

BAB 4

HASIL DAN PEMBAHASAN

4.1 Pengujian Logika Fuzzy

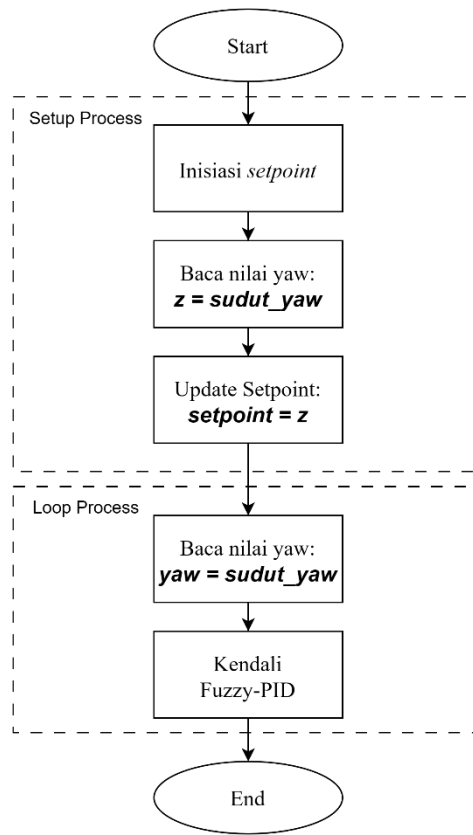
Aturan logika fuzzy yang telah ditentukan kemudian diimplementasikan ke dalam program algoritma pada mikrokontroler ESP32 menggunakan software Arduino IDE. Pengujian logika fuzzy dilakukan dengan pengambilan 100 data dari masing-masing keluaran fuzzy yaitu kiri, lurus, dan kanan. Analisis hasil keluaran fuzzy menggunakan *Confusion Matrix* dengan tujuan untuk mengetahui seberapa akurasi dan presisi dari model fuzzy yang diimplementasikan ke dalam mobile robot. Hasil pengujian dapat dilihat pada table 4.1.

Tabel 4.1 Confusion Matrix Logika Fuzzy

		Nilai Prediksi		
		Kiri	Lurus	Kanan
Nilai Aktual	Kiri	99	0	1
	Lurus	0	98	2
	Kanan	6	0	94

Berdasarkan hasil pengujian logika fuzzy dan pengolahan menggunakan *confusion matrix*, maka didapatkan nilai akurasi 97% dan presisi 97%. Akurasi dan presisi tidak mencapai 100%, hal tersebut disebabkan oleh kesalahan pembacaan sensor ultrasonik yang menyebabkan kesalahan pengambilan keputusan di beberapa kondisi. Pembacaan sensor menjadi tidak stabil ketika objek terlalu jauh dari sensor ultrasonik dan objek yang ramping.

