

## Integrasi Odometri LiDAR dan Sensor IMU untuk Peningkatan Lokalisasi pada Robot Bergerak *Indoor*

Muhammad Fauzan Ridho<sup>1</sup>, Ali Zainal Abidin<sup>2</sup>, Belen Septian<sup>2</sup>

<sup>1</sup> Fakultas Sains dan Teknik, Teknik Elektro, Universitas Bangka Belitung, Pangkal Pinang, Indonesia

<sup>2</sup> Fakultas Ilmu Komputer dan Rekayasa, Teknik Elektro, Universitas Multi Data Palembang, Palembang, Indonesia

### INFORMASI ARTIKEL

Diterima Redaksi: 26 Februari 2024

Revisi Akhir: 26 Maret 2024

Diterbitkan Online: 27 Maret 2024

### KATA KUNCI

Odometri; Lokalisasi; Mobil Bergerak;  
Fusi Sensor

### KORESPONDENSI

Phone: +62 821-7532-4972

E-mail: [muhammadfauzanridho@ubb.ac.id](mailto:muhammadfauzanridho@ubb.ac.id)

### A B S T R A K

Lokalisasi adalah salah satu kompon penting dalam navigasi robot bergerak. Tugas tersebut dapat dilakukan dengan memanfaatkan data odometri LIDAR dan sensor unit pengukuran inersia (IMU). Odometri lidar memberikan informasi tentang gerak berbasis cahaya, sedangkan IMU memberikan data tentang percepatan dan kecepatan sudut. Keduanya mempunyai kelebihan, namun juga mempunyai kelemahan. Akumulasi kesalahan odometri LiDAR dan penyimpangan IMU seiring waktu sering terjadi pada odometri yang berbasis pada dua sensor tersebut. Penelitian ini mencoba mengatasi permasalahan tersebut dengan mengembangkan metode fusi sensor dengan mengntegrasi data odometri LIDAR dan IMU bertujuan untuk meningkatkan akurasi dan keandalan lokalisasi robot. Penelitian meliputi kalibrasi sensor, pemrosesan data, dan evaluasi kinerja pendekatan fusi yang diusulkan. Dengan memanfaatkan literatur terbaru tentang pengembangan teknologi sensor dan strategi fusi sensor, penelitian ini berkontribusi pada pemahaman yang lebih baik tentang cara meningkatkan navigasi robot bergerak dalam berbagai situasi. Kami berharap hasil penelitian ini dapat membuka jalan bagi pengembangan robot bergerak yang lebih akurat dan andal, terutama ketika melakukan pergerakan pada lingkungan yang kompleks.

## PENDAHULUAN

Dalam perjalanan dinamika menuju era robotika bergerak yang semakin berkembang, kehandalan dan ketepatan navigasi robot menjadi aspek yang semakin penting. Menyusuri ragam aplikasi, mulai dari kendaraan otonom hingga pengiriman otomatis, ketersediaan informasi lokalisasi, dalam hal ini estimasi posisi dan orientasi yang akurat membentuk dasar untuk keberhasilan setiap robot bergerak. Sebagai respons terhadap kompleksitas dan variasi lingkungan yang dihadapi oleh robotik bergerak, penelitian dan pengembangan dalam meningkatkan kemampuan lokalisasi menjadi semakin menarik.

Odometri berbasis *wheel encoder* adalah salah satu metode untuk mengestimasi posisi dan orientasi robot memanfaatkan perhitungan rotasi roda robot untuk memperkirakan perubahan posisi [1]–[3]. Meskipun metode ini telah lama digunakan dan terbukti efektif dalam banyak konteks, tetapi tetap rentan terhadap beberapa kendala. Akurasi odometri berbasis *wheel encoder* dapat terpengaruh oleh berbagai faktor seperti gesekan, *slippage*, atau deformasi roda [1]. Oleh karena itu, integrasi dengan sensor lain menjadi penting untuk mengatasi keterbatasan ini dan meningkatkan ketepatan navigasi robot.

Di sisi lain, LiDAR (*Light Detection and Ranging*) dan sensor *Inertial Measurement Unit* (IMU) telah menjadi salah komponen standar dalam sistem navigasi robot [1], [4]. Odometri dari LiDAR memberikan informasi jarak dan sudut berbasis cahaya yang sangat berguna untuk mengidentifikasi perubahan posisi robot ketika odometri berbasis *wheel encoder* tidak dapat dipakai di lingkungan tertentu [1], [5]. Sementara itu IMU memberikan data percepatan dan kecepatan

sudut untuk memonitor perubahan orientasi [6]. Namun, keduanya memiliki keterbatasan dan kelemahan masing-masing. Odometri LiDAR rentan terhadap kesalahan akumulasi selama perjalanan, terutama di lingkungan dengan hambatan atau permukaan yang tidak rata serta permukaan yang memiliki material reflektif [5]. Sementara itu, IMU memiliki kecenderungan terhadap *drift*, di mana kesalahan perlahan-lahan terakumulasi seiring waktu. Hal ini dapat menyebabkan estimasi posisi dan orientasi yang dilakukan murni hanya dengan menggunakan IMU akan memiliki kecenderungan terjadinya penurunan akurasi seiring waktu [6]. Oleh karena itu, untuk meningkatkan ketepatan dan keandalan lokalisasi robot, diperlukan integrasi efektif antara odometri LiDAR dan sensor IMU.

