

RANCANG BANGUN KONTROL PROPORSIONAL MENGGUNAKAN METODE ROOT LOCUS DAN KONSEP LEADER-FOLLOWER UNTUK PENJEJAK FORMASI PADA MOBILE ROBOT

PROPORTIONAL DESIGN CONTROL USING ROOT LOCUS METHOD AND LEADER-FOLLOWER CONCEPT ON MOBILE ROBOT FOR FORMATION TRACKING

M Maulana Riswandha¹

Drs. Suwandi, M.si²

Ahmad Qurthobi, ST., MT.³

¹⁻³ Prodi S1 Teknik Fisika, Fakultas Teknik, Universitas Telkom

¹m.maulana.riswandha@gmail.com

²suwandi.sains@gmail.com

³qurthobi@gmail.com

ABSTRAK

Perkembangan teknologi saat ini mengarah pada otomatisasi di berbagai bidang, salah satunya yaitu teknologi robotika. Robot saat ini menjadi solusi alternatif di tengah keterbatasan manusia dalam mengerjakan aktifitas secara kontinyu. Salah satu konsep yang menarik dalam dunia robotika adalah konsep *leader-follower*. Konsep *leader-follower* adalah salah satu hal yang coba diterapkan dalam robot dengan mengadaptasi kebiasaan makhluk hidup. *Leader-follower* sendiri merupakan suatu konsep sederhana yang memposisikan salah satu robot berperan untuk memberikan arahan terhadap robot lain. Untuk mendapatkan respon yang baik antar *mobile robot* diperlukan suatu pengendalian yang mampu mengendalikan kecepatan robot untuk bergerak berdasarkan perbedaan jarak dan sudut. PID dipilih sebagai kontroler pada sistem *mobile robot* karena responnya terhadap perubahan yang cukup cepat. Dengan menggunakan metode *root locus* terhadap model matematis sistem diketahui bahwa sistem memiliki satu *pole* di sebelah kiri sumbu real pada bidang S yang bergerak menuju tak hingga. Sehingga digunakanlah kontrol proporsional untuk mempercepat respon sistem dengan konstanta $K_p = 1.5048$. Dari pengujian yang telah dilakukan didapatkan hasil $error X_{lurus} = 0.84\%$, $error X_{kiri} = 2.58\%$, $error X_{kanan} = 3.67\%$

Kata kunci : *mobile robot, leader follower, PID, root locus, penjejak formasi.*

ABSTRACT

Nowadays the development of technology is heading to industrial otomation in many things, one of them is robotic. Robot is an alternative solution behind the limitation of human ability to doing their activity continuously. One of interesting thing in robotic is leader-follower concept. The leader-follower concept is a thing that is tried to be implemented in robot to adapt human habit. Leader-follower is simple concept that placed one of the robot acts to give command to another robot. In order to get good respon from the system for the mobile robot, it needs controller that can organize the velocity of robot depend on the differences of distance and angle. PID is choosen as controller on mobile robot system because of it capability to produce fast respond. By using root locus on mathematical model of the system can be known that system has one pole in the left of the real axis on S plane that move to $-\infty$. Hence proportional control is used to raise respon of the system with K_p constant = 1.5048. From the testing that has been done the result is $X_{straight}$ error = 0.84%, X_{left} error = 2.58%, X_{right} error = 3.67%.

Keywords : *mobile robot, leader follower, PID, root locus, formation tracking*

1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi saat ini mengarah pada otomatisasi di berbagai bidang, salah satunya yaitu teknologi robotika. Robot saat ini menjadi solusi alternatif di tengah keterbatasan manusia dalam mengerjakan aktifitas secara kontinyu dan tidak mampu bekerja selama 24 jam penuh^[1]. Arah dari perkembangan teknologi ini salah satunya mengarah pada konsep robot swarm, yaitu robot dengan jumlah banyak dengan kesamaan fisik.