4.2 Kalibrasi Sensor IMU

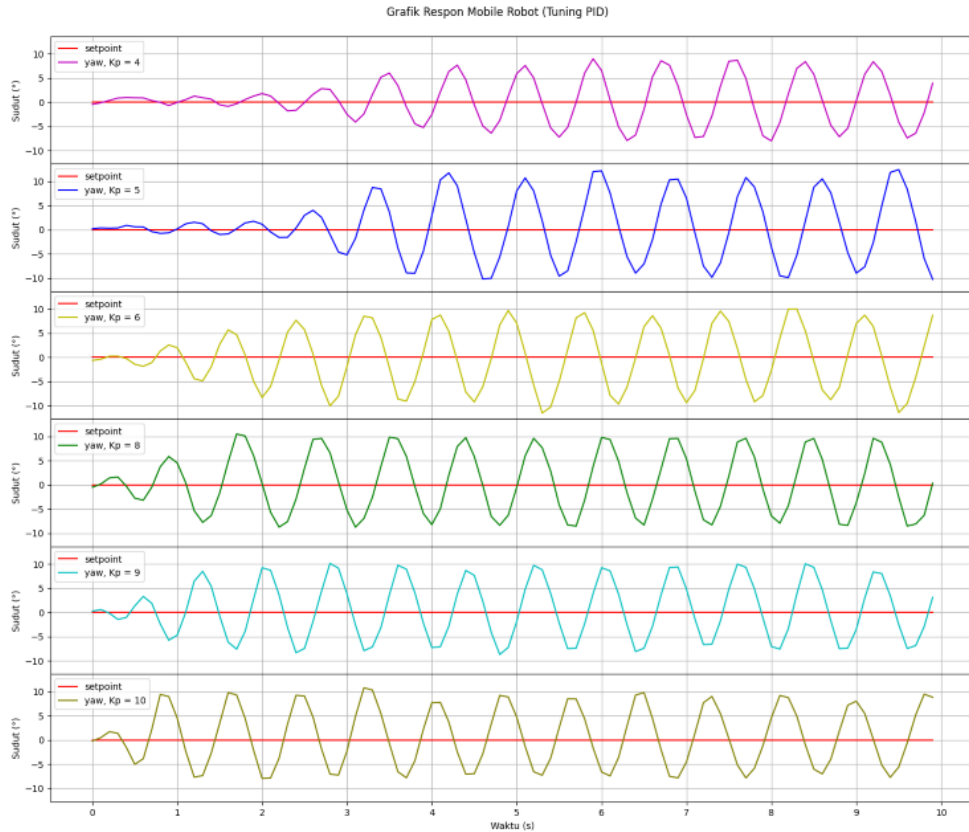


Gambar 4.1 Alur kalibrasi sensor IMU

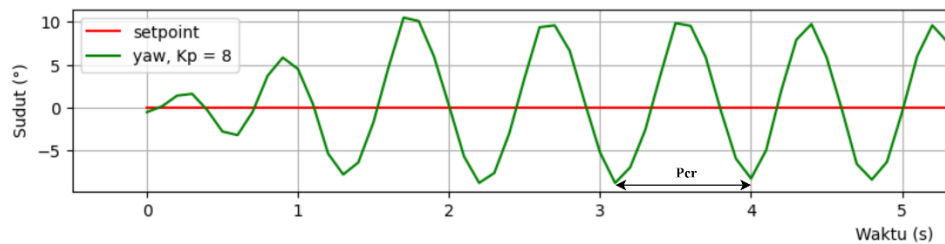
Kalibrasi sensor IMU diperlukan karena ketika sistem pertama kali diaktifkan, pembacaan awal sensor pada sudut yaw tidak selalu bernilai nol. Hal tersebut menyebabkan robot akan bergerak menuju $\text{setpoint} = 0$ di mana sudut nol pembacaan sensor tidak selalu di posisi yang sama. Dalam mengatasi hal tersebut, gambar 4.1 menunjukkan alur kalibrasi sensor IMU dengan menjadikan pembacaan awal sudut sebagai *setpoint*.

4.3 Tuning Kendali PID

Pada proses tuning PID, dilakukan percobaan dengan menggunakan metode Ziegler-Nichols 2 dimana nilai KP dari nilai minimal dinaikan sampai mendapat respon yang cepat dan berosilasi yang berkelanjutan. Grafik respon pergerakan mobile robot yang dibaca sensor IMU berdasarkan nilai KP ditunjukkan pada gambar 4.2.



Gambar 4.2 Grafik Respon Mobile Robot pada Tuning PID



Gambar 4.3 Nilai periode kritis (Pcr) dari grafik respon

Berdasarkan gambar di atas, dapat dilihat bahwa pada konstanta proposional (KP) dengan nilai sebesar 8, robot mengalami gerakan osilasi secara berkelanjutan dengan amplitudo yang stabil. Nilai periode Pcr bisa didapatkan dari grafik respon tersebut seperti yang ditunjukkan pada gambar 4.3. Dengan demikian, nilai PID dapat dihitung dengan metode Ziegler-Nichols, yaitu:

