

RANCANG BANGUN DAN IMPLEMENTASI SISTEM *TRACKING* PADA ROBOT TANK MENGGUNAKAN ANDROID

Design and Implementation Tracking System on Tank Robot Using Android

Ali Akhmad Ghifari¹, Angga Rusdinar², Ramdhan Nugraha³

^{1,2,3}Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

¹aliakhmadghifari@gmail.com, ²angga.rusdinar@telkomuniversity.ac.id, ³ramdhan@telkomuniversity.ac.id

Abstrak

Pada jurnal ini dijelaskan perancangan sistem *tracking* menggunakan android pada robot berbasis roda tank / *track wheel*. Sistem *tracking* ini menggunakan sensor *quadrature encoders* dan *magnetometer* sebagai penunjuk arah muka robot. Sistem *tracking* ini akan mengeluarkan jalur pergerakan robot dalam bentuk dua dimensi. Dengan sistem yang telah diuji pada sekala (kotak) 1:1 (meter) dapat disimpulkan pembuatan jalur *tracking* memiliki persentasi kesalahan rata rata pada lintasan lurus sepanjang 2 meter adalah 8.589%. Kesalahan tersebut terjadi karena komunikasi antara *server* ke *client* mengalami permasalahan yaitu data hilang dan *delay* pengiriman / penerimaan data.

Kata Kunci : Robot, *Tracking*, Android, *Encoders*, *Smartphone*

Abstract

In this paper described the design of the tracking system using android on tank robot / track wheel based . These tracking systems use quadrature encoders and magnetometer sensor as a direction to face the robot . This tracking system will issue a movement path in the form of two-dimensional robot . With systems that have been tested on the scale (box) 1 : 1 (meter) can be concluded manufacture tracking path has an average percentage error in a straight line along the 2 meter was 8,589 % . The error occurred because the communication between the server to the client to experience problems that the lost data and delay sending / receiving data.

Keywords : Robot, *Tracking*, Android, *Encoders*, *Smartphone*

1. Pendahuluan

Robot Exploration adalah sebuah robot yang ditugaskan secara khusus untuk melakukan eksplorasi terhadap tempat tempat tertentu yang bertujuan untuk mengambil data lingkungan sekitar dan juga kondisi tertentu yang tengah terjadi di lingkungan tersebut. Salah satu fitur yang sangat dibutuhkan pada *robot exploration* adalah sistem *tracking*, sehingga manusia tidak perlu turun langsung pada saat robot diterjunkan untuk melakukan eksplorasi pada lingkungan tertentu. Ataupun jika robot hilang pada saat eksplorasi, data pergerakan robot nantinya akan dikirimkan ke android untuk diolah menjadi titik / garis yang menyimbolkan hasil dari pengolahan data dari pergerakan robot yang di keluarkan pada android. data dari pengolahan pada android juga dapat digunakan untuk mencari keberadaan robot jika hilang, pada saat melakukan eksplorasi. Untuk mendapatkan data perpindahan / perubahan posisi robot, dibutuhkan dua sensor utama yaitu magnetometer sebagai kompas dan *quadrature encoders* sebagai penunjuk kecepatan motor, yang nantinya data tersebut di konversi menjadi jarak tempuh sementara magnetometer akan diolah menjadi posisi arah muka robot terhadap magnet bumi. kedua data tersebut nantinya akan dijadikan sebagai data acuan dalam pembuatan sistem *tracking* pada android yang stabil.

2. Dasar Teori

2.1 Robot

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dahulu (kecerdasan buatan) [1]. Robot Tank adalah robot yang memiliki ciri khas seperti Tank yang sebenarnya menggunakan aktuator motor dan *track wheel* untuk menggerakkan keseluruhan badan robot, sehingga robot tersebut dapat berjalan atau berpindah posisi dari satu titik ke titik lainnya. Kelebihan dari *track wheel* pada robot tank memungkinkan robot bergerak lebih halus dan dapat melalui berbagai medan.