Dalam upaya mengatasi tantangan tersebut, penelitian ini mengusulkan pendekatan untuk mengintegrasikan data odometri LiDAR dan IMU dengan membentuk suatu framework fusi sensor yang dapat merespons secara adaptif terhadap dinamika lingkungan. Dengan menggunakan algoritma *Extended Kalman Filter* (EKF) untuk mengintegrasikan data odometri LiDAR dan IMU, serta mengurangi kelemahan dari kedua sensor tersebut untuk lokalisasi robot bergerak, diharapkan cara tersebut dapat menghasilkan estimasi posisi dan orientasi yang lebih akurat dan responsif terhadap perubahan kondisi sekitar. Dengan mengintegrasikan fusi sensor tersebut kedalam *framework Robot Operating System* (ROS), memberikan modularitas dan skabilitas pada sistem fusi untuk dapat dipakai dan dikembangkan lebih lanjut pada berbagai jenis robot bergerak dengan hanya beberapa penyesuaian parameter.

## TINJAUAN PUSTAKA

### *Extended Kalman Filter sebagai Algoritma Fusi Sensor*

Penggunaan *Extended Kalman Filter* (EKF) dalam fusi sensor telah menjadi perhatian utama dalam pengembangan sistem navigasi robotika mobile. EKF merupakan varian dari algoritma Kalman Filter yang dikembangkan untuk menangani sistem nonlinier [7], [8]. Dalam konteks fusi sensor, EKF berfungsi untuk mengintegrasikan dan menyatukan data dari berbagai sensor, menghasilkan estimasi pengukuran yang lebih akurat dan konsisten. EKF didasarkan pada dua proses utama, yaitu proses prediksi dan proses pembaruan. Proses prediksi menghasilkan perkiraan posisi berikutnya berdasarkan model dinamis dan data odometri sebelumnya. Proses pembaruan kemudian menggabungkan data sensor terkini, seperti data lidar atau IMU, untuk memperbaiki estimasi posisi dan mengurangi ketidakpastian [7]–[9].

Algoritma EKF dimulai dengan mendefinisikan sistem dinamik non-linear dari robot yang dideskripsikan pada persamaan (1) [7], [9]:

$$x_k = f(x_{k-1}) + w_{k-1} \quad (1)$$

Dimana dalam ekspresi tersebut,  $x_k$  mewakili keadaan sistem robot (*state vector*), khususnya posisi dan orientasi robot, pada waktu  $k$ . Fungsi  $f$  adalah fungsi transisi keadaan nonlinier, dan  $w_{k-1}$  mewakili *noise* proses, diasumsikan mengikuti distribusi normal.

$$z_k = h(x_k) + v_k \quad (2)$$

Pada persamaan (2) [7],  $z_k$  menunjukkan pengukuran yang diambil pada waktu  $k$ ,  $h$  menggambarkan model sensor nonlinier yang bertugas menerjemahkan keadaan ke dalam ruang pengukuran, dan  $v_k$  adalah *noise* pengukuran yang diasumsikan mengikuti distribusi normal.

Langkah awal pada algoritma EKF terlihat pada persamaan (3) dan (4) [7], [8], adalah melakukan langkah prediksi yang memproyeksikan estimasi keadaan dan kovarian kesalahan saat ini ke masa depan (*prior state vector* dan *prior error covariance*).

$$\hat{x}_k = f(x_{k-1}) \quad (3)$$

$$\hat{P}_k = F P_{k-1} F^T + Q \quad (4)$$

Dalam aplikasinya,  $f$  mewakili model kinematika robot yang berasal dari prinsip mekanika Newton. Proyeksi kovarian kesalahan perkiraan,  $P$ , dilakukan melalui  $F$ , yaitu matriks Jacobian dari  $f$ , dan selanjutnya disesuaikan oleh  $Q$ , kovarian *noise* proses yang harus disesuaikan terhadap aplikasi sistem tertentu atau disebut dengan *tuning*.

Tahap kedua adalah langkah koreksi dengan mengupdate persamaan sebelumnya dengan data pengukuran terbaru, yang diperlihatkan pada persamaan (5) [7], [9], [10]:

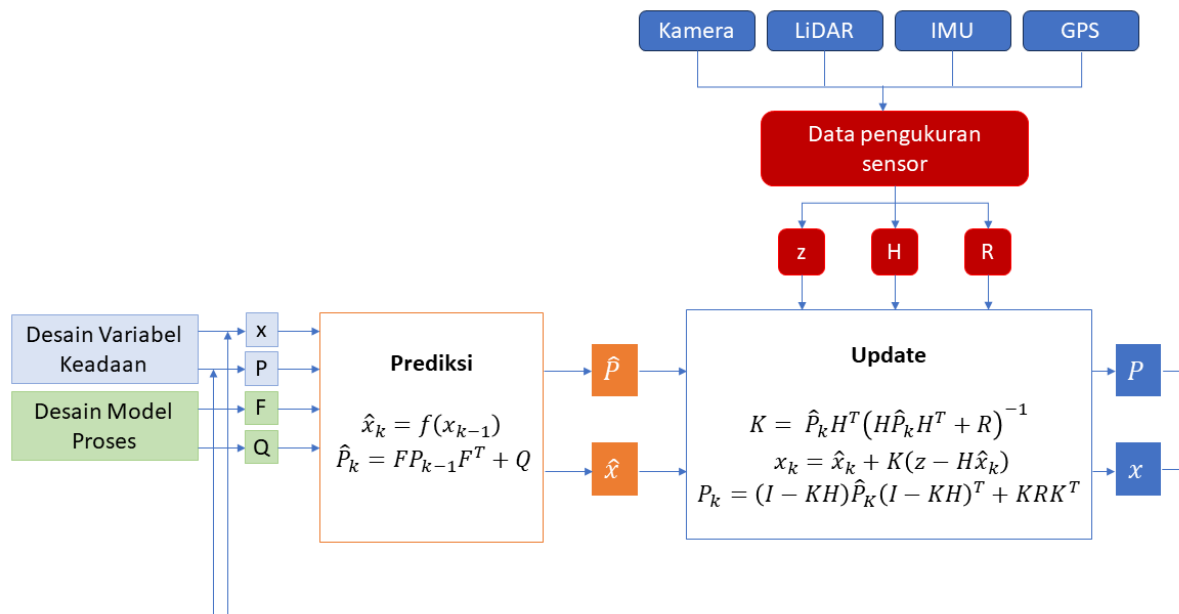
$$K = \hat{P}_k H^T (H \hat{P}_k H^T + R)^{-1} \quad (5)$$