$$K_{cr} = 8$$

$$P_{cr} = (4.0 - 3.1) \times 1 = 0.9 \text{ s}$$

$$T_i = 0.5 \times 0.9 = 0.45 \text{ s}$$

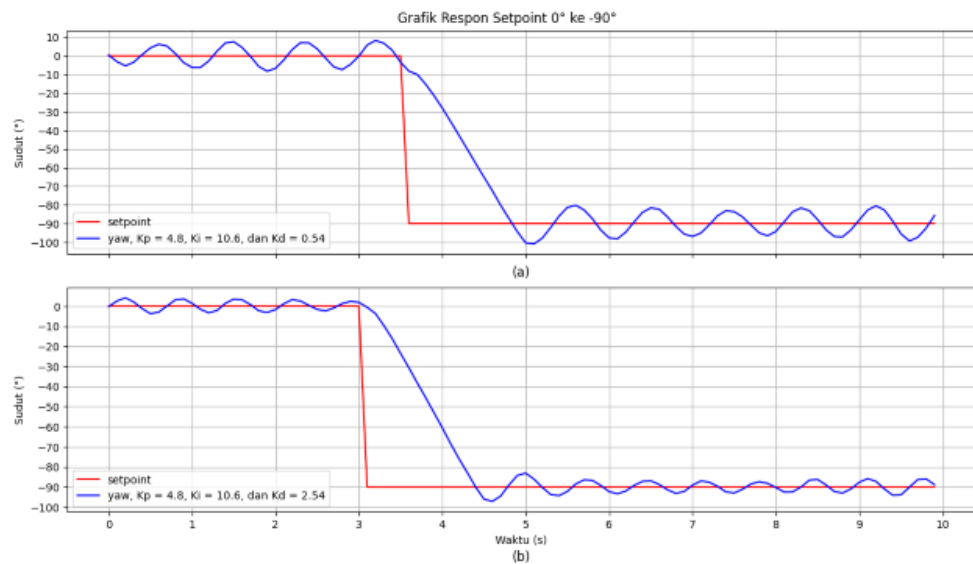
$$T_d = 0.125 \times 0.9 = 0.1125 \text{ s}$$

$$K_p = 0.6 \times 8 = 4.8$$

$$K_i = \frac{4.8}{0.45} = 10.6$$

$$K_d = 4.8 \times 0.1125 = 0.54$$

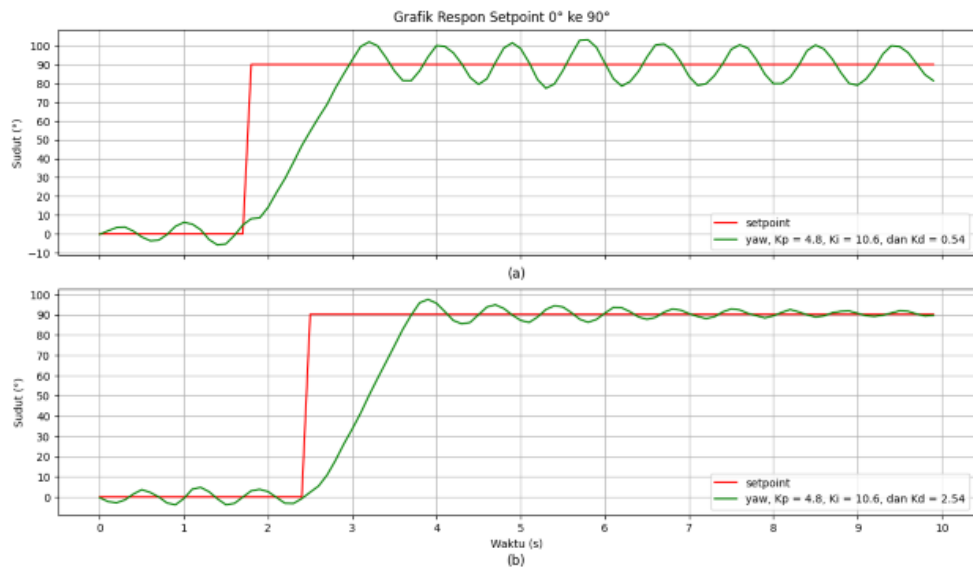
Berdasarkan hasil perhitungan tuning PID dengan menggunakan metode Ziegler-Nichols 2 didapatkan nilai $K_p = 4.8$, $K_i = 10.6$, dan $K_d = 0.54$. Hasil perhitungan nilai tersebut diuji dengan memasukkan ke dalam program kendali PID dengan melakukan dua percobaan yaitu untuk mengetahui respon ketika setpoint dari 0 ke -90 (belok kanan) dan setpoint dari 0 ke 90 (belok kiri). Perubahan nilai setpoint tersebut merupakan hasil keputusan dari logika fuzzy berdasarkan pembacaan objek disekitar. Respon pergerakan robot ditunjukkan pada gambar 4.4 dan 4.5.