Salah satu konsep robot swarm yang menarik untuk diteliti adalah skema *leader follower*. Secara umum skema *leader follower* adalah skema yang sangat sering dijumpai dalam kebiasaan hidup makhluk hidup. Konsep mengikuti pemimpin didasarkan pada kebutuhan suatu kelompok untuk mencapai keteraturan. Hal terpenting dalam konsep *leader follower* adalah mengontrol formasi dari robot, sehingga robot dapat mencapai bentuk formasi yang diinginkan. Topik penelitian mobile robot dengan skema leader follower ini memiliki tantangan karena harus mengintegrasikan antara pengukuran, komunikasi dan pengendalian dalam waktu yang relatif singkat^[2].

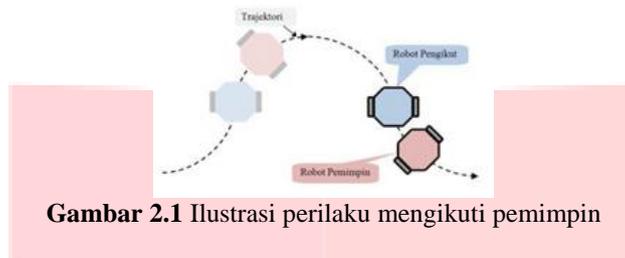
Tujuan yang ingin dicapai dari penelitian ini adalah menjaga formasi antar robot dengan *setpoint* sejauh 15 cm serta mengetahui kelakuan robot selama robot bergerak. Untuk merealisasikan skema tersebut dibutuhkan pengendalian yang konvensional agar robot pemimpin dapat memberikan komunikasi yang baik kepada robot

pengikut. Pengontrolan yang digunakan pada penelitian ini adalah kontrol PID. Penggunaan kontrol PID pada penelitian ini adalah untuk melanjutkan penelitian sebelumnya yang berkaitan dengan *mobile robot*, sehingga dapat mempermudah dalam menganalisis permasalahan yang terjadi dengan membandingkan penelitian yang telah tercapai sebelumnya.

2. PERANCANGAN SISTEM

2.1 Konsep Leader Follower

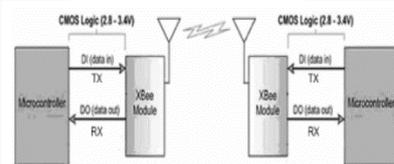
Perilaku mengikuti pemimpin atau lebih dikenal dengan *Follow The Leader (leader-follower)* merupakan salah satu skema dari *Robot Swarm*. Dalam formasi mengikuti pemimpin ini terdapat satu robot yang berperan sebagai robot *leader* dan sebagian yang lain sebagai robot *follower*. Tujuan dari pengontrolan pada *follower* adalah merancang agen yang mampu berkomunikasi dengan *leader* dan menambahkan kompensasi yang diperlukan untuk mempertahankan formasi^[3]. Ilustrasi perilaku mengikuti pemimpin ditunjukkan pada gambar 2.1



Gambar 2.1 Ilustrasi perilaku mengikuti pemimpin

2.2 Modul Wireless Xbee

Modul Wireless Xbee adalah modul komunikasi yang akan digunakan untuk menghubungkan komunikasi dari robot leader ke robot follower. Modul Wireless Xbee merupakan modul *Radio Frequency transceiver*. *Radio Frequency Transceiver* atau pengirim dan penerima frekuensi radio ini berfungsi untuk komunikasi secara *full duplex*. Salah satu modul komunikasi *wireless* dengan frekuensi 2.4Ghz adalah Xbee- ZigBee/IEEE 802.15.4 2.4GHz^[4]. Ilustrasi kerja modul Xbee ditunjukkan pada gambar 2.2