$$x_k = \hat{x}_k + K(z - H\hat{x}_k) \quad (6)$$

$$P_k = (I - KH)\hat{P}_k(I - KH)^T + KRK^T \quad (7)$$

Kalman Gain  $K$  dihitung dengan menggunakan matriks observasi  $H$ , kovarian pengukuran  $R$ , dan  $\hat{P}_k$  (kovarian matriks posterior). Pada persamaan (6) [7], [10], gain tersebut digunakan untuk memperbarui vektor keadaan dan matriks kovarian dengan menggunakan bentuk update kovarian Joseph pada persamaan (7) [7], [10]. Pada persamaan standar EKF,  $H$  adalah sebuah matriks Jacobian yang mengakomodir elemen-elemen nonlinear dari persamaan keadaan sistem.

Proses akan terus berulang dengan mengakomodir pengukuran yang baru. Diagram proses algoritma EKF sebagai algoritma fusi sensor diperlihatkan pada Gambar 1.



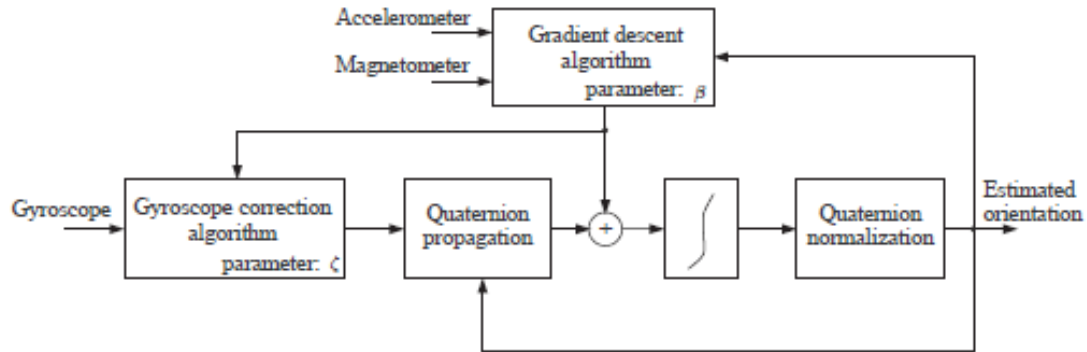
Gambar 1. Diagram proses algoritma EKF untuk fusi sensor

### Filter Madgwick untuk Sensor Orientasi MARG

Salah satu masalah terjadi dari suatu sensor orientasi dan inersia adalah, ketika pengukuran sensor memiliki sumbu yang sejajar dengan medan bumi sehingga mencegah penggunaan fungsi trigonometri untuk menentukan orientasi karena  $\tan(90)$  tidak terdefinisi. Fusi dari sensor-sensor tersebut menjadi kritis untuk akurasi orientasi dan untuk transformasi lengkap dari kerangka inersial sensor ke kerangka bumi [11].

Filter *Madgwick* adalah suatu jenis filter orientasi atau algoritma fusi sensor yang digunakan untuk memperkirakan orientasi atau posisi suatu objek berdasarkan data yang diperoleh dari sensor inersia dan orientasi. Madgwick Filter memanfaatkan pengukuran MARG (*Magnetic, Angular Rate, and Gravity*) dari Giroskop (*Angular Rate*), Akselerometer (*Gravity*), dan Magnetometer (*Magnetic*). Dengan menggunakan representasi kuaternion dan mengintegrasikan kompensasi distorsi magnetik, algoritma ini memungkinkan penggunaan data dari Akselerometer dan Magnetometer dalam algoritma *Gradien Descent* yang dianalisis dan dioptimalkan untuk menghitung arah kesalahan pengukuran Giroskop sebagai derivatif kuaternion [11], [12].

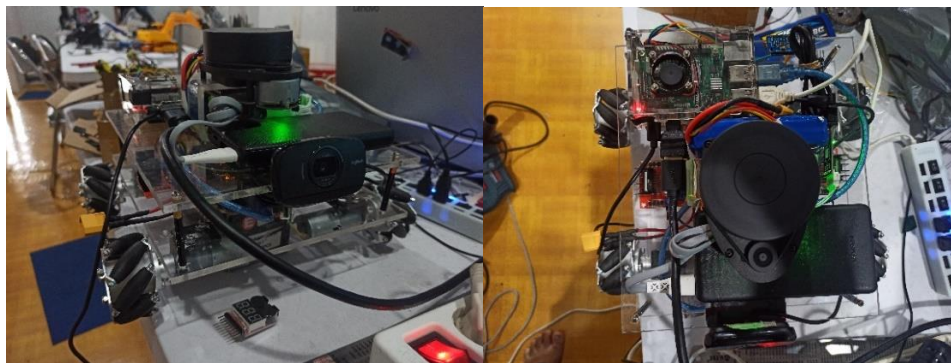
Terdapat dua metode utama yang digunakan untuk menghitung orientasi benda kaku. Untuk mengurangi efek kesalahan bias dan *drift*, pengukuran Girooskop pertama-tama dielaborasi menggunakan algoritma koreksi yang bergantung pada parameter  $\zeta$ . Pengukuran ini kemudian digunakan untuk menghitung orientasi benda menggunakan propagasi Kuaternion, dimulai dari orientasi yang diestimasi pada langkah sebelumnya. Selanjutnya, algoritma *Gradient Descent* yang rumusnya diberikan dalam [11], menggabungkan pembacaan Akselerometer dan magnetometer dengan parameter konfigurasi  $\beta$ . Kemudian, orientasi yang dihitung hanya berdasarkan pengukuran Girooskop dikoreksi menggunakan keluaran algoritma *Gradient Descent* [11], [12]. Gambar 2 menunjukkan blok algoritma filter Madgwick



