

# Sistem Navigasi pada Throwing Robot ABU Robocon 2021 dengan Integrasi Wheel Odometry dan Lidar Scan Matching Menggunakan Iterative Closest Point

Furqan Aliyuddien<sup>1</sup>, Chastine Fatichah<sup>1</sup>, dan Djoko Purwanto<sup>2</sup>

<sup>1</sup>Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS)

<sup>2</sup>Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS)

*e-mail:* chastine@if.its.ac.id

**Abstrak**—Sistem navigasi hingga saat ini menjadi fokus utama dalam pengembangan robot otonom. Mengetahui lokasi di mana robot sedang berada merupakan syarat agar robot dapat menjalankan tugasnya. Robot otonom darat, pada umumnya digunakan untuk transportasi, survei lokasi yang berbahaya, atau pekerjaan berat yang repetitif. Sistem navigasi dengan metode dead reckoning seperti Inertial Navigation System (INS) memiliki masalah dengan akumulasi error. Untuk mengatasi masalah tersebut, penggunaan kamera atau Lidar sebagai referensi eksternal sangat cocok untuk diaplikasikan di lingkungan indoor. Penelitian ini bertujuan mengkombinasikan Wheel Odometry dengan Lidar Scan Matching menggunakan Iterative Closest Point untuk navigasi robot yang akan digunakan untuk ABU Robocon 2021, kontes robot antar perguruan tinggi se-Asia Pasifik. Iterative Closest Point merupakan salah satu metode Scan Matching pada Point Cloud Lidar yang sederhana namun cukup akurat. Integrasi Wheel Odometry dan Scan Matching dilakukan dengan menggunakan Extended Kalman Filter. Data mentah posisi dari Wheel Odometry akan digunakan sebagai dasar transformasi Point Cloud Lidar, sehingga dapat memangkas jumlah iterasi ICP yang diperlukan. Hasil ICP antara data Lidar dengan peta referensi akan digunakan sebagai input posisi pengukuran sensor pada EKF. Data mentah posisi Wheel Odometry juga akan digunakan sebagai input posisi EKF pada waktu dimana data Lidar tidak didapatkan. Data estimasi posisi dari EKF menjadi output dari sistem pada waktu  $t$  dan juga akan digunakan sebagai input prediksi pada waktu  $t+1$ . Dari hasil uji coba yang dilakukan, integrasi sistem navigasi dengan Scan Matching menggunakan Lidar dapat mengatasi masalah akumulasi error yang menjadi masalah utama Inertial Navigation System. Setelah digunakan selama sepuluh menit, error sudut pada gyro dapat mencapai  $3^\circ$ , sedangkan sistem navigasi yang dibangun dapat dengan konsisten memiliki error sudut dibawah  $0.7^\circ$ .

**Kata Kunci**—Extended Kalman Filter, Iterative Closest Point, Lidar, Mobile Robot.

## I. PENDAHULUAN

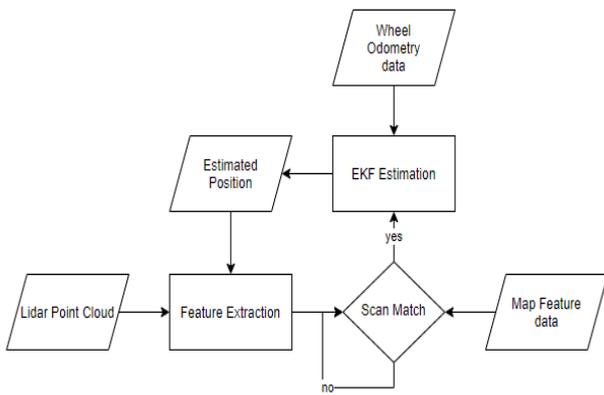
PERKEMBANGAN teknologi informasi mempengaruhi hampir segala aspek kehidupan. Termasuk dalam bidang transportasi, kendaraan tak berawak banyak dikembangkan. Robot otonom dikembangkan untuk berbagai kebutuhan, baik sebagai alat transportasi, survei lokasi yang berbahaya, ataupun melakukan pekerjaan berat yang repetitif. Pengembangan robot otonom berfokus pada perkembangan sistem navigasi, karena robot harus dapat menentukan posisi dirinya secara cepat dan akurat agar dapat menyelesaikan tugasnya secara otonom.

