak mobil sedangkan roda bagian belakang berfungsi sebagai penentu gerakan dan kecepatan laju mobil, hal tersebut yang disebut dengan *two wheel drive*. Untuk merancang bagian bodi CLMR maka diperlukan suatu acuan. Gambar 1 adalah desain atau *blueprint* dari mobil Honda Civic dalam satuan meter. Pada penelitian ini, desain dari CLMR yang dirancang akan mengacu pada *blueprint* dari mobil Honda Civic.



Gambar 1. *Blueprint* Mobil Honda Civic, (a). Tampak Samping; (b). Tampak Belakang

Ackermann steering geometry adalah susunan geometris dalam kemudi mobil atau kendaraan lain yang dirancang untuk memecahkan masalah roda di bagian dalam dan di luar dari belokan yang mampu menelusuri lingkaran dengan radius yang berbeda (**Angeles, 2002**). Ilustrasi pada Gambar 2(a) merupakan desain dasar yang sering digunakan pada mobil. Geometri *Ackermann* terdiri dari *kingpins*, *steering arms*, *linkage pivot point*, dan *tie rod*. *Kingpins* merupakan *ball joint* yang berfungsi untuk sumbu putar kemudi yang terhubung dengan setiap roda bagian depan, *steering arms* merupakan perpanjangan penghubung *tie rod* dan juga sebagai penentu perbedaan sudut putar antara roda kanan dan roda kiri. *Linkage pivot point* merupakan *ball joint* sebagai sumbu putar penghubung antara *steering arms* dan *tie rod*.



Gambar 2. (a). Ilustrasi Geometri *Ackermann* (b). Radius Putar pada Kemudi *Ackermann*

Sudut untuk *steering arms* akan ditentukan berdasarkan *center of axle*. *Axle* merupakan garis lurus yang menghubungkan antara roda bagian kanan dengan roda bagian kiri. Jika diperhatikan pada Gambar 2(a) kedua *steering arms* dari roda kanan dan roda kiri akan mengarah ke titik *center of axle*. Terdapat beberapa hal yang perlu dipertimbangkan saat

merancang *Ackermann*, hal pertama yang perlu dipertimbangkan adalah panjang dari jarak antara roda depan dan roda belakang atau *wheel base* dan yang kedua adalah jarak antar *ball joints* atau *kingpins* di bagian roda kiri dan kanan depan. Hal lain yang perlu dipertimbangkan adalah dimana *steering arms* akan diposisikan, baik itu diposisikan di depan dari sumbu kemudi maupun di belakang dari sumbu kemudi. *Ackermann* akan sangat sulit untuk dihasilkan dengan penempatan *steering arms* pada bagian depan, sehingga penempatan *steering arms* akan ditempatkan di belakang dari sumbu kemudi atau disebut dengan *rear steering arms*.

Dengan memberikan sudut pada sumbu *kingpins* yang berada di sepanjang *steering arms*, maka *Ackermann* aktual dapat dihasilkan sesuai dengan yang dibutuhkan. Terdapat beberapa hal yang perlu dipertimbangkan, yaitu jarak antara *axle* bagian depan dengan *axle* bagian belakang atau *wheelbase* (L) dan jarak antara roda bagian kanan dan roda bagian kiri. Dengan mengetahui parameter tersebut maka dapat dihitung sudut belokan *Ackermann*. Gambar 2(b) berikut merupakan ilustrasi kalkulasi dari geometri *Ackermann*.

Berdasarkan pada Gambar 2(b), radius (R) merupakan jari-jari lingkaran yang dihasilkan oleh mobil saat berbelok satu putaran penuh. R dihitung mulai dari *center of axle* hingga titik potong perpanjangan garis yang tegak lurus dengan setiap roda. Persamaan (1) digunakan ketika mobil sedang berbelok ke kiri atau *counter clock wise*, sedangkan Persamaan (2) digunakan ketika mobil sedang berbelok ke kanan atau *clock wise*.

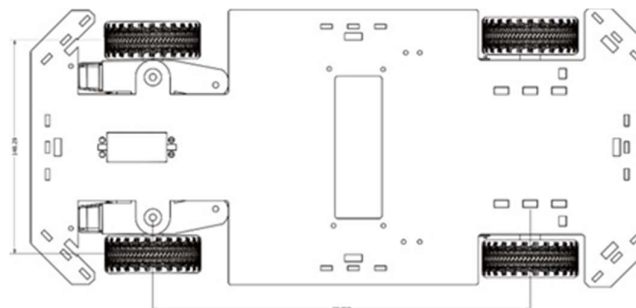
$$\alpha_{Left} = \tan^{-1} \left(\frac{L}{R - t/2} \right) \quad \beta_{Left} = \tan^{-1} \left(\frac{L}{R + t/2} \right) \quad (1)$$

$$\alpha_{Right} = \tan^{-1} \left(\frac{L}{R + t/2} \right) \quad \beta_{Right} = \tan^{-1} \left(\frac{L}{R - t/2} \right) \quad (2)$$