Gambar 2.2 Ilustrasi Kerja Modul Xbee

2.3 Sistem Kontrol PID

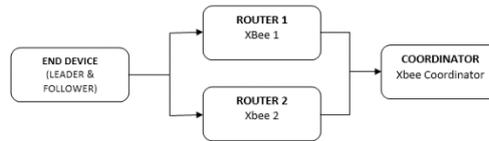
PID (Proporsional-Integral-Derivatif) merupakan kontrol untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem. Komponen kontrol PID ini terdiri dari tiga jenis yaitu Proporsional, Integratif, dan Derivatif. Ketiganya dapat digunakan bersamaan maupun sendiri-sendiri bergantung dari respon yang diinginkan terhadap suatu *plant*^[6]. Fungsi alih $H(s)$ pada kontrol PID ditunjukkan pada persamaan 2.1

$$H(s) = \frac{Kd s^2 + Kp s + Ki}{s^3 + Kd s^2 + Kp s + Ki} \quad (2.1)$$

Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran system terhadap masukan tertentu sebagaimana yang diinginkan. Untuk merancang sistem kontrol PID, kebanyakan dilakukan dengan metoda coba-coba atau (*trial & error*). Hal ini disebabkan karena parameter K_p , K_i dan K_d tidak independen. Untuk mendapatkan aksi kontrol yang baik diperlukan langkah coba-coba dengan kombinasi antara P, I dan D sampai ditemukan nilai K_p , K_i dan K_d seperti yang diinginkan.

2.4 Perancangan Robot

Perancangan robot dilakukan dengan berdasarkan pada diagram blok pada gambar Robot pengikut dibuat dengan spesifikasi sensor yang lengkap dengan tujuan untuk mengukur parameter-parameter yang berkaitan dengan gerak *mobile robot*. Baik robot pemimpin dan pengikut juga dilengkapi dengan modul komunikasi nirkabel untuk bisa berkomunikasi dengan *station ground*.



Gambar 2.3 Gambaran Sistem Keseluruhan

2.5 Blok Diagram

Blok diagram sistem secara keseluruhan dibagi menjadi tiga bagian utama yaitu *node end device*, *node router*, dan *node coordinator*.

1. Node Coordinator

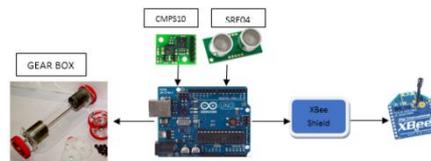
Blok sistem ini terdiri dari Arduino Mega dengan satu modul Xbee yang difungsikan sebagai coordinator. Fungsi dari blok ini adalah untuk melakukan pemantauan data-data pengukuran yang masuk seperti jarak dan selisih antar robot.

2. Node Router

Blok diagram ini terdiri dari arduino mega dan 2 buah modul Xbee sebagai *router*. Modul Xbee diletakkan masing-masing dikoordinat (3,0) dan (0,3) dengan dihubungkan kabel untuk memfasilitasi komunikasi antara Xbee dengan arduino mega. Modul dalam *node* ini berfungsi untuk menerima data yang dikirim oleh masing-masing robot.

3. Node End Device

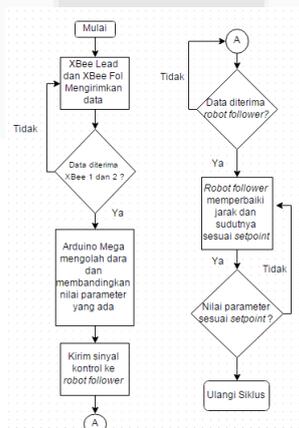
Blok sistem pada *node* ini merupakan robot *leader* dan *follower* yang dilengkapi dengan modul Xbee, Xbee shield, CMPS10, HC SR04, driver motor, LCD, baterai lithium, gear box dan arduino. Tujuan dari *node* ini adalah untuk mendapatkan nilai perbandingan antara sudut dan jarak dari masing-masing robot. Blok sistem dari *node* ini ditunjukkan pada gambar