Sistem navigasi robot pada awal perkembangannya hanya menggunakan sensor-sensor yang melakukan perhitungan relatif terhadap kondisi atau posisi awal, biasa disebut dengan *Inertial Navigation System* (INS). Sensor yang biasa digunakan untuk INS yaitu *Inertial Measurement Unit* (IMU) dan *Rotary Encoder*. Kekurangan dari sistem INS yaitu akumulasi *error* dari perhitungan akan semakin besar seiring dengan lama waktu penggunaannya. Hal tersebut terjadi dikarenakan tidak adanya faktor eksternal yang dapat memperbaiki *error* tersebut.

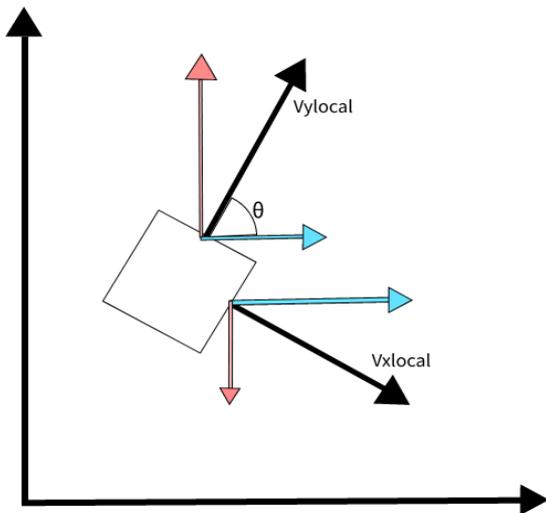
Untuk menangani masalah tersebut, dikembangkan sistem navigasi yang mengkombinasikan INS dengan sensor-sensor eksternal, terutama *Global Positioning System* (GPS). Namun penggunaan GPS memiliki masalah tersendiri, yaitu sulitnya mendapat data yang akurat untuk kondisi tertentu, seperti wilayah yang dikelilingi gedung-gedung tinggi, ataupun kondisi dalam ruangan. Selain menggunakan GPS, dikembangkan pula penggunaan kamera dan juga Lidar untuk mendeteksi *landmark* posisi tertentu. Penggunaan kamera untuk navigasi, biasa disebut dengan *Visual Odometry* sudah lebih berkembang dibandingkan dengan penggunaan Lidar, dikarenakan harga kamera yang relatif lebih murah dibandingkan dengan Lidar. Namun karena belakangan ini Lidar dengan harga terjangkau mulai dikembangkan dan diproduksi, pengembangan sistem navigasi dengan Lidar pun terus berkembang.

Karena kemampuan Lidar untuk memperoleh data yang luas, Lidar juga biasa digunakan untuk melakukan pemetaan. Hubungan antara pemetaan dan lokalisasi dengan menggunakan Lidar tidak bisa dipisahkan, pada pengembangannya sistem yang melakukan pemetaan bersamaan dengan lokalisasi biasa disebut dengan *Simultaneous Localization and Mapping* (SLAM). Adapula SLAM yang menggunakan kamera dikenal dengan *Visual SLAM* (V-SLAM). Sistem SLAM dapat melakukan pemetaan sekaligus menentukan posisi robot terhadap peta yang telah dibangun. Sudah banyak pengembangan algoritma Lidar SLAM dengan berbagai metode yang dikembangkan dan dipublikasikan, seperti EKF-SLAM [1-2], UKF-SLAM [3-4], LOAM [5], dan FastSLAM [6]. Algoritma SLAM secara garis besar terdiri dari 3 tahap proses, yaitu *map building*, *position estimation*, dan *loop closure*.

*Map building* dilakukan dengan mendaftarkan fitur-fitur yang didapatkan sensor ke dalam suatu array atau list, berdasarkan pada estimasi posisi yang didapatkan pada tahap



Gambar 1. Diagram alir sistem.



Gambar 2. Ilustrasi state update pada model.

