

Sistem Distribusi Data Kontrol Pada *Differential Drive Mobile Robot* Menggunakan *Robot Operating System*

Gillang Al Azhar^{1*}, Totok Winarno², Syarifatul Izza³

^{1,2}Politeknik Negeri Malang, Malang

³Politeknik Unisma, Malang

*Penulis Korespondensi, email: gillang_al_azhar@polinema.ac.id

Received:28/10/2022

Revised:29/10/2022

Accepted:29/10/2022

Abstract. Generally, the microcontroller to drive the robot system only uses one unit. This can result in excessive load received by the microcontroller. An overload on the microcontroller results in inaccurate sampling times used in the robot system and can lead to errors or hank in the robot. To reduce errors, the multi-controller system was tested on the Differential Drive (DDMR) robot. The data distribution system between controllers utilizes the I2C serial communication protocol, and communication between the controller and the mini-computer uses the rosserial protocol provided by the Robot Operating System (ROS). The results obtained to distribute data with serial I2C takes 1.4 ms, while to communicate with the three slaves it takes 4.18 ms, with the time to communicate with each other between master and slave being 8.36 ms. For distribution with rosserial, it takes 4.7 ms. So the total time for data distribution is 13.06 ms. Tests are also carried out when the robot runs other required algorithms and produces an average sampling time of 63.5 ms. These results indicate that the use of a multi-controller system with data distribution using Serial I2C and rosserial can run optimally because the required sampling time is still faster than 80 ms.

Keywords: DDMR, I2C, rosserial, data distribution, microcontrollers

Abstrak. Pada umumnya mikrokontroler untuk menggerakkan sistem robot hanya menggunakan satu buah saja. Hal ini dapat mengakibatkan berlebihnya beban yang diterima oleh mikrokontroler. Kelebihan beban pada mikrokontroler mengakibatkan tidak akuratnya waktu sampling yang digunakan pada sistem robot tersebut dan dapat berujung pada error atau hank pada robot. Untuk mengurangi terjadinya error, maka diujicobakan sistem multikontroler pada robot *Differential Drive*(DDMR). Sistem distribusi data antar kontroler memanfaatkan protokol komunikasi serial I2C, dan komunikasi antara kontroler dengan mini komputer menggunakan protokol rosserial yang disediakan oleh *Robot Operating System* (ROS). Hasil yang didapatkan untuk melakukan distribusi data dengan serial I2C membutuhkan waktu sebesar 1.4 ms, sedangkan untuk melakukan komunikasi ketiga slave dibutuhkan waktu sebesar 4.18 ms, dengan waktu untuk saling berkomunikasi antara master dan slave didapatkan sebesar 8.36 ms. Untuk distribusi dengan rosserial dibutuhkan sebesar 4.7 ms. Sehingga total waktu untuk distribusi data sebesar 13.06 ms. Pengujian juga dilakukan ketika robot menjalankan algoritma lain yang dibutuhkan, dan menghasilkan rata - rata waktu sampling sebesar 63.5 ms. Hasil tersebut menunjukkan bahwa penggunaan sistem multi kontroler dengan distribusi data menggunakan Serial I2C dan rosserial dapat berjalan secara optimal, karena waktu sampling yang dibutuhkan masih lebih cepat dari 80 ms.

Kata Kunci: DDMR, I2C, rosserial, distribusi data, mikrokontroler

I. PENDAHULUAN

Semakin *robust* sebuah sistem kontrol dari robot, semakin dibutuhkannya sistem yang memiliki ketahanan dalam mengolah data serta memberikan algoritma dan perintah untuk mengendalikan sebuah robot [1] [2]. Penggunaan perangkat dengan yang memiliki spesifikasi mempunyai contohnya

menggunakan mini komputer [3], kemudian mikrokontroler yang memiliki clock yang tinggi [4] sudah menjadi kebutuhan yang mutlak. Selain itu dibutuhkan juga sensor – sensor yang memiliki kegunaan khusus, seperti penggunaan kamera [5] [6] yang digunakan untuk mendapatkan input data berupa gambar, sensor laser [7] [8] dan sensor

ultrasonik yang digunakan untuk mengetahui jarak benda disekitar robot, sensor kompas yang dapat digunakan untuk memberikan koreksi terhadap gerak dan arah hadap robot [9] [10], serta sensor – sensor lain yang dapat dimanfaatkan oleh sistem kontrol robot [11]. Sensor sensor tersebut dibutuhkan, karena robot tidak bisa hanya menggunakan sensor odometry roda saja [12] [13].

