



TESIS - EE185401

**DETEKSI DAN RINTANGAN
MENGGUNAKAN PROBABILISTIC DATA
ASSOCIATION FILTER PADA LINGKUNGAN
INDOOR MOBILE ROBOT**

MUHAMMAD AZRIEL RIZQIFADHIILAH
NRP 6022221047

DOSEN PEMBIMBING
Dr. Trihastuti Agustinah, S.T., M.T
Prof. Dr. Ir. Achmad Jazidie, M.Eng.

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK SISTEM PENGATURAN
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER



TESIS - EE185401

**DETEKSI DAN TRACKING RINTANAN
MENGGUNAKAN PROBABILISTIC DATA
ASSOCIATION FILTER PADA LINGKUNGAN
INDOOR MOBILE ROBOT**

MUHAMMAD AZRIEL RIZQIFADHIILAH
NRP 6022221047

DOSEN PEMBIMBING
Dr. Trihastuti Agustinah, S.T., M.T
Prof. Dr. Ir. Achmad Jazidie, M.Eng.

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK SISTEM PENGATURAN
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2022

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Teknik (MT)

di

Institut Teknologi Sepuluh Nopember

Oleh:

MUHAMMAD AZRIEL RIZQIFADHIILAH

NRP: 6022221047

Tanggal Ujian: 12 July 2023

Periode Wisuda: September 2023

Disetujui oleh:

Pembimbing:

1. Dr. Trihastuti Agustinah, S.T., M.T
NIP: 196898121994032001

.....

2. Prof. Dr. Ir. Achmad Jazidie, M.Eng
NIP: 195902191986101001

.....

Pengaji:

1. Prof. Ir. Abdullah Alkaf, M.Sc, Ph.D
NIP: 195501231980031002

.....

2. Dr. Ir. Ari Santoso, DEA
NIP: 196602181991021001

.....

3. Yurid Eka Nugraha, S.T., M.Eng., Ph.D
NPP: 2023199511059

.....

Kepala Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas

Dedet Candra Riawan, ST., M.Eng., Ph.D.

NIP: 197311192000031001

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul "**Deteksi dan Tracking Ringtangan Menggunakan Probabilistic Data Association Filter pada Lingkungan Indoor Mobile Robot**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 12 Juli 2022



Muhammad Azriel R
NRP. 6022221047

Halaman ini sengaja dikosongkan

DETEKSI DAN TRACKING RINTANGAN MENGGUNAKAN PROBABILISTIC DATA ASSOCIATION FILTER PADA LINGKUNGAN INDOOR MOBILE ROBOT

Nama mahasiswa : Muhammad Azriel Rizqifadhiilah
NRP : 6022221047
Pembimbing : 1. Dr. Trihastuti Agustinah, S.T, M.T
 2. Prof. Dr. Ir. Achmad Jazidie, M.Eng.

ABSTRAK

Penelitian ini menyoroti pengembangan dan penerapan teknologi robotika canggih pada *differential drive* mobile robot, sebuah jenis mobil robot yang saat ini menjadi subjek banyak penelitian karena berbagai aplikasinya dalam memfasilitasi kegiatan manusia. Kunci teknologi yang dikaji adalah *Simultaneous Localization and Mapping* (SLAM), yang merupakan aspek esensial dalam robotika yang memungkinkan robot untuk menginterpretasi dan menavigasi lingkungannya. Metode *Euclidean Clustering*, yang terbukti efektif dalam menangani data berdimensi tinggi yang dihasilkan oleh sensor LIDAR, juga digunakan dalam penelitian ini. Selanjutnya, penelitian ini mengeksplorasi kombinasi *Probabilistic Data Association Filter* (PDAF) dan *Ensemble Kalman Filter* (EnKF) untuk melacak dan memprediksi trajektori penghalang. Hasil penelitian menunjukkan bahwa metode SLAM berhasil dalam mapping lingkungan, sedangkan *Euclidean Clustering* berhasil mendeteksi dan mengcluster objek seperti lingkaran, persegi, dan dinding dengan menghitung kesalahan menggunakan metrik *Mean Absolute Error* (MAE) dalam tiga skenario berbeda. Untuk skenario rintangan bergerak, MAE lingkaran adalah 0.17, MAE persegi adalah 26.04. Sementara itu, pada skenario rintangan multiobjek, MAE lingkaran adalah 0.19, MAE persegi adalah 38.44. Akhirnya, untuk skenario rintangan hybrid, MAE lingkaran adalah 0.19, MAE persegi adalah 31.10. Modifikasi PDAF dengan EnKF berhasil mengurangi kesalahan dalam pelacakan, dengan rata-rata *Root Mean Square Relative Error* (RMRSE) 0.03 untuk skenario rintangan bergerak, 0.04 untuk skenario multi objek bergerak, dan 0.04 untuk skenario rintangan hybrid. Hasil ini menunjukkan potensi signifikan teknologi ini dalam mengembangkan robotika *diferensial drive* yang lebih efisien dan efektif.

Kata kunci: *Differential drive* mobile robot, SLAM, *Euclidean Clustering*, PDAF, Deteksi objek, EnKF

Halaman ini sengaja dikosongkan

OBSTACLE DETECTION AND TRACKING USING PROBABILISTIC DATA ASSOCIATION FILTER IN INDOOR MOBILE ROBOT ENVIRONMENTS

By : Muhammad Azriel Rizqifadhiilah
Student Identity Number : 6022221047
Supervisors : 1. Dr. Trihastuti Agustinah, S.T, M.T
 2. Prof. Dr. Achmad Jazidie, M.Eng

ABSTRACT

This research emphasizes the development and application of advanced robotics technology in differential drive mobile robots, a type of robotic vehicle currently undergoing extensive investigation due to its diverse applications in facilitating human activities. The central technology under examination is Simultaneous Localization and Mapping (SLAM), an essential aspect in robotics that allows robots to interpret and navigate their surroundings. The Euclidean Clustering method, proven effective in managing high-dimensional data produced by LIDAR sensors, is also utilized in this study. Furthermore, this research explores the combination of the Probabilistic Data Association Filter (PDAF) and Ensemble Kalman Filter (EnKF) for tracking and predicting the trajectory of obstacles. The research outcomes indicate that the SLAM method successfully maps the environment, whereas Euclidean Clustering succeeds in detecting and Clustering objects such as circles, squares, and walls by calculating errors using Mean Absolute Error (MAE) metrics across three different scenarios. For the moving obstacle scenario, the MAE of the circle is 0.17, the square is 26.04. Meanwhile, in the multi-object obstacle scenario, the MAE of the circle is 0.19, the square is 38.44. Finally, for the hybrid obstacle scenario, the MAE of the circle is 0.19, the square is 31.10. The modification of the PDAF with EnKF successfully reduced *tracking* errors, with an average Root Mean Square Relative Error (RMRSE) of 0.03 for the moving obstacle scenario, 0.04 for the multi-object moving scenario, and 0.04 for the hybrid obstacle scenario. These results demonstrate the significant potential of these technologies in developing more efficient and effective differential drive robotics.

Key words: Differential drive Mobile robot, SLAM, Euclidean *Clustering*, PDAF, Obstacle detection, EnKF

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji Syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat yang telah dilimpahkan-Nya sehingga penulis mampu menyelesaikan tesis dengan judul **“Deteksi dan Tracking Ringtangan Menggunakan Probabilistic Data Association Filter pada Lingkungan Indoor Mobile Robot”** dengan tepat waktu.

Tesis ini disusun sebagai salah satu persyaratan untuk menyelesaikan pendidikan S2 Bidang Studi Teknik Sistem Pengaturan, Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember. atas selesainya penyusunan tesis ini, penulis mengucapkan terima kasih kepada:

1. Allah SWT atas limpahan rahmat, karunia dan petunjuk-Nya.
2. Keluarga Penulis terutama bapak Herri Imanto, ibu Ratna Sari Triwidiastuti, Kakak Annisa Nabilah Rizqaviany yang selalu memberikan dukungan secara materiil maupun non-materiil.
3. Ibu Dr. Trihastuti Agustinah, S.T, M.T dan Bapak Prof. Dr. Ir. Achmad Jazidie, M.Eng selaku dosen pembimbing yang telah memberikan arahan dan petunjuk dalam menyelesaikan laporan tesis.
4. Bapak dan Ibu Dosen atau Guru saya yang telah memberikan ilmu dan pelajarannya selama menjalani masa perkuliahan.
5. Teman-teman bermain yang selalu menemani dan memberikan kebahagiaan sejak awal masa perkuliahan hingga saat ini.
6. Teman-teman e-58 yang saling berjuang dan memberi semangat satu sama lain dalam penggerjaan tesis.
7. Seluruh teman-teman saya yang berasal dari berbagai departemen di ITS yang telah memberikan banyak pengalaman selama mengenyam bangku perkuliahan ini.

Penulis telah berusaha maksimal dalam penyusunan tugas akhir ini, namun besar harapan penulis akan saran dari pembaca agar nantinya laporan tesis ini dapat menjadi referensi yang berguna bagi penelitian selanjutnya.

Surabaya, 12 Juli 2023

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN TESIS.....	iii
PERNYATAAN KEASLIAN TESIS	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	5
1.3 Tujuan	5
1.4 Batasan Masalah	6
1.5 Kontribusi	6
BAB 2 KAJIAN PUSTAKA.....	7
2.1 Kajian Penelitian Terkait	7
2.1.1 Modified Headed Social Force Model Based on Hybrid Velocity Obstacles for Mobile Robot to Avoid Disturbed Groups of Pedestrians [16]	7
2.1.2 Collision Avoidance of Multi Modal Moving Objects for Mobile Robot Using Hybrid Velocity Obstacles [17]	15
2.1.3 Real Time Obstacle Prediction Using Neural Network Based on Extended Kalman Filter for Robot Path Planning [18].....	22
2.2 Teori Dasar.....	28
2.2.1 Ensemble Kalman Filter.....	28
2.2.2 PDAF.....	30
BAB 3 DETEKSI DAN TRACKING RINTANGAN MENGGUNAKAN PDAF	33

3.1	Model Objek Rintangan Dinamis.....	33
3.2	<i>Trajectory Tracking</i> Mobile Robot	36
3.3	<i>Simultaneous Localization and Mapping (SLAM)</i>	37
3.4	<i>Euclidean Clustering</i>	39
3.5	Filtering Data PDAF Dengan Ensemble Kalman Filter.....	41
3.6	Analisa Data	47
3.7	Identifikasi Lidar	48
3.8	Skema Sistem Keseluruhan.....	49
3.9	Hipotesa Penelitian.....	49
3.10	Kriteria Pengujian.....	50
	BAB 4 HASIL DAN PEMBAHASAN	51
4.1	Estimasi, Prediksi dan <i>Tracking</i> Terhadap Rintangan Bergerak.....	51
4.1.1	Hasil Pengujian Algoritma SLAM	51
4.1.2	Hasil Pengujian Algoritma <i>Euclidean Clustering</i>	52
4.1.3	Hasil Pengujian Estimasi Menggunakan PDAF	55
4.2	Estimasi, prediksi dan <i>Tracking</i> Terhadap Rintangan Bergerak dan Berinteraksi Dengan Benda Statis.....	59
4.2.1	Hasil Pengujian Algoritma SLAM	59
4.2.2	Hasil Pengujian Algoritma <i>Euclidean Clustering</i>	60
4.2.3	Hasil Pengujian Estimasi Menggunakan PDAF	63
4.3	Estimasi, Prediksi dan <i>Tracking</i> Terhadap Rintangan Bergerak Multi Modalitas Serta Kelompok Pedestrian	66
4.3.1	Hasil Pengujian Algoritma SLAM	67
4.3.2	Hasil Pengujian Algoritma <i>Euclidean Clustering</i>	67
4.3.3	Hasil Pengujian Estimasi Menggunakan PDAF	71
	BAB 5 KESIMPULAN DAN SARAN	75
5.1	Kesimpulan.....	75
5.2	Saran.....	76
	DAFTAR PUSTAKA.....	77
	LAMPIRAN	79

DAFTAR GAMBAR

Gambar 2. 1. Kemungkinan Pergerakan Manusia Tunggal.....	7
Gambar 2. 2. Interaksi Antara Manusia dan Objek Statis.....	8
Gambar 2. 3. Kinematika Mobile Robot.....	9
Gambar 2. 4. Struktur Sistem Navigasi.....	10
Gambar 2. 5. Konfigurasi path planning menggunakan PRM.....	10
Gambar 2. 6. Perbandingan lintasan referensi dan aktual dan performa keselamatan dan kenyamanan sekenario 1	14
Gambar 2. 7. Perbandingan lintasan referensi dan aktual dan performa keselamatan dan kenyamanan sekenario 2	15
Gambar 2. 8. Interaksi antara mobile robot dengan objek yang bergerak dalam ruangan	16
Gambar 2. 9. Jalur perpindahan objek bentuk cloverleaf.....	17
Gambar 2. 10. Struktur HVO sistem penghindaran rintangan untuk trajektori traking pada mobile robot dengan dua roda differential-steering..	17
Gambar 2. 11. Konfigurasi mobile robot dua roda differential-steering.....	18
Gambar 2. 12. HVO berdasarkan penghindaran rintangan	20
Gambar 2. 13. Skenario 1 menggunakan algoritma VO trajektori robot dan agen-agen lain.....	21
Gambar 2. 14. Definisi permasalahan	23
Gambar 2. 15. Performa kalman filter dalam penelitian tersebut	27
Gambar 2. 16. Gambar kanan path planing dan gambar kiri ketidakpastian prediksi.....	27
Gambar 2. 17. Uji coba dengan algoritma penghindaran rintangan	27
Gambar 2. 18. Perbandingan performa prediksi dengan penghindaran rintangan skenario 3	28
Gambar 2. 19. Flowchart sistem <i>tracking</i> dengan PDAF	32
Gambar 3.1. Ruang interaksi antar manusia dalam F-Formation	34
Gambar 3.2. Objek bergerak dengan lintasan oval	35
Gambar 3.3. Objek bergerak dengan lintasan Lissajous	35
Gambar 3.4. Objek bergerak dengan lintasan Cloverleaf [21].....	36

Gambar 3.5. Langkah umum SLAM dilakukan pada differential drive mobile robot	39
Gambar 3.6 Perhitungan untuk mencari titik centroid dari objek setiap objek	42
Gambar 3.7. Flowchart tracking sistem PDAF termodifikasi EnKF.....	43
Gambar 3.8. Algoritma EnKF pada PDAF.....	44
Gambar 3.9. Flowchart penghindaran rintangan dengan prediksi dan estimasi state.	50
Gambar 4. 1 SLAM dengan 10 scan dalam frame data.....	51
Gambar 4. 2 SLAM secara keseluruhan scan dalam frame data	52
Gambar 4. 3 Pengujian frame ke-8 skenario rintangan bergerak	53
Gambar 4. 4 Pengujian frame ke-30 skenario rintangan bergerak	54
Gambar 4. 5 Pengujian frame ke-48 sekanario rintangan bergerak	54
Gambar 4. 6 Pengujian frame ke-130 skenario rintangan bergerak	54
Gambar 4. 7 Kesalahan ukuran pada objek lingkaran skenario rintangan bergerak	55
Gambar 4. 8 Kesalahan ukuran pada objek persegi skenario rintangan bergerak .	55
Gambar 4. 9 Prediksi, estimasi dan tracking objek dinamis sekanrio rintangan bergerak.....	57
Gambar 4. 10 Error posisi dalam sumbu x dan y pada skenario rintangan bergerak	57
Gambar 4. 11 Rata-rata error estimasi posisi skenario rintangan bergerak.....	58
Gambar 4. 12 Prediksi posisi objek ke-1 skenario rintangan bergerak	58
Gambar 4. 13 SLAM dengan 10 scan dalam frame data.....	58
Gambar 4. 14 SLAM secara keseluruhan scan dalam frame data	59
Gambar 4. 15 Pengujian frame ke-4 skenario rintangan multiobjek.....	61
Gambar 4. 16 Pengujian frame ke-30 skenario rintangan multiobjek.....	61
Gambar 4. 17 Pengujian frame ke-66 sekanario rintangan multiobjek	62
Gambar 4. 18 Pengujian frame ke-126 skenario rintangan multiobjek	62
Gambar 4. 19 Kesalahan ukuran pada objek lingkaran skenario rintangan multiobjek.....	62
Gambar 4. 20 Kesalahan ukuran pada objek persegi skenario rintangan multiobjek	63

Gambar 4. 21 Prediksi, estimasi dan tracking objek dinamis sekanrio rintangan multiobjek.....	65
Gambar 4. 22 Error posisi dalam sumbu x dan y pada skenario rintangan multiobjek	65
Gambar 4. 23 Rata-rata error estimasi posisi skenario rintangan multiobjek	66
Gambar 4. 24 Prediksi posisi objek ke-1 skenario rintangan multiobjek	66
Gambar 4. 25 SLAM dengan 10 scan dalam frame data	69
Gambar 4. 26 SLAM secara keseluruhan scan dalam frame data.....	69
Gambar 4. 27 Pengujian frame ke-4 skenario rintangan hybrid	69
Gambar 4. 28 Pengujian frame ke-68 skenario rintangan hybrid	70
Gambar 4. 29 Pengujian frame ke-88 sekanario rintangan hybrid.....	70
Gambar 4. 30 Pengujian frame ke-120 skenario rintangan hybrid	70
Gambar 4. 31 Kesalahan ukuran pada objek lingkaran skenario rintangan hybrid	71
Gambar 4. 32 Kesalahan ukuran pada objek persegi skenario rintangan hybrid..	71
Gambar 4. 33 Prediksi, estimasi dan tracking objek dinamis sekanrio rintangan hybrid	73
Gambar 4. 34 Error posisi dalam sumbu x dan y pada skenario rintangan hybrid	73
Gambar 4. 35 Rata-rata error estimasi posisi skenario rintangan hybrid.....	74
Gambar 4. 36 Prediksi posisi objek ke-1 skenario rintangan hybrid	74
Gambar 4. 37 Prediksi posisi objek ke-2 skenario rintangan hybrid	74

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2. 1. Performa metode HVO untuk penghindaran rintangan komparasi APF,VFH,VO dan HRVO	22
Tabel 3.1. Parameter lingkungan kerja robot.....	50
Tabel 4. 1 Parameter SLAM skenario 1.....	51
Tabel 4. 2 Hasil perhitungan kesalahan skenario rintangan bergerak.....	53
Tabel 4. 3 Posisi awal objek terdeteksi skenario rintangan bergerak	53
Tabel 4. 4 RMSRE skenario rintangan bergerak	57
Tabel 4. 5 Parameter SLAM skenario2.....	59
Tabel 4. 6 Hasil perhitungan kesalahan skenario rintangan hybrid	63
Tabel 4. 7 Posisi awal objek terdeteksi skenario rintangan multi objek	63
Tabel 4. 8 RMSRE skenario rintangan multiobjek	66
Tabel 4. 9 Parameter SLAM skenario 3.....	66
Tabel 4. 10 Hasil perhitungan kesalahan skenario rintangan hybrid	68
Tabel 4. 11 Posisi awal objek terdeteksi skenario rintangan hybrid	72
Tabel 4. 12 RMSRE skenario rintangan hybrid	73

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pada era sekarang ini banyak di lakukan penelitian dan pengembangan teknologi *autonomous vehicle* mulai dari kendaraan laut, udara, hingga darat. Penelitian tentang mobil robot khususnya jenis *differential drive* merupakan salah satu permasalahan yang sekarang sedang banyak diteliti, karena dari perkembangan teknologi *differential drive* banyak membantu kegiatan-kegiatan manusia mulai dari mengerjakan pekerjaan ringan hingga berat, agar *differential drive* dapat berjalan di tanamkan teknologi yang menjadi faktor utama yang perlu di perhatikan [1].

Dengan menggunakan kecerdasan buatan robot *differential drive* dapat beroperasi dalam pengaturan yang berbeda. Seperti mengikuti jalur yang diberikan dan navigasi itu sendiri untuk menghindari rintangan dan pindah ke titik tujuan adalah masalah utama [2]. Manusia memiliki kapasitas yang luar biasa untuk tugas-tugas fisik dan mental tanpa harus bergantung pada pengukuran atau perhitungan eksplisit. Contoh tugas sehari-hari adalah memarkir mobil, mengemudi di medan yang berbeda, menghindari rintangan statis dan dinamis. Bagi manusia, kemampuan untuk bernavigasi adalah yang terbaik, untuk robot; tugas semacam ini kompleks dan menantang. Untuk mengenali lintasan; rintangan dan target, robot harus mempertimbangkan informasi dari lingkungan kerja yang diperoleh dari sensor atau sistem penglihatannya, hasil tangkapan sensor ini akan di proses oleh algoritma yang sudah di bangun sehingga menghasilkan gerak oleh *differential drive* yang mengantarkan *differential drive* arah tujuannya [3].

SLAM adalah teknologi kunci dalam bidang robotika yang memungkinkan robot, seperti *differential drive* mobile robot, untuk memahami dan menavigasi lingkungannya [4]. Metode ini memungkinkan robot untuk membangun atau memperbarui peta lingkungan yang belum dipetakan atau tidak diketahui sebelumnya secara langsung dalam simulasi ketika melacak lokasi relatifnya dalam peta tersebut. SLAM menyelesaikan dua masalah sekaligus, yaitu pemetaan

lingkungan (*mapping*) dan menentukan posisi robot dalam lingkungan tersebut (*localization*).

Differential drive mobile robot adalah jenis robot yang menggunakan dua roda yang bergerak secara independen untuk manuver [5]. Teknologi ini memungkinkan robot untuk melakukan gerakan yang sangat presisi, membuatnya ideal untuk aplikasi di mana navigasi yang akurat diperlukan. SLAM digunakan dalam konteks ini untuk memungkinkan robot untuk menavigasi dengan akurat di lingkungannya, sekaligus membangun peta yang akurat dari lingkungan tersebut.

Dibandingkan dengan metode lain, SLAM menawarkan beberapa keunggulan. Misalnya, metode seperti *Extended Kalman Filter* (EKF) mengharuskan model lingkungan dan gerakan robot diketahui sebelumnya dan masalah ini sangat sulit untuk dipenuhi dalam praktik. Di sisi lain, metode seperti Particle Filter atau FastSLAM dapat menangani lingkungan dan gerakan yang tidak pasti dengan lebih baik, tetapi mereka bisa menjadi berat secara komputasi, terutama dalam lingkungan yang besar dan kompleks [6]. Sebaliknya, SLAM dapat bekerja dengan baik dalam lingkungan yang tidak pasti dan besar, sambil menjaga biaya komputasi pada level yang dapat diterima.

Euclidean Clustering adalah metode yang digunakan dalam pengolahan data LIDAR untuk mengidentifikasi objek dalam data *point cloud* [7]. Metode ini memanfaatkan konsep jarak *Euclidean* yaitu, jarak "garis lurus" antara dua titik dalam ruang untuk mengelompokkan titik-titik yang berdekatan menjadi satu "*cluster*". Dalam konteks penelitian ini, *differential drive mobile* robot, *Euclidean Clustering* dapat digunakan untuk mengidentifikasi dan melacak objek dalam lingkungan robot, seperti penghalang atau target.

Dalam konteks penggunaan data LIDAR, *Euclidean Clustering* adalah metode yang efektif karena dapat menangani data berorde tinggi yang dihasilkan oleh sensor LIDAR [8]. Selain itu, metode ini juga cukup efisien secara komputasi, yang memungkinkan digunakannya dalam aplikasi penelitian. Bagaimanapun, *Euclidean Clustering* bergantung pada penentuan jarak *threshold* yang tepat untuk mengelompokkan titik, dan mungkin perlu disesuaikan tergantung pada aplikasi dan lingkungan spesifik.

Pendeteksian objek bergerak menggunakan data LIDAR adalah salah satu tantangan besar dalam bidang robotika. *Euclidean Clustering* dapat digunakan untuk identifikasi awal objek, tetapi untuk mendeteksi apakah objek tersebut bergerak atau tidak, diperlukan metode tambahan. Biasanya, ini melibatkan perbandingan data LIDAR dari beberapa titik waktu berbeda untuk melihat perubahan dalam posisi objek.

Dibandingkan dengan metode lain, seperti *K-Nearest Neighbors* (k-NN), *Euclidean Clustering* memiliki beberapa keunggulan. K-NN, misalnya, memerlukan penentuan jumlah tetangga (K) yang harus dipertimbangkan. Penentuan nilai K yang tidak tepat bisa menghasilkan hasil yang kurang akurat. Selain itu, K-NN bisa menjadi mahal secara komputasi, terutama jika jumlah data yang harus diproses sangat besar. Di sisi lain, *Euclidean Clustering* tidak memerlukan penentuan jumlah *cluster* sebelumnya, dan bisa mengelompokkan titik berdasarkan kedekatan jarak mereka, yang bisa lebih intuitif dan mudah diinterpretasikan dalam konteks robotika [9].

PDAF adalah metode yang digunakan dalam *tracking* objek, seperti untuk melacak objek pada *differential drive mobile robot* [10]. Metode ini berfungsi dengan menghitung probabilitas gabungan dari semua asosiasi pengukuran-prediksi yang mungkin, berdasarkan pada pengetahuan sebelumnya tentang sistem dan pengukuran yang diterima. PDAF dibuat untuk menangani situasi di mana beberapa pengukuran mungkin berasal dari target yang sama yang sering terjadi dalam situasi dengan banyak objek bergerak, seperti lingkungan perkantoran atau kampus.

Dalam konteks *tracking* penghalang, PDAF juga bisa digunakan untuk estimasi *trajektori* [11]. Setelah penghalang terdeteksi dan melacaknya sepanjang waktu, mendapatkan rangkaian data posisi sepanjang waktu yang bisa digunakan untuk memprediksi *trajektori* penghalang di masa mendatang. Ini sangat penting dalam konteks penelitian ini, di mana prediksi tentang pergerakan penghalang di masa mendatang dapat digunakan untuk merencanakan jalur yang aman dan efisien.

Bandingkan dengan metode lain, seperti Kalman Filter atau Particle Filter, PDAF memiliki beberapa keunggulan. Kalman Filter hanya efektif dalam situasi

di mana hanya ada satu penghalang yang perlu dilacak dan lingkungan sekitarnya cukup sederhana. Di sisi lain, Particle Filter dapat menghandle kasus dengan banyak penghalang dan lingkungan yang lebih kompleks, tetapi bisa menjadi berat secara komputasi, terutama dalam kasus dengan banyak penghalang. Sebaliknya, PDAF mampu menangani situasi dengan banyak penghalang dengan biaya komputasi yang relatif rendah, membuatnya ideal untuk digunakan dalam situasi simulasi ataupun langsung pada lingkungan yang kompleks [12].

EnKF di sisi lain, adalah variasi dari Kalman Filter tradisional yang dirancang untuk mengatasi masalah non-linear dan non-Gaussian. EnKF menghasilkan "ensemble" atau kumpulan sampel dari state sistem, dan kemudian menggunakan sampel ini untuk mengestimasi state sistem dan keadaan ketidakpastian [13]. Dalam konteks *tracking* penghalang dan estimasi trajektori, EnKF dapat digunakan untuk menghasilkan perkiraan yang lebih akurat dan robust tentang posisi penghalang di masa mendatang.

Meskipun keduanya merupakan metode yang kuat untuk melacak penghalang dan memprediksi trajektori, ada perbedaan penting antara keduanya. PDAF lebih fokus pada penanganan asosiasi pengukuran yang ambigu dan bisa lebih efisien dalam kasus dengan banyak target. Di sisi lain, EnKF lebih mampu menangani non-linearitas dan non-Gaussianity dalam sistem dan pengukuran, yang bisa membuatnya lebih akurat dalam kasus di mana asumsi linear dan Gaussian dari Kalman Filter tradisional tidak berlaku.

Dalam beberapa tahun terakhir, ada peningkatan signifikan dalam penggunaan PDAF dan EnKF dalam penelitian pengejadian objek. Misalnya, dalam sebuah studi oleh [14], PDAF digunakan untuk melacak fitur dalam lingkungan yang dinamis untuk robot percepatan. Hasilnya menunjukkan bahwa PDAF mampu melacak fitur dengan akurat, bahkan di hadapan asosiasi pengukuran yang ambigu.

EnKF telah digunakan dalam berbagai penelitian untuk melacak dan memprediksi penghalang. Dalam studi oleh [15], EnKF digunakan untuk kombinasi estimasi state dan parameter dalam pengendalian sistem. Hasilnya menunjukkan bahwa EnKF mampu menghasilkan perkiraan yang lebih akurat dan

robust tentang state sistem dan ketidakpastian, bahkan di hadapan non-linearitas dan non-Gaussianity dalam sistem dan pengukuran.

Dalam penelitian ini, data *filtering* pada PDAF menggunakan EnKF untuk mengamati penghalang dan prediksi trajektori dikaji. Studi ini berfokus pada bagaimana kombinasi dalam data *filtering* PDAF menggunakan EnKF dapat meningkatkan estimasi dan pelacak trajektori dalam komputasi yang besar.

1.2 Rumusan Masalah

Tesis ini merumuskan permasalahan sebagai berikut:

1. SLAM adalah salah satu teknologi penting dalam *differential drive* yang membuat peta sekitar dan melacak robot secara bersamaan. Sensor yang sering dipakai adalah Lidar 2D, karena bisa mengukur jarak dengan tepat dalam ruang 2D. Namun, Lidar 2D punya tantangan dalam SLAM, seperti bagaimana menangani lingkungan dengan banyak objek bergerak atau berubah.
2. Mengenali objek adalah masalah lain dalam *differential drive*. *Euclidean Clustering* sering dipakai untuk mengelompokkan data dari lidar 2D menjadi kumpulan yang merepresentasikan objek berbeda. Tapi metode ini punya batasan, seperti penanganan objek yang berdekatan dan tampak sebagai satu kumpulan, atau objek yang bergerak atau berubah seiring waktu.
3. Melacak dan memprediksi pergerakan objek adalah kunci dalam *differential drive*, khususnya untuk navigasi dan perencanaan jalur. Teknik yang biasa digunakan adalah PDAF. Tapi, penggunaan PDAF punya tantangan, seperti menangani banyak objek untuk dilacak dan memastikan keakuratan estimasi dan prediksi ketika objek bergerak secara non-linear.

1.3 Tujuan

Penelitian ini bertujuan untuk meningkatkan akurasi dan efisiensi lokalisasi, deteksi, dan pelacakan objek dalam *differential drive* mobile robot. Ini mencakup pengembangan metode SLAM yang berbasis lidar dua dimensi untuk mapping dan lokalisasi robot terhadap perubahan lingkungan. *Euclidean Clustering* untuk deteksi dan prediksi objek berdasarkan data lidar 2D, terutama dalam kasus di mana objek sangat dekat atau berubah seiring waktu. Akhirnya, tujuan penelitian

ini adalah untuk prediksi, estimasi, dan pelacakan PDAF dalam melacak dan memprediksi pergerakan objek.

1.4 Batasan Masalah

Dalam penelitian ini terdapat batasan masalah untuk membatasi permasalahan yang muncul ketika penelitian dilakukan di antaranya,

1. Mobile robot *differential drive*.
2. Lingkungan kerja robot adalah indoor karena batas maximum pembacaan LIDAR
2. Rintangan terdiri dari pedestrian, kelompok manusia, dan benda statis.
3. Objek manusia dianggap lingkaran ketika terdeteksi.
4. Gerakan sosial manusia dinyatakan dalam gerak non-linear.
5. Rintangan diprediksi menggunakan teknik *Clustering* posisi, dan bentuknya.
6. Lokalisasi dan mapping dicoba dengan beberapa skenario untuk menguji algoritma SLAM
7. *Noise* pengukuran pada saat *tracking* menggunakan model *mutually independent gaussian white-noise* dengan zero-mean dan standar deviasi.
8. Menentukan maximum batas aman kecepatan rintangan dinamis.
8. Kecepatan objek rintangan 2 kali dari kecepatan mobile robot.
9. Evaluasi metode dilihat dari perbandingan kesalahan prediksi dan aktual *trajektori* rintangan bergerak.

1.5 Kontribusi

Penelitian ini diharapkan memberikan kontribusi signifikan dalam penelitian sebelumnya, khususnya pada *differential drive mobile robot*. Dengan meningkatkan efisiensi dan akurasi dari metode SLAM berbasis lidar 2D, teknik *Euclidean Clustering* untuk deteksi dan prediksi objek, serta perbaikan dalam estimasi dan pelacakan menggunakan PDAF dengan EnKF pada filter data, penelitian ini dapat membantu dalam menghadapi tantangan lokalisasi dan pemetaan, deteksi objek, serta prediksi dan *tracking* dalam lingkungan yang berubah atau kompleks. Selain itu, penelitian ini juga bisa memperkaya literatur akademis dan teknis dalam bidang SLAM, deteksi objek, dan pelacakan objek, serta memberikan dasar untuk penelitian lebih lanjut dalam bidang-bidang ini.

BAB 2

KAJIAN PUSTAKA

2.1 Kajian Penelitian Terkait

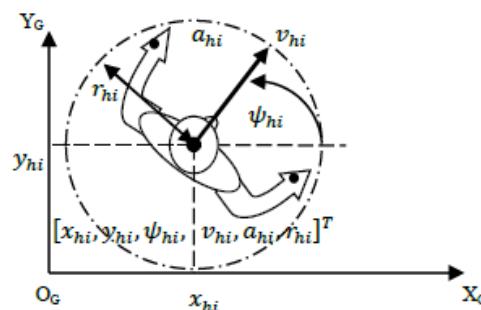
Kajian penelitian terkait memuat tentang hasil penelitian terdahulu yang telah dilakukan oleh penulis sendiri ataupun peneliti lain. Hal-hal yang mendasari ide dari penelitian yang akan dilakukan dijelaskan pada sub-bab ini.

2.1.1 Modified Headed Social Force Model Based on Hybrid Velocity Obstacles for Mobile Robot to Avoid Disturbed Groups of Pedestrians

[16]

Metode yang diusulkan penulis dalam makalah ini Headed modifikasi berbasis *Hybrid Velocity Obstacle Social Force Model* (HSFM modifikasi berbasis HVO) untuk menghindari gangguan kelompok pejalan kaki saat bernaligasi di ruang kerja yang kompleks dan padat. HVO digunakan untuk menghitung kecepatan linier dan sudut untuk menghindari rintangan dengan non-linier lintasan. Pengelompokan gangguan dari pejalan kaki dikelompokkan dari jenis pejalan kaki yang berjalan dengan kecepatan lebih cepat dari pejalan kaki normal. Individu manusia pada Gambar 2.1 dapat dideskripsikan dengan pose manusia X-Y koordinat $[x_{hi}, y_{hi}, \psi_{hi}]^T$, kecepatan v_{hi} , akselerasi a_{hi} , dan radius individu tubuh manusia r_{hi} dapat diformulasikan sebagai berikut:

$$S_{hi} = [x_{hi}, y_{hi}, \psi_{hi}, v_{hi}, a_{hi}, r_{hi}]^T \quad (2.1)$$



Gambar 2.1. Kemungkinan Pergerakan Manusia Tunggal

Manusia dapat berinteraksi dengan benda-benda statis di lingkungannya dapat dilihat pada Gambar 2.2. Beberapa objek statis di lingkungan adalah papan pesan, cat, dan tampilan layar. Interaksi dinyatakan dalam bentuk persamaan

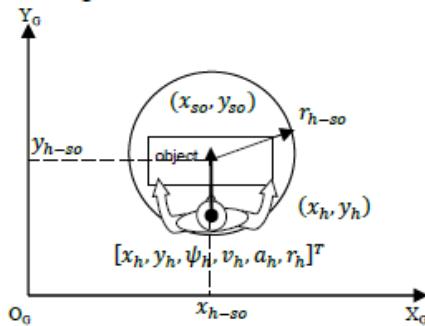
$$\mathbf{S}_{h-so_m} = [x_{h-so_m}, y_{h-so_m}, r_{h-so_m}]^T \quad (2.2)$$

di mana (x_{h-so_m}, y_{h-so_m}) merepresentasikan m^{th} adalah titik poin dari area *Human Object Interaction*. Radius dalam area ini dinotasikan dengan r_{h-so_m} titik tengah dan radius HOI dapat di kalkulasikan menggunakan persamaan sebagai berikut

$$(x_{h-so_m}, y_{h-so_m}) = (x_{so}, y_{so}) \quad (2.3)$$

$$r_{h-so_m} = \sqrt{(x_{so} - x_h)^2 + (y_{so} - y_h)^2} \quad (2.4)$$

di asumsikan bahwa lokasi objek statis (x_{so}, y_{so}) adalah diposisikan di tengah area HOI.



Gambar 2.2. Interaksi Antara Manusia dan Objek Statis

Dalam makalah ini mendeskripsikan geometri penggerak manusia dengan model *non-holonomic* yang digambarkan sebagai berikut:

$$\begin{bmatrix} \dot{x}_h \\ \dot{y}_h \\ \dot{\psi}_h \end{bmatrix} = \begin{bmatrix} \cos\psi_h & 0 \\ \sin\psi_h & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_h \\ \omega_h \end{bmatrix} \quad (2.5)$$

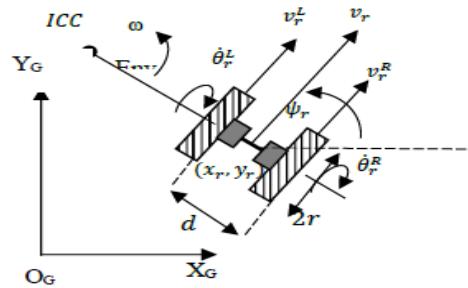
di mana \mathbf{v}_h adalah kecepatan manusia bergerak dengan $[v_h \ \omega_h]^T$. Pose pundak terhadap nilai tengah dinotasikan dengan $[x_h, y_h, \psi_h]^T$ yang merepresentasikan koordinat umum manusia. Persamaan ini dapat membantu membuat transisi dari koordinat tubuh manusia ke koordinat global.

Forward kinematics model manusia *non-holonomic* pada Gambar 2.3 direpresentasikan sebagai berikut

$$\begin{bmatrix} v_h \\ \omega_h \end{bmatrix} = r_h \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{d_h} & -\frac{1}{d_h} \end{bmatrix} \begin{bmatrix} \dot{\theta}_h^R \\ \dot{\theta}_h^L \end{bmatrix} \quad (2.6)$$

di mana langkah kaki manusia r_h , diameter tubuh d_h , kecepatan dari langkah kaki manusia $\dot{\mathbf{q}}_h = [\dot{\theta}_h^R \dot{\theta}_h^L]^T$ digunakan untuk pergerakan manusia dengan pengaruh dari kecepatan linear dan *angular* \mathbf{v}_h . Model *inverse kinematics* digunakan agar mendapatkan kecepatan langkah kaki manusia dengan $\dot{\mathbf{q}}_h$ dari kecepatan linear dan *angular* \mathbf{v}_h didefinisikan sebagai berikut

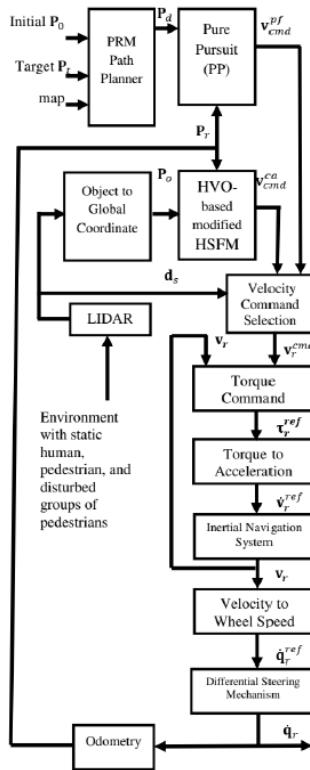
$$\begin{bmatrix} \dot{\theta}_h^R \\ \dot{\theta}_h^L \end{bmatrix} = \frac{1}{r_h} \begin{bmatrix} 1 & \frac{d_h}{2} \\ 1 & -\frac{d_h}{2} \end{bmatrix} \begin{bmatrix} v_h \\ \omega_h \end{bmatrix} \quad (2.7)$$



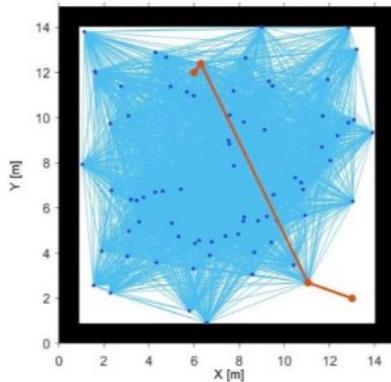
Gambar 2.3. Kinematika Mobile Robot

Desain struktur sistem penghindaran lingkungan untuk *mobile* robot diilustrasikan pada Gambar 2.4 yang diinginkan sebagai referensi $P_d = [x_d, y_d, \psi_d]^T$ dihasilkan dari *probabilistic road map* (PRM) perencanaan lintasan pada Gambar 2.5. Referensi ini dapat dihubungkan dengan menentukan *way-point* secara acak. *Waypoint* dapat dideteksi dengan *mobile* robot, sebenarnya pose *mobile* robot dinotasikan $P_r = [x_r, y_r, \psi_r]^T$, pose rintangan dinyatakan $P_o = [x_o, y_o, \psi_o]^T$.

Penentuan jarak dan kenyamanan aman menggunakan metode *Theat Level Index* (TLI) untuk mengevaluasi jarak dan kenyamanan aman untuk masing-masing objek yang ada di lingkungan $TLI = e^{\frac{d_{r-o}}{\sigma}}$. Indeks evaluasi ini membandingkan antara standar deviasi dan jarak sebenarnya masing-masing keberatan *mobile* robot d_{r-o} . Standar deviasi didefinisikan pada 0,65 m. Nilai parameter dapat memberikan ruang untuk robot bermanuver.



Gambar 2.4. Struktur Sistem Navigasi



Gambar 2.5. Konfigurasi path planning menggunakan PRM

Desain kinematika dan dinamika mobile robot ditentukan pada pusat massa digambarkan sebagai G. titik ini mewakili robot dalam koordinat XY. Pose mobile robot dan transisi konfigurasinya adalah

$$\mathbf{P}_r = [x_r, y_r, \psi_r]^T \quad (2.8)$$

$$\dot{\mathbf{P}}_r = [\dot{x}_r, \dot{y}_r, \dot{\psi}_r]^T \quad (2.9)$$

di mana robot dinyatakan dengan $[x_r, y_r]^T$ dan orintasi di simbolkan ψ_r . Konfigurasi transisi pose robot dari body ke koordinat global

$$\begin{bmatrix} \dot{x}_h \\ \dot{y}_h \\ \psi_h \end{bmatrix} = \begin{bmatrix} \cos\psi_h & 0 \\ \sin\psi_h & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_h \\ \omega_h \end{bmatrix} \quad (2.10)$$

kecepatan mobile robot $V_r = [v_r, \omega_r]^T$ karena perubahan pose dalam koordinat global $\dot{p}_r = [\dot{x}_r, \dot{y}_r, \dot{\psi}_r]^T$. Model forward kinematic

$$\begin{bmatrix} v_h \\ \omega_h \end{bmatrix} = r_h \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{d_h} & -\frac{1}{d_h} \end{bmatrix} \begin{bmatrix} \dot{\theta}_h^R \\ \dot{\theta}_h^L \end{bmatrix} \quad (2.11)$$

kecepatan roda dinotasikan $\dot{q}_r = [\dot{\theta}_r^R, \dot{\theta}_r^L]^T$. Kecepatan roda diperoleh dari linear dan kecepatan sudut V_r dengan aplikasi inverse kinematik

$$\begin{bmatrix} \dot{\theta}_h^R \\ \dot{\theta}_h^L \end{bmatrix} = \frac{1}{r_h} \begin{bmatrix} 1 & \frac{d_h}{2} \\ 1 & -\frac{d_h}{2} \end{bmatrix} \begin{bmatrix} v_h \\ \omega_h \end{bmatrix} \quad (2.12)$$

Integrasi Eulerian digunakan untuk memperbarui pose robot sebagai berikut

$$\begin{bmatrix} x_r(k+1) \\ y_r(k+1) \\ \psi_r(k+1) \end{bmatrix} = \begin{bmatrix} x_r(k) \\ y_r(k) \\ \psi_r(k) \end{bmatrix} + kT \begin{bmatrix} \dot{x}_r(k) \\ \dot{y}_r(k) \\ \dot{\psi}_r(k) \end{bmatrix} \quad (2.13)$$

model dinamis mobile robot dapat ditunjukkan menggunakan newton-euler sebagai berikut

$$\mathbf{F}_r = m_r a_r = m_r \ddot{v}_r \quad (2.14)$$

$$\mathbf{N}_r = I_r \ddot{\omega}_r \quad (2.15)$$

$$\ddot{v}_r = \frac{1}{m_r r} (\tau_R + \tau_L) \quad (2.16)$$

$$\ddot{\omega}_r = \frac{d}{I_r r} (\tau_R + \tau_L) \quad (2.17)$$

Di mana gaya total yang bekerja pada robot di pusat massanya diwakili, ia akan bergerak dalam garis lurus dengan \mathbf{F}_r , total torsi dinotasikan dengan \mathbf{N}_r , massa badan robot disimbolkan dengan m_r , akselerasi di deklarasikan dengan a_r , inersia dari robot dinyatakan I_r , torsi kanan dan kiri motor di deskripsikan dengan τ_R dan τ_L .

Persamaan yang digunakan mobil robot terdiri dari *repulsive force* dan *attractive force*

$$\mathbf{F}_r^{HVO-modHSFM} = \sum \mathbf{F}_{r-obs}^{rep} + \mathbf{F}_{r-tar}^{att} \quad (2.18)$$

dan untuk *repulsive force* di bagi menjadi

- Objek gangguan statis dideskripsikan sebagai berikut

$$\begin{aligned} \mathbf{F}_{h-so}^{rep} &= K_h^{str} \exp\left(\frac{(r_{h-so} - d_{h-so})}{K_h^{rng}}\right) \mathbf{n}_{h-so} \\ &\quad + K_h^{cmp} g(r_{h-so} - d_{h-so}) \mathbf{n}_{h-so} \\ &\quad + K_h^{fri} g(r_{h-so} - d_{h-so}) \Delta v_{so-h}^t \mathbf{t}_{h-so} \end{aligned} \quad (2.19)$$

- *Single* statis atau pergerakan manusia diformulasikan

$$\begin{aligned} \mathbf{F}_{h-sh}^{rep} &= K_h^{str} \exp\left(\frac{(r_{h-sh} - d_{h-sh})}{K_h^{rng}}\right) \mathbf{n}_{h-sh} \\ &\quad + K_h^{cmp} g(r_{h-sh} - d_{h-sh}) \mathbf{n}_{h-sh} \\ &\quad + K_h^{fri} g(r_{h-sh} - d_{h-sh}) \Delta v_{sh-h}^t \mathbf{t}_{h-sh} \end{aligned} \quad (2.20)$$

- Interaksi objek manusia di formulasikan sebagai berikut

$$\begin{aligned} \mathbf{F}_{h-hoi}^{rep} &= K_h^{str} \exp\left(\frac{(r_{h-hoi} - d_{h-hoi})}{K_h^{rng}}\right) \mathbf{n}_{h-hoi} \\ &\quad + K_h^{cmp} g(r_{h-hoi} - d_{h-hoi}) \mathbf{n}_{h-hoi} \\ &\quad + K_h^{fri} g(r_{h-hoi} - d_{h-hoi}) \Delta v_{hoi-h}^t \mathbf{t}_{h-hoi} \end{aligned} \quad (2.21)$$

- Hasil interaksi dengan kelompok manusia di notasikan sebagai berikut

$$\begin{aligned} \mathbf{F}_{h-gh}^{rep} &= K_h^{str} \exp\left(\frac{(r_{h-gh} - d_{h-gh})}{K_h^{rng}}\right) \mathbf{n}_{h-gh} \\ &\quad + K_h^{cmp} g(r_{h-gh} - d_{h-gh}) \mathbf{n}_{h-gh} \\ &\quad + K_h^{fri} g(r_{h-gh} - d_{h-gh}) \Delta v_{gh-h}^t \mathbf{t}_{h-gh} \end{aligned} \quad (2.22)$$

penjumlahan untuk seluruh persamaan *repulsive force* dengan berbagai macan skenario sehingga menjadi

$$\sum \mathbf{F}_{r-obs}^{rep} = \mathbf{F}_{h-so}^{rep} + \mathbf{F}_{h-sh}^{rep} + \mathbf{F}_{h-hoi}^{rep} + \mathbf{F}_{h-gh}^{rep} \quad (2.23)$$

sedangkan untuk memberikan robot bantuan menemukan target dengan *attractive force* adalah

$$\mathbf{F}_{r-tar}^{att} = K_r^v (v_r^{des} - v_r^{act}) \quad (2.24)$$

Robot dapat berpindah antara direksi target dengan menggunakan v_r^{des} . Waktu konstan K_r^v diperlukan untuk *mobile* robot berpindah dari kecepatan aktual v_r^{act} dan kecepatan yang diinginkan v_r^{des} . Hasilkan kecepatan yang diinginkan berdasarkan pemindaian LIDAR di sekitar robot seperti yang dijelaskan di bawah ini

$$HVO = \bigcup_{o \in O} VO_{r|o} \cup \bigcup_{ro \in R} RVO_{r|ro} \cup \bigcup_{h \in P} NLVO_{r|h} \quad (2.25)$$

$$V_{safe\Omega} = \overline{HVO_\Omega} = \{v_r \in v | v_r \notin HVO_\Omega\} \quad (2.26)$$

$$v_r^{des} = \arg \min_{v_r^{HVO^*} \in V_{safe\Omega}} (v_{\max, v_r^{HVO^*}}^{ca}) \quad (2.27)$$

kecepatan sudut *angular* di tentukan dengan memanfaatkan pengindraan LIDAR maka di dapatkan persamaan sebagai berikut

$$\omega_{cmd}^{ca} = \begin{cases} \arg \max_{d_{\psi_r^*} \in ds} (\mathbf{d}_s) \\ \psi_r^* \in \psi \mid \arg \max_{\psi_r^* \in \psi} (\Delta \psi_r^*) \end{cases} \quad (2.28)$$

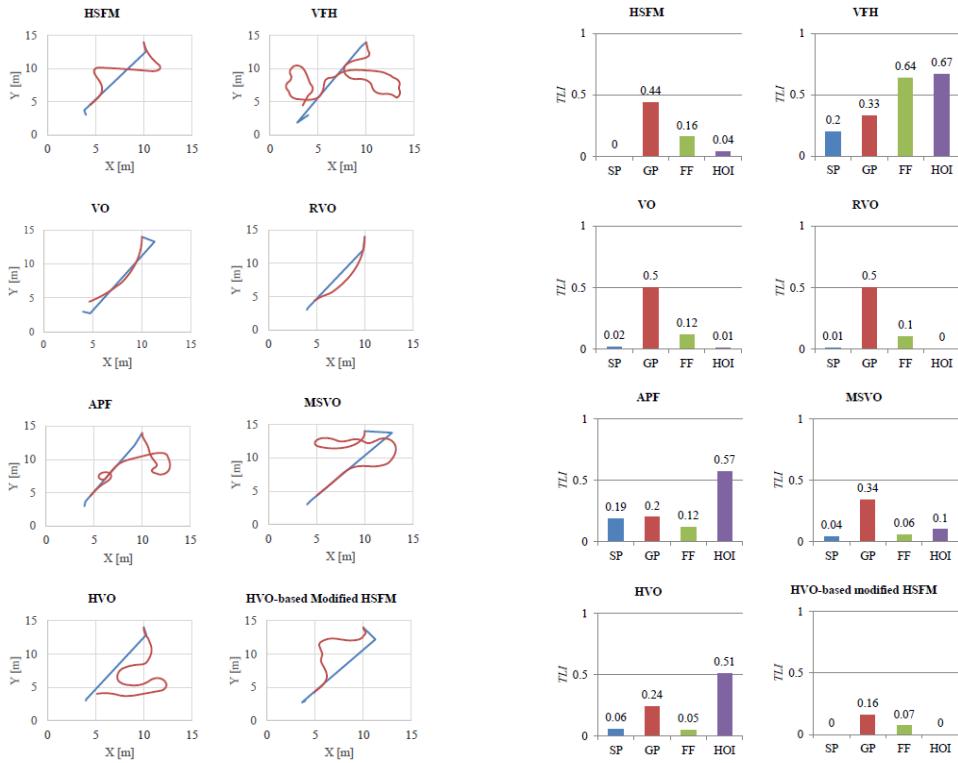
- Skenario Pertama

Simulasi sistem navigasi dalam ruangan dengan kinematika dan dinamikanya hasilnya nilai referensi ada di garis biru dan aktualnya pada garis merah. Metode penghindaran tabrakan untuk menghindari sekelompok pejalan kaki GP, tunggal pejalan kaki SP, F-Formasi FF dan individu manusia berinteraksi dengan objek statis HOI yang diilustrasikan pada Gambar 2.6.

- Skenario kedua

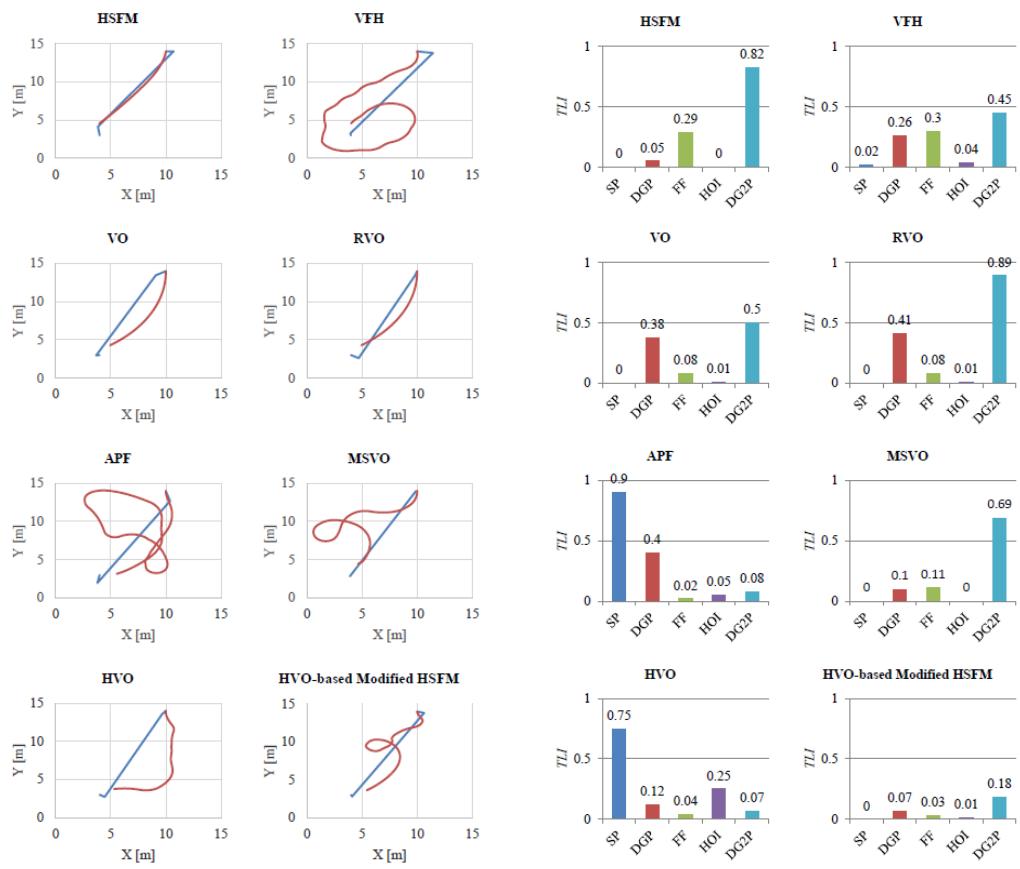
Simulasi dalam ruangan dengan memperhatikan kinematikanya untuk menghindari dua kelompok pejalan kaki yang terganggu, tunggal pejalan kaki, sekelompok formasi F terdiri dari tiga orang, dan interaksi objek manusia yang diilustrasikan pada Gambar 2.7.

Skenario pertama meliputi pejalan kaki dengan kecepatan normal. Sedangkan yang kedua skenario melibatkan kelompok pejalan kaki yang terganggu dengan kecepatan yang lebih cepat. Dari ringkasan tabrakan ini evaluasi kinerja penghindaran, berbasis HVO HSMF yang dimodifikasi memiliki nilai *threat level index* (TLI) terendah.



Gambar 2.6. Perbandingan lintasan referensi dan aktual dan performa keselamatan dan kenyamanan sekenario 1

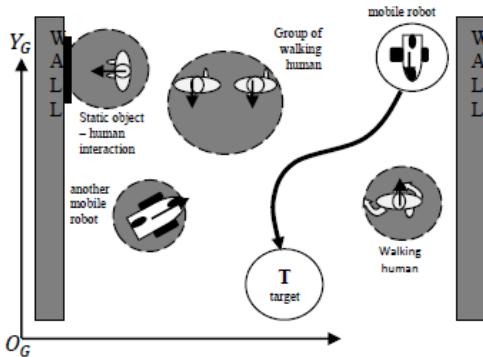
Pada journal ini merupakan penelitian untuk penghindaran rintangan dengan lebih fokus pada HVO HSFM dengan penggunaan TLI sebagai pengukuran keselamatan dan kenyamanan di lingkungan sekitar, penggunaan TLI dapat dikembangkan dengan keselamatan, kenyamanan, kecepatan, dan efisien waktu untuk mencapai target. Kelebihan dari penelitian tersebut adalah metode yang digunakan berhasil mengaplikasikan metode HVO HSFM dengan penggunaan TLI dengan menghasilkan nilai keselamatan dan aman pada mobile robot pada lingkungan sekitar. Namun penelitian tersebut menghasilkan jalur baru dengan memakan waktu terbilang lama karena, banyak rintangan dalam lingkungan, jika penelitian menggunakan metode di update dengan algoritma prediksi trajektori objek maka robot dapat meminimalkan waktu lebih baik dan jalur actual akan bermanuver tidak jauh jalur referensi. Prediksi bentuk objek yang tidak diketahui oleh mobile robot jika bentuknya, diusulkan untuk diatasi dengan estimasi *state* dan prediksi EnKF, sehingga penelitian selanjutnya dengan ditambahkan estimasi dan prediksi dapat menguji keberhasilan algoritma penghindaran rintangan.



Gambar 2.7. Perbandingan lintasan referensi dan aktual dan performa keselamatan dan kenyamanan sekenario 2

2.1.2 Collision Avoidance of Multi Modal Moving Objects for Mobile Robot Using Hybrid Velocity Obstacles [17]

Multi modal objek bergerak adalah rintangan yang bergerak dengan non-linear. Gambar di bawah ilustrasi dari skenario penghindaran rintangan mobile robot dalam ruangan tertutup dengan multi modal objek bergerak. Mobile robot harus memastikan keamanan setiap objek di sekitarnya saat mengangkut barang ke posisi target tertentu. Garis putus-putus pada batas setiap objek abu-abu menunjukkan area aman yang harus dihindari robot. Gerakan multi modal mengacu pada semua jenis gerakan yang menggunakan dua atau lebih jenis gerakan yang berbeda. Ini termasuk orang, kelompok orang, dan robot. Masing-masing jenis gerakan ini dapat membantu memahami cara kerja gerakan dan bagaimana gerakan itu dapat digunakan dalam konteks yang berbeda, di mana diilustrasikan pada Gambar 2.8.



Gambar 2.8. Interaksi antara mobile robot dengan objek yang bergerak dalam ruangan
Lingkaran dinotasikan dengan persamaan

$$x_o(k) = x_c + r_{path} \cos(k\omega_o T) \quad (2.29)$$

$$y_o(k) = y_c + r_{path} \sin(k\omega_o T) \quad (2.30)$$

Pusat jalur lingkaran adalah representasi dari x_c dan y_c dengan radius r_{path} , $k = 0, 1, 2, \dots$, ketika ω_o . Tipe lain adalah gerak loop pada gambar di bawah dengan gerak jalur cloverleaf. Jalur cloverleaf direpresentasikan dengan persamaan sebagai berikut

$$x_o(k) = x_c + a \sin(k\omega_o T) \sin(2k\omega_o T) \quad (2.31)$$

$$y_o(k) = y_c + a \cos(k\omega_o T) \sin(2k\omega_o T) \quad (2.32)$$

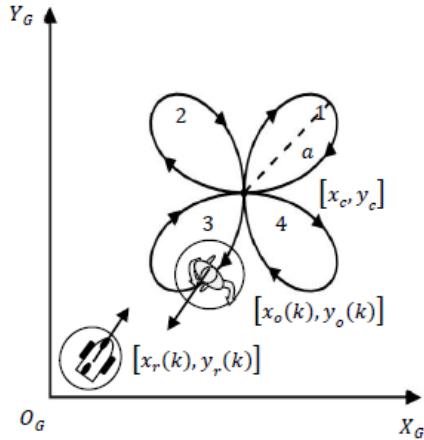
Titik tengah dari jalur cloverleaf direpresentasikan x_c dan y_c amplitudo a . Bentuk jalur cloverleaf dapat dilihat pada Gambar 2.9.

Gerak tanpa loop terdiri dari beberapa jalur non-linier. Perencanaan jalur dapat menggunakan Probabilistic Road Map (PRM), untuk menentukan jalur gerak non-linear tanpa loop.

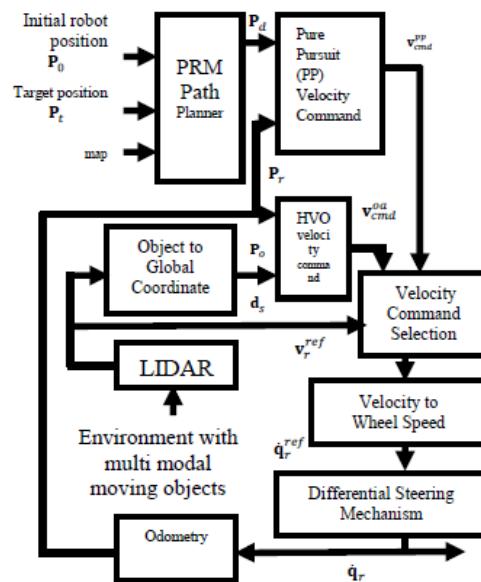
Node ini direpresentasikan sebagai titik penting untuk pembuatan jalur gerak mobile robot. banyaknya node dan jarak antar node ini berhubungan antara node lainnya yang dapat di atur pada setiap jaraknya. Jalur non-linier sebagai kumpulan node yang saling terkait $[x_o(k), y_o(k)]$ diambil dari start awal ke titik akhir tujuan yang dapat ditentukan dengan menentukan initial posisi dan target posisi.

Struktur HVO berdasarkan sistem penghindaran rintangan. Struktur trakjektori traking bebas rintangan dengan kinematik mobile robot menggunakan dua roda pada Gambar 2.10. Mobile robot di setting dengan menggunakan titik referensi diinginkan, sehingga direpresentasikan dengan persamaan $P_d = [x_d \ y_d \ \psi_d]^T$.

Waypoint yang di bentuk melalui pose robot akan membentuk jalur yang akan dilalui oleh robot dari titik awal hingga target



Gambar 2.9. Jalur perpindahan objek bentuk cloverleaf



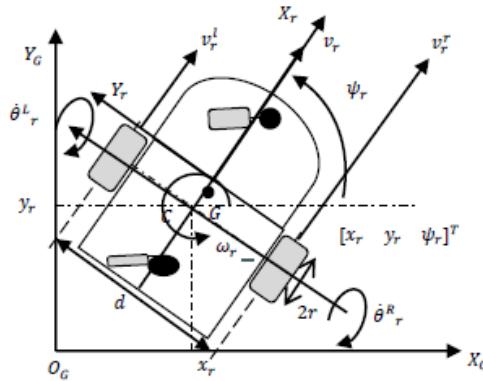
Gambar 2.10. Struktur HVO sistem penghindaran rintangan untuk trajektori traking pada mobile robot dengan dua roda differential-steering

Ketika mobile robot menghindari rintangan maka, standar jarak aman perlu dinotasikan. Multi modal objek bergerak yang ada di sekitar mobile robot adalah rintangan didefinisikan dengan $P_o = [x_o \ y_o \ \psi_o]^T$ dengan batasan luar setiap objek. Penelitian ini mengusulkan Proximity Indeks (PI) untuk evaluasi jarak aman untuk setiap objek pada sekitar lingkungan

$$PI = \frac{\sigma}{d_{r-o}} \quad (2.33)$$

Hasilnya dibandingkan dengan jarak aman terdekat terhadap objek σ dan jarak saat ini terhadap setiap objek terhadap mobile robot d_{r-o} . Jarak terdekat σ adalah 0.35 m . Parameter ini di pilih menyesuaikan ukuran robot. Untuk d_{r-o} ditentukan dengan jarak Euclidean. Standarisasi garis aman adalah 0.5 m atau sama dengan 0.7 dari PI. Semakin kecil nilai PI, semakin aman kondisinya. Ketika nilai PI lebih besar dari 0.7 keamanan objek terancam.

Penelitian ini mendefinisikan mobile robot sebagai robot differential-steering beroda dua yang dibutuhkan untuk bergerak dari posisi awal ke posisi target yang di tunjukan pada Gambar 2.11.



Gambar 2.11. Konfigurasi mobile robot dua roda differential-steering

dengan d adalah lebar robot, C adalah titik tengah dari sudut ban, G adalah Center of Gravity (COG) dan titik representasi robot pada sumbu koordinat X-Y. Diameter ban adalah $2r$. Model mobile robot dan transisi konfigurasi adalah

$$P_r = [x_r \ y_r \ \psi_r]^T \quad (2.34)$$

$$\dot{P}_r = [\dot{x}_r \ \dot{y}_r \ \dot{\psi}_r]^T \quad (2.35)$$

posisi robot adalah x_r dan y_r dan orientasi adalah ψ_r .

Konfigurasi transisi robot dari bentuk ke koordinat umum adalah

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\psi}_r \end{bmatrix} = \begin{bmatrix} \cos \psi_r & 0 \\ \sin \psi_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \quad (2.36)$$

di mana kecepatan mobile robot $V_r = [v_r \ \omega_r]^T$ memicu perubahan posisinya dalam koordinat global $\dot{P}_r = [\dot{x}_r \ \dot{y}_r \ \dot{\psi}_r]^T$. Model forward kinematik dari mobile robot adalah

$$\begin{bmatrix} v_r \\ \omega_r \end{bmatrix} = r \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{d} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r^R \\ \dot{\theta}_r^L \end{bmatrix} \quad (2.37)$$

di mana kecepatan roda $\dot{q}_r = [\dot{\theta}_r^R \ \dot{\theta}_r^L]^T$ di pakai sebagai steering mobile robot dengan efek linear dan kecepatan angular V_r .

Disisi lain, inverse kinematik digunakan untuk menentukan kecepatan roda \dot{q}_r dari efek linear dan kecepatan angular V_r adalah sebagai berikut

$$\begin{bmatrix} \dot{\theta}_r^R \\ \dot{\theta}_r^L \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & \frac{d}{2} \\ \frac{1}{2} & d \\ 1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \quad (2.38)$$

pose robot dapat di perbaharui dengan menggunakan Euler ke dalam persamaan berikut

$$\begin{bmatrix} x_r(k+1) \\ y_r(k+1) \\ \psi_r(k+1) \end{bmatrix} = \begin{bmatrix} x_r(k) \\ y_r(k) \\ \psi_r(k) \end{bmatrix} + kT \begin{bmatrix} \dot{x}_r(k) \\ \dot{y}_r(k) \\ \dot{\psi}_r(k) \end{bmatrix} \quad (2.39)$$

Mobile robot menggunakan sensor LIDAR (Light Detection and Ranging) maximal sensing di jarak 20 m. Di pasang pada sudut mobile robot dengan total 8 titik, setiap titik memiliki jarak 45°. Berdasarkan pembacaan LIDAR, menentukan kecepatan mana yang perlu digunakan untuk menghasilkan kecepatan yang sesuai dengan Algoritma-1.

Pure Pursuit (PP) adalah metode pelacakan yang efektif untuk mengikuti sebuah jalur. Ini menghitung perintah kecepatan linier dan sudut yang menggerakkan robot dari posisinya saat ini untuk mencapai beberapa sudut pandang maju di depan robot. PP dapat digunakan sebagai waypoint spesifik. Linear dan kecepatan angular dari path tracking $V_{