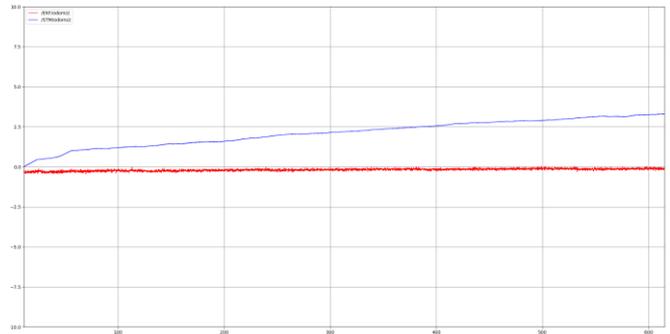
position estimation. Loop closure adalah proses mengenali lokasi yang sebelumnya telah dilalui, sehingga penyesuaian dan koreksi pada *map* dapat dilakukan. Pada kondisi di mana peta lingkungan sudah diketahui, navigasi dapat dilakukan dengan mengabaikan proses *map building* dan *loop closure* untuk mengurangi kompleksitas sistem. Pada tahap *position estimation*, beberapa algoritma yang biasa digunakan yaitu *Iterative Closest Point (ICP) scan matching Algorithm*, *pose-graph estimation*, dan lain sebagainya. ICP merupakan metode yang sederhana, namun cukup akurat.

Berdasarkan penelitian yang sudah ada, penelitian ini mengkombinasikan antara Wheel Odometry dengan algoritma Lidar ICP untuk navigasi robot yang akan digunakan untuk ABU Robocon 2021. ABU Robocon sendiri merupakan kontes Robot tahunan tingkat Perguruan Tinggi se-Asia Pasifik, yang diselenggarakan oleh *Asia-Pacific Broadcasting Union (ABU)*.

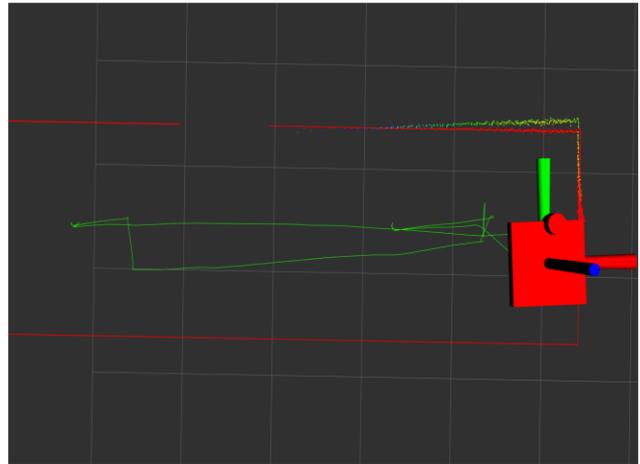
## II. PERANCANGAN SISTEM

### A. Rancangan Umum Sistem

Secara umum sistem navigasi yang dibangun pada penelitian ini dapat digambarkan dengan diagram alir pada Gambar 1. Terdapat tiga masukan pada sistem berupa nilai-nilai dari sensor yaitu *point cloud* Lidar, data *wheel odometry*, dan data *point cloud* referensi lapangan. Luaran sistem berupa estimasi posisi. Sistem yang dibangun merupakan *closed-*



Gambar 3. Grafik perubahan sudut pada posisi diam. garis biru merupakan sudut gyro dan garis merah menunjukkan sudut pada sistem dengan EKF.



Gambar 4. Visualisasi path dan posisi akhir robot pada pengujian sistem dengan gerak manual.

*loop control system*, di mana luaran tahap  $t$  akan digunakan sebagai input tahap  $t+1$ , dan seterusnya. Luaran estimasi posisi akan digunakan sebagai input odometry pada tahap selanjutnya.