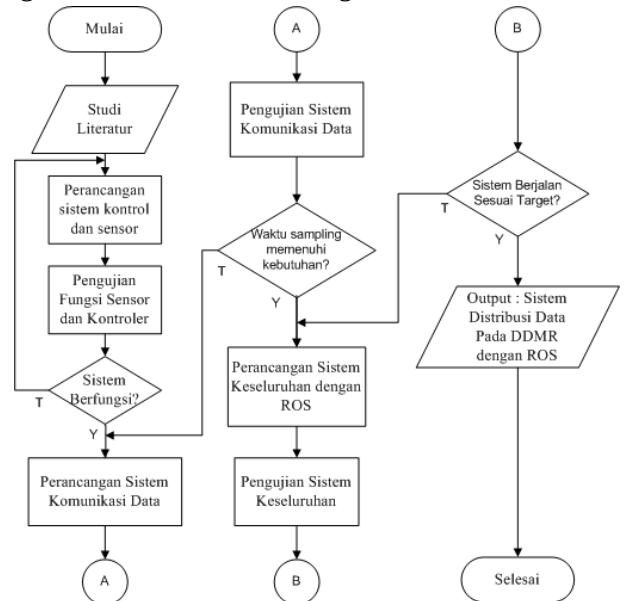
Namun dibalik bannyaknya perangkat yang dibutuhkan serta banyaknya sensor yang digunakan, maka semakin kompleks sistem kontrol robot tersebut [14]. Salah satunya adalah diperlukan tingkat akurasi waktu sampling yang digunakan untuk mengontrol serta membaca sensor – sensor yang digunakan, dimana seluruh proses tersebut dilakukan secara simultan atau secara bersamaan [15]. Hal tersebut dapat memicu terjadinya keterlambatan dalam hal waktu sampling, karena pembacaan data sensor juga membutuh waktu, yang rata – rata membutuhkan waktu sebesar 10 ms – 50 ms. Sedangkan kebutuhan waktu sampling yang dibutuhkan sebuah robot paling lambat adalah 100 ms. Sehingga jika terjadi pembacaan sensor yang dilakukan secara bersamaan dengan menggunakan kontroler yang sama dapat membuat terjadinya keterlambatan waktu sampling pada sistem robot, dan berakibat pada tidak optimalnya sistem kontrol dari robot tersebut.

Untuk itu, pada jurnal ini akan dibahas sistem kontrol robot yang memanfaatkan multi kontroler yang digunakan untuk mengurai kebutuhan waktu dalam pembacaan sensor dan pengiriman sinyal kontrol, yang diaplikasikan pada robot beroda dengan jenis *Differential Drive Mobile Robot* (DDMR) yang juga memanfaatkan *Robot Operating System* (ROS) sebagai sarana distribusi data antara mini komputer dan mikrokontroler. Sehingga dengan menggunakan pemisahan kontroler dan pemanfaatan ROS, waktu sampling yang ada pada robot menjadi lebih akurat dan optimal yang dibuktikan melalui pengujian yang datanya disajikan pada jurnal ini.

II. METODOLOGI

Pada penelitian yang telah dilaksanakan, metodologi yang dilakukan adalah seperti yang ada

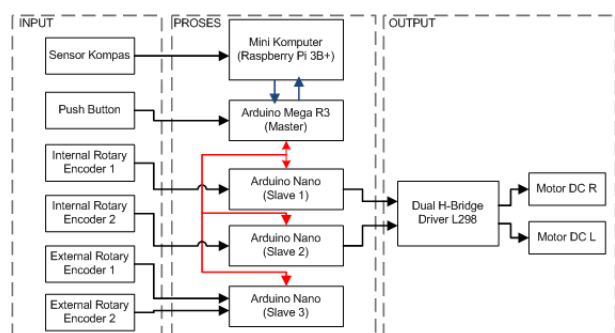
pada diagram alir pada gambar 1. Untuk tahapan inti yang dilaksanakan adalah sebagai berikut.



