

# KONTROL PERGERAKAN ROBOT BERODA MENGGUNAKAN PENGENDALI FUZZY-PID BERBASIS LINE FOLLOWER DAN IMAGE PROCESSING

## WHEELED ROBOT MOVEMENT CONTROL USING FUZZY-PID BASED ON LINE-FOLLOWER AND IMAGE PROCESSING

Muhammad Nurul Fadhil<sup>1</sup>, Sigit yuwono<sup>2</sup>,Angga Rusdinar<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

nurulfadhil@gmail.com <sup>1</sup> yuwono@telkomuniversity.ac.id <sup>2</sup> anggarusdinar@telkomuniversity.ac.id <sup>3</sup>

### Abstrak

Pada umumnya sistem kendali robot line follower menggunakan kontroler konvensional atau yang biasa disebut PID (Proportional Integral Derivativ). Hasil output dari penggunaan sistem ini kadang dapat membuat pergerakan robot semakin tidak stabil jika diberikan nilai acuan (konstanta) yang sensitif. Itu disebabkan karena semakin sensitif nilai acuan yang diberikan, dapat menyebabkan nilai overshoot yang tinggi jika ada gangguan. Jika nilai acuan kurang sensitif dapat mengurangi nilai overshoot, namun itu akan memperpanjang recovery time. Agar dapat mengatasi permasalahan ini, diterapkan metode gabungan dari kontroler PID dengan Fuzzy Logic. Fuzzy logic disini berfungsi untuk membantu menentukan parameter nilai error dan perubahan nilai error sebelumnya seta memperkecil recovery time.

Pada penelitian ini, dirancang sebuah robot line follower yang dapat mengenali objek garis berwarna menggunakan metode image processing dan Pi camera sebagai sensor input. Pada proses tersebut didapatkan data mengenai koordinat obyek, jalur yang akan dilalui, serta luasan objek yang terdeteksi. Actuator yang digunakan adalah motor servo dan motor DC. Sedangkan otak dari sistem tersebut menggunakan Raspberry Pi.

**Kata kunci:** *Fuzzy-PID, Robot, Line Follower, Image Processing .*

### Abstract

*In general, line follower robot control systems use conventional controllers or called PID (Proportional Integral Derivatives). The output results from the use of this system can sometimes make the movement of the robot unstable if given sensitive reference values (constants). That is because the more sensitive the reference value is given, it can cause a high overshoot and undershoot value if there is a disturbance. However, if the reference value is less sensitive, it can reduce the overshoot and undershoot values, but it will extend recovery time. In order to overcome this problem, we can use both methods from PID controller and Fuzzy logic. Fuzzy logic use to help determine the parameters of the error value and change the value of the previous error and reduce team recovery.*

*In this study, a line follower robot will be designed that can recognize colored line objects using the image processing method and a webcam as an input sensor. In the process we can obtained data of object coordinates, the path to be passed, and the area of the object detected. The actuator used is a servo motor and DC motor. While the brain of the system uses Raspberry Pi.*

**Keywords:** *Fuzzy-PID, Robot, Line Follower, Image Processing.*

## 1. Pendahuluan

Teknologi mobil kendali otonom saat ini sangat berkembang pesat, memanfaatkan teknik robotika dan perkembangan teknologi sensor membuat mobil dapat berjalan secara otomatis mengurangi beban kerja manusia terutama di sektor transportasi dan industri. Salah satu teknologi yang dimanfaatkan dalam mobil kendali otonom adalah Line follower. Dalam penelitian ini akan dirancang pengembangan kendali miniatur robot mobil penggerak 8 roda memanfaatkan sensor citra membaca garis lintasan mobil.

Salah satu metode yang akan digunakan untuk membaca garis sekaligus kendali gerakan mobil adalah kendali PID (Proportional Integral Derivative), input berupa simpangan error pembacaan kelurusan garis dan simpangan garis terhadap sumbu tengah roda robot beroda, kemudian proses akan mengatur aktuasi steering dan mesin penggerak untuk menyesuaikan gerak kendaraan agar tetap tegak diatas garis.