Gambar 2.4 Blok Sistem *node end device*

2.6 Perancangan Model Komunikasi Mobile Robot

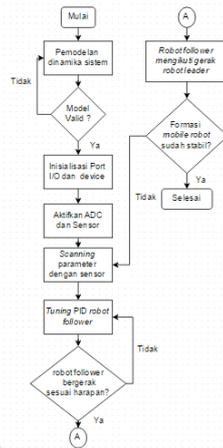
Alur komunikasi yang dibuat adalah komunikasi dua arah antara modul komunikasi di tiap robot. Namun dalam hal ini antara robot pengikut dan pemimpin tidak berkomunikasi secara langsung. Untuk format data kiriman dari komputer kepada pengikut berupa kiriman PWM (0-255), tujuannya agar data kiriman cepat sampai sehingga yang dikirim hanya berupa angka. Gambar 2.5 adalah model komunikasi untuk sistem robot.



Gambar 2.5 Sistem Komunikasi *Leader Follower*

2.7 Perancangan Perangkat Lunak Mobile Robot Skema Leader Follower

Sistem kendali yang digunakan pada perancangan *mobile robot* ini adalah dengan menggunakan PID, tujuannya adalah untuk menjaga kestabilan formasi antara robot pemimpin dengan robot pengikut. Gambar 2.6 adalah diagram alir pengendalian robot dengan PID.



Gambar 2.6 Diagram Alir Perangkat Lunak Pengendali *mobile robot*

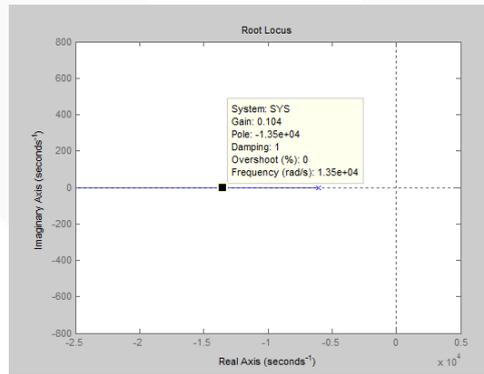
3. Pembahasan

3.1 Implementasi Kontrol

Sebelum menentukan pengontrolan untuk *mobile robot*, perlu dicari terlebih dahulu model matematis dari motor dc[7]. Dengan menemukan parameter seperti torsi maksimum dan kecepatan maksimum motor dc didapatkan model matematis sistem motor DC yaitu sebagai berikut :

$$\frac{\omega_m(s)}{E_a(s)} = \frac{716000}{s + 60400} \tag{3.1}$$

Dengan *root locus* dapat diketahui pergerakan pole pada loop tertutup akibat perubahan besarnya penguatan *gain*. Dari plot *root locus* fungsi transfer yang didapat bisa dilihat pada gambar 4.7 bahwa sistem hanya mempunyai satu *pole* di -60400 yang nilainya menuju minus tak hingga sehingga sistem bisa dikontrol menggunakan kontrol proporsional saja. Dan karena rentang penguatannya berada di sebelah kiri sampai menuju minus tak hingga maka pada dasarnya sistem ini akan tetap stabil dengan penguatan *gain* berapapun. Sehingga dilakukan *tuning* konstanta proporsional untuk mendapatkan nilai *Kp* yang tepat.