2.2 Android

Android (sistem operasi) merupakan sebuah sistem operasi berbasis linux untuk telepon seluler seperti *smartphone*. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi sendiri yang nantinya dapat digunakan sesuai dengan kebutuhan penggunanya. Karena berbasis linux juga maka android bersifat *open source* sehingga dapat digunakan dan dikembangkan secara bebas oleh para penggunanya. [2]

2.3 Mikrokontroler ATmega128

Mikrokontroler adalah chip yang digunakan untuk mengontrol sesuatu yang dapat diprogram sesuai dengan kebutuhan penggunanya. Pemilihan mikrokontroler AT-Mega 128 dan 32 karena jenis mikrokontroler tersebut memiliki kelebihan dari jumlah pin dan fitur yang disajikan mikrokontroler dan juga dapat diprogramnya pin *input / output* sesuai keinginan dan kebutuhan. Bahasa pemrograman mikrokontroler yang akan digunakan adalah bahasa C.[3]

Tabel 1 Fitur ATmega128 [3]

Menggunakan arsitektur RISC 8-bit dengan 32KB internal flash memory.
Memiliki frekuensi kerja maksimum 16 MHz
Memiliki 4KBytes internal SRAM
Memiliki 2 USART
Memiliki 53 programmable I/O
Tegangan kerja 4,5V – 5,5V
Memiliki 8 channel 10-bit ADC

2.4 FT232R

FT232R adalah IC yang biasa digunakan pada rangkaian komunikasi serial antara mikrokontroler dengan *personal computer* maupun *smartphone*. IC ini dibuat oleh perusahaan Future Technology Device International Ltd dan memiliki spesifikasi seperti berikut. [4]

2.5 Quadrature Encoder

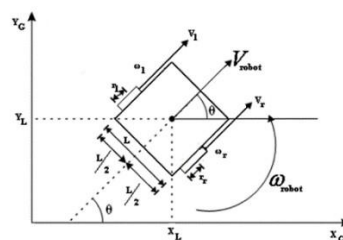
Quadrature Encoders atau yang biasa disebut *incremental rotary encoder* adalah jenis sensor *encoders* yang bekerja dengan cara membaca jumlah pulsa yang terjadi pada saat roda berputar[5]. Data tersebut nantinya dapat dikonversi kedalam kecepatan motor.

2.6 Sensor Magnetometer

Magnetometer adalah sebuah alat yang berfungsi untuk mendeteksi medan magnet di lingkungan sekitarnya, sehingga Magnetometer dapat digunakan untuk mendeteksi arah muka angin layaknya kompas. Sensor ini akan digunakan untuk mendeteksi arah muka robot terhadap arah mata angin berdasarkan medan magnet bumi sebagai acuannya.[6]

2.7 Permodelan Pergerakan [7]

Dalam melakukan implementasi sistem *tracking*, pergerakan dari robot Tank harus dimodelkan terlebih dahulu. Pergerakan robot dapat dipetakan dengan mengetahui kecepatan dan arah gerak robot melalui data kecepatan masing-masing roda. Kecepatan roda (ω) dan arah gerak (θ) robot dapat diketahui melalui rumus berikut:



Gambar 1 Permodelan Gerak Robot [13]

$$v = \frac{v_{kiri} + v_{kanan}}{2} \tag{2.7.1}$$

$$\omega = \frac{v_{kanan} - v_{kiri}}{L} \tag{2.7.2}$$

$$\theta = \theta_0 + \omega t \tag{2.7.3}$$

Keterangan:

v : kecepatan roda (m/s)

ω : Kecepatan sudut robot (rad/s)

θ_0 : Besar Sudut Awal (derajat)

θ : Besar Sudut (derajat)

L : jarak antar roda

v_{kiri} : kecepatan roda kiri

v_{kanan} : kecepatan roda kanan

Setelah diketahui kecepatan dan arah gerak robot, dapat dilakukan pemetaan posisi robot melalui persamaan berikut:

$$x_t = x_{t-1} + v \cos(\theta) \tag{2.7.4}$$

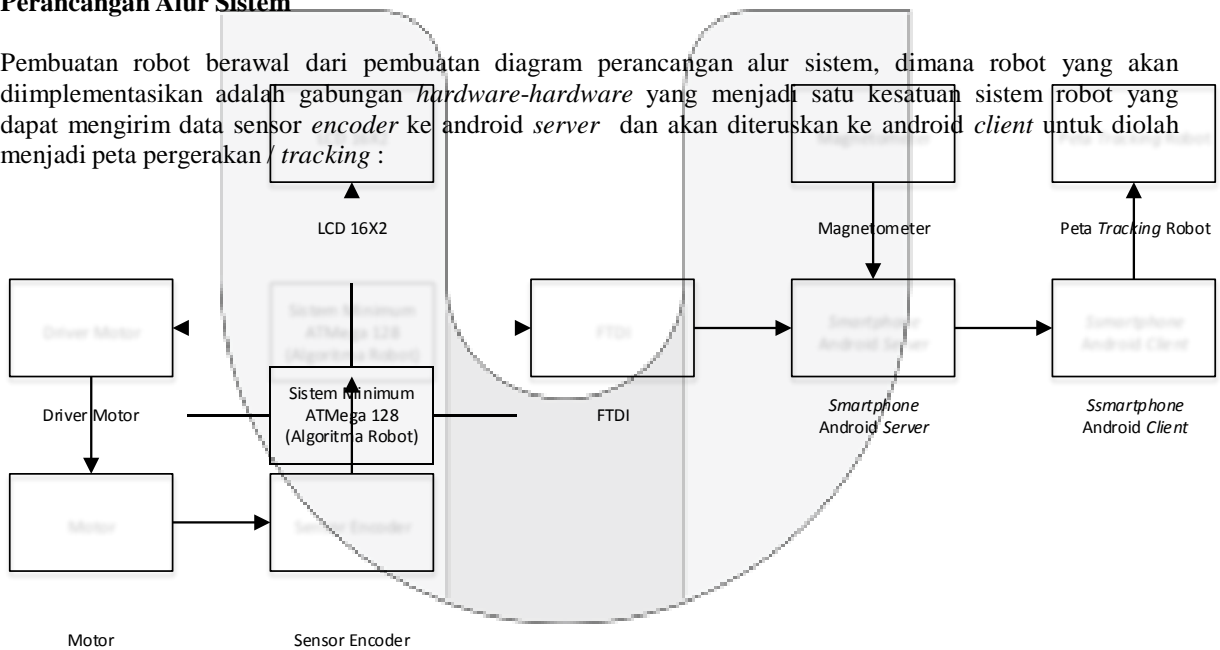
$$y_t = y_{t-1} + v \sin(\theta) \tag{2.7.5}$$

Dengan x dan y merupakan posisi robot pada sumbu X dan Y.

3. Pembahasan

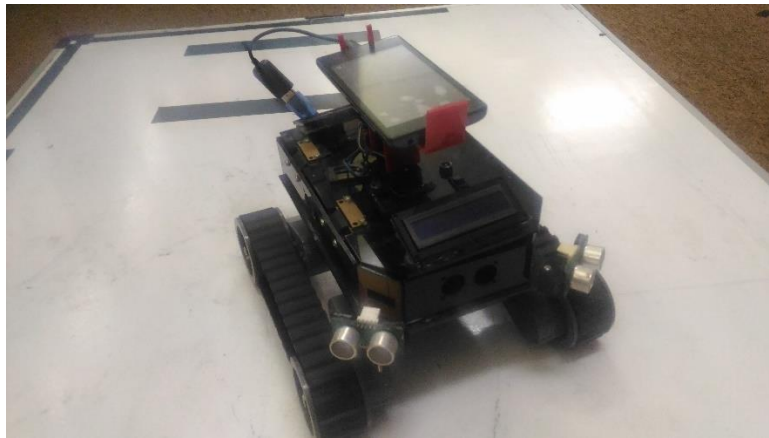
3.1 Perancangan Alur Sistem

Pembuatan robot berawal dari pembuatan diagram perancangan alur sistem, dimana robot yang akan diimplementasikan adalah gabungan *hardware-hardware* yang menjadi satu kesatuan sistem robot yang dapat mengirim data sensor *encoder* ke android *server* dan akan diteruskan ke android *client* untuk diolah menjadi peta pergerakan / *tracking* :