3. PERANCANGAN CLMR PENJEJAK GARIS

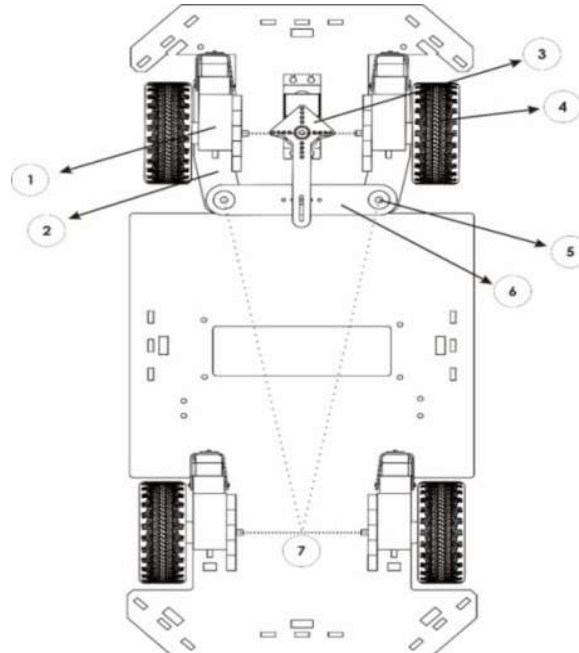
3.1. Perancangan Mekanika

Dari *blueprint* mobil Honda Civic pada Gambar 1, diketahui mobil Honda Civic memiliki dimensi jarak antara roda bagian depan dengan roda bagian belakang sepanjang 2,62 meter. Sedangkan jarak antara roda bagian kanan dan roda bagian kiri sepanjang 1,45 meter. Dengan menggunakan dimensi tersebut sebagai acuan, maka Gambar 3 adalah desain robot mobil CLMR dengan perbandingan rasio 1:10 dari dimensi mobil Honda Civic. Dengan menerapkan rasio tersebut akan dihasilkan desain rancangan robot mobil CLMR dengan dimensi 26,2 x 14,5 cm.



Gambar 3. Desain Rancangan Dimensi CLMR

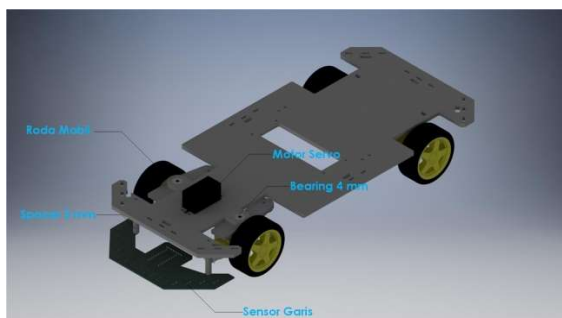
Dengan mengacu pada Gambar 3, *steering arms* harus diarahkan pada satu titik pusat yang merupakan titik tengah dari *axle* bagian belakang. Dalam merancang sistem kemudi *Ackermann*, terdapat beberapa komponen penyusun yang harus ada. Tabel 1 merupakan deskripsi komponen penyusun dari Gambar 4 dan Gambar 5 adalah hasil desain CAD CLMR.



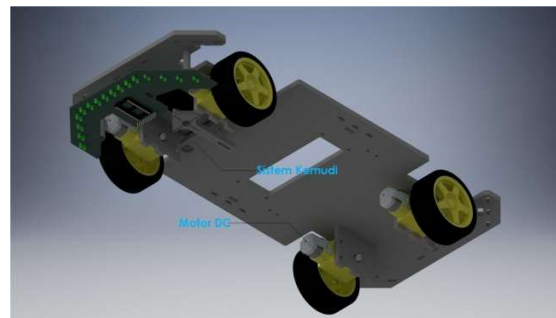
Gambar 4. Komponen Penyusun Sistem Kemudi *Ackermann*

Tabel 1. Deskripsi Komponen penyusun sistem kemudi *Ackermann*

Nomor	Keterangan
1	Kingpins atau ball joint
2	Steering arms
3	Front axle
4	Roda
5	Linkage pivot point
6	Tie rod
7	Center of axle



(a)