Gambar 1. Diagram Alir Metodologi Penelitian

A. Perancangan kontroler dan sensor

Kontroler robot terbagi menjadi 2 bagian, yaitu *High Level Control* (HLC) dan *Low Level Control* (LLC). Dimana HLC digunakan untuk menjalankan algoritma khusus seperti *robot localization* [16] [17], kontrol kinematik, *sensor fusion* [18], serta algoritma lain yang tertanam pada robot [19] [20]. Sedangkan untuk LLC, digunakan untuk membaca internal rotary encoder, dan eksternal rotary encoder, serta digunakan untuk mengendalikan kecepatan dan arah putar roda penggerak dari robot. Berikut merupakan diagram blok sistem dari DDMR yang dibuat.



Gambar 2. Diagram Blok Sistem Rangkaian Sensor dan Kontroler

Berdasarkan gambar 2. perakitan ditunjukkan melalui arah panah dengan warna yang berbeda – beda. Dapat dilihat untuk robot yang digunakan menggunakan 4 buah kontroler yaitu 1 buah Arduino Mega 2560 R3 sebagai master kontroler, dan 3 buah Arduino nano sebagai Slave. Komunikasi data antara Arduino master dan Arduino slave adalah menggunakan komunikasi data serial I2C, dimana hubungan antar Arduino tersebut dilakukan melalui pin SDA dan SCL (Arah panah dengan warna merah). Kemudian untuk HLC digunakan mini komputer dengan jenis raspberry Pi 3b+, dimana untuk komunikasi data dengan Arduino master menggunakan protokol yang telah disediakan oleh ROS, yaitu dengan menggunakan roserial. Sedangkan untuk perangkaan lainnya (arah panah warna hitam) diubungkan melalui pin GPIO yang dimiliki antar perangkat.

B. Perancangan Sistem Komunikasi Data

Sistem Komunikasi data yang digunakan pada sistem robot terbagi menjadi 2 macam komunikasi, yaitu komunikasi Serial I2C, dan komunikasi roserial.

- *Komunikasi Serial I2C*

Komunikasi Serial I2C digunakan untuk distribusi data pada LLC, dimana terdapat 4 buah Arduino yang terhubung melalui komunikasi ini, yaitu 1 Buah Arduino Mega 2560 R3, dan 3 buah Arduino Nano R3. Komunikasi Serial I2C merupakan sistem komunikasi yang bersifat half duplex, dimana dibutuhkan 1 kontroler utama yang dijadikan sebagai master dan kontroler lainnya digunakan sebagai slave. Hubungan yang digunakan antara Arduino adalah melalui pin I2C yaitu ada pada pin SDA dan SCL. Berikut merupakan konfigurasi pin yang digunakan pada masing – masing Arduino yang digunakan untuk komunikasi Serial I2C

Tabel 1. Konfigurasi Pin Arduino Untuk I2C

	Konfigurasi Pin	
	SDA	SCL
Arduino Mega	GPIO 20	GPIO 21
Arduino Nano	GPIO 18	GPIO 19

Data yang dikirimkan melalui protokol ini adalah data kecepatan putar motor aktual, data kecepatan roda encoder eksternal, dan nilai output dari algoritma kontrol kinematik yang digunakan untuk mengatur kecepatan dan arah putaran motor.

- *Komunikasi melalui roserial*

Komunikasi roserial merupakan salah satu protokol yang telah disediakan oleh ROS (*Robot Operating System*). Dimana komunikasi ini digunakan untuk melakukan distribusi data antara mini komputer dan kontroler master. Pada dasarnya komunikasi ini menggunakan protokol yang sama dengan protokol komunikasi serial, namun paket data yang digunakan adalah paket data yang standar digunakan oleh ROS. Berikut merupakan paket data yang digunakan komunikasi antara mini komputer dan mikrokontroler