Gambar 2. Blok diagram algoritma filter Madgwick [12]

## METODOLOGI

Metodologi penelitian ini berfokus pada implementasi integrasi odometri LiDAR dan sensor IMU dengan memanfaatkan algoritma EKF untuk fusi odometri sebagai estimator posisi robot, yang dibuat di dalam kerangka kerja *Robot Operating System* (ROS). Implementasi ini dijalankan dalam lingkungan ROS untuk memfasilitasi pengembangan robotika yang modular dan mempermudah integrasi beberapa data yang berasal dari pengukuran kedua sensor. Hasil dari fusi odometri tersebut digunakan pada robot bergerak menggunakan 4 roda mecanum yang diperlihatkan pada Gambar 3.



Gambar 3. Robot yang digunakan sebagai wahana pengujian fusi odometri

Robot yang digunakan memiliki persamaan kinematik balik dan kinematik maju yang diperlihatkan pada persamaan (8) dan (9) [13], [14] :

Kinematika Maju :

$$\begin{aligned}\omega_1 &= \frac{1}{r}(v_X - v_Y - (l_x + l_y)\omega_z) \\ \omega_2 &= \frac{1}{r}(v_X + v_Y + (l_x + l_y)\omega_z) \\ \omega_3 &= \frac{1}{r}(v_X + v_Y - (l_x + l_y)\omega_z) \\ \omega_4 &= \frac{1}{r}(v_X - v_Y + (l_x + l_y)\omega_z)\end{aligned}\quad (8)$$

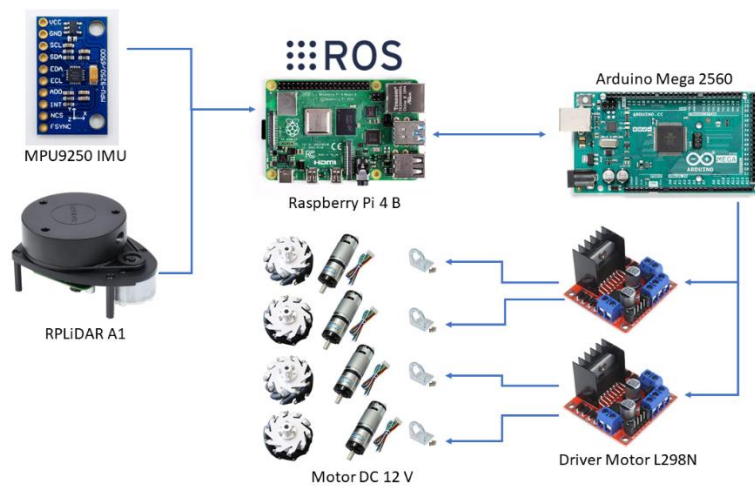
Kinematika Balik :

$$\begin{aligned}v_X &= \frac{r}{4}(\omega_1 + \omega_2 + \omega_3 + \omega_4) \\ v_Y &= \frac{r}{4}(-\omega_1 + \omega_2 + \omega_3 - \omega_4)\end{aligned}$$

$$\omega_z = \frac{r}{4(l_x + l_y)}(-\omega_1 + \omega_2 - \omega_3 + \omega_4) \quad (9)$$

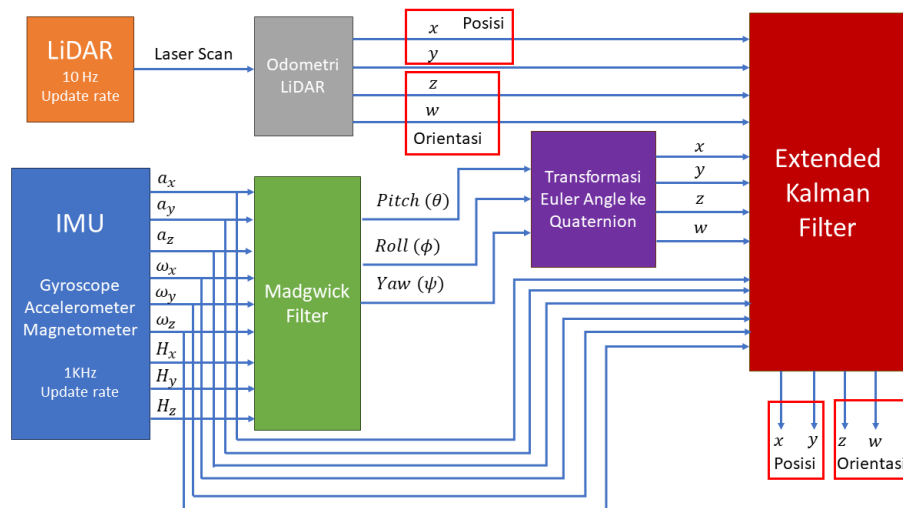
Dimana,  $\omega_i, i = 1, 2, 3, 4$ , adalah kecepatan masing-masing roda;  $v_x, v_y$  kecepatan linier robot pada sumbu x,y;  $\omega_z$ , kecepatan angular robot;  $l_x$ , jarak antara pusat robot dan garis tengah roda mecanum sepanjang sumbu x;  $l_y$ , jarak antara pusat robot dan garis tengah roda mecanum sepanjang sumbu y.