Gambar 3.1 Root Locus Fungsi Transfer Mobile Robot

Tabel 3.1 Tabel Tuning Parameter *Kp*

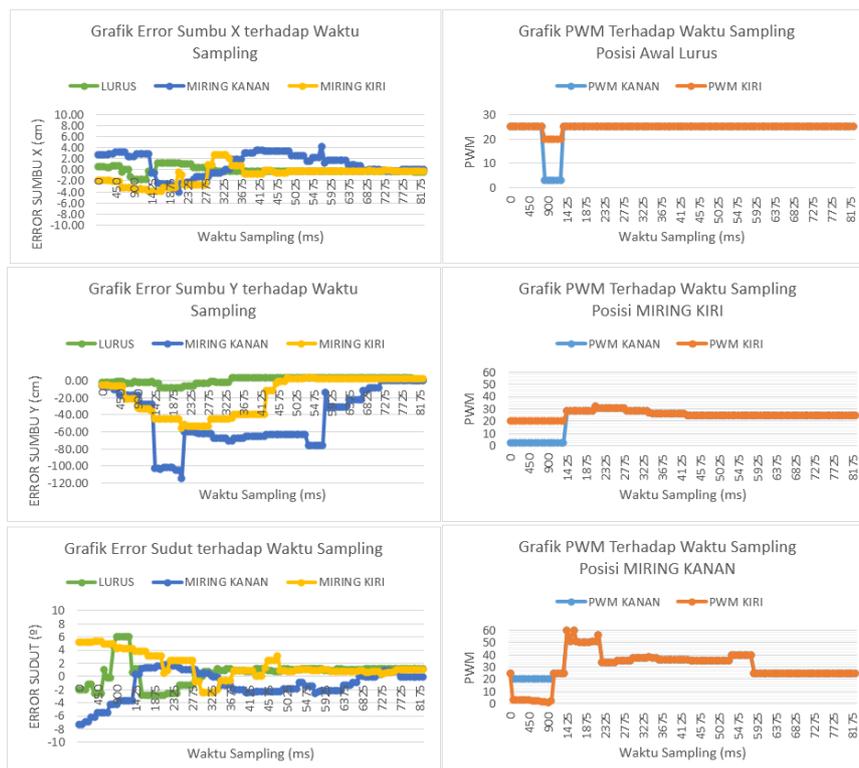
<i>Kp</i>	<i>Tr</i> (S)	<i>Ts</i> (S)	Peak Amplitude	OS (%)	Steady State
0.11085	1.57 x 10 ⁻⁶	2.80 x 10 ⁻⁶	0.558	0	0.558
0.2157	1.02 x 10 ⁻⁶	1.82 x 10 ⁻⁶	0.718	0	0.718
0.31142	7.55 x 10 ⁻⁶	1.38 x 10 ⁻⁶	0.787	0	0.787
0.40408	6.28 x 10 ⁻⁶	1.12 x 10 ⁻⁶	0.827	0	0.827
0.5046	5.21 x 10 ⁻⁶	9.28 x 10 ⁻⁶	0.857	0	0.857
0.60404	4.46 x 10 ⁻⁶	7.94 x 10 ⁻⁶	0.877	0	0.877
0.72403	3.80x10 ⁻⁶	6.76 x 10 ⁻⁶	0.896	0	0.896

0.82994	3.36×10^{-6}	5.98×10^{-6}	0.908	0	0.908
0.90922	3.09×10^{-6}	5.50×10^{-6}	0.915	0	0.915
1	2.83×10^{-6}	5.04×10^{-6}	0.922	0	0.922
1.1428	2.50×10^{-6}	4.45×10^{-6}	0.931	0	0.931
1.2525	2.30×10^{-6}	4.09×10^{-6}	0.937	0	0.937
1.3113	2.2×10^{-6}	3.91×10^{-6}	0.94	0	0.94
1.5048	1.93×10^{-6}	3.44×10^{-6}	0.947	0	0.947
2.531	1.17×10^{-6}	2.09×10^{-6}	0.968	0	0.968

Berdasarkan tabel 3.1, nilai K_p yang paling cepat responnya adalah sebesar 2.531, akan tetapi pada saat implementasi ke kontroler nilai tersebut menyebabkan *mobile robot* terlalu sensitif, sehingga perubahan *error* sedikit saja direspon berlebihan. Oleh karenanya dipilih K_p sebesar 1.5048 dengan kondisi yang lebih stabil setelah diimplementasikan ke dalam sistem, *rise time* dan *settling time* yang cepat, *Overshoot* sebesar 0 dan nilai tunak yang mendekati nilai *setpoint* sebesar 1.