Akan tetapi penggunaan kendali PID umumnya dipakai pada plant sistem dengan sistem mekanis mumpuni dengan variabel simpangan error dan input linear terhadap output yang diinginkan, maka apabila terjadi variabel tambahan atau gangguan tidak terduga masuk kedalam proses maka akan terjadi overshoot tidak terduga dan dapat menghambat waktu respon sehingga harus dilakukan tuning ulang untuk controller PID tersebut. Dari sini diusulkan menggunakan logika fuzzy sebagai metode tuning adaptif konstanta P, I, D mengkompensasi error plant yang tidak linear seperti yang telah dijelaskan sebelumnya, sehingga kendali PID dapat menghasilkan overshoot dan waktu respon ke posisi threshold garis menjadi lebih kecil.

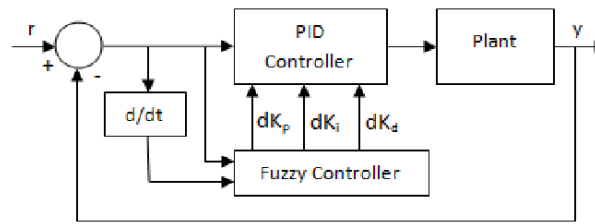
## 2. Dasar Teori dan Perancangan

### 2.1 Deskripsi Cara Kerja dan Konsep Solusi

Prinsip kerja pembentuk gerakan pada robot ini terdiri dari beberapa tahap aktivitas. Aktivitas pertama adalah pengambilan variabel input, input yang diambil adalah citra warna dan garis lintasan, aktivitas kedua adalah pemrosesan, pemrosesan pertama dilakukan di raspberry pi untuk mengolah input citra yang masuk, proses seleksi perbandingan gradasi warna dijalankan untuk membedakan warna jalur dan latar belakang, kemudian logika fuzzy dengan formula F-transform dari pustaka opencv diterapkan untuk membuat citra garis menjadi blur, tujuannya adalah mengkompensasi bentuk lintasan yang rusak dan tidak berbentuk namun tetap berbeda dari latar belakang sehingga kriteria garis masih tetap dapat dibedakan, apabila kriteria garis sudah didapatkan kemudian dipakai untuk membaca posisi jalur terhadap tegak kendaraan, hasil inferensi error posisi kemudian di proses dengan mikrokontroler arduino untuk mengelola kendali steering robot melalui servo dan motor driver untuk kecepatannya.

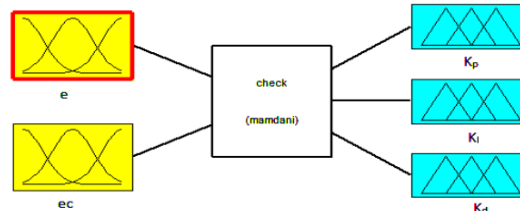
### 2.2 Fuzzy tuned PID controller

Kelemahan utama dari controller PID konvensional adalah tidak cukup memberikan performa baik terhadap sistem yang memiliki output tidak menentu, memiliki delay waktu dan tidak linear, sehingga dibutuhkan tuning parameter PID untuk mendapatkan respon yang diinginkan[12], dengan bekal pengetahuan dasar sifat proses maka logika fuzzy dapat diterapkan untuk tuning adaptif dari kendali PID saat controller sedang berjalan. Secara garis besar berikut adalah diagram kontrol Fuzzy PID



Gambar 2.8 Diagram Blok Fuzzy tuned PID

Dari diagram diatas tujuan dari logika fuzzy adalah memberi nilai koreksi terhadap nilai Konstanta P,I dan D dalam rentang nilai didalam dKp, dKi, dan dKd terhadap simpangan error tidak tentu (d/dt). Sebelumnya nilai Kp, Ki dan Kd sudah disetting dalam angka yang dianggap paling baik memberikan respon untuk sistem, kemudian angka Kp, Ki dan Kd tersebut diberi toleransi untuk mengkompensasi error yang tidak linear atau tidak menentu (*error change*). Nilai rentang kompensasi tersebut kemudian dipetakan kedalam *rule-based fuzzy* untuk kemudian difuzzifikasi sehingga didapatkan inferensi (ec) terhadap nilai rentang konstanta dKp,dKi, dKd. Berikut adalah penjabarannya melalui diagram dibawah



Gambar 2.9 fuzzifikasi error terhadap Konsanta P,I, dan D

Dengan nilai keanggotaan tiap Konstanta didapatkan dari persamaan berikut:

$$K'_p = \frac{K_p - K_{p\min}}{K_{p\max} - K_{p\min}} \quad (\text{persamaan 2.12})$$

$$K'_i = \frac{K_i - K_{i\min}}{K_{i\max} - K_{i\min}} \quad (\text{persamaan 2.13})$$

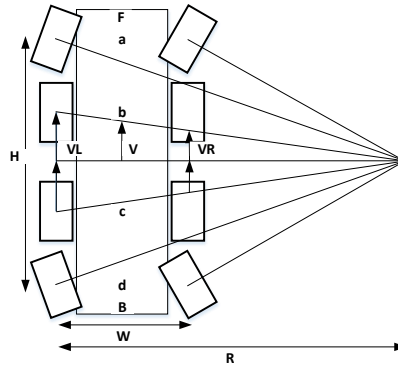
$$K'_d = \frac{K_d - K_{d\min}}{K_{d\max} - K_{d\min}} \quad (\text{persamaan 2.14})$$

Sehingga hasil inferensi dari fuzzyfikasi tersebut menghasilkan nilai Konstanta P,I, dan D yang cocok untuk tiap perubahan error di kontroler PID yang digunakan.

### 2.3 Hybrid Ackerman-Differential Steering

Penggunaan *Ackerman steering* sering digunakan berdampingan dengan differential steering karena roda penggerak belakang mengalami friksi saat berbelok. Penggunaan sistem ackerman atau *differential steering* mudah diaplikasikan pada rancang kendaraan roda 4 dengan ukuran panjang dan lebar antar sumbu roda yang

sama, karena variabel friksi roda saat berbelok terhitung minimum. Akan tetapi untuk robot 8 roda atau kendaraan dengan panjang jarak antar sumbu roda yang tidak sama atau lebih panjang maka perlu menggunakan sistem belok gabungan Ackerman-Differential. Berikut ini adalah skema kendaraan penggerak 8 roda dengan chassis panjang.



Gambar 2.5 Skema kendali rancang kendaraan penggerak 8 roda

Dengan radius belok sejajar perpotongan tengah chassis kendaraan maka nilai pergerakan sisi roda depan kendaraan (F) di sumbu notasi a menjadi koorelasi aksen cermin untuk input gerakan roda belakang (B) di sumbu notasi d, kemudian sisi roda dengan motor penggerak dibaris sumbu roda ke-2 dan ke-3 memiliki kecepatan paralel.

$$\theta F \text{ Left } d = \theta F \text{ Left } a' \quad (\text{persamaan 2.6})$$

$$\theta F \text{ Right } d = \theta F \text{ Right } a' \quad (\text{persamaan 2.7})$$

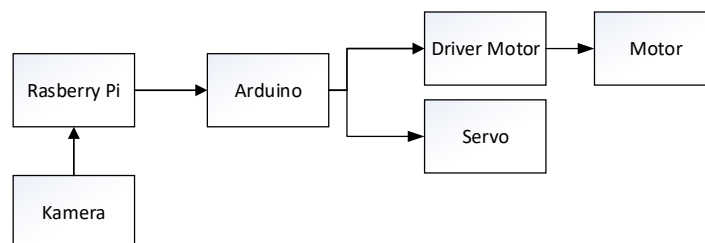
$$V \text{ Left } b = V \text{ Left } c \quad (\text{persamaan 2.8})$$

$$V \text{ Right } b = V \text{ Right } c \quad (\text{persamaan 2.9})$$

Jika diketahui sudut belok yang diinginkan (R) beserta kecepatan robot linear yang diinginkan, maka sudut belok sumbu roda a dan d dapat dicari dengan persamaan 2.4 dan 2.5 dan kecepatan roda b dan c dapat diketahui dengan menggunakan persamaan 2.2 dan 2.3.

### 3. Perancangan Sistem

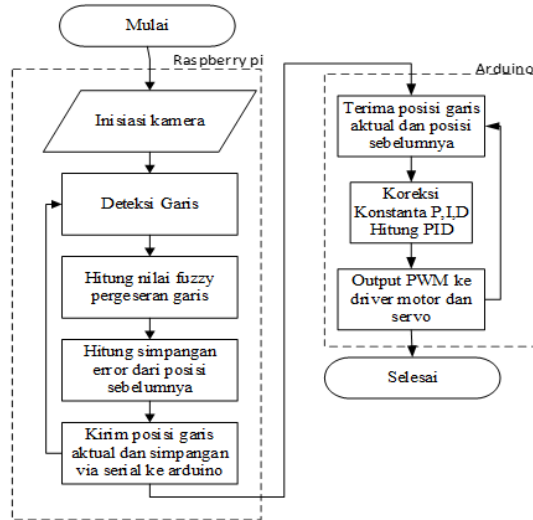
#### 3.1 Desain Sistem



Gambar 2. Diagram Blok Sistem Secara Keseluruhan

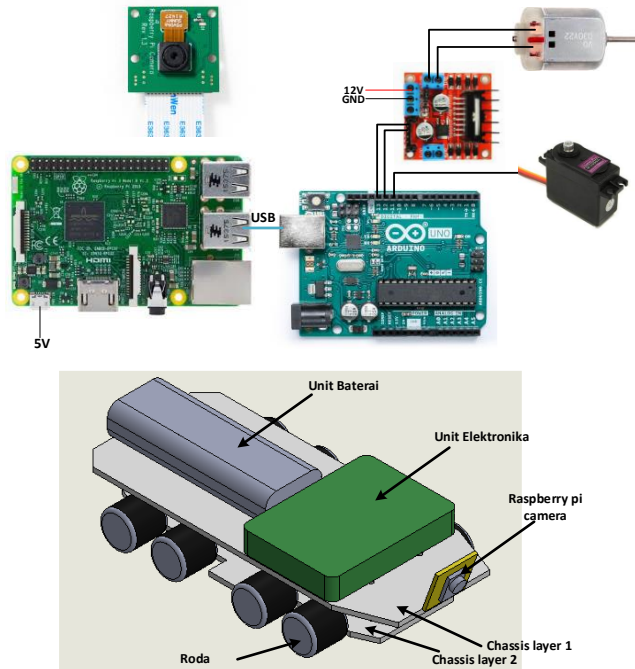
### 3.1.1 Perancangan Perangkat Lunak

Perangkat lunak yang digunakan untuk proses produksi adalah Arduino IDE dengan bahasa pemrograman yaitu C kemudian di sisi Single board computer menggunakan Python. Diagram Alir dari sistem seperti yang ditunjukkan pada gambar 2.



Gambar 3. Diagram Alir Sistem Secara Keseluruhan

### 3.1.2 Perancangan Perangkat Keras



Gambar 4. Gambaran Perancangan Sistem Secara Keseluruhan

#### 4. Pengujian dan Analisis

##### 4.1 Pengujian

Dari percobaan menunjukkan bahwa gerak servo ( $\alpha$ ) terhadap posisi garis (S) adalah linear, tiap derajat gerakan servo menunjukkan perpindahan tegak sebesar rata-rata 2.5 pixel terhadap posisi garis. Jadi fungsi matematika yang dapat dipakai

$$\int_t \alpha = 2.5 S$$

Nilai ini dapat dipakai untuk fase tuning iterasi ke 2 proporsi pergerakan servo terhadap garis, maka diterapkan nilai  $K_p=2.5$ ,  $K_i=0$ ,  $K_d=0$ .

Di iterasi 5 ini kurva  $K_p$  4 diujicobakan, akan tetapi nilai belok servo cenderung mengalami overshoot melebihi batas belok maksimum, sehingga diputuskan bahwa nilai gain  $K_p$  awal yang dipilih ialah 3.5.

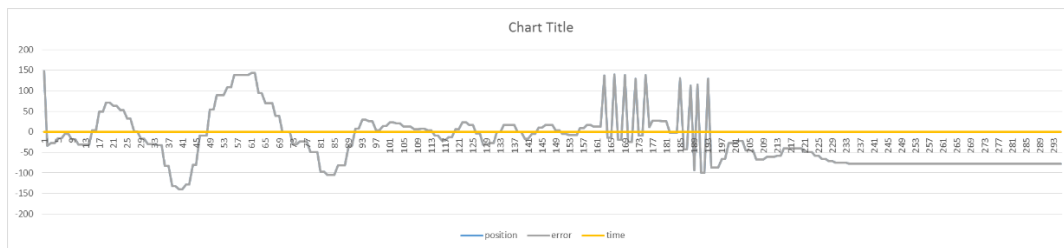
Dari gambar 4.10 Diketahui  $K_u$  ialah 3.5, jarak antar puncak osilasi step adalah 2.2 sekon. Dengan menggunakan persamaan tuning Ziegler Nichols Maka:

$$K_p = 0.6 * K_u = 2.1$$

$$K_i = (2 * K_p) / T_u = 1.9090 \sim 1.91$$

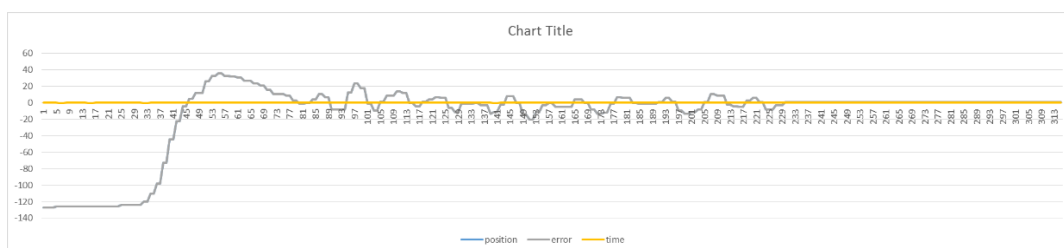
$$K_d = (K_p * T_u) / 8 = 0.5775 \sim 0.58$$

Nilai tuning ini kemudian diujicobakan untuk iterasi selanjutnya.



Gambar 4.12 Kurva respon garis terhadap sudut belok mobil Gain  $K_p=2.1$ ,  $K_i=1.91$ ,  $K_d=0.58$

Di Iterasi 6 ini respon pergerakan mobil diujicobakan dengan parameter gain hasil perhitungan Ziegler Nichols, dari gambar 4.11 terlihat bahwa terjadi siklus stabilisasi yang cukup baik, namun terjadi jitter tak terduga saat fase stabilisasi, dari sini disimpulkan untuk mengubah parameter  $K_i$  menjadi 0 dan menaikkan parameter  $K_d$  menjadi 0.8 untuk meredam overshoot.



Gambar 4.13 Kurva respon garis terhadap sudut belok mobil Gain  $K_p=2.1$ ,  $K_i=0$ ,  $K_d=0.8$

Di Iterasi 7 ini respon pergerakan menjadi lebih stabil, dengan mengubah parameter  $K_i$  menjadi 0 dan  $K_d=0.8$ , overshoot dapat diminimalisir, jitter menghilang dan sistem cukup stabil di garis setpoint.

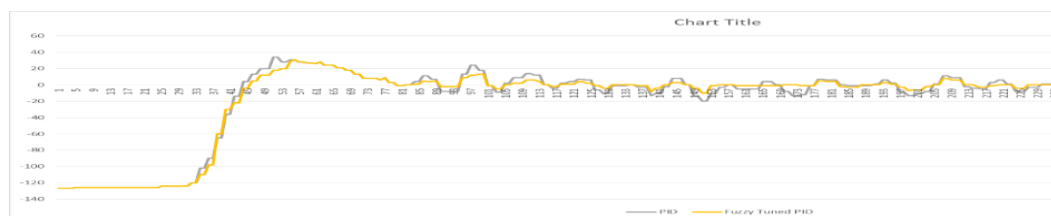
Iterasi 8 membahas tuning variabel gain kedalam logika fuzzy. Penalaran rule ditentukan dari hasil perilaku sistem dari iterasi-iterasi sebelumnya, diketahui bahwa kenaikan tiap 0.1 dari konstanta P menghasilkan respon lebih cepat untuk memulihkan error posisi dari semula 3 detik ke 1.5 detik, hanya saja menghasilkan overshoot posisi garis dan sudut servo yang lebih tinggi dari toleransi sudut servo, sehingga perlu diredam

dengan menaikkan atau menurunkan konstanta D sebanyak 0.1 point dari semula yang dapat meredam nilai simpangan sudut hingga 10 derajat.

Tabel 4.1 Peta Rule Input error posisi terhadap gain konstanta output dan konstanta Output

error (input)	Konstanta P (output)	Konstanta D (Output)
Negative Big	$K_p=K_p+0.15$	Kd= +0.1
Negative Small	$K_p=K_p+0.05$	
Zero	$K_p=0$	Kd=0
Positive Small	$K_p=K_p-0.05$	Kp=-0.1
Positive Big	$K_p=K_p-0.15$	

## 4.2 Analisis



Terlihat bahwa dengan kendali Fuzzy-PID , gerakan overshoot dapat diredam dengan adanya minimasi overshoot hingga 40% dan jeda stabil dari waktu ke 41-163, sedangkan dengan PID waktu jeda pulih dari 41 hingga 189. Dengan memberikan  $K_p$  dan  $K_d$  pada jeda setpoint terlihat bahwa robot cenderung lebih stabil dapat mempertahankan posisi diatas garisnya.

## 5. Kesimpulan

Dari hasil pengujian dan rancangan kendali fuzzy-PID memperlihatkan reaksi kendali yang lebih baik, dibanding dengan kendali PID menggunakan konstanta yang tidak di Tuning dengan PID, konsep Adaptive Tuning ini dapat mengkompensasi perilaku tidak terduga yang diakibatkan oleh kesalahan rancang mekanis robot.

**Daftar Pustaka:**

- [1] Chicoki, A., Amari, S.i. (2002). *Adaptive Blind Signal and Image processing: Learning Algorithms and Applications*. Chicester, John Wiley & Sons, Ltd.
- [2] Darfyma. P. M., Darmawan,D., Rusdinar.A. (2016). *Kendali Kecepatan dan Posisi Automated Guide Vehicle Menggunakan Fuzzy Logic dan PID Control*. E-proceeding of engineering, Vol.3, No.3 Desember 2016
- [3] Nugraha, R., D. (2016). *Rancang Bangun Mobile Robot Pengikut Objek Berdasarkan Warna dan Bentuk Menggunakan Template Matching Berbasis Mini PC*. Universitas Andalas, Padang
- [4] Parker, J. (2011). *Algorithms for Image processing and Computer Vision*. Indianapolis., Wiley Publishing, Inc.