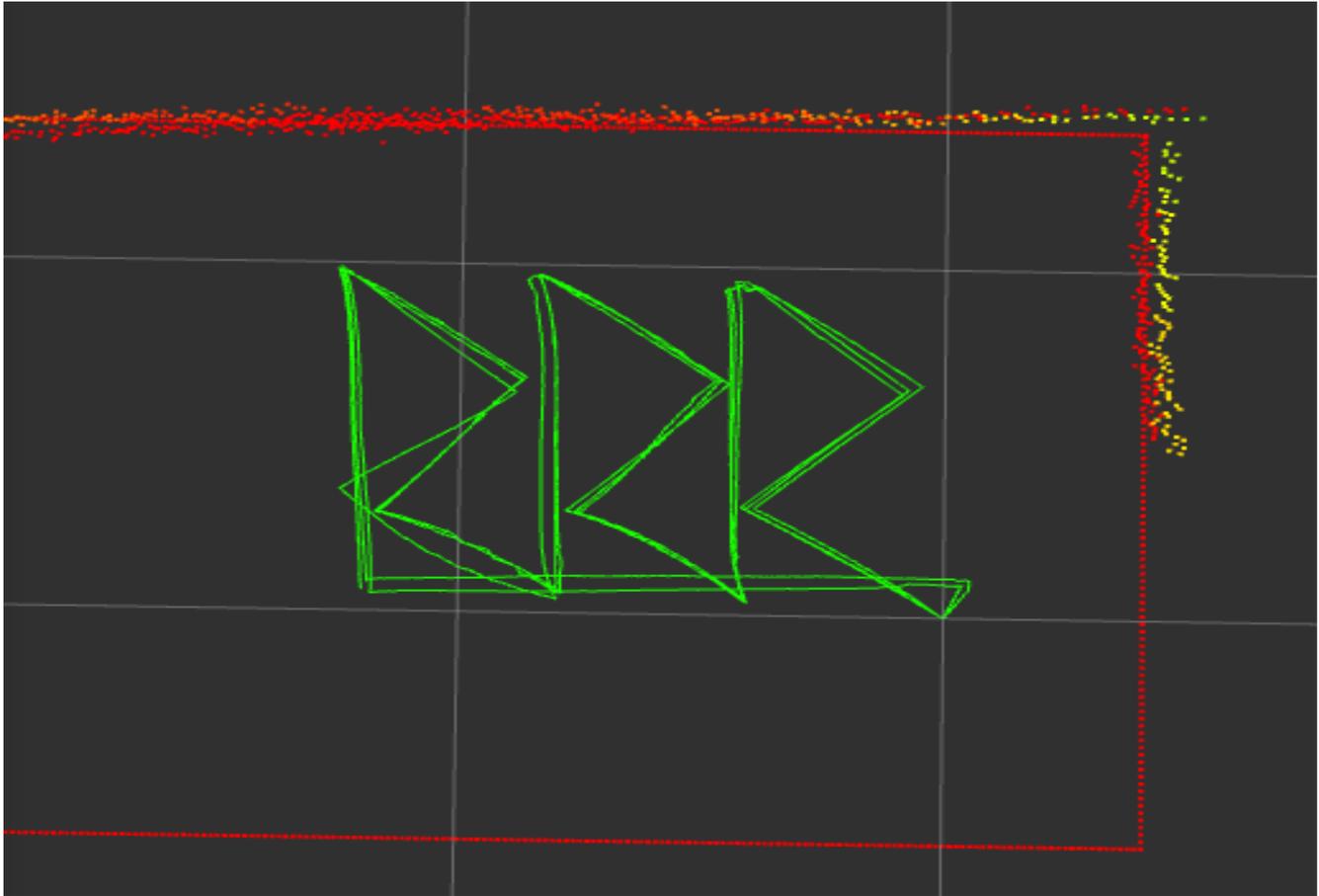
Secara garis besar, diagram alir pada Gambar 1 dapat dijelaskan sebagai berikut: *Lidar Point Cloud*: *Lidar Point Cloud* merupakan input sistem yang merupakan data dari sensor Lidar. Data ini bertipe data *point cloud* dan diperoleh secara *realtime* dan digunakan pada proses *Scan Matching* setelah dilakukan ekstraksi fitur.

*Map Feature Data*: Data ini merupakan data input sistem yang bersifat statis, bertipe data *point cloud*. Data ini di-generate sesuai dengan ukuran lapangan pada lomba ABUROBOCON 2021. Data ini digunakan pada proses *Scan Matching*.

*Odometry Data*: Merupakan data Posisi robot hasil perhitungan dari *wheel odometry*, bertipe *pose2D*, atau posisi titik pada koordinat kartesian dua dimensi ditambah dengan sudut. Data *odometry* ini digunakan untuk estimasi posisi robot pada tahap *predict kalman filter* saat data Lidar tidak tersedia.

*Estimated Position*: Merupakan keluaran sistem pada saat  $t$ , dan menjadi input untuk sistem saat  $t+1$ . Bertipe data *pose2D*, dan merupakan hasil dari tahap *update kalman filter*. Data ini digunakan sebagai input pada tahap *feature extraction*, digunakan untuk mentransformasi input data Lidar ke *frame* posisi robot.

*Feature Extraction*: Pada tahap ini data *point cloud* Lidar akan diambil fitur yang bersesuaian dengan data peta yang tersedia. Pada tahap ini *point cloud* juga di-transform dari *frame* robot ke *frame* global, berdasarkan posisi robot pada hasil estimasi EKF. Posisi estimasi ini bernilai (0,0,0) pada



Gambar 5. Visualisasi jalur kontrol posisi dengan referensi kontrol menggunakan *wheel odometry*.

awal sistem dijalankan, atau robot berada pada posisi awal yang sudah ditentukan.