3.2 Hasil Pengujian

Grafik hasil pengujian dapat terlihat di gambar 3.2, dalam grafik tersebut terdapat data pengujian untuk tiga kondisi yaitu lurus (hijau), biru (miring kanan) dan kuning (miring kiri). Parameter yang diukur adalah error sudut dan posisi (x,y) dari robot sementara keluaran yang dihasilkan adalah dalam bentuk PWM.



Gambar 3.2 Grafik hasil pengujian sistem

Pada saat waktu 900-1425 ms hasil yang ditunjukkan oleh grafik PWM saat kondisi awal “lurus” terlihat ada perbedaan nilai PWM. Hal tersebut dikarenakan pada rentang waktu tersebut *error* sudut antar *robot leader* dan *robot follower* melebihi 4° , akhirnya *robot follower* yang dikontrol dengan kontrol proporsional merespon dengan memberikan PWM kanan dan PWM kiri yang berbeda agar *robot follower* kembali ke formasinya. Setelah 1425 ms maka sistem bergerak lurus (PWM kanan=PWM kiri) karena sudut ygn diterima sistem setelah waktu tersebut masih dalam jangkauan $-4^\circ < \theta < 4^\circ$.

Pada saat waktu 0-900 ms hasil yang ditunjukkan oleh grafik PWM saat kondisi awal “miring kiri” terlihat terjadi perbedaan PWM yang artinya *robot follower* sedang berusaha mengembalikan posisi sesuai formasi karena

posisi awalnya menghadap kiri. Pada waktu tersebut sudut yang terukur awalnya adalah lebih besar dari 4° , oleh karenanya *robot follower* langsung merespon dengan memberikan PWM kiri lebih besar dibanding PWM kanan. Setelah waktu 1425 ms rentang perubahan sudut masih dalam rentang $-4^\circ < \Theta < 4^\circ$, sehingga robot akan bergerak lurus dengan PWM kiri sama dengan PWM kanan.

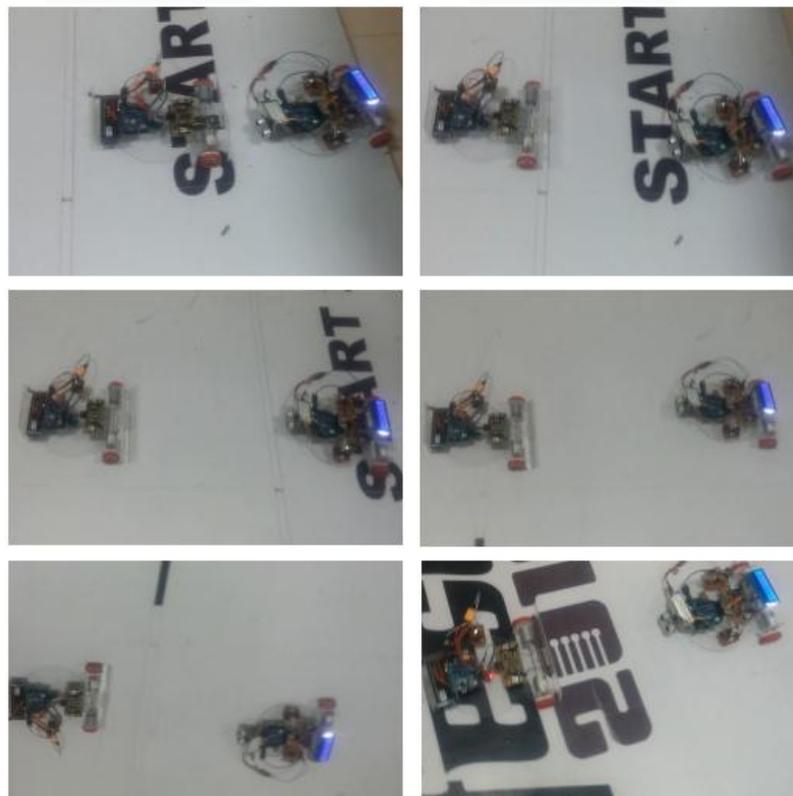
Kemudian pada saat waktu 0-1435 ms hasil yang ditunjukkan oleh grafik PWM saat kondisi awal “miring kanan” juga terlihat adanya perbedaan PWM (PWM kiri < PWM kanan) yang artinya robot follower sedang berusaha mengembalikan posisi sesuai formasi karena posisi awalnya menghadap kanan. Pada waktu tersebut sudut yang terukur awalnya adalah kurang dari -4° , oleh karenanya *robot follower* langsung merespon dengan memberikan PWM kanan lebih besar dibanding PWM kiri. Setelah waktu 1435 ms rentang perubahan sudut masih dalam rentang $-4^\circ < \Theta < 4^\circ$, sehingga robot akan bergerak lurus dengan PWM kiri sama dengan PWM kanan.