Robot yang digunakan pada pengujian, dilengkapi dengan komputer papan tunggal Raspberry Pi 4 model B+ ; sensor IMU MPU9250 yang terdapat sensor Akselerometer, Giroskop, dan Magnetometer didalamnya; sensor LiDAR RPLiDAR A1; serta kamera yang berfungsi sebagai visualisasi sudut pandang robot. Raspberry Pi 4 menggunakan sistem operasi Ubuntu 20.04 yang telah dilengkapi juga dengan ROS versi Noetic. Sementara untuk kendali level bawah menggunakan *development board* mikrokontroller Arduino Mega 2560 yang terhubung dengan Raspberry Pi 4 melalui komunikasi serial, dan driver motor L298N untuk mengendalikan putaran motor DC 12 V pada robot. Pada Raspberry Diagram perangkat keras yang ada pada robot diperlihatkan pada Gambar 4.



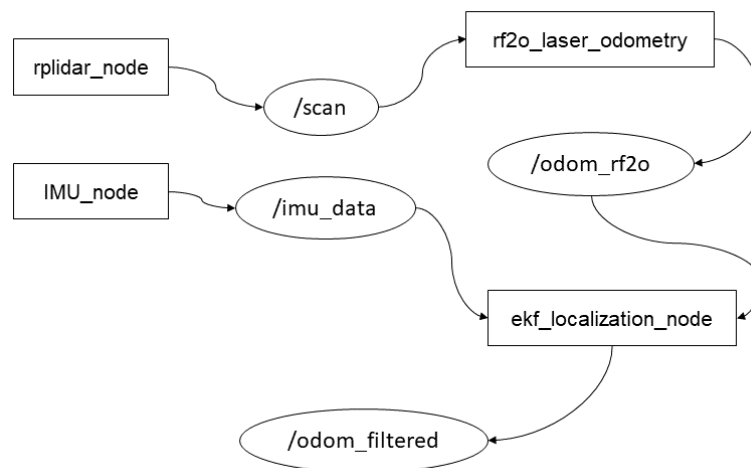
Gambar 4. Digram perangkat keras robot

Desain filter melibatkan beberapa data dari kedua sensor dengan paradigma desain fusi dua tahap. Untuk mendapatkan data sudut Euler (*Pitch, Yaw, Roll*) perlu dilakukan fusi sensor Akselerometer, Giroskop, dan Magnetometer dengan menggunakan filter Madgwick pada tahap pemrosesan data MARG. Update pengukuran dari IMU dilakukan dengan frekuensi 1 KHz. Hasil pemrosesan IMU harus di konversikan dari bentuk sudut Euler ke dalam bentuk Kuaternion yaitu rotasi pada sumbu x, y, dan z dan sudut rotasi w. Data tersebut diteruskan ke EKF untuk diintegrasikan dengan data odometri LiDAR dari hasil pemindaian LiDAR dengan frekuensi 10 Hz. Data akselerasi sumbu x, y, dan z ( $a_x, a_y, a_z$ ) dari akselerometer dan kecepatan sudut sumbu x, y, dan z ( $\omega_x, \omega_y, \omega_z$ ) juga diintegrasikan ke dalam algoritma EKF pada bagian update. Estimasi posisi yang dihasilkan oleh EKF hanya mencakup sumbu x dan y dikarenakan robot hanya bergerak pada sumbu tersebut. Sementara untuk orientasi memunculkan data rotasi pada sekitar sumbu z dan sudut orientasi w. Desain inetgrasi odometri LiDAR dan sensor IMU menggunakan EKF diperlihatkan pada Gambar 5



Gambar 5. Desain algoritma fusi untuk estimasi posisi

Algoritma fusi yang masuk pada bagian desain perangkat lunak robot, dilakukan pengintegrasian dengan menggunakan *framework* ROS, sehingga setiap fungsi pada robot dibuat kedalam sebuah ROS *Node* yang mempublish *Topic* tertentu yang berisikan data yang dibutuhkan *Node* lain dan menjadi saluran komunikasi antar *Node*. Komunikasi antar ROS *Node* diatur dengan menggunakan paradigma komunikasi *Publish-Subscribe*. Yang artinya setiap ROS *Node* dapat melakukan *Subscribe* pada *Topic* yang di-*Publish* *Node* tertentu sesuai parameter yang ditetapkan dan dibutuhkan pada *Source code* program *Node* tersebut. Pada penelitian ini, algoritma EKF diprogram ke dalam sebuah *Node* bernama *ekf\_localization\_node*. *Node* ini men-*Subscribe* *Topic* *\imu\_data* dan *\odom\_rf2o* yang di-*Publish* *Node* *rf2o\_laser\_odometry* dan *IMU\_Node*. Kedua *node* berisi data sensor dari IMU dan odometri LiDAR. *Node* *rf2o\_laser\_odometry* sendiri adalah sebuah *package* ROS yang menyediakan program algoritma odometri LiDAR dengan memanfaatkan pemindaian radial pada LiDAR 2 dimensi untuk memberikan informasi posisi ( $x, y$ ) dan orientasi ( $z, w$ ) berdasarkan sudut pandang LiDAR tertentu [15]. Berdasarkan integrasi dari kedua sensor tersebut melalui algoritma fusi EKF, *ekf\_localization\_node* akan mem-*Publish* *Topic* *odometry/filtered* yang berisi data posisi dan orientasi yang telah terintegrasi dari kedua sensor. Skenario publish-subscribe fusi sensor pada ROS diperlihatkan pada Gambar 6.