*Scan Matching*: Tahap ini membandingkan fitur hasil ekstraksi dari tahap sebelumnya dengan data peta yang tersedia. Perbandingan dilakukan dengan menggunakan ICP. ICP menghasilkan matriks transformasi untuk data *point cloud* yang didapat dari sensor lidar agar cocok dengan data *point cloud* referensi lapangan. *Matching ICP* dilakukan dengan melakukan transformasi kecil secara iteratif hingga mencapai *threshold* tertentu, atau mencapai batas iterasi. Pada kasus *threshold* tidak tercapai, data *scan* akan dianggap sebagai *noise* dan transformasi akan diabaikan. Matriks transformasi ini mentransformasi posisi dari *odometry* untuk mendapatkan pengukuran sensor yang akan digunakan pada tahap *update kalman filter*.

*EKF Estimation*: Pada tahap ini dilakukan prediksi posisi robot berdasarkan data dari scan matching dan *wheel odometry*. Tahap ini mengimplementasikan dua tahap *Kalman Filter*, yaitu *prediction* dan *update*. Dalam *estimator EKF* ini, digunakan dua model, yaitu *state space model* dan *observation model*. Kedua model tersebut dapat dijabarkan sebagai berikut.

1) *State Space Model*

Robot  $X$  bergerak dengan sinyal kontrol  $u$  dan memiliki perturbasi atau *error*  $n$ , dan memperbaiki *state* secara berkala dinotasikan dengan

$$X = f(R, u, n) \tag{1}$$

Karena robot bergerak di bidang datar, *state* didefinisikan dengan posisi dan orientasi sebagai

$$X = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{2}$$

Karena Robot bergerak secara *omni-directional*, sinyal kontrol  $u$  didefinisikan dengan vektor kecepatan posisi dan orientasi sebagai

$$u = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{3}$$

Berdasarkan ilustrasi pada Gambar 2 *state* robot  $X$  akan diperbarui pada tiap *step* dengan persamaan

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_{yt-1} \cos \theta_{t-1} \delta t + v_{xt-1} \sin \theta_{t-1} \delta t \\ y_{t-1} + v_{yt-1} \sin \theta_{t-1} \delta t + v_{xt-1} \cos \theta_{t-1} \delta t \\ \theta_{t-1} + \omega_{t-1} \delta t \end{bmatrix} \tag{4}$$

Karena *Kalman filter* membutuhkan *state space model* dalam bentuk persamaan linear, maka persamaan harus diubah ke dalam bentuk sebagai berikut.

$$X_t = A_{t-1}X_{t-1} + B_{t-1}u_{t-1} \tag{5}$$

$A$  merupakan matriks perubahan *state* dalam keadaan tanpa kontrol dan  $B$  merupakan matriks perubahan *state* pengaruh dari kontrol vektor  $u$ . Karena sistem merupakan robot beroda yang posisinya tetap jika tidak mendapat kontrol,  $A$  merupakan matriks identitas. Sedangkan  $B$  bisa didapatkan dengan menghitung bentuk *jacobian* dari persamaan 4 terhadap vektor kontrol  $u$ . Maka didapatkan *state space model* dalam bentuk persamaan linear sebagai berikut.

Tabel 1.  
Error sudut pada kondisi diam

Menit ke- <i>n</i>	Error Sudut Gyro	Error Sudut EKF
1	0.81 °	-0.32 °
2	0.98 °	-0.36 °
3	1.51 °	-0.34 °
4	1.73 °	-0.35 °
5	1.99 °	-0.26 °
6	2.16 °	-0.20 °
7	2.55 °	-0.27 °
8	2.81 °	-0.25 °
9	3.02 °	-0.21 °
10	3.21 °	-0.17 °

Tabel 2.  
Error sudut pada kondisi diam

Sistem	Kecepatan	Komponen	Rata-rata error
Wheel Odometry	5	X	13.58 mm
		Y	13.49 mm
		θ	0.96 °
	10	X	28.33 mm
		Y	38.05 mm
		θ	0.91 °
Sistem ICP-EKF	5	X	14.33 mm
		Y	16.86 mm
		θ	0.87 °
	10	X	39.83 mm
		Y	40.00 mm
		θ	1.54 °