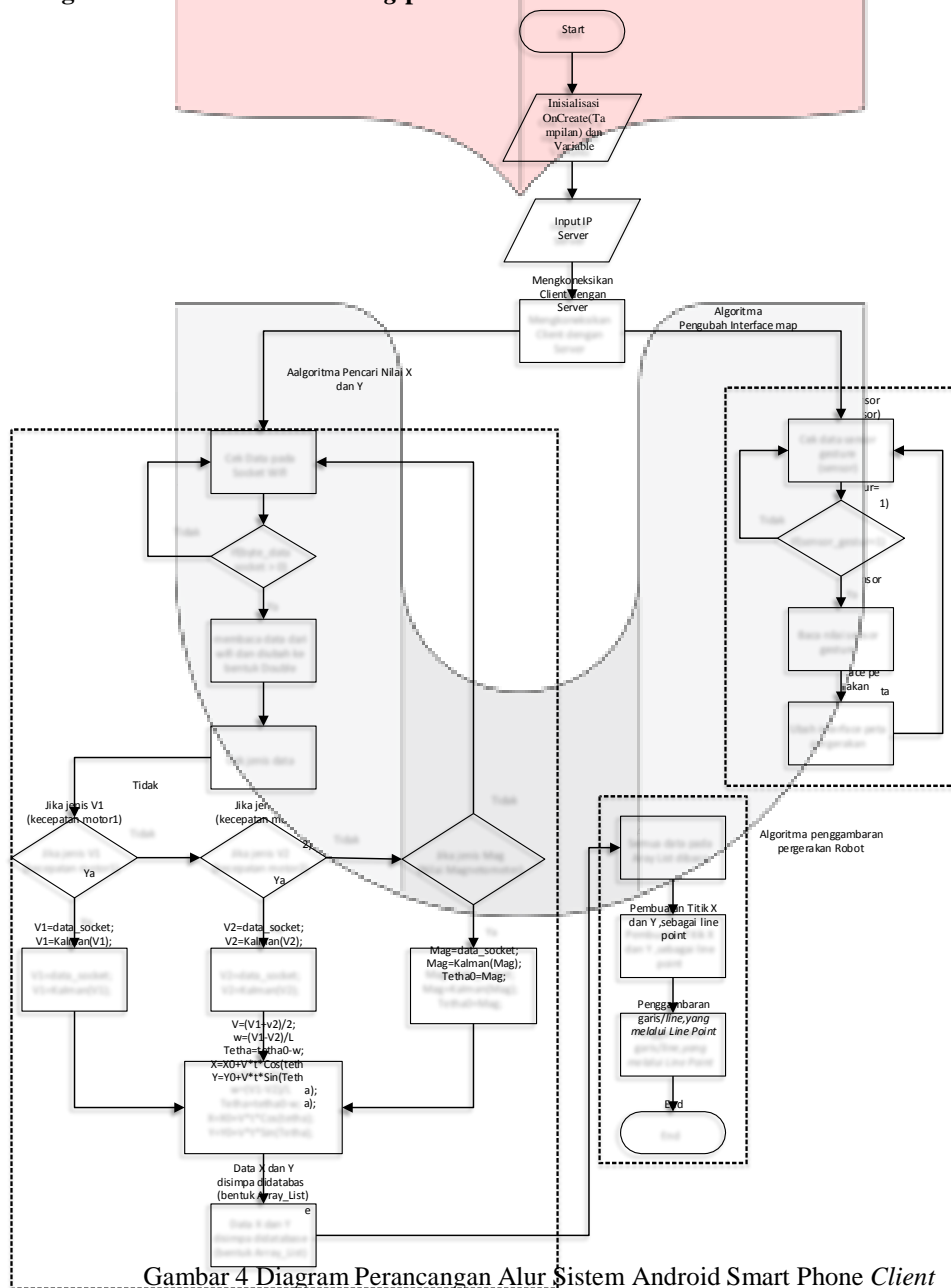
Gambar 2 Diagram Perancangan Alur Sistem

Ketika Robot mulai bergerak Sistem minimum yang ada pada robot akan mengirimkan data berupa kecepatan motor, data tersebut didapat dari nilai sensor *encoder* yang telah diolah menjadi kecepatan. Setelah itu data dikirim melalui protokol serial FTDI ke smartphone