Hasil keseluruhan dari ketiga pengujian tersebut kemudian diolah untuk bisa mengetahui seberapa jauh *robot follower* menyimpang dari *robot leader*. Formasi yang dibentuk antara *robot leader* dan *robot follower* adalah formasi segaris atau dengan kata lain jika robot bergerak sepanjang sumbu Y, maka untuk mengetahui penyimpangan formasi *robot follower* terhadap *robot leader* dapat dilihat dari error posisi x *robot follower*. Berikut adalah error sumbu X dari masing-masing pengujian.

Tabel 3.2 Hasil akhir pengolahan data

	Error X
Lurus (-4° sampai 4°)	0.84%
Miring kanan ($>4^\circ$)	3.67%
Miring kiri ($<-4^\circ$)	2.58%

Pada saat kondisi awal “lurus” perubahan sudut tidak begitu besar sehingga robot follower cenderung mempertahankan formasi segarisnya dengan error 0.84%. Selain itu dapat dilihat juga dari keluaran PWM saat kondisi awal “lurus” lebih dari 90% keluaran yang dihasilkan nilai PWM nya adalah sama, hal tersebut menunjukkan bahwa kondisi *robot follower* sudah segaris dengan *robot leader*. Sementara pada kondisi “miring kanan” dan “miring kiri” terlihat *error*nya berturut-turut adalah 3.67% dan 2.58%. Hal tersebut terjadi karena pembacaan sensor kompas yang lebih lambat dibandingkan respon sistem, sehingga terjadi keterlambatan sinyal kontrol yang diterima robot. Karena keterlambatan tersebut konsekwensinya adalah *error* sudut semakin membesar, ketika *error* sudut membesar *error* sumbu x pun semakin membesar.



Gambar 3.3 Hasil Implementasi *mobile robot*

Data yang dihasilkan sebelumnya adalah saat pengujian dengan *delay program* dari *robot follower* adalah 100 ms *Station garound* 75 ms dan *robot leader* sebesar 100 ms. *Delay* tersebut dipilih karena hasil pergerakan mobil sesuai dengan yang diinginkan yaitu mengikuti *robot leader* dengan setpoint tertentu. Berikut data beberapa pengujian *delay* terhadap kelakuan *robot* saat implementasi.