Tabel 3.  
Error sudut pada kondisi diam

No.	Waktu	Komponen	W. Odometry	ICP-EKF
1	30 detik	X	26.94mm	26.88mm
		Y	132.73mm	132.51mm
		θ	2.81°	0.28°
2	1 menit	X	44.76mm	44.67mm
		Y	42.81mm	42.74mm
		θ	0.65°	-0.16°
3	2 menit	X	71.58mm	71.36mm
		Y	504.24mm	502.66mm
		θ	9.50°	0.69°
4	3 menit	X	80.46mm	80.33mm
		Y	11.46mm	11.40mm
		θ	2.80°	-0.41°

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \sin\theta\delta t & \cos\theta\delta t & 0 \\ -\cos\theta\delta t & \sin\theta\delta t & 0 \\ 0 & 0 & \delta t \end{bmatrix} \begin{bmatrix} v_{xt-1} \\ v_{yt-1} \\ \omega_{t-1} \end{bmatrix} \quad (6)$$

2) Observation Model

Observation Model merupakan representasi pengukuran sensor yang didapatkan sistem dalam format sebagai berikut

$$\begin{bmatrix} y_1^t \\ \vdots \\ y_n^t \end{bmatrix} = H_{n \times 3}^t \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_n \end{bmatrix} \quad (7)$$

$y_n^t$  merupakan estimasi pengukuran sensor ke-*n*,  $H_{n \times 3}^t$  merupakan matriks yang mentransformasi pengukuran sensor ke bentuk *state model*, dan  $\omega_n$  merupakan varian atau toleransi *error* sensor ke-*n*. Karena observasi fitur lapangan dengan sensor Lidar menggunakan ICP berupa matriks transformasi posisi dan berhubungan langsung dengan *state vector X*, *H* merupakan matriks identitas dan *observation model* dapat direpresentasikan sebagai berikut.

$$\begin{bmatrix} y_1^t \\ y_2^t \\ y_3^t \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (8)$$

3) Tahap Predict

Pada tahap ini dilakukan prediksi *state vector X* dengan input *control vector u* dengan persamaan 6. Selain itu, pada tahap ini juga akan dihitung matriks kovarian *P* dengan persamaan 9. *Q* menunjukkan seberapa dipercaya hasil prediksi posisi yang akan digunakan.