Gambar 3 Prototype Robot Tank

3.2 Perancangan Software Sistem Tracking pada Android



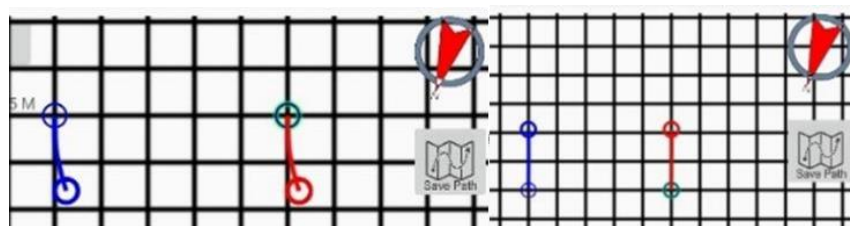
Gambar 4 Diagram Perancangan Alur Sistem Android Smart Phone Client

Perancangan *Software* Android terdiri dari tiga bagian program yang telah ada dan dimodifikasi untuk digabungkan menjadi satu program yang dibutuhkan dalam sistem *tracking* robot dalam tugas akhir ini antara lain : gambar garis dan pengolahan paket data pergerakan robot pada *smartphone* Android *client*, pengiriman data socket *client server* antar *smartphone* Android dan pengiriman data serial dari ATmega 128 robot ke android *server* menggunakan FTDI. Penggabungan program menggunakan Android studio yang terdiri dari bagian. java (program algoritma) dan .xml sebagai tampilan program. Berikut flow chart sederhana algoritma program pada *smartphone client* dan *server*.

4. Pengujian

• Cara Pengujian :

Pengujian ini dilakukan dengan cara membandingkan kesesuaian skala jarak antara Peta pergerakan robot pada *smartphone* Android *client* dengan pergerakan robot. Kesesuaian tersebut dilakukan dengan pengujian untuk jalur. Pengujian dilakukan beberapa kali untuk melihat akurasi kesesuaian jarak sesuai skala pergerakan robot dan pengukuran jarak pergerakan robot diukur secara manual menggunakan meteran. Penyesuaian skala jarak pergerakan antara peta pergerakan pada *smartphone* Android *client* dengan jalan robot menggunakan kecepatan yang sudah ditentukan saat melewati jalur yang dibuat. Sekala pada peta *smartphone*. Untuk sekala yang digunakan tiap satu blok / kotak pada android *client* mewakili pergerakan robot satu meter. Pada saat pengujian robot akan bergerak lurus sejauh satu meter dan nantinya hasilnya akan dibandingkan dengan panjang garis yang dibuat pada android *client*. Pengujian dilakukan dengan dua metode, metode dengan menggunakan magnetometer dan tidak menggunakan magnetometer.



Gambar 5 Aplikasi *Tracking* Pada *Smartphone Client* (Kiri Menggunakan *Magnetometer*, kanan Tidak Menggunakan *Magnetometer*).

• Hasil Pengujian :

Pada pengujian ini dibandingkan jarak sesungguhnya yang dilalui robot dengan skala jarak jalur yang dibuat aplikasi *tracking* pada *smartphone client*. Pengujian pertama dilakukan dengan menggunakan nilai magnetometer sebagai penentu derajat arah muka robot. Pengujian komunikasi keseluruhan sistem pun dilakukan untuk mengetahui apakah komunikasi dapat berjalan dengan baik atau tidak.

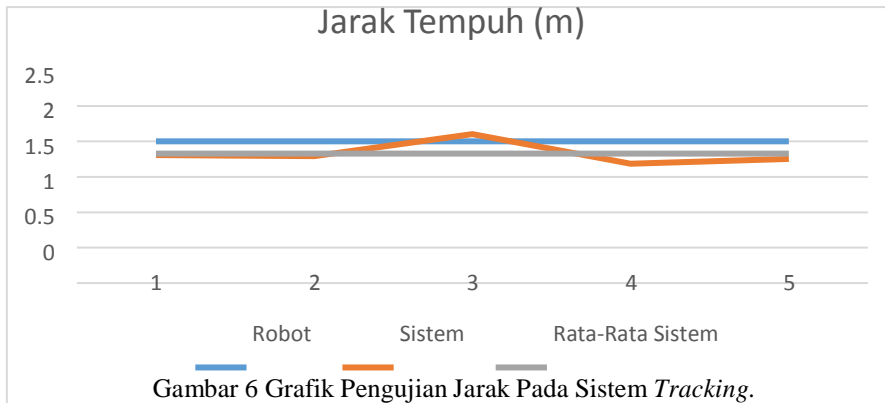
Tabel 2 Hasil Pengujian Komunikasi Data Keseluruhan Sistem.