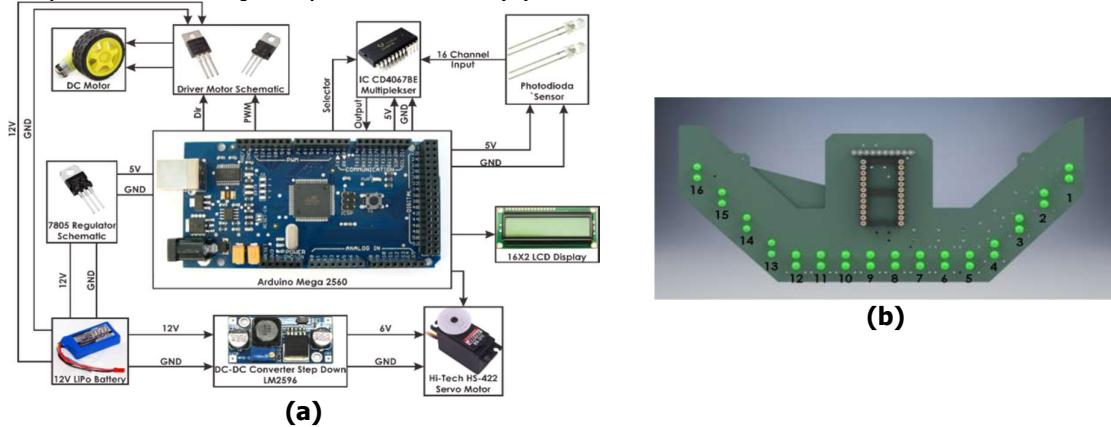


(b)

Gambar 5. Hasil Desain CAD CLMR, (a). Tampak Atas; (b). Tampak Bawah

3.2. Perancangan Elektronika

Sistem elektronika yang terdapat pada CLMR ialah terdiri dari perangkat dari sensor penjejak garis, perangkat *embeded microncontroller* menggunakan Arduino Mega dan satu buah motor servo serta dua buah motor DC sebagai *actuator* pergerakan. Diagram blok dari sistem elektronika rancangan CLMR disajikan pada Gambar 6(a). Tata letak sensor penjejak garis pada CLMR disajikan pada Gambar 6(b).



Gambar 6. Sistem Elektronika CLMR (a). Sistem Elektronika Keseluruhan, (b). Distribusi penomoran Sensor Garis

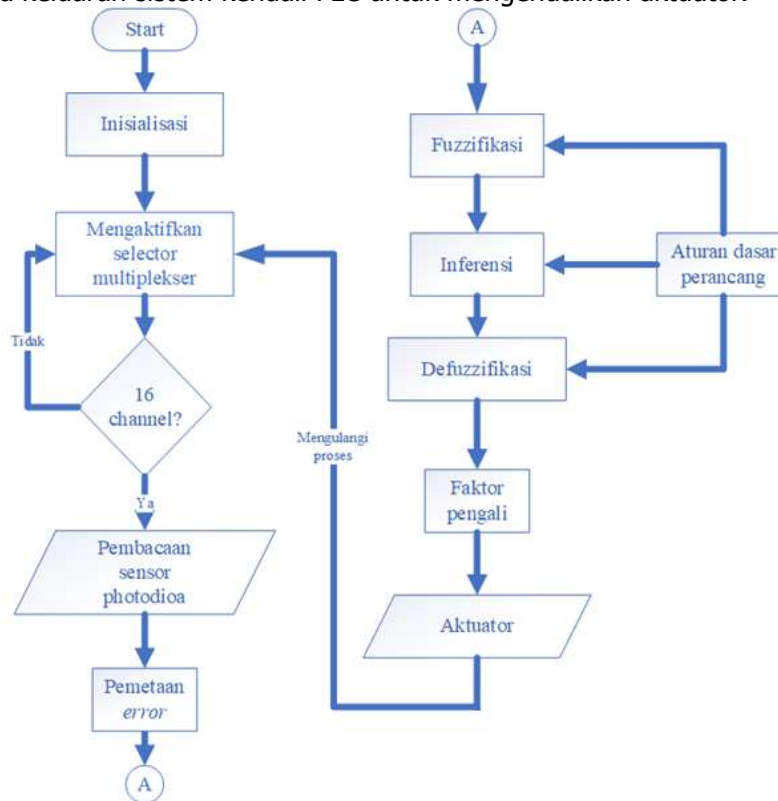
Selanjutnya adalah memetakan kondisi *error* dengan cara memetakan setiap kemungkinan kondisi yang dianggap sebagai *error* beserta nilainya. Namun untuk dapat memetakan *error*, variabel sensor kanan (nomer 1 hingga 8) dan sensor kiri (nomer 9-16) akan dibandingkan dengan fungsi "OR". Selanjutnya, nilai tersebut akan dipetakan berdasarkan nomor urut posisi sensor seperti yang disajikan pada Gambar 6(b). Tabel 2 menginformasikan pemetaan pembacaan *error*. Indikasi nilai logika "1" adalah dimana kondisi sensor berada garis berwarna hitam.