Gambar 4.4 Grafik Respon Mobile Robot dengan setpoint 0 ke -90,
(a) $K_d = 0.54$, dan (b) $K_d = 2.54$

Gambar di atas menunjukkan respon dimulai dengan nilai setpoint 0 dan robot bergerak berosilasi pada setpoint tersebut. Pada detik ke 3.5, robot belok ke kanan untuk mengubah setpoint 0 menjadi -90 dari keputusan logika fuzzy. Pada gambar 4.4 (a) didapatkan nilai *settling time* = 4.7 s, *risetime* = 1.2 s, *overshoot* = 12.16%, dan *steady state error* = 8.34% yang menyebabkan

osilasi robot masih cukup besar. Langkah yang dilakukan untuk mengurangi osilasi atau mengurangi nilai *steady state error* yaitu bisa dengan menaikkan nilai konstanta K_d seperti yang dilakukan oleh A. N. Albab dkk. [32] dengan hasil pengujian yang mendapatkan nilai *steady state error* yang menurun, namun perlu diketahui jika nilai K_d terlalu besar dapat menyebabkan ketidakstabilan hingga akhir. Berdasarkan hal tersebut, maka nilai K_d dinaikan hingga didapatkan menjadi 2.54, sehingga didapatkan nilai *settling time* = 3.3 s, *risetime* = 1.3 s, *overshoot* = 7.97%, dan *steady state error* = 3.06%. Dapat dilihat pada gambar 4.4 (b) yang menunjukkan respon robot dengan *steady state error* yang berkurang dan meredam osilasi dari hasil K_d yang dinaikan.



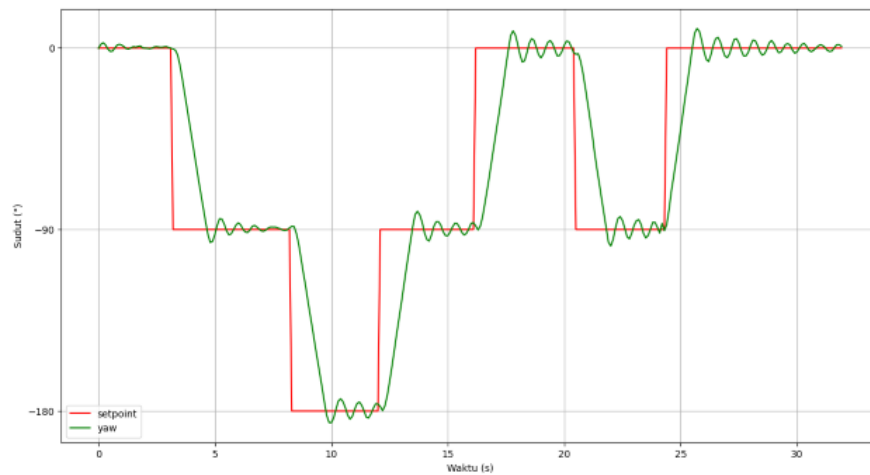
Gambar 4.5 Grafik Respon Mobile Robot dengan setpoint 0 ke 90,
(a) $K_d = 0.54$, dan (b) $K_d = 2.54$

Pengujian kendali PID dilakukan juga untuk robot belok kiri dengan nilai setpoint 0 menjadi 90 dengan respon robot yang ditunjukkan gambar 4.4. Pada gambar 4.5 (a) dengan kondisi yang sama seperti sebelumnya, didapatkan nilai *settling time* = 4.9 s, *risetime* = 1.2 s, *overshoot* = 13.31%, dan *steady state error* = 10.89% yang menyebabkan osilasi robot masih cukup besar. Nilai K_d dinaikan menjadi 2.54 untuk mengurangi *steady state error* dengan respon seperti yang ditunjukkan gambar 4.5 (b), didapatkan nilai

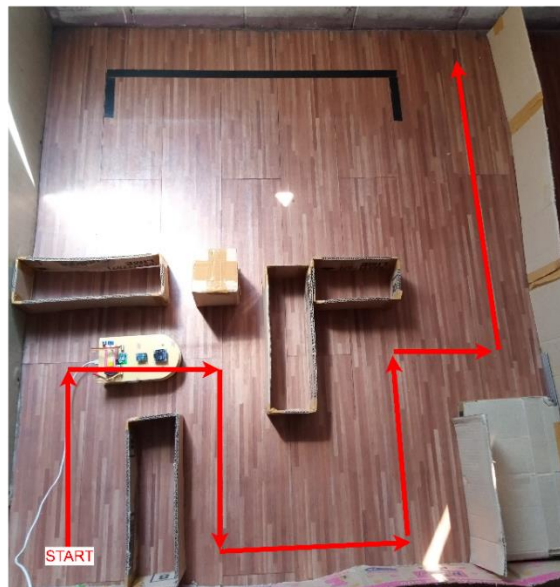
settling time = 5.7 s, *risetime* = 1.2 s, *overshoot* = 8.36%, dan *steady state error* = 2.54%.

