

## ANALISIS SIMULASI EXTENDED KALMAN FILTER PADA MOBILE ROBOT NAVIGATION MENGGUNAKAN LASER

### SIMULATION ANALYSIS OF EXTENDED KALMAN FILTER ON MOBILE ROBOT NAVIGATION USING LASER

Adlirrahman Hasfi Aji<sup>1</sup>, Agung Nugroho Jati, S.T,M.T<sup>2</sup>, Casi Setianingsih, S.T,M.T<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[adlirha@student.telkomuniversity.ac.id](mailto:adlirha@student.telkomuniversity.ac.id), <sup>2</sup>[agungnj@telkomuniversity.ac.id](mailto:agungnj@telkomuniversity.ac.id),

<sup>3</sup>[setiacasie@telkomuniversity.ac.id](mailto:setiacasie@telkomuniversity.ac.id)

---

#### Abstrak

Perkembangan Mobile Robot Navigation berkembang sangat cepat yang mana sekarang sudah diterapkan mobil tanpa pengemudi dengan mendeteksi objek yang akurat sehingga meminimalisir tabrakan. Simultaneous Localization and Mapping (SLAM) merupakan suatu teknik yang digunakan oleh mobile robot untuk membangun peta dari daerah yang sedang dijelajahi, dimana daerah tersebut tidak diketahui bentuk medan sebelumnya. Pada Jurnal ini, disimulasikan dan dianalisa algoritme Extended Kalman Filter(EKF), yang berfungsi untuk memperkirakan peta lingkungan dan memperkirakan posisi robot relatif terhadap lingkungannya, pada robot turtlebot. Simulasi dilakukan pada simulator Gazebo. Operating system pada robot ini menggunakan ROS (Robot Operating System). Dengan memanfaatkan ROS (Robot Operating System) sebagai basisnya, operating system ini bertugas sebagai system navigasi robot yang memetakan secara real time dan menggerakkan robot sekaligus melakukan pemetaan lingkungan robot dan memperkirakan posisi robot terhadap lingkungan.

**Kata kunci :** SLAM, EKFSLAM , Turtlebot, Gazebo,ROS

---

#### Abstract

The development of Mobile Robot Navigation developed very rapidly, which now has been implemented cars without driver by detecting accurate objects so as to minimize collisions. Simultaneous Localization and Mapping (SLAM) is one of the areas of research in the field of robotics and artificial intelligence. SLAM is used by Mobile Robots to build maps of areas that are being explored, where the area is not known beforehand. In this Paper simulation and analyze EKFSLAM algorithm, which is used to estimate the environmental map and estimate the position of the robot relative to its environment, on Turtlebot. Simulations performed on Gazebo simulator. The operating system on the robot using Robot Operating System (ROS). By utilizing ROS as a basis, the operating system functions as a robot navigation system that maps in real time and drive the robot and conduct environmental mapping and estimate the position of the robot on the environment.

**Keywords:** SLAM, EKFSLAM, Turtlebot, Gazebo, ROS

---

#### 1. Pendahuluan

Sistem navigasi merupakan bagian terpenting dari sebuah mobile robot, sehingga robot mampu bergerak secara mandiri dari suatu tempat ke tempat lain yang dituju tanpa melakukan kontak langsung dengan objek pada lingkungan, melalui penentuan posisi dan arah pergerakannya. Namun ketika robot dijalankan pada suatu tempat yang belum diketahui kondisi lingkungannya, sistem akan berjalan sesuai tujuannya atau mengalami kegagalan adalah tergantung pada sistem navigasinya[1].

Simultaneous Localization And Mapping(SLAM) memiliki banyak metode dalam mengambil ciri informasi dari ruang yang akan dijadikan peta, mulai dari menggunakan sensor ultrasonic, infra merah, kamera dan juga Light Detection and Ranging(LIDAR) dan encoder untuk mendapatkan data odometry dari robot untuk informasi lokasi awal dimana robot berada. Penggunaan odometry biasanya akan menghasilkan error yang semakin lama semakin membesar. Untuk mengatasi hal tersebut, digunakan filter berupa Extended Kalman Filter(EKF)[2][4].

Ada beberapa pendekatan untuk memecahkan masalah SLAM. Dua metode utama adalah Kalman Filtering dan partikel filter. Memperkenalkan Filter Particle Rao-Blackwellized (RBPF) sebagai pendekatan solusi untuk masalah SLAM. Pendekatan ini menggunakan sebuah filter Partikel dimana setiap partikel memiliki peta sendiri dan menggambarkan setiap jalur yang memungkinkan [3][5].

## 2. Dasar Teori

### 2.1. Simultaneous Localization and Mapping (SLAM)

Seperti yang sudah dinyatakan, SLAM adalah singkatan dari Simultaneous Localization and Mapping. SLAM merupakan pengembangan dari pekerjaan sebelumnya yang dilakukan oleh Hugh Durrant-Whyte dan John J. Leonard [9] berdasarkan pekerjaan yang dilakukan oleh Smith, Self, dan Cheeseman [10]. SLAM bersangkutan dengan masalah membangun peta lingkungan yang tidak diketahui oleh mobile robot sementara pada saat yang sama melakukan navigasi di dalam peta tersebut [11].

SLAM terdiri dari beberapa langkah, yaitu ekstraksi landmark, asosiasi data, state estimation, state update, dan landmark update. Ada banyak cara untuk mengerjakan langkah-langkah tersebut [11].

SLAM terdiri dari beberapa langkah. Tujuan dari keseluruhan proses tersebut adalah memperbaharui posisi dari robot dengan menggunakan lingkungan sekeliling. Karena odometry (yang memberikan pengukuran posisi robot) dari robot sering mengalami kesalahan, tidak dapat bergantung sepenuhnya dengan pengukuran odometry, dapat digunakan scan dari laser untuk memperbaiki pengukuran posisi robot. Hal tersebut dapat dilakukan dengan ekstraksi landmark yang ada di lingkungan dan mengamatinya kembali selagi robot bergerak [11].

SLAM memiliki beberapa jenis metode, salah satunya metode Extended Kalman Filter (EKFSLAM). Metode Extended Kalman Filter merupakan pengembangan dari metode Kalman Filter. Perbedaan diantara keduanya adalah, metode Kalman Filter digunakan untuk mengestimasi parameter-parameter yang bersifat linear, sedangkan Extended Kalman Filter ditujukan untuk kasus-kasus non linear.

### 2.2. Extended Kalman Filter (EKFSLAM)

EKFSLAM, sebuah pendekatan alternatif terhadap SLAM yang berdasarkan particle filtering. EKFSLAM memanfaatkan independensi kondisional yang disebabkan oleh struktur SLAM untuk memfaktorisasi probabilitas ke dalam probabilitas yang mempunyai dimensi rendah [1].

#### a) Particle Filtering

Particle filter, merepresentasikan distribusi probabilitas menggunakan himpunan sampel kondisi atau particle. Daerah dengan probabilitas yang tinggi memiliki kepadatan particle yang tinggi, sedangkan daerah dengan probabilitas rendah memiliki kepadatan particle yang rendah atau bahkan tidak memiliki particle sama sekali [1].

#### b) Representasi Posterior Terfaktorisasi

Mayoritas dari pendekatan terhadap masalah SLAM yang lain adalah dengan memperkirakan terhadap peta dan robot *pose*.

$$p(s_t, \theta | z^t, u^t, n^t)$$

EKFSLAM sedikit berbeda, FastSLAM memperkirakan terhadap peta dan robot *path*.

$$p(s^t, \theta | z^t, u^t, n^t)$$

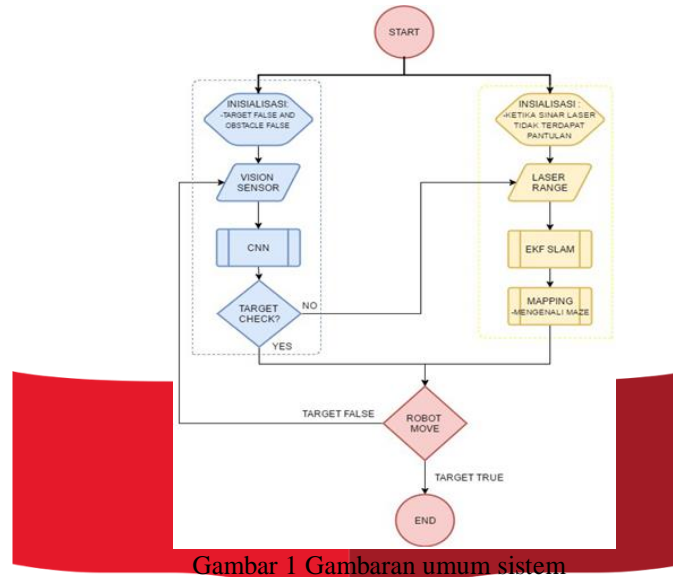
Independensi bersyarat memiliki akibat yang penting dalam EKFSLAM. Jika diketahui jalur robot, sebuah pengamatan dari 1 *landmark* tidak akan menyediakan informasi apa pun tentang posisi dari *landmark* lainnya. Ini artinya, perkiraan EKFSLAM diatas dapat difaktorisasi ke dalam bentuk yang lebih sederhana [1]:

$$p(s^t, \theta | z^t, u^t, n^t) = \underbrace{p(s^t | z^t, u^t, n^t)}_{\text{path estimator}} \underbrace{\prod_{n=1}^N p(\theta_n | s^t, z^t, u^t, n^t)}_{\text{landmark estimator}}$$

### 3. Analisis dan Perancangan Sistem

#### 3.1. Perancangan Sistem Keseluruhan

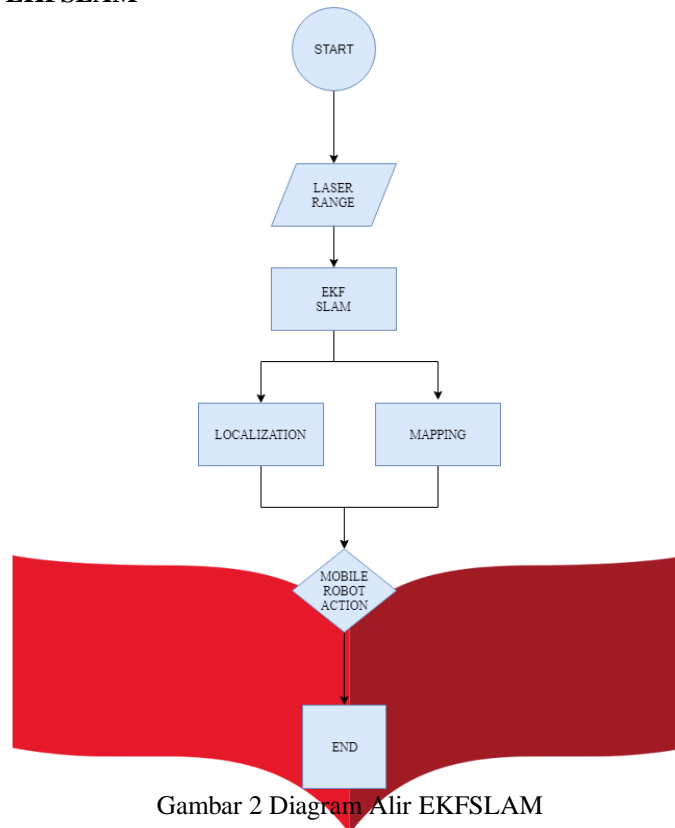
Implementasi EKFSLAM pada robot turtlebot ini merupakan penelitian Tugas Akhir yang bertujuan untuk menganalisis keakuratan dari algoritme EKFSLAM



Gambar 1 Gambaran umum sistem

Pada gambaran umum sistem diatas menunjukkan bahwa Mobile Robot Navigation memulai mengenali suatu lingkungan yang belum dikenali dengan inisialisasi yaitu jika Target = False And Obstacle = False artinya jika tidak ada target dan Obstacle maka robot mulai bergerak mengaktifkan vision sensor berupa kamera dan pada saat bersamaan laser range akan aktif sebagai tanda mapping sudah berjalan. Lalu diproses pada algoritma Convolutional Neural Network(CNN) untuk mengklasifikasi objek berupa Target atau Obstacle jika, ditemukan Obstacle robot akan menghindari dengan sudut konstan tertentu dan menggunakan laser sebagai tanda mapping telah berubah akibat obstacle, dan menggunakan metode Extended Kalman Filter (EKF) sampai ke sistem kendali pada mobile robot. Output dari EKF ini adalah membuat suatu gambaran hasil dari mapping sehingga pada saat bernavigasi bisa meminimalisir tabrakan pada saat lintasan berbelok. Bila Target ditemukan maka, robot akan berhenti.

### 3. 2. Perancangan EKFSLAM



Gambar 2 Diagram Alir EKFSLAM

Pada diagram alir sistem diatas menunjukkan bahwa Mobile Robot Navigation disertakan laser range yang bertujuan untuk mengenali lingkungan yang belum pernah diketahui. Kemudian dengan menggunakan algoritma Extended Kalman filter untuk melakukan Localization dan Mapping lokasi yang tidak dikenali. Robot akan memuat instruksi STOP saat robot kembali ke titik awal.

## 4. Pengujian Sistem

### 4. 1. Hasil Pengujian

Hasil pengujian dibawah ini merupakan hasil pemetaan dengan tipe gambar PGM yang diubah menjadi data JPEG. Hasil yang didapatkan ialah peta hasil pemetaan lingkungan buatan dengan garis hitam sebagai dinding, warna putih merupakan jalur yang dapat dilalui, warna abu-abu sebagai daerah yang belum diketahui. Hasil proses lokalisasi dapat dilihat melalui RViz. Proses pergerakan berdasarkan grid yang berukuran 1m x 1m. Setiap partikel menghasilkan proses pembentukan peta dari peta berukuran kecil hingga peta secara lengkap. Proses pembentukan didapat dari robot yang telah melakukan pengamatan didaerah tersebut, terlihat dari titik merah yang juga ikut bergerak seiring dengan pembentukan peta. Titik merah ini juga menggambarkan proses lokalisasi terhadap peta. Proses pembuatan peta memakan waktu yang berbeda sesuai dengan jumlah partikel yang digunakan. Berikut merupakan data hasil, dengan melakukan pengujian sebanyak enam kali. Jumlah partikel yang dimasukkan adalah 20, 40, 60, 80, 100, dan 120.

### 4. 2. Pengujian 20 partikel

Pengujian dengan 20 partikel

Pada awalnya saat robot berada di posisi awal yaitu di (0,0) atau di tengah peta partikel yang merepresentasikan adanya peluang robot belum terlihat lalu ketika robot berjalan mengikuti dinding sebelah kiri dan menyusuri semua peta partikel akan berkumpul dan mulai menggambarkan posisi robot yang sebenarnya. Posisi robot di dalam peta hanya dapat dilihat melalui RViz sedangkan hasil pemetaan

dapat disimpan dalam bentuk gambar bertipe PGM. Sistem navigasi menggunakan left wall following akan lebih terlihat ketika robot menghadapi halangan di depannya. Robot akan mendeteksi keberadaan dinding sebelah kiri, jika tidak ada maka robot akan berbelok ke kiri. Namun jika terdapat dinding disebelah kiri dan depan robot maka robot akan berbelok ke kanan. Proses pengambilan keputusan ini dapat memakan waktu dan terkadang dapat berdampak pada akurasi peta yang dihasilkan. Proses pemetaan dan lokalisasi akan berhenti jika mendapat masukan dari user. Waktu simulasi yang dibutuhkan untuk 20 partikel yaitu 30 menit.

#### **4. 3. Pengujian 40 partikel**

Pada awalnya saat robot berada di posisi awal yaitu di (0,0) atau di tengah peta partikel yang merepresentasikan adanya peluang robot belum terlihat lalu ketika robot berjalan mengikuti dinding sebelah kiri dan menyusuri semua peta partikel akan berkumpul dan mulai menggambarkan posisi robot yang sebenarnya. Posisi robot di dalam peta hanya dapat dilihat melalui RViz sedangkan hasil pemetaan dapat disimpan dalam bentuk gambar bertipe PGM. Sistem navigasi menggunakan left wall following akan lebih terlihat ketika robot menghadapi halangan di depannya. Robot akan mendeteksi keberadaan dinding sebelah kiri, jika tidak ada maka robot akan berbelok ke kiri. Namun jika terdapat dinding disebelah kiri dan depan robot maka robot akan berbelok ke kanan. Proses pengambilan keputusan ini dapat memakan waktu dan terkadang dapat berdampak pada akurasi peta yang dihasilkan. Proses pemetaan dan lokalisasi akan berhenti jika mendapat masukan dari user. Waktu simulasi yang dibutuhkan untuk 40 partikel yaitu 50 menit.

#### **4. 4. Pengujian 60 partikel**

Pada awalnya saat robot berada di posisi awal yaitu di (0,0) atau di tengah peta partikel yang merepresentasikan adanya peluang robot belum terlihat lalu ketika robot berjalan mengikuti dinding sebelah kiri dan menyusuri semua peta partikel akan berkumpul dan mulai menggambarkan posisi robot yang sebenarnya. Posisi robot di dalam peta hanya dapat dilihat melalui RViz sedangkan hasil pemetaan dapat disimpan dalam bentuk gambar bertipe PGM. Sistem navigasi menggunakan left wall following akan lebih terlihat ketika robot menghadapi halangan di depannya. Robot akan mendeteksi keberadaan dinding sebelah kiri, jika tidak ada maka robot akan berbelok ke kiri. Namun jika terdapat dinding disebelah kiri dan depan robot maka robot akan berbelok ke kanan. Proses pengambilan keputusan ini dapat memakan waktu dan terkadang dapat berdampak pada akurasi peta yang dihasilkan. Proses pemetaan dan lokalisasi akan berhenti jika mendapat masukan dari user. Waktu simulasi yang dibutuhkan untuk 60 partikel yaitu 1 jam 20 menit.

#### **5. Kesimpulan**

Dari hasil pengujian dan Analisa yang telah dilakukan, maka dapat disimpulkan bahwa akurasi tertinggi dari algoritme EKFSLAM dan peta didapatkan saat pengujian dengan jumlah partikel yang digunakan yaitu 60, semakin banyak partikel yang digunakan maka peta yang dihasilkan semakin presisi dan juga semakin tinggi partikel yang digunakan maka semakin tinggi akurasi algoritme EKFSLAM, dan hubungan jumlah partikel dan waktu komputasi berbanding lurus, karena semakin banyak partikel yang digunakan membutuhkan waktu komputasi yang semakin tinggi.

**Daftar Pustaka:**

- [1] M. W. M. Gamini Dissanayake, P. Newman S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, 2001.
- [2] A. R. Khairuddin, M. S. Talib, and H. Haron, "Review on simultaneous localization and mapping (SLAM)," *Proc. - 5th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2015*, no. November, pp. 85–90, 2016.
- [3] S. Prayoga, K. R. Sumantri, A. B. K. A, and P. N. Batam, "Ekf slam menggunakan lidar," pp. 135–140.
- [4] M. Calonder, "EKF SLAM vs. FastSLAM – A Comparison.pdf," pp. 1–5.
- [5] B. Cinaz and H. Kenn, "Head SLAM - Simultaneous localization and mapping with head-mounted inertial and laser range sensors," *Proc. - Int. Symp. Wearable Comput. ISWC*, pp. 3–10, 2008.
- [6] J. Guivant, E. Nebot, and S. Baiker, "High accuracy navigation using laser range sensors in outdoor applications," *Proc. 2000 ICRA. Millenn. Conf. IEEE Int. Conf. Robot. Autom. Symp. Proc. (Cat. No.00CH37065)*, vol. 4, no. April 2000, pp. 3817–3822, 2000.
- [7] J. Z. Sasiadek, a. Monjazeb, and D. Neculescu, "Navigation of an autonomous mobile robot using EKF-SLAM and FastSLAM," *2008 16th Mediterr. Conf. Control Autom.*, no. 1, pp. 517–522, 2008.
- [8] A. Koubâa et al., "Turtlebot at Office: A Service-Oriented Software Architecture for Personal Assistant Robots Using ROS," *Proc. - 2016 Int. Conf. Auton. Robot Syst. Compet. ICARSC 2016*, pp. 270–276, 2016.
- [9] E. H. Maciel, R. V. B. Henriques, and W. F. Lages, "Control of a Biped Robot Using the Robot Operating System," *2014 Jt. Conf. Robot. SBR-LARS Robot. Symp. Rob.*, pp. 247–252, 2014