Tabel 2. Pemetaan Nilai Error

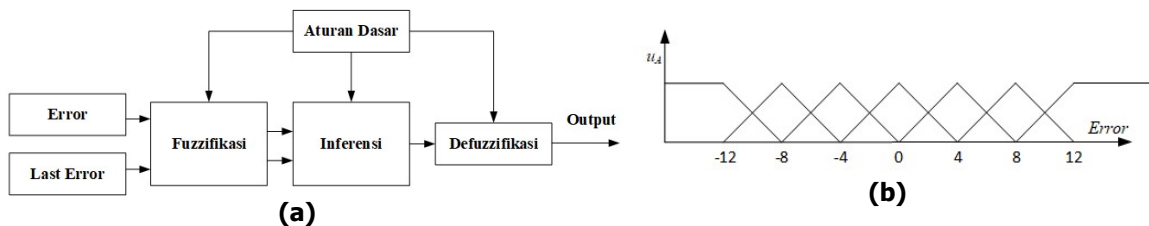
	Urutan Sensor															
<i>error</i>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
-12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-12	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-11	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
-10	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
-9	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
-8	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
-7	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
-6	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
-5	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
-4	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
-3	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

3.1. Perancangan Sistem Kendali Logika Fuzzy

Sistem kendali sangat dibutuhkan agar robot dapat berjalan dengan stabil, maka digunakan kendali cerdas untuk mengendalikan robot mobil CLMR penjejak garis. Diagram alir kendali cerdas tersebut adalah *Fuzzy Logic Controller* (FLC) seperti yang dijelaskan pada Gambar 7. Hasil pembacaan sensor yang telah dipetakan akan menghasilkan nilai *error* dan *last error*. Nilai tersebut akan menjadi nilai masukan untuk FLC dalam mengendalikan kecepatan putar motor DC dan sudut motor servo. Dalam sistem kendali FLC memiliki tiga tahapan proses, yaitu fuzzifikasi, inferensi, dan defuzzifikasi seperti yang ditunjukkan pada blok diagram Gambar 8(a). Tahapan proses tersebut digunakan untuk mengolah nilai *error* dari masukan yang berupa nilai krip menjadi nilai linguistik fuzzy yang kemudian diubah kembali menjadi nilai krip pada keluaran sistem kendali FLC untuk mengendalikan aktuator.



Gambar 7. Diagram Alir Sistem CLMR Penjejak Garis



Gambar 8. (a). Blok Diagram Sistem Kendali Logika Fuzzy, (b). Fungsi Keanggotaan Nilai *Error*

Fuzzifikasi

Pada tahapan proses ini nilai *error* yang merupakan nilai krip akan diubah menjadi nilai linguistik agar dapat di proses di tahapan selanjutnya, yaitu inferensi. Pengubahan nilai krip menjadi nilai linguistik adalah dengan cara menggolongkan nilai *error* ke nilai derajat

keanggotaan antara 0 dan 1. Terdapat dua jenis bentuk fungsi keanggotaan yang digunakan, yaitu fungsi segitiga dan fungsi trapesium. Sedangkan untuk jumlah keanggotaan yang digunakan terdapat lima fungsi segitiga dan dua fungsi trapesium seperti yang ditunjukkan pada Gambar 8(b).

Inferensi

Setelah didapatkan nilai linguistik berupa derajat keanggotaan dari setiap fungsi, selanjutnya nilai linguistik tersebut akan diolah pada tahapan proses inferensi (**Zadeh, 1965**). Pada tahapan ini diperlukan dua kondisi agar proses inferensi dapat berjalan dengan semestinya, dikarenakan hal itulah terdapat dua buah fungsi keanggotaan yang sama untuk memenuhi syarat inferensi tersebut. Jika suatu kondisi sudah tersedia maka diperlukan akibat untuk merespon kondisi tersebut. Seperti layaknya sebab-akibat, hal ini merupakan komponen utama dari FLC yang biasa disebut dengan aturan dasar. Untuk dapat membuat aturan dasar ini, memerlukan pengetahuan dasar tentang aksi yang harus dilakukan untuk berbagai macam kondisi yang mungkin terjadi. Salah satu keunikan dari FLC adalah perancang dapat menggunakan bahasa sendiri dengan bebas dalam merancang aturan dasar untuk inferensi FLC. Pada makalah ini aturan dasar yang digunakan disajikan pada Tabel 3, dimana suatu kondisi dilambangkan dengan "IF" dan aksi dilambangkan dengan "THEN".