Gambar 6. Skenario *Publish-Subscribe* fusi sensor pada ROS

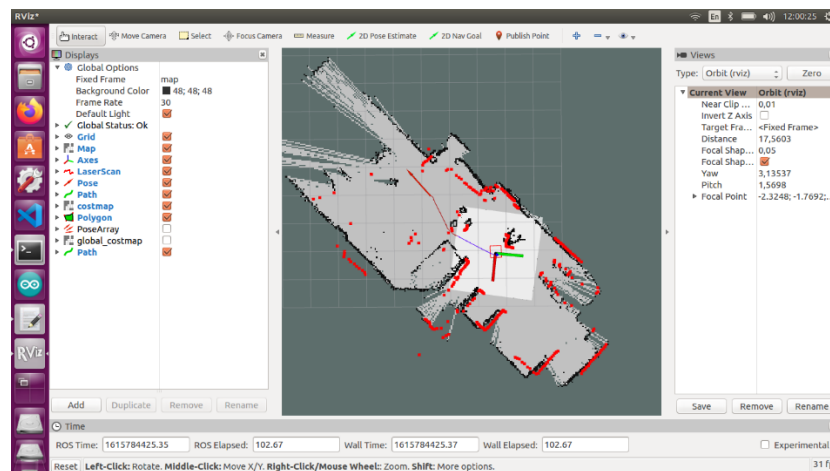
## HASIL DAN PEMBAHASAN

Pengujian estimasi posisi dari odometri hasil EKF dilakukan untuk melihat tingkat seberapa jauh error estimasi dengan posisi robot sebenarnya. Pengujian dilakukan pada lingkup ruangan indoor yang diperlihatkan pada Gambar 7. Data hasil pengujian serta posisi robot relatif terhadap posisi awal divisualisasikan melalui Rviz, yaitu perangkat visualisasi yang tergabung dalam instalasi ROS. Visualisasi posisi robot pada Rviz diperlihatkan pada Gambar 8





Gambar 7. Pengujian estimasi posisi robot indoor



Gambar 8. Tampilan posisi dan orientasi robot pada Rviz

Data numerik yang dihasilkan oleh algoritma EKF pada `ekf_localization_node`, diakses pada file data logger ROS atau Roslog dengan tampilan pada Gambar 9.

Odom data	
pose:	pose:
position:	position:
x: -0.920187115669	x: -0.895615675797
y: -0.056956410408	y: -0.0833493462189
z: 0.0	z: 0.0
orientation:	orientation:
x: 0.0	x: 0.0
y: 0.0	y: 0.0
z: -0.759757286524	z: -0.997587154265
w: 0.650206786779	w: 0.0694252810303