Tabel 2. Nama topic dan tipe data ROS yang digunakan untuk komunikasi roserial

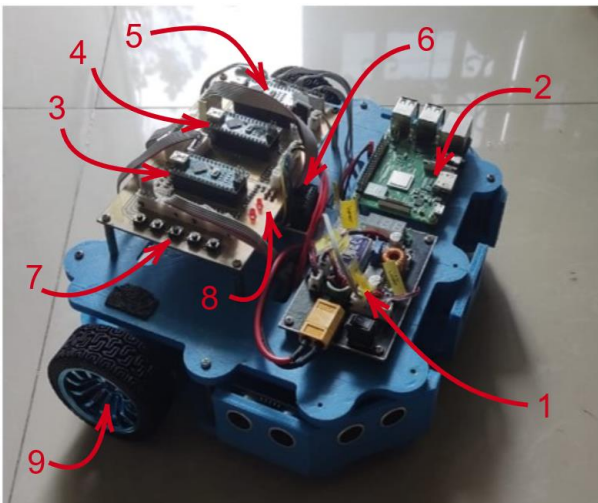
Nama Topic	Tipe Data	Keterangan
cmd_vel	Vector3	Data yang berisikan Output dari algoritma kinematik
wheels_enc	Vector3	Data yang berisikan kecepatan putar motor robot
ext_enc	Vector3	Data yang berisikan kecepatan putar roda yang terpasang pada encoder eksternal yang digunakan sebagai input Forward Kinematik
button_val	Int32	Data yang menunjukkan informasi tombol yang ditekan
Imu_val	Imu	Data yang berisikan informasi dari sensor kompas yang terdiri dari data akselerometer, gyroscope, magnetometer yang ada pada sensor kompas.

C. Pengujian kecepatan waktu sampling dan komunikasi data

Terdapat 4 buah pengujian yang dilakukan pada penelitian yang dilakukan, dimana 4 pengujian tersebut dilakukan untuk mengetahui waktu yang dibutuhkan untuk transmisi data melalui masing – masing protokol komunikasi serta waktu sampling ketika robot bekerja secara *full system*. Dimana waktu yang paling lambat untuk waktu sampling adalah 100 ms, tetapi agar sistem dapat lebih *robust* dan tidak mudah mengalami error atau *hank* maka waktu dari waktu sampling yang boleh terpakai adalah sebesar 80%. Sehingga dapat dihitung sebagai berikut

$$\begin{aligned} \text{Max Time} &= 80\% \times T_s \\ &= 80\% \times 0.1s \\ &= 0.08 s \end{aligned} \quad (1)$$

Sehingga bila ditentukan waktu sampling yang digunakan adalah 100 ms, maka berdasarkan persamaan (1) didapatkan maksimal untuk keseluruhan sistem berjalan adalah sebesar 80 ms. Untuk robot DDMR yang digunakan dapat dilihat seperti yang digunakan pada gambar 3.



Gambar 3. Prototipe Robot DDMR yang digunakan dalam penelitian

Pada gambar 3 tersebut dapat dilihat dari komponen – komponen yang digunakan pada robot tersebut. Berikut penjelasan masing – masing komponen yang digunakan pada robot tersebut yang disajikan pada tabel 3.

Tabel 3. List Komponen yang digunakan pada robot berdasarkan gambar 3.

No. Gambar	Penjelasan
1	Power Manajemen
2	Raspberry Pi 3B+
3	Arduino Nano (Slave 1)
4	Arduino Nano (Slave 2)
5	Arduino Nano (Slave 3)
6	Sensor Kompas BNO055
7	Push Button
8	Arduino Mega 2560 Board (Master)
9	Velg dan Motor DC + Rotary encoder

Waktu yang digunakan saat sistem berjalan secara penuh dapat dihitung dengan menggunakan persamaan berikut

$$\sum I2C_TC = I2C_T_1 + I2C_T_2 + I2C_T_3 \quad (2)$$

$$I2C_TC = 2 \times \sum I2C_TC \quad (3)$$

$$TC = I2C_TC + Rosserial_TC \quad (4)$$

Dimana TC merupakan *Time Consumption* atau waktu total yang digunakan untuk menjalankan sistem keseluruhan. $I2C_TC$ merupakan waktu yang dibutuhkan untuk komunikasi data I2C antara master dan slave. Serta $Rosserial_TC$ adalah waktu yang dibutuhkan untuk komunikasi *roserial*.

III. HASIL DAN PEMBAHASAN

Pengujian dibagi menjadi 3 bagian, dimana masing – masing pengujian dilakukan untuk mengetahui peforma akurasi data dan kecepatan transmisi data dari masing – masing protokol yang digunakan.

A. Pengujian Waktu Komunikasi data Serial I2C Antar Kontroler

Pengujian komunikasi antar kontroler dilakukan bertujuan mengetahui tingkat akurasi data yang dikirimkan serta kecepatan transmisi data yang dilakukan melalui waktu transmisi yang dibutuhkan. Pengujian dilakukan dengan cara mengirimkan data dari master ke masing – masing slave. Pada komunikasi data ini error data yang ditargetkan adalah 0%, atau seluruh data yang dikirim dan diterima haruslah sama.