$$P_t = AP_{t-1}A^T + Q \quad (9)$$

4) Tahap Update

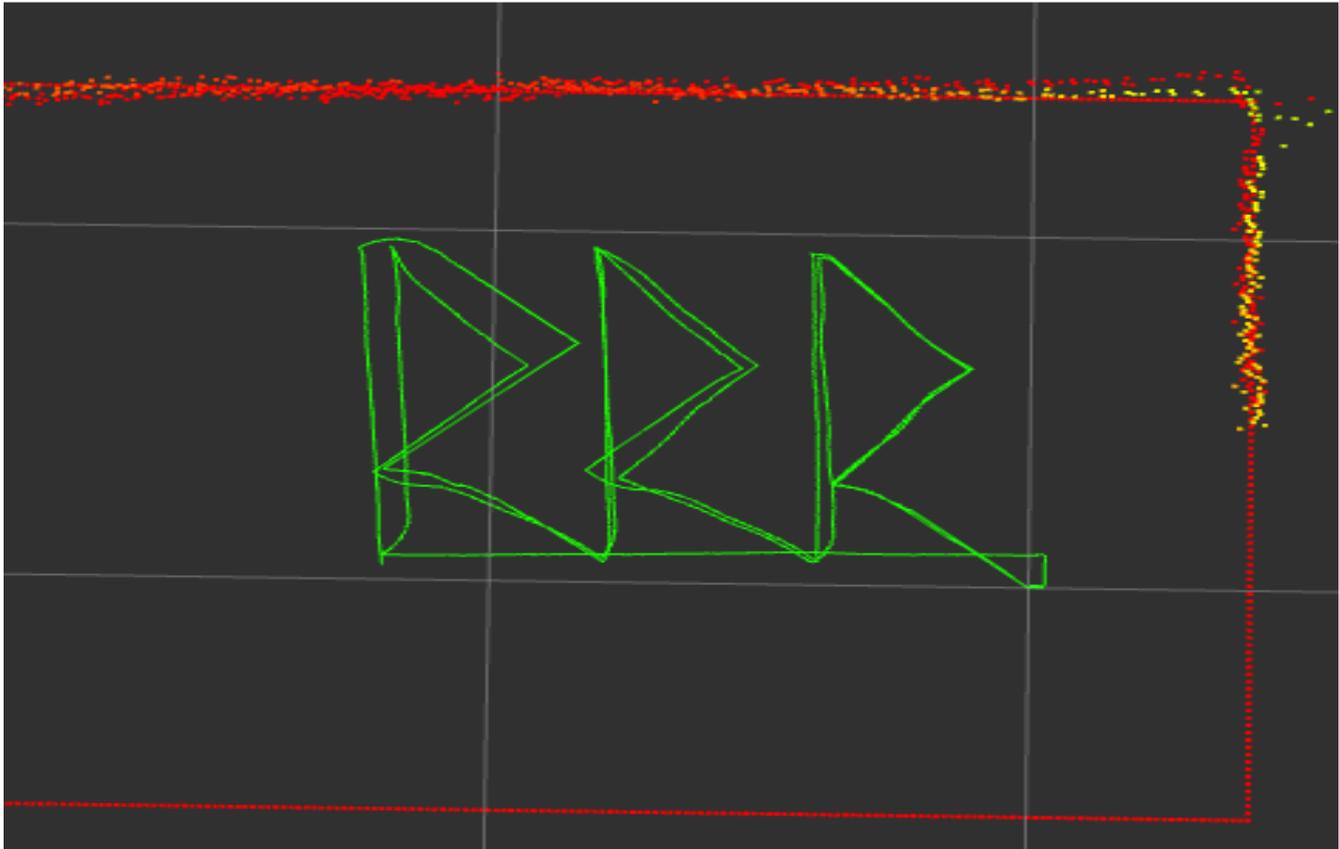
Pada Tahap ini dihitung prediksi observasi menggunakan model observasi 8. Kemudian dihitung residual pengukuran  $\hat{Y}$  dengan mengurangi nilai prediksi observasi dengan nilai observasi yang didapat dari sensor. *Kalman Gain* dihitung dengan persamaan 10, lalu digunakan untuk mengupdate nilai *state vector X* dengan persamaan 11 dengan *S* merupakan matriks kovarian residual.

$$K = P_t H^T S^{-1} \quad (10)$$

$$X = X_t + K_t \hat{Y} \quad (11)$$

III. UJI COBA DAN ANALISIS

Untuk menguji kinerja sistem yang telah dibuat, dilakukan



Gambar 6. Visualisasi jalur kontrol posisi dengan referensi kontrol menggunakan sistem ICP-EKF.

tiga skenario pengujian sebagai berikut.

#### A. Uji Coba Posisi Diam

Salah satu kelemahan terbesar dari sistem navigasi *wheel odometry* dan *gyro* adalah akumulasi *error* yang terjadi, terutama pada sudut. *Drifting* pada data *gyro* terjadi seiring dengan berjalannya waktu, meskipun sistem berada dalam kondisi diam. Maka pada skenario ini dibandingkan performa sistem dalam kondisi diam selama satu sampai sepuluh menit. Data *error* lengkap hingga menit ke sepuluh dapat dilihat pada Tabel 1. Grafik perubahan sudut selama sepuluh menit dapat dilihat pada Gambar 3. Dari percobaan yang dilakukan, didapatkan bahwa integrasi sistem *wheel odometry* dengan ICP-EKF dapat mengatasi masalah *drifting* data sudut pada *gyro*.

#### B. Skenario Uji Coba Gerak Manual

Akumulasi *error* pada INS tidak hanya berpengaruh pada sudut, namun juga posisi. Pada skenario ini sistem akan diuji dengan beberapa pergerakan sederhana, lalu dikembalikan ke posisi semula. Pergeseran posisi pada posisi awal menjadi penilaian pada pengujian ini. Uji coba dilakukan dengan menggerakkan robot secara manual dengan *remote controller*, dan dikembalikan ke posisi semula.

Jalan yang ditempuh robot dan posisi akhir dapat dilihat pada Gambar 4. Posisi akhir setelah robot dikembalikan ke posisi awal dapat dilihat pada Tabel 2. Dapat dilihat bahwa perbaikan yang signifikan hanya terjadi pada sudut. Hal ini disebabkan *Kalman Filter* masih mendapatkan pengaruh yang besar dari *wheel odometry*, karena *data rate* Lidar yang digunakan hanya 10 Hz, sangat rendah dibandingkan dengan *data rate wheel odometry* yang mencapai 100 Hz.