4.4 Pengujian Sistem Fuzzy-PID dengan Rintangan

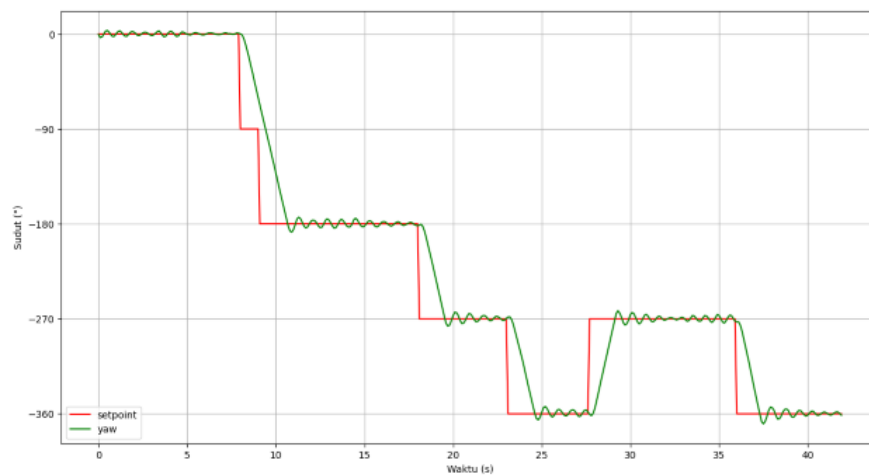
Pengujian keseluruhan sistem yaitu penggabungan logika fuzzy dan kendali PID pada mobile robot dilakukan untuk mengetahui performa kombinasi sistem tersebut. Logika fuzzy diterapkan dengan tujuan agar robot dapat bergerak dengan menghindari rintangan di sekitarnya. Oleh karena itu, sistem kendali robot diuji dengan memberikan dua kondisi rintangan yang berbeda seperti yang ditunjukkan gambar 4.7 dan 4.9.



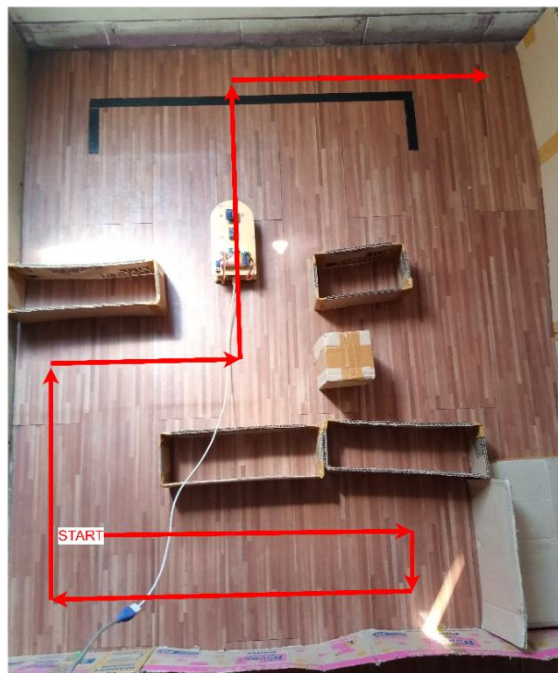
Gambar 4.6 Pengujian Hasil Fuzzy-PID pada Rintangan 1



Gambar 4.7 Ilustrasi Pergerakan Robot pada rintangan 1



Gambar 4.8 Pengujian Hasil Fuzzy-PID pada Rintangan 2



Gambar 4.9 Ilustrasi Pergerakan Robot pada rintangan 2

Berdasarkan pengujian dari kedua rintangan yang berbeda tersebut, dapat dilihat bahwa robot dapat bergerak menghindari rintangan dengan sudut belok -90 untuk ke kanan dan 90 untuk ke kiri. Ketika setpoint berubah berdasarkan keputusan fuzzy, robot akan berbelok dengan kelipatan 90 derajat kemudian berjalan lurus dengan distabilkan oleh kendali PID.