Tabel 4. Hasil Pengujian Akurasi dan kecepatan komunikasi data serial I2C antar Arduino Master dan Arduino Slave 1

No	Data yang dikirim	Data yang diterima	Waktu
1	x : 20 y : 10	x : 20 y : 10	1.2 ms
2	x : 10.4 y : 20.05	x : 10.4 y : 20.05	1.5 ms
3	x : 100 y : 29	x : 100 y : 29	1.2 ms
4	x : 3.13 y : 4.125	x : 3.13 y : 4.125	1.6 ms
5	x : 6 y : 7.4	x : 6 y : 7.4	1.4 ms

Tabel 5. Hasil pengujian akurasi dan kecepatan komunikasi data serial I2C antar Arduino Master dan Arduino Slave 2

No	Data yang dikirim	Data yang diterima	Waktu
1	x : 16 y : 72	x : 16 y : 72	1.2 ms
2	x : 55 y : 3.18	x : 55 y : 3.18	1.4 ms
3	x : 6.72 y : 1.12	x : 6.72 y : 1.12	1.6 ms
4	x : 10.77 y : 784	x : 10.77 y : 784	1.5 ms
5	x : 912 y : 610	x : 912 y : 610	1.3 ms

Tabel 6. Hasil Pengujian Akurasi dan kecepatan komunikasi data serial I2C antar Arduino Master dan Arduino Slave 3

No	Data yang dikirim	Data yang diterima	Waktu
1	x : 814 y : 0.03	x : 814 y : 0.03	1.4 ms
2	x : 391 y : 441	x : 391 y : 441	1.3 ms
3	x : 1.27 y : 0.294	x : 1.27 y : 0.294	1.6 ms
4	x : 8 y : 1.87	x : 8 y : 1.87	1.4 ms
5	x : 62 y : 99	x : 62 y : 99	1.3 ms

Berdasarkan hasil yang didapatkan pada tabel 4 – 6. dapat dilihat bahwa akurasi data dari masing – masing komunikas antar master dan slave memiliki nilai error 0%, hal ini menunjukkan bahwa sistem komunikasi serial I2C antara master dan slave telah berjalan sesuai dengan yang dibutuhkan. Sedangkan untuk waktu yang dibutuhkan, untuk komunikasi antara master dan slave 1 memiliki rata – rata waktu sebesar **1.38 ms** atau sebesar **0.00138 s**. Sedangkan untuk komunikasi antara master dan slave 2 memiliki rata – rata waktu sebesar **1.4 ms** atau sebesar **0.0014 s**. Kemudian untuk komunikasi data master dan slave 3 dibutuhkan rata – rata waktu sebesar **1.4 ms** atau sebesar **0.0014 s**. Sehingga untuk total waktu yang dibutuhkan untuk distribusi data dari master ke slave dapat diaplikasikan persamaan (2) didapatkan

$$\begin{aligned}\sum I2C_TC &= 1.38 + 1.4 + 1.4 \\ &= 4.18 \text{ ms}\end{aligned}$$

Sedangkan untuk waktu transmisi data secara keseluruhan dapat dihitung dengan persamaan (3) yaitu

$$\begin{aligned}I2C_TC &= 2 \times 4.18 \text{ ms} \\ &= 8.36 \text{ ms}\end{aligned}$$

B. Pengujian Komunikasi Data Rosserial Antara Kontroler dan Mini Komputer

Pengujian komunikasi antara mini komputer dan kontroler dilakukan bertujuan mengetahui tingkat akurasi data yang dikirimkan serta kecepatan transmisi data yang dilakukan melalui waktu transmisi yang dibutuhkan. Pengujian dilakukan dengan cara mengirimkan data dari mini komputer ke Arduino master dengan menggunakan protokol komunikasi roserial. Pada komunikasi data ini error data yang ditargetkan adalah 0%, atau seluruh data yang dikirim dan diterima haruslah sama, serta kecepatan komunikasi data yang diperlukan tidak lebih dari 50 ms.