Tabel 3.3 Tabel pengujian *delay*

Delay Program Leader (ms)	Delay Program Follower (ms)	Delay Program Station Ground (ms)	Implementasi
550	550	525	Respon robot sedikit lama. Hal tersebut menandakan bahwa ada data yang tidak dikenali masuk. Karena pada program di atur apabila ada data dengan <i>header</i> (pengenal) yang tidak didefinisikan maka PWM Kanan=PWM kiri yaitu sama dengan PWM sebelumnya. Sehingga robot responnya kurang bagus yaitu baru membaca pengukuran jauh ketika jaraknya sudah dekat. Pada jarak dekat harusnya robot mengurangi kecepatan tapi pada kondisi ini robot malah memberikan PWM yang paling maksimal, sehingga robot bertabrakan. Data yang dikenali adalah <i>header</i> seperti F (untuk follower), L (untuk leader), @ (untuk jalan lurus), #(untuk berbelok kanan atau kiri).
350	350	325	Sama halnya dengan kondisi <i>delay</i> sebelumnya, pada posisi dekat robot follower mendapat PWM yang maksimal akibatnya terjadi tabrakan.
100	100	75	Respon robot cepat dan pergerakan sesuai yang diharapkan, ketika ada header yang tidak dikenali diterima masuk ke <i>follower</i> , nilai PWM kanan = PWM kiri. Karena <i>delay</i> nya cepat sehingga respon kontrol selanjutnya segera masuk, sehingga <i>follower</i> menerima respon sesuai dengan kondisi jaraknya terhadap <i>leader</i> dan langsung mengeksekusi perubahan yang terjadi. Sehingga tidak terjadi tabrakan antar robot.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem dapat disimpulkan bahwa :

1. Pengontrolan yang cocok dengan sistem mobile robot ini adalah kontrol proporsional dengan besar konstanta K_p adalah sebesar 1.5048.
2. Error sumbu X yang paling kecil adalah saat kondisi awal lurus yaitu sebesar 0.84%, kemudian saat kondisi awal miring kiri yaitu sebesar 2.58% dan saat kondisi awal miring kanan yaitu sebesar 3.67%.
3. Komunikasi menggunakan modul XBee membutuhkan *delay* sehingga respon kontrol yang dikirimkan melalui koordinat diterima oleh *robot follower* sedikit lama. Sehingga perlu diatur *delay* yang tepat agar mobil *follower* dapat menerima respon kontrol di waktu yang tepat. Pada tugas akhir ini *delay* yang tepat untuk *program leader* adalah 100 ms *program follower* adalah 100 ms dan *program station ground* adalah 75 ms

4.2 Saran

Pada perancangan tugas akhir ini terdapat beberapa hal yang dapat dikembangkan lebih lanjut, diantaranya adalah :

1. Sistem kontrol leader-follower bisa disempurnakan lagi dengan penambahan sensor HCSR04 jauh lebih banyak, sehingga sisi yang diukur tidak hanya satu arah saja.
2. Untuk mempercepat respon sistem, diperlukan pengaturan *delay* yang lebih baik agar tidak adanya data yang berbenturan ketika mobil *follower* mengirim dan menerima data

Daftar Pustaka:

- [1] Pitowarno, Endra, (2006). Robotika: Desain, Kontrol, Dan Kecerdasan Buatan. Buku Teks. Yogyakarta: ANDI. 1-3
- [2] McLurkin, James. (2008). *Analysis and Implementation of Distributed Algorithms for Multi-Robot Systems*. Desertasi Massachusetts institute of technology.
- [3] Nofriyanti. 2011. *Rancang Bangun Robot Leader Dan Robot Follower Dengan Sistem Navigasi Sensor Infra Merah.*, PENS, Surabaya.
- [4] Garaudy, Hendrit. *Perancangan Sistem Monitoring Kelembaban dan Temperatur menggunakan Komunikasi Zigbee 2.4 GHz*. Tugas akhir Universitas Diponegoro, Semarang.
- [5] Ahmad Qurthobi. 2011. Perancangan Kontrol Optimal Penjejak Lintasan dan Formasi dengan Konsep Leader - Follower pada Agen Jamak Helikopter Otonom Skala Kecil. Thesis ITB, Bandung.
- [6] Ali, Muhammad. 2004. Pembelajaran Perancangan Sistem Kontrol PID dengan Software Matlab. Jurnal UNY, Yogyakarta.
- [7] S. Norman. *Control Systems Engineering Sixth Edition*. California State Polytechnic University, Pomona.