Tabel 3. Aturan dasar FLC

IF	THEN
Sangat Kiri (SKr)	Big Steer Kiri (BSKr)
Agak Kiri (AKr)	Medium Steer Kiri (MSKr)
Kiri (Kr)	Little Steer Kiri (LSKr)
Tengah (T)	Stable (St)
Kanan (Kn)	Little Steer Kanan (LSKn)
Agak kanan (AKn)	Medium Steer Kanan (MSKn)
Sangat Kanan (SKn)	Big Steer Kanan (BSKn)

Setelah aturan dasar telah dipastikan, maka langkah selanjutnya adalah membuat susunan tabel untuk melakukan inferensi untuk mengolah data sebab-akibat yang ada untuk kemudian akan dievaluasi. Tabel yang sering digunakan untuk inferensi mengacu pada tabel MacViear Wheelan (**MacVicar-Whelan, 1976**) dengan menggunakan metode *Mamdani Inference Rule*. Table 4 merupakan hubungan relasi besar derajat dari nilai *error* dan *last error* dengan tujuh fungsi keanggotaan.

Tabel 4. Aturan inferensi mengacu pada tabel Macviear Wheelan

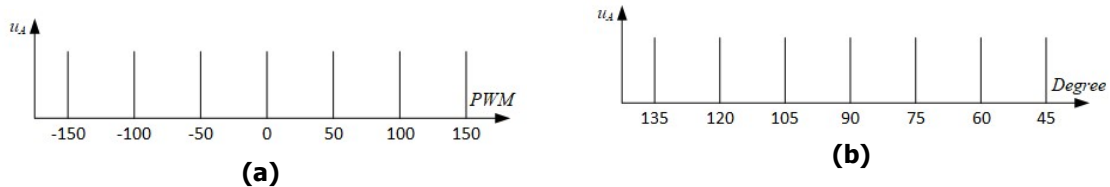
	Skr	Akr	Kr	T	Kn	Akn	SKn
Skr	BSKr	BSKr	MSKr	MSKr	LSr	LSr	St
Akr	BSKr	MSKr	MSKr	LSr	LSr	St	LSn
Kr	MSKr	MSKr	LSr	LSr	St	LSn	LSn
T	MSKr	LSr	LSr	St	LSn	LSn	MSKn
Kn	LSr	LSr	St	LSn	LSn	MSKn	MSKn
Akn	LSr	St	LSn	LSn	MSKn	MSKn	BSKn
SKn	St	LSn	LSn	MSKn	MSKn	BSKn	BSKn

SKr = Sangat Kiri
 AKr = Agak Kiri
 Kr = Kiri
 T = Tengah
 Kn = Kanan
 AKn = Agak Kanan
 SKn = Sangat Kanan

BSKr = Big Steer Kiri
 MSKr = Medium Steer Kiri
 LKr = Little Steer Kiri
 St = Stable
 LKn = Little Steer Kanan
 MSKn = Medium Steer Kanan
 BSKn = Big Steer Kanan

Defuzzifikasi

Nilai hasil pemetaan dari inferensi fuzzy tersebut akan diteruskan menuju tahapan terakhir dari FLC, yaitu defuzzifikasi. Pada tahapan ini, nilai tersebut akan diubah kembali menjadi nilai krusip dari tahapan proses sebelumnya. Metode yang digunakan adalah *Weight of Area* (WA). Untuk mengubah nilai linguistik tersebut diperlukan fungsi yang disesuaikan dengan rentang yang diperlukan oleh aktuator. Gambar 9(a) merupakan fungsi keanggotaan defuzzifikasi untuk motor DC, sedangkan Gambar 9(b) merupakan fungsi keanggotaan defuzzifikasi untuk menentukan sudut motor servo. Bentuk fungsi keluaran ini menggunakan bentuk yang disebut dengan *singleton*.



Gambar 9. Fungsi Keanggotaan Defuzzifikasi; (a). Defuzzifikasi untuk Motor DC, (b). Defuzzifikasi untuk Menentukan Sudut Motor Servo

4. PENGUJIAN DAN PEMBAHASAN

Pengujian sistem kendali fuzzy dilakukan dengan tiga tahap, yaitu pengujian variasi fungsi keanggotaan nilai masukan fuzzifikasi yang disajikan pada Tabel 5, variasi fungsi keanggotaan keluaran untuk motor DC yang disajikan pada Tabel 6 dan variasi fungsi keanggotaan keluaran untuk motor servo yang disajikan pada Tabel 7. Metode pengujian yang digunakan adalah dengan cara memvariasikan satu parameter secara bergantian sedangkan kedua parameter lainnya tidak divariasikan terlebih dahulu dan dilakukan secara bertahap. Pengujian ini akan dilakukan pada lintasan uji yang disajikan pada Gambar 10. Nilai acuan pengujian berupa perbandingan waktu tempuh (s) dan kecepatan ($\frac{m}{s}$) tiap variasi parameter dengan panjang lintasan (m). Kecepatan (v) diperoleh dari Persamaan (3).