Tabel 7. Hasil Pengujian Akurasi dan kecepatan komunikasi data roserial antar Arduino Master dan mini komputer

No	Data yang dikirim	Data yang diterima	Waktu
1	x : 814	x : 814	4.6 ms
	y : 0.03	y : 0.03	
	z : 50.9	z : 50.9	
2	x : 17.24	x : 17.24	4.9 ms
	y : 3.572	y : 3.572	
	z : 784	z : 784	
3	x : 1.27	x : 1.27	4.8 ms
	y : 0.294	y : 0.294	
	z : 0.002	z : 0.002	
4	x : 307	x : 307	5.0 ms
	y : 217.16	y : 217.16	
	z : 77.29	z : 77.29	
5	x : 37.65	x : 37.65	4.7 ms
	y : 44.12	y : 44.12	
	z : 0.218	z : 0.218	
6	x : 11.79	x : 11.79	4.6 ms
	y : 400	y : 400	
	z : 379	z : 379	
7	x : 18.12	x : 18.12	4.8 ms
	y : 64.76	y : 64.76	
	z : 1040	z : 1040	

Berdasarkan hasil yang didapatkan dari 7 kali ujicoba dengan cara mengirimkan data dari mini komputer ke Arduino master, serta secara

sebaliknya, didapatkan bahwa akurasi data yang diterima memiliki nilai error sebesar 0%, hal ini menunjukkan komunikasi data telah berjalan sesuai dengan kebutuhan sistem. Serta memiliki kecepatan distribusi data dengan rata – rata waktu sebesar **4.7 ms** atau sebesar **0.0047 s**. Sehingga didapatkan nilai *Roserial_TC* sebesar 4.7 ms. Hal ini membuktikan bahwa komunikasi data dengan menggunakan roserial antara mini komputer dan Arduino master sesuai dengan kebutuhan yaitu kurang dari 50 ms.

C. Pengujian Time Sampling Keseluruhan Sistem Robot

Pengujian time sampling keseluruhan sistem dilakukan untuk mengetahui berapa waktu yang dibutuhkan dari sistem robot untuk menjalankan algoritma yang dibutuhkan. Dengan berbekal nilai sampling untuk distribusi data yang dapat dihitung melalui persamaan (4) sehingga didapatkan

$$TC = 8.36 + 4.7 \\ = 13.06 \text{ ms}$$

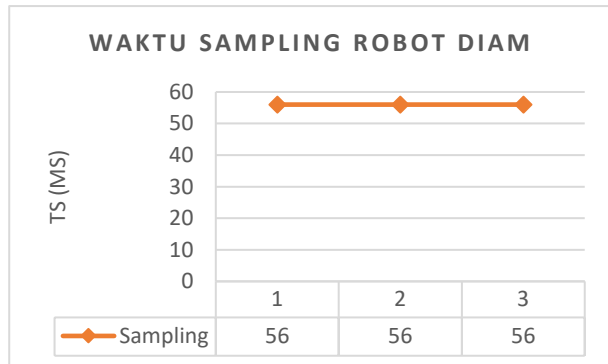
Dengan batasan maksimal dari waktu sampling adalah 80 ms, maka pengujian ini dilakukan untuk mengetahui waktu yang dibutuhkan ketika robot menjalankan algoritma kontrol kinematik hingga output algoritma tersebut didistribusikan ke masing – masing kontroler motor yang terpasang pada sistem. Selain itu dijalankan juga algoritma *sensor fusion* untuk menggabungkan data odometry dengan data sensor kompas.

Tabel 8. Hasil pengujian waktu sampling pada robot dengan seluruh algoritma yang dijalankan

Pengujian	Jarak Pergerakan (meter)	Rata – rata Waktu Sampling
Robot Diam	0	56 ms
	0	56 ms
	0	56 ms
Robot Bergerak	0.5	58 ms
	1	62.3 ms
	2	63.35 ms

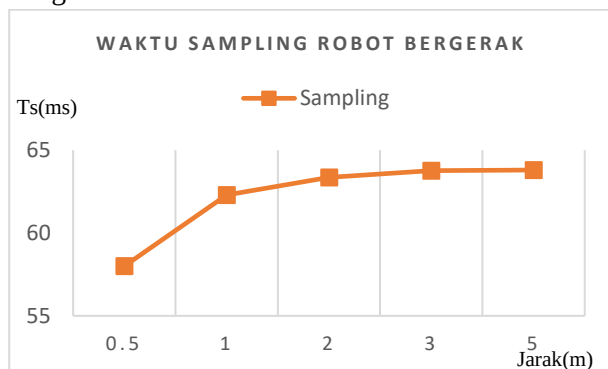
3	63.76 ms
5	63.8 ms

Berdasarkan hasil yang didapatkan, terlihat ketika robot sedang diam tidak bergerak, rata – rata waktu sampling yang dibutuhkan adalah sebesar **56 ms**.