#### C. Skenario Uji Coba Gerak Otomatis

Pada skenario ini akan diuji kemampuan sistem untuk digunakan sebagai acuan *positioning control*, karena diasumsikan dengan menggunakan data Lidar yang frekuensi *data rate*-nya jauh lebih lambat dari *wheel odometry*, akan ada perlambatan *response time* sistem. Pada skenario ini *response time* dilihat dengan melihat *error* posisi sistem setelah posisi tujuan dicapai. Uji coba dilakukan dengan menjalankan robot secara otomatis menggunakan kontrol PID dengan referensi posisi kedua sistem. Pengujian dilakukan dengan dua kecepatan kontrol yang berbeda untuk masing-masing sistem. Rata-rata *error* posisi sistem dapat dilihat pada Tabel 3. Visualisasi posisi kontrol dapat dilihat pada Gambar 5 dan Gambar 6.

## IV. KESIMPULAN

#### A. Kesimpulan

Berdasarkan pada pengujian yang dilakukan, dapat disimpulkan beberapa poin mengenai performa sistem Navigasi integrasi ICP-EKF dengan *Wheel Odometry*. Integrasi *Wheel Odometry* dengan ICP-EKF dapat mengatasi masalah *drifting* pada sudut *gyro*. Setelah digunakan selama sepuluh menit, *error* sudut pada *gyro* dapat mencapai  $3^\circ$ , sedangkan sistem navigasi yang dibangun dapat dengan konsisten memiliki *error* sudut di bawah  $0.7^\circ$ .

Sistem yang dibangun dapat mengatasi akumulasi *error* perjalanan yang jauh karena memiliki referensi dari luar sistem untuk memperbaiki data. Posisi dari sistem memiliki sedikit *delay* dari data *realtime* karena memiliki tahap proses tambahan. Namun masih dapat digunakan untuk kontrol

posisi pada kecepatan rendah. Pada kecepatan tinggi, dapat terjadi *overshoot* karena frekuensi *refresh* posisi kurang dari frekuensi kontrol.

### B. Saran

Berdasarkan pengalaman yang dialami penulis dalam mengerjakan penelitian ini, terdapat beberapa saran yang bisa diambil untuk pengembangan yang dapat dilakukan terhadap penelitian ini di masa yang akan datang. Arsitektur ROS *nodes* yang lebih simpel dapat mempercepat kecepatan proses sistem. Penggunaan *Iterative Closest Line*, variasi dari ICP dapat mengurangi pengaruh *noise* pada data *point cloud* dengan melakukan segmentasi garis terlebih dahulu untuk mendapatkan fitur yang relevan.

Pembuatan model untuk EKF masih dapat dioptimalkan lagi untuk mendapatkan hasil estimasi yang lebih baik. Konversi satuan variabel kontrol ke satuan baku yang konsisten agar lebih mudah dilakukan *debugging* dan penyetelan sistem. Penggunaan variasi *estimator* lain seperti *Unscented Kalman Filter* dapat dilakukan untuk

dibandingkan performanya dengan sistem yang sudah ada.

### DAFTAR PUSTAKA

- [1] D. Wang, H. Liang, T. Mei, H. Zhu, J. Fu, and X. Tao, "Lidar Scan Matching EKF-SLAM Using the Differential Model of Vehicle Motion," in *IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 908-912.
- [2] J. Lv, Y. Kobayashi, A. A. Ravankar, and T. Emaru, "Straight line segments extraction and EKF-SLAM in indoor environment," *J. Autom. Control Eng.*, vol. 2, no. 3, 2014.
- [3] H. Wang, G. Fu, J. Li, Z. Yan, and X. Bian, "An adaptive UKF based SLAM method for unmanned underwater vehicle," *Math. Probl. Eng.*, vol. 2013, 2013, [Online]. Available: <https://doi.org/10.1155/2013/605981>.
- [4] Z. Kurt-Yavuz and S. Yavuz, "A Comparison of EKF, UKF, FastSLAM2.0, and UKF-based FastSLAM Algorithms," in *IEEE 16th International Conference on Intelligent Engineering Systems (INES)*, 2012, pp. 37-43.
- [5] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*. 2014.
- [6] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *J. Mach. Learn. Res.*, vol. 4, no. 3, pp. 380-407, 2004.