Gambar 9. File Roslog untuk nilai odometri EKF

Pada bagian pemrograman filter, nilai awal *gain* filter  $\beta$  pada filter Madgwick di atur menjadi 0.1, sedangkan untuk nilai awal matriks kovarian *noise* proses  $\mathbf{Q}$  dan matriks kovarian kesalahan perkiraan  $\mathbf{P}$  diatur menjadi nilai larik berikut :

```
process_noise_covariance: [0.05, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.05, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.06, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.03, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

0.0, 0.0, 0.0, 0.0, 0.0, 0.06, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.025, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.025, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.04, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0, 0.0, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.02, 0.0, 0.0, 0.0, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.01, 0.0, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.01, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.015]

initial\_estimate\_covariance: [1e-9, 1e-9, 1e-9, 1e-9, 1e-9, 1e-9, 1e-9, 1e-9, 1e-9, 1e-9, 1e-9, 1e-9, 1e-9, 1e-9, 1e-9]

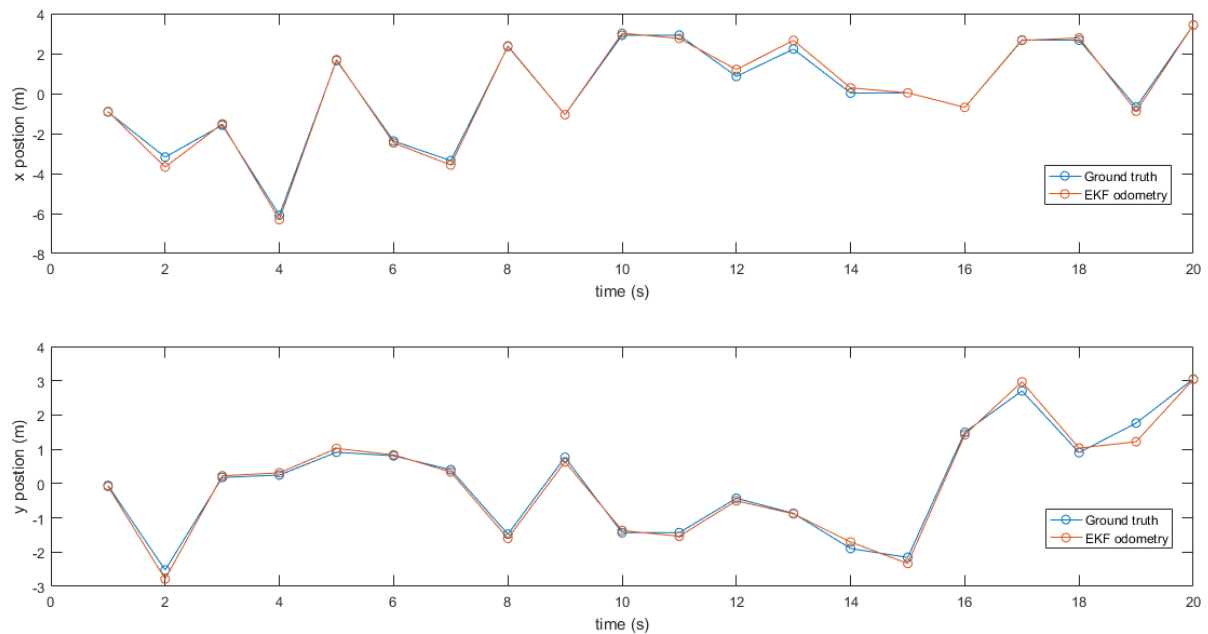
secara berurutan nilai tersebut adalah x, y, z, roll, pitch, yaw,  $v_x$ ,  $v_y$ ,  $v_z$ ,  $\omega_x$ ,  $\omega_y$ ,  $\omega_z$ ,  $a_x$ ,  $a_y$ ,  $a_z$ .

Setelah dilakukan pengujian estimasi posisi dan orientasi pada robot didapatkan data posisi robot yang dibandingkan dengan data posisi asli (*ground truth*) yang ditunjukkan pada Tabel 1. Sedangkan grafik odometri EKF berbanding dengan posisi asli robot diperlihatkan pada Gambar 10.

Tabel 1. Hasil estimasi posisi odometri EKF dan *ground truth* robot

Ground Truth		Odometri EKF	
Posisi			
x (m)	y (m)	x (m)	y (m)
-0.920	-0.057	-0.895	-0.083
-3.176	-2.522	-3.673	-2.783
-1.580	0.179	-1.513	0.227
-6.082	0.249	-6.304	0.314
1.656	0.915	1.694	1.024
-2.375	0.809	-2.469	0.839
-3.353	0.404	-3.572	0.338
2.355	-1.476	2.381	-1.598
-1.055	0.765	-1.055	0.635
2.921	-1.433	3.018	-1.370
2.922	-1.432	2.758	-1.537
0.860	-0.428	1.202	-0.503
2.227	-0.868	2.662	-0.883
0.026	-1.895	0.285	-1.706
0.039	-2.155	0.039	-2.331
-0.691	1.498	-0.691	1.428
2.681	2.703	2.660	2.968
2.672	0.904	2.785	1.034
-0.669	1.766	-0.879	1.222
3.429	3.051	3.439	3.039





Gambar 10. Grafik posisi odometri EKF vs posisi asli

Dari data Tabel 1 didapatkan nilai Akar Kuadrat Galat Rata-Rata (*Root Mean Square Error*) atau RMSE untuk estimasi posisi x adalah :

$$RMSE\ x = \sqrt{\frac{(\sum_i^N \text{ground truth } x_i - \text{odometri EKF } x_i)^2}{N}} \approx 0.003$$

Sementara untuk nilai RMSE untuk estimasi posisi y adalah :

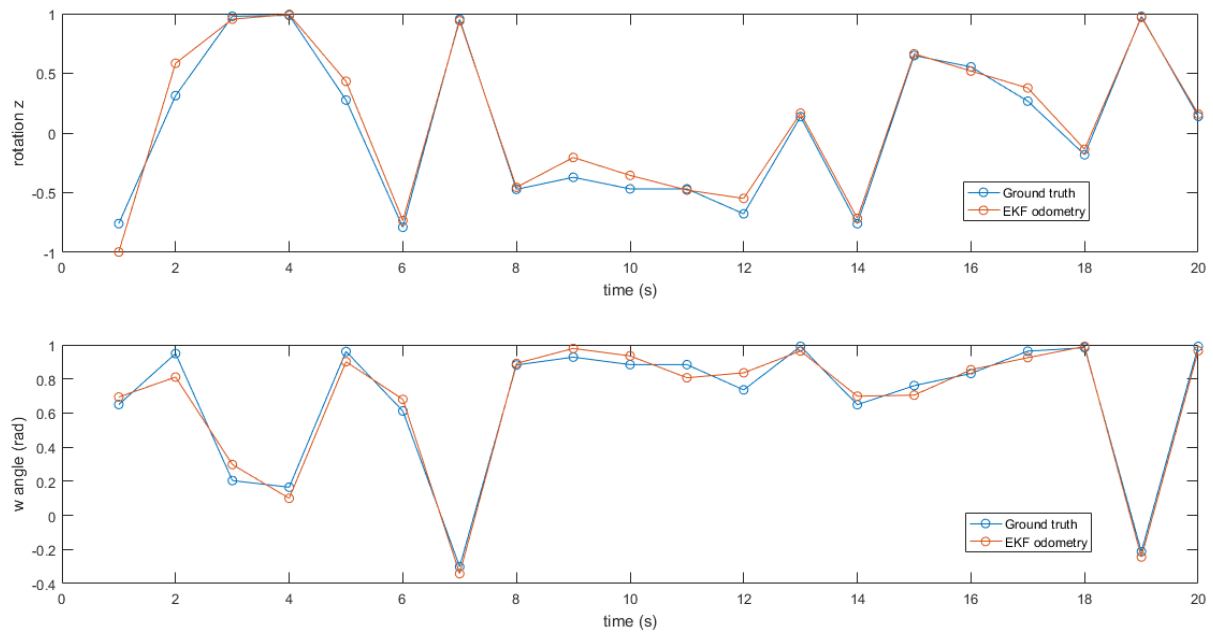
$$RMSE\ y = \sqrt{\frac{(\sum_i^N \text{ground truth } y_i - \text{odometri EKF } y_i)^2}{N}} \approx 0.157$$

Sedangkan untuk data estimasi orientasi dinyatakan dengan orientasi pada sumbu rotasi z , serta besar sudut orientasi w berdasarakan hasil perhitungan kuaternion odometri EKF. Dalam hal ini rotasi hanya terjadi pada sumbu z dikarenakan robot hanya berada pada posisi bidang datar. Tabel 2 memperlihatkan hasil odometri EKF dibandingkan dengan *ground truth*, sementara Gambar 11 memperlihatkan grafik dari data pada Tabel 2.

Tabel 2. Hasil estimasi orientasi odometri EKF dan *ground truth*

<i>Ground truth</i>		Odometri EKF	
Orientasi			
<i>z</i>	<i>w</i>	<i>z</i>	<i>w</i>
-0.759	0.650	-0.997	0.694
0.313	0.949	0.584	0.811
0.978	0.204	0.954	0.298
0.986	0.165	0.994	0.100
0.276	0.961	0.433	0.901
-0.790	0.612	-0.733	0.680
0.953	-0.301	0.940	-0.342
-0.472	0.882	-0.455	0.891

-0.371	0.928	-0.205	0.979
-0.468	0.884	-0.355	0.935
-0.468	0.884	-0.478	0.807
-0.677	0.736	-0.548	0.836
0.137	0.991	0.166	0.964
-0.760	0.649	-0.715	0.699
0.649	0.761	0.663	0.705
0.554	0.832	0.519	0.854
0.267	0.963	0.377	0.925
-0.182	0.983	-0.136	0.991
0.977	-0.213	0.969	-0.244
0.139	0.991	0.159	0.965



Gambar 11. Grafik orientasi odometri EKF vs oreintasi asli

Dari data Tabel 2 didapatkan nilai RMSE untuk rotasi pada sumbu z adalah sebagai berikut :