Gambar 4. Grafik waktu sampling ketika robot diam

Ketika robot dalam kondisi bergerak, rata – rata waktu sampling akan bertambah menjadi lambat dengan rata – rata **63.5 ms**.



Gambar 5. Grafik waktu sampling ketika robot bergerak berdasarkan jarak tempuh

Hal ini dapat terjadi dikarenakan ketika robot sedang berjalan data yang didistribusikan akan semakin banyak, baik data yang dikirimkan melalui protokol serial I2C, serta yang dikirimkan melalui roserial. Ditambah lagi algoritma kinematik (*forward velocity* dan *invers velocity*) akan lebih banyak mengupdate data dikarenakan data sensor rotary encoder dan sensor kompas yang selalu mengalami perubahan. Namun dilihat dari hasil pengujian, dimana robot diperintahkan untuk bergerak dengan jarak yang berbeda – beda, rata – rata waktu sampling yang dibutuhkan adalah sebesar **63.5 ms**, hal ini menunjukkan bahwa

dengan mengaplikasikan multi kontroler dengan distribusi data Serial I2C dan roserial, dapat membuat waktu sampling dari robot memenuhi kebutuhan yaitu masih dibawah **80 ms**.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang didapatkan, dapat disimpulkan bahwa

- Sistem multi kontroler yang diaplikasikan pada sistem robot dapat membantu mengurangi beban yang ditanggung satu unit kontroler
- Komunikasi Serial I2C dan roserial dapat diimplementasikan dengan baik, hal ini terbukti bahwa error data yang didapatkan untuk masing – masing protokol komunikasi adalah sebesar 0% (**Akurasi data 100%**).
- Sistem multi kontroler dapat bekerja secara optimal dengan menggunakan distribusi data secara serial I2C dan roserial, terbukti dengan batasan maksimal waktu sampling 100 ms, alokasi waktu yang harus dipenuhi adalah dibawah 80 ms, dengan hasil pengujian yang telah dilakukan terbukti dapat memenuhi kebutuhan tersebut, yaitu membutuhkan waktu sebesar **63.5 ms**.

Saran yang diberikan penulis untuk pengembangan selanjutnya adalah

- Dapat digunakan mini komputer yang memiliki spesifikasi diatas Raspberry Pi 3B+ untuk dapat menjalankan algoritma kinematik dan sensor fushion secara lebih cepat
- Untuk dapat mengurangi jumlah mikrokontroler yang digunakan, dapat digunakan mikrokontroler dengan spesifikasi clock yang lebih tinggi dari 16 MHz, sebagai contoh menggunakan ESP32 atau STM32 yang memiliki clock lebih dari 80 MHz.

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan pada Politeknik Negeri Malang yang telah mendukung proses penelitian yang dilakukan, serta kepada pihak laboratorium tefa teknik elektronika yang telah memberikan tempat untuk melakukan pengembangan dan uji coba pada penelitian yang telah selesai dilaksanakan.