no	metode pengiriman <i>server</i> ke <i>client</i>	Waktu perbedaan penerima dan pengirim (s)	waktu rata rata		jumlah data	
			pengiriman <i>server</i> (ms)	penerimaan <i>client</i> (ms)	hilang di penerima	tidak terkirim
1	tergantung input data FTDI	6.961	55.667	85.667	3	0
2	10ms	8.839	10.333	0.5	5	0
3	100 ms	8.758	100.571	206.333	6	2

Pada tabel 2 diperlihatkan pengujian komunikasi antara *server* dan *client*, dapat disimpulkan bahwa dengan metode pengiriman yang telah dicoba tetap mengalami data hilang, kejadian tersebut mengakibatkan sistem menjadi kurang responsif terhadap perubahan kecepatan pergerakan robot sehingga terjadi kesalahan pembacaan jarak ataupun penggambaran jalur *tracking*.

Tabel 3 Hasil pengujian jarak 2 m Dengan Pergerakan Lurus.

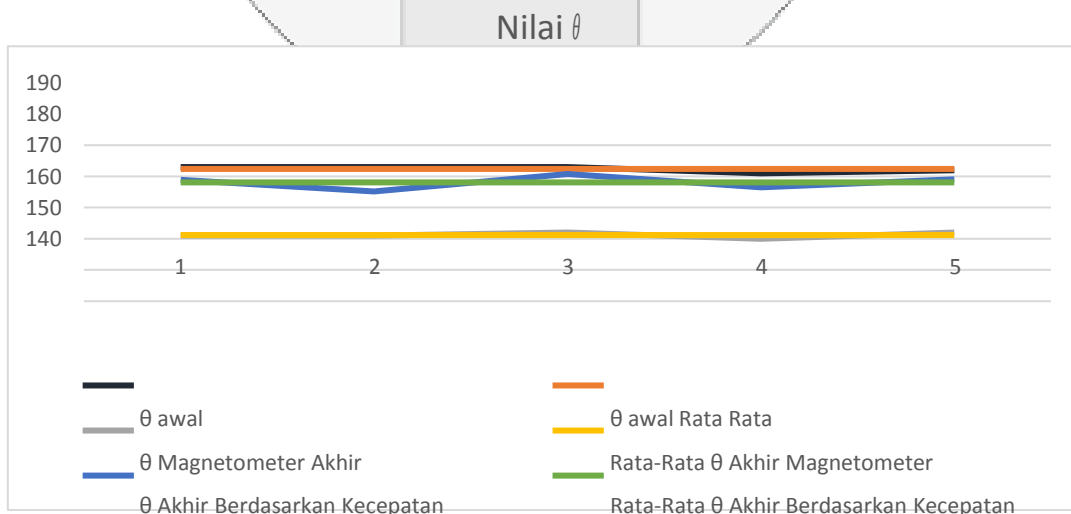
No.	Jarak Tempuh Robot (m)	Jarak Tempuh sistem (m)	Persentase Kesalahan Jarak (%)
1	2	1.806	9.680
2	2	1.795	10.230
3	2	2.102	-5.095
4	2	1.684	15.780
5	2	1.753	12.350
Rata-Rata	2	1.828	8.589



Dari tabel 3 nilai akhir pembacaan jarak pada sistem kurang begitu sesuai dengan jarak tempuh yang dilalui robot menyebabkan penggambaran pada sistem *tracking* menjadi kurang sesuai dengan pergerakan robot.

Tabel 4 Hasil Pengujian Nilai θ Pada Sistem *Tracking* Pada Saat Gerak Lurus Sejauh 2 m.