$$RMSE \text{ rot}_z = \sqrt{\frac{(\sum_i^N \text{ground truth rot}_{z_i} - \text{odometri EKF rot}_{z_i})^2}{N}} \approx 0.191$$

Sedangkan untuk nilai RMSE untuk sudut rotasi dalam radian w adalah sebagai berikut :

$$RMSE \text{ w} = \sqrt{\frac{(\sum_i^N \text{ground truth } w_i - \text{odometri EKF } w_i)^2}{N}} \approx 0.014$$

## KESIMPULAN DAN SARAN

Pada penelitian ini fusi odometri dari odometri LiDAR dan sensor IMU untuk robot bergerak pada lingkungan did dalam ruangan, menghasilkan nilai RMSE sebesar 0.003 untuk posisi x, 0.157 untuk posisi y, 0.191 untuk besar rotasi pada

sumbu z, dan 0.014 untuk besar sudut rotasi w. Dari nilai tersebut estimasi posisi dan orientasi dari algoritma EKF menghasilkan estimasi dengan performa yang cukup baik dengan rata-rata nilai yang tidak begitu jauh dari nilai asli. Untuk perbaikan lebih lanjut, disarankan untuk mempertimbangkan penambahan data dari sensor posisi dan orientasi yang lain seperti odometri visual dari kamera.

## DAFTAR PUSTAKA

- [1] S. A. S. Mohamed, M. H. Hagbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A Survey on Odometry for Autonomous Navigation Systems," *IEEE Access*, vol. 7, no. July, pp. 97466–97486, 2019, doi: 10.1109/ACCESS.2019.2929133.
- [2] R. A. Priambudi and M. Mobed Bachtari, "Penentuan Posisi Menggunakan Odometry Omniwheel," *Indones. Symp. Robot. Syst. Control*, no. August 2019, pp. 1–3, 2018, [Online]. Available: <https://www.researchgate.net/publication/335311518>.
- [3] A. N. Albab, E. Rahmawati, M. Yantidewi, I. Sucahyo, Dzulkiflih, and R. R. Firmansyah, "Control Position of Mobile Robot Based on Odometry Method and PID Controller," *J. Phys. Conf. Ser.*, vol. 1491, no. 1, 2020, doi: 10.1088/1742-6596/1491/1/012039.
- [4] S. Garg et al., *Semantics for Robotic Mapping, Perception and Interaction: A Survey*, vol. 8, no. 1–2, 2020.
- [5] D. Lee, M. Jung, W. Yang, and A. Kim, "LiDAR Odometry Survey: Recent Advancements and Remaining Challenges," *Intell. Serv. Robot.*, 2023, doi: 10.1007/s11370-024-00515-8.
- [6] A. Etzion and I. Klein, "MoRPI: Mobile Robot Pure Inertial Navigation," *IEEE J. Indoor Seamless Position. Navig.*, vol. 1, no. May, pp. 141–150, 2023, doi: 10.1109/jispin.2023.3334697.
- [7] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the robot operating system," *Adv. Intell. Syst. Comput.*, vol. 302, pp. 335–348, 2016, doi: 10.1007/978-3-319-08338-4\_25.
- [8] M. B. Alatis and G. P. Hancke, "Pose estimation of a mobile robot based on fusion of IMU data and vision data using an extended kalman filter," *Sensors (Switzerland)*, vol. 17, no. 10, 2017, doi: 10.3390/s17102164.
- [9] N. C. Basjaruddin, "Penggunaan Extended Kalman Filter Sebagai Estimator Sikap pada Sistem Kendali Servo Visual Robot," *J. Mechatronics, Electr. Power, Veh. Technol.*, vol. 2, no. 1, pp. 23–30, 2012, doi: 10.14203/j.mev.2011.v2.23-30.
- [10] O. J. Montañez, M. J. Suarez, and E. A. Fernandez, "Application of Data Sensor Fusion Using Extended Kalman Filter Algorithm for Identification and Tracking of Moving Targets from LiDAR–Radar Data," *Remote Sens.*, vol. 15, no. 13, 2023, doi: 10.3390/rs15133396.
- [11] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," *IEEE Int. Conf. Rehabil. Robot.*, no. June, 2011, doi: 10.1109/ICORR.2011.5975346.
- [12] A. Cirillo, P. Cirillo, G. De Maria, C. Natale, and S. Pirozzi, "A comparison of multisensor attitude estimation algorithms," *Multisens. Attitude Estim. Fundam. Concepts Appl.*, no. June, pp. 529–540, 2016, doi: 10.1201/9781315368795.
- [13] H. Taheri, B. Qiao, and N. Ghaeminezhad, "Kinematic Model of a Four Mecanum Wheeled Mobile Robot," *Int. J. Comput. Appl.*, vol. 113, no. 3, pp. 6–9, 2015, doi: 10.5120/19804-1586.
- [14] D. S. Alhanov and V. I. Rubtsov, "Development of the Laboratory Work: 'Modeling of a Mobile Robot on Mecanum Wheels Kinematics,'" *ITM Web Conf.*, vol. 35, p. 04001, 2020, doi: 10.1051/itmconf/20203504001.
- [15] M. Jaimez, J. G. Monroy, and J. Gonzalez-Jimenez, "Planar odometry from a radial laser scanner. A range flow-based approach," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2016-June, pp. 4479–4485, 2016, doi: 10.1109/ICRA.2016.7487647.