REFERENSI

- [1] J. Zhang, R. Liu, K. Yin, Z. Wang, M. Gui and S. Chen, "Intelligent Collaborative Localization Among Air-Ground Robots for Industrial Environment Perception," *IEEE Transactions on Industrial Electronics*, vol. 66, pp. 9673-9681, 2019.
- [2] X. Bai, L. Dong, L. Ge, H. Xu, J. Zhang and J. Yan, "Robust Localization of Mobile Robot in Industrial Environments With Non-Line-of-Sight Situation," *IEEE Access*, vol. 8, pp. 22537-22545, 2020.
- [3] I. Siradjuddin, I. M. Fitriani, R. A. Asmara, M. Junus, T. S. Patma, G. A. Azhar and H. Setiawan, "A study of a discrete Bayes and a Kalman filter computational Complexity and performance in the case of 1D robot localization," *Journal of Physics: Conference Series*, vol. 1402, p. 044026, 12 2019.
- [4] M. Cui, "Observer-Based Adaptive Tracking Control of Wheeled Mobile Robots With Unknown Slipping Parameters," *IEEE Access*, vol. 7, pp. 169646-169655, 2019.
- [5] T. Shu, S. Gharaaty, W. Xie, A. Joubair and I. A. Bonev, "Dynamic Path Tracking of Industrial Robots With High Accuracy Using Photogrammetry Sensor," *IEEE/ASME Transactions on Mechatronics*, vol. 23, pp. 1159-1170, 2018.
- [6] W. Budiharto, V. Andreas, J. S. Suroso, A. A. S. Gunawan and E. Irwansyah, "Development of Tank-Based Military Robot and Object Tracker," in *2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, 2019.
- [7] H. Dong, C.-Y. Weng, C. Guo, H. Yu and I.-M. Chen, "Real-Time Avoidance Strategy of Dynamic Obstacles via Half Model-Free Detection and Tracking With 2D Lidar for Mobile Robots," *IEEE/ASME Transactions on Mechatronics*, vol. 26, pp. 2215-2225, 2021.
- [8] I. Siradjuddin, G. A. Azhar, C. Rahmad, F. Rohman, I. M. Fitriani, E. R. K. Pradani, E. Rohadi, F. Ronilaya and R. A. Asmara, "A Non-Iterative Solution for Rigid Body Transformation Estimation," in *2018 International Conference on Applied Science and Technology (iCAST)*, 2018.
- [9] I. Siradjuddin, G. A. Azhar, A. Murdani and M. L. M. Faizin, "Desain dan pemodelan kontrol kinematik pergerakan robot beroda dengan menggunakan 6 roda omni-wheels," *Jurnal Eltek*, vol. 18, pp. 116-127, 2020.
- [10] Z. Dan, Z. Jisheng, H. Guohua and Y. Jie, "Navigation stabilization technology of mobile robot based on compass and gyro," in *2009 Chinese Control and Decision Conference*, 2009.
- [11] I. Siradjuddin, T. Winarno, M. Khairuddin, M. N. Achmadiah, R. P. Wicaksono and G. Al Azhar, "An Exponential Decreased Kinematic and PID Low Level Control Schemes for an Omni-Wheeled Mobile Robot," in *2021 International Conference on Electrical and Information Technology (IEIT)*, 2021.
- [12] I. Siradjuddin, G. Azhar, S. Wibowo, F. Ronilaya, C. Rahmad and R. O. H. A. D. I. Erfan, "A General Inverse Kinematic Formulation and Control Schemes for Omnidirectional Robots," *Engineering Letters*, vol. 29, 1 2022.
- [13] I. Siradjuddin, L. Kamajaya, S. Wibowo, A. A. Rofiq, G. A. Azhar and M. Khairuddin, "Near Real Time Simulation of an Independent Steering Independent Driving Mobile Robot," in *2021 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, 2021.
- [14] G. A. Azhar, T. Winarno and A. Komarudin, "Kontrol Sudut Elevasi Robot Pelontar Softsaucer dengan Metode PID," *Jurnal Elektronika Otomasi Industri*, vol. 4, pp. 9-14, 2020.

- [15] M.-A. Chung and C.-W. Lin, "An Improved Localization of Mobile Robotic System Based on AMCL Algorithm," *IEEE Sensors Journal*, vol. 22, pp. 900-908, 2022.
- [16] S. Di Cairano and I. V. Kolmanovsky, "Real-time optimization and model predictive control for aerospace and automotive applications," in *2018 Annual American Control Conference (ACC)*, 2018.
- [17] S. Xu and W. Chou, "An Improved Indoor Localization Method for Mobile Robot Based on WiFi Fingerprint and AMCL," in *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, 2017.
- [18] B. Kim and K. Yi, "Probabilistic and Holistic Prediction of Vehicle States Using Sensor Fusion for Application to Integrated Vehicle Safety Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, pp. 2178-2190, 2014.
- [19] J. L. Mata-Machuca, L. F. Zarazua and R. Aguilar-López, "Experimental Verification of the Leader-Follower Formation Control of Two Wheeled Mobile Robots with Obstacle Avoidance," *IEEE Latin America Transactions*, vol. 19, pp. 1417-1424, 2021.
- [20] G. Al Azhar, T. Winarno and S. Izza, "Implementasi gh Filter Pada Sensor Kompas Sebagai Peningkatan Akurasi Trajectory Tracking Robot Differential Drive," *Metrotech (Journal of Mechanical and Electrical Technology)*, vol. 1, p. 21–30, 2022.