No.	Pembacaan <i>Magnetometer</i> ($^{\circ}$)			pembacaan tetha melalui kecepatan ($^{\circ}$)		
	Awal	Akhir	Delta	Awal	akhir	Delta
1	183.0	161.0	-22.0	183.0	178.855	-4.145
2	183.0	161.0	-22.0	183.0	175.171	-7.829
3	183.0	162.0	-21.0	183.0	180.763	-2.237
4	181.0	160.0	-21.0	181.0	176.461	-4.539
5	182.0	162.0	-20.0	182.0	179.106	-2.894
Rata-rata	182.4	161.2	-21.2	182.4	178.071	-4.329



Gambar 7 Grafik Pengujian Nilai θ Sistem *Tracking*.

Pada tabel 4 dapat disimpulkan bahwa nilai θ yang berdasarkan nilai kecepatan memiliki kesesuaian yang cukup jauh dibandingkan nilai dari *magnetometer*. Maka dari itu penggunaan *magnetometer* pada sistem *tracking* cukup baik untuk mengatasi ketidak sesuaian pembuatan jalur *tracking* terhadap pergerakan robot sesungguhnya.

5. Kesimpulan

- Pembuatan jalur dengan metode bantuan magnetometer memiliki hasil penggambaran yang lebih realistis, karena kemiringan garis yang dibuat berdasarkan kemiringan robot sesungguhnya melalui pembacaan magnetometer.
- Rata-rata kesalahan pembacaan jarak untuk jalur lurus sepanjang 2 meter sebesar 8.589% hal tersebut terjadi karena permasalahan pada pertukaran data seperti pada tabel 2, terjadi data hilang dan waktu penerimaan dan pengiriman yang cukup lama menyebabkan sistem menjadi kurang responsif terhadap perubahan yang terjadi pada robot. Maka dari itu sistem kurang cepat mendapatkan *update* data dari robot.
- Pembacaan arah muka robot menggunakan data kecepatan dari encoders saja kurang cukup bisa dilihat pada tabel 4. Perbedaan pembacaan derajat arah muka robot antara *magnetometer* dan berdasarkan kecepatan terpaut cukup jauh rata-rata pembacaan magnetometer menunjukkan perubahan arahmuka robot sebesar -21.2^0 sementara dengan menggunakan encoders sebesar -4.239^0 .

Daftar Pustaka :

- [1] T. D. S. Suyhadi, Buku Pintar Robotika, Yogyakarta: Andi, 2010.
- [2] N. S. H, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, Bandung: Informatika, 2011.
- [3] Atmel Corporation, "ATmega128/L Datasheet," San Jose, 2011.
- [4] Future Technology Devices International Limited, "FT232R - FTDI," Glasgow, 2015.
- [5] DFRobot, "Rover 5 Introduction," [Online]. Available: <http://www.dfrobot.com/image/data/ROB0055/Rover%20%20Introduction.pdf>. [Accessed 15 June 2016].
- [6] Inovative Electronic Datasheet. (n.d.). *DT-Sense Accelero Magnetometer*.
- [7] Rusdinar, A., & Kim, S. (2012). Model of Vision Based Robot Formation Control Using Fuzzy Logic Controller and Extended Kalman Filter. *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 12, no 3, Sept 2012, pp. 238-244.
- [8] Bräunl, T. (2006). *Embedded Robotics : Mobile Robot Design and Applications with Embedded Systems Second Edition*. Germany: Springer.
- [9] santoso, h. d. (2014). *KONTROL TAKE-OFF LANDING dan HOVERING AUTOMATIS UNTUK QUADCOPTER MENGGUNAKAN STM32 dan KALMAN-FILTER*. Bandung: Telkom University.
- [10] "Canvas," [Online]. Available: <https://developer.android.com/reference/android/graphics/Canvas.html>. [Accessed 1 Juni 2016].
- [11] D. Singh, P. Arora, R. and T. Saket, "Study of Data Transmission Using Sockets," *International Journal of Computer Science and Mobile Applications*, vol. 2, no. 11, pp. 151-154, 2014.
- [12] Tri Nugroho, Chandra. *Perancangan dan Implementasi Robot Tank Yang Berjalan Dengan Jalur Virtual Dari Smartphone Android*. Bandung: Telkom University, 2016.
- [13] STMicroelectronics, "L298," Italy, 2000.