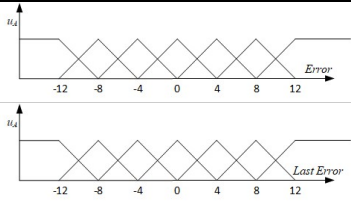
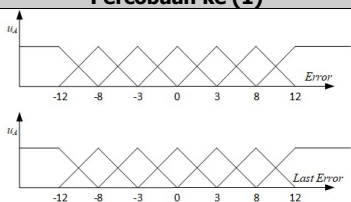
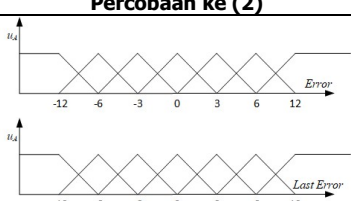
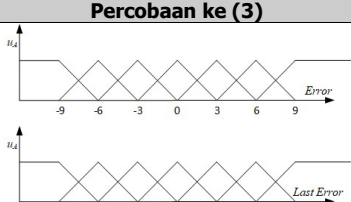
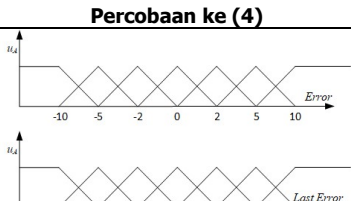
$$v = \frac{s}{t} \tag{3}$$

Tabel 5 merupakan lima buah variasi fungsi keanggotaan *error* dan *last error* dalam menelusuri lintasan satu putaran penuh. Dalam pengujian ini hanya parameter fuzzifikasi yang akan divariasikan, sedangkan parameter lainnya seperti fuzzifikasi motor DC dan motor servo dengan nilai tetap seperti yang disajikan pada Gambar 9.

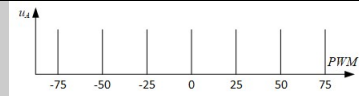
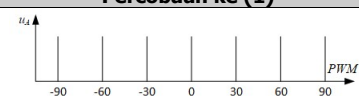
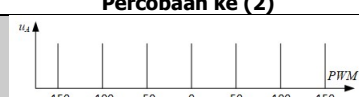
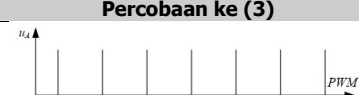
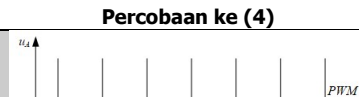


Gambar 10. Lintasan Uji

Tabel 5. Data pengujian variasi fungsi keanggotaan *error* dan *last error*

Fungsi keanggotaan yang digunakan	Waktu (s) - Kecepatan (m/s)
 <p>Percobaan ke (1)</p>	10,16 – 0,571
 <p>Percobaan ke (2)</p>	10,32 – 0,562
 <p>Percobaan ke (3)</p>	10,28 – 0,565
 <p>Percobaan ke (4)</p>	10,34 – 0,561
 <p>Percobaan ke (5)</p>	10,32 – 0,562

Tabel 6. Data pengujian variasi fungsi keanggotaan keluaran defuzzifikasi untuk motor DC

Fungsi keanggotaan yang digunakan	Waktu (s) - Kecepatan (m/s)
 <p>Percobaan ke (1)</p>	13,62 – 0,426
 <p>Percobaan ke (2)</p>	11,90 – 0,488
 <p>Percobaan ke (3)</p>	10,16 – 0,571
 <p>Percobaan ke (4)</p>	10,13 – 0,573
 <p>Percobaan ke (5)</p>	9,75 – 0,595

Tabel 6 pengujian dengan memvariasikan fungsi keanggotaan keluaran untuk motor DC dalam menelusuri lintasan satu putaran penuh. Dalam pengujian ini hanya parameter motor DC yang akan divariasikan, sedangkan parameter lainnya seperti fuzzifikasi masukan *error* dan *last error* yang digunakan adalah hasil dari percobaan ke satu pada Tabel 5 dan motor servo tetap seperti yang digunakan pada sub bab perancangan yang disajikan pada Gambar 9.

Tabel 7 pengujian dengan cara memvariasikan fungsi keanggotaan keluaran untuk motor servo dalam menelusuri lintasan satu putaran penuh. Dalam pengujian ini hanya parameter defuzzifikasi motor servo yang akan divariasikan, sedangkan parameter lainnya seperti defuzzifikasi motor DC percobaan ke-lima pada Tabel 6 dan fuzzifikasi *error* dan *last error* yang digunakan adalah hasil dari percobaan ke satu pada Tabel 5.

Berdasarkan hasil pengujian yang telah dilakukan pada Tabel 5, Tabel 6, dan Tabel 7, parameter yang paling optimal merupakan parameter yang menghasilkan catatan waktu tempuh tercepat, sedangkan parameter yang tidak menghasilkan catatan waktu tempuh artinya robot tidak dapat menyelesaikan satu putaran penuh pada lintasan uji. Bila keseluruhan parameter paling optimal seperti yang ditunjukkan pada Tabel 8 dikombinasikan

maka akan menghasilkan catatan waktu tempuh selama 9,38 detik dengan kecepatan 0,619 (m/s).

Tabel 7. Data pengujian variasi fungsi keanggotaan keluaran defuzzifikasi untuk motor servo

Fungsi keanggotaan yang digunakan	Waktu (s) - Kecepatan (m/s)
<p>Percobaan ke (1)</p>	10,16 – 0,571
<p>Percobaan ke (2)</p>	9,75 – 0,595
<p>Percobaan ke (3)</p>	-
<p>Percobaan ke (4)</p>	9,38 – 0,619
<p>Percobaan ke (5)</p>	-

Tabel 8. Kumpulan parameter FLC paling optimal

Fungsi keanggotaan yang digunakan	Waktu (s) - Kecepatan (m/s)
<p>Fuzifikasi <i>error</i> dan <i>last error</i></p>	9,38 – 0,619
<p>Defuzifikasi Motor DC</p>	
<p>Defuzifikasi Motor servo</p>	

Selanjutnya sebagai bahan perbandingan, diterapkan kendali PID. Proses penentuan konstanta kendali PID yang terdiri dari K_p sebagai konstanta proporsional, K_i sebagai konstanta integratif dan K_d sebagai konstanta differensial dilakukan dengan *tuning* eksperimen. Sebagai acuan pengujian parameter konstanta kendali PID diamati saat CLMR melintasi belokan pada lintasan uji Gambar 10. Terdapat dua pengamatan respon, yaitu terhadap motor servo dan motor DC dari CLMR seperti yang disajikan pada Tabel 9-12.

Tabel 9. Data pengujian konstanta proporsional untuk motor servo

K_p	K_i	K_d	Respon
1	0	0	CLMR tidak mampu berbelok tajam mengikuti lengkungan yang ada pada lintasan.
2,5	0	0	Saat melintasi belokan membaik namun CLMR masih belum mampu menyelesaikan lintasan satu putaran penuh.
4	0	0	Sudut belokan sudah terlihat optimal dan CLMR dapat menyelesaikan lintasan satu putaran penuh.
5	0	0	Osilasi berkurang dan CLMR menjadi lebih stabil
7	0	0	CLMR mengalami osilasi pada lintasan berbelok.

Tabel 10. Data pengujian konstanta *derivative* untuk motor servo

K_p	K_i	K_d	Respon
5	0	10	CLMR masih berosilasi.
5	0	20	CLMR membaik, osilasi berkurang.
5	0	30	CLMR menjadi lebih stabil lagi dan hampir tidak berosilasi.
5	0	40	CLMR menjadi lebih stabil lagi.
5	0	50	Penambahan konstanta K_d tidak menghasilkan perbedaan begitu <i>significant</i> pada respon CLMR.

Tabel 11. Data pengujian konstanta proporsional untuk motor DC

Kp	Ki	Kd	Respon
5	0	0	CLMR terlihat lambat.
7	0	0	Kecepatan CLMR meningkat namun osilasi cukup terasa.
10	0	0	CLMR stabil dan hampir tidak ada osilasi.
12	0	0	Kecepatan CLMR bertambah dan osilasi meningkat.
15	0	0	Kecepatan CLMR bertambah lagi dan osilasi semakin meningkat.

Tabel 12. Data pengujian konstanta *derivative* untuk motor DC

Kp	Ki	Kd	Respon
10	0	30	CLMR lambat dan berosilasi
10	0	40	CLMR menjadi lebih stabil
10	0	50	Kecepatan CLMR terlihat cukup optimal dan stabil
10	0	60	CLMR menjadi lebih stabil
10	0	70	Penambahan kontanta Kd tidak menghasilkan perbedaan begitu <i>significant</i> pada respon CLMR.

Berdasarkan dari hasil pengujian menggunakan sistem kendali PID, catatan waktu yang dihasilkan adalah selama 9,91 detik dibandingkan dengan saat menggunakan sistem kendali logika fuzzy yang menghasilkan catatan waktu selama 9,38 detik. Artinya sistem kendali logika fuzzy memiliki kemampuan yang lebih baik dibandingkan dengan sistem kendali PID untuk di implementasikan pada CLMR penjejak garis.

5. KESIMPULAN

Dari serangkaian pengujian, dan analisis yang telah dilakukan pada sistem CLMR penjejak garis maka dapat disimpulkan bahwa sistem telah berhasil dibuat menggunakan kendali logika fuzzy. Tahapan pertama pada pengujian kendali fuzzy adalah variasi pada proses masukan fuzzifikasi, yang kedua variasi pada tahapan defuzzifikasi kecepatan motor DC CLMR dan yang terakhir variasi pada tahapan defuzzifikasi sudut *steering* CLMR. Dari hasil pengujian diperoleh bahwa kendali logika fuzzy yang diaplikasikan pada sistem mampu membuat pergerakan CLMR sukses mengikuti lintasan uji selama 9,38 detik dan menghasilkan performansi yang lebih baik 0,53 detik bila dibandingkan dengan kendali PID.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Departemen Teknik Elektro dan Informatika Sekolah Vokasi Universitas Gadjah Mada atas dukungan yang telah diberikan kepada penulis.

DAFTAR RUJUKAN

- Angeles, J., (2002). *Fundamentals of robotic mechanical systems* (Vol. 2). New York: Springer-Verlag.
- Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zhang, X. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.
- Maślak, W., & Butkiewicz, B. S. (2013). Autonomous vehicle with fuzzy control. In *Signal Processing Symposium (SPS)*, (pp. 1-6).

- Hunt, K. H. (1978). *Kinematic geometry of mechanisms* (Vol. 7). USA: Oxford University Press.
- Baturone, I., Moreno-Velo, F. J., Sánchez-Solano, S., & Ollero, A. (2004). Automatic design of fuzzy controllers for car-like autonomous robots. *IEEE Transactions on Fuzzy Systems*, *12*(4), 447-465.
- Chiu, C. S., Lian, K. Y., & Liu, P. (2005). Fuzzy gain scheduling for parallel parking a car-like robot. *IEEE Transactions on Control Systems Technology*, *13*(6), 1084-1092.
- El-Khatib, M. M., & Hamilton, D. J. (2006, July). A layered fuzzy controller for nonholonomic car-like robot motion planning. In *Mechatronics, 2006 IEEE International Conference on*, (pp. 194-198).
- Gucrcro, H. B., Velasquez, A. E. B., Barrero, J. F., Côco, D. Z., Risardi, J. C., Magalhães, D. V., & Becker, M. (2014, October). Orientation (Yaw) Fuzzy controller applied to a car-like mobile robot prototype. In *Circuits and Systems (CWCAS), 2014 IEEE 5th Colombian Workshop on*, (pp. 1-6).
- Liu, Z., Wang, Y., & Lu, T. F. (2011, November). Car-like mobile robot reverse parking: Using fuzzy logic control approach. In *Robot, Vision and Signal Processing (RVSP), 2011 First International Conference on*, (pp. 77-80).
- Ouahah, N., Ourak, L., Hamerlain, M., & Boudjema, F. (2006, November). Implementation of an oriented positioning on a car-like mobile robot by fuzzy control. In *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*, (pp. 4076-4081).
- Scicluna, N., Gatt, E., Casha, O., Grech, I., & Micallef, J. (2012, December). FPGA-based autonomous parking of a car-like robot using Fuzzy Logic Control. In *Electronics, Circuits and Systems (ICECS), 2012 19th IEEE International Conference on*, (pp. 229-232).
- Abatari, H. T., & Tafti, A. D. (2013, February). Using a fuzzy PID controller for the path following of a car-like mobile robot. In *Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on*, (pp. 189-193).
- Hasan, K. M., & Al Mamun, A. (2012, May). Implementation of autonomous line follower robot. In *Informatics, Electronics & Vision (ICIEV), 2012 International Conference on*, (pp. 865-869).
- Nath, A. S., Kumar, A., & Malik, T. (2013). Implementation of PID Control to Reduce Wobbling in a Line Following Robot.

Normey-Rico, J. E., Alcalá, I., Gómez-Ortega, J., & Camacho, E. F. (2001). Mobile robot path tracking using a robust PID controller. *Control Engineering Practice*, 9(11), 1209-1214.

Zadeh, L. A. (1965). *Fuzzy sets. Information and control*, 8, 3.

MacVicar-Whelan, P. J. (1976). Fuzzy sets for man-machine interaction. *International Journal of Man-Machine Studies*, 8(6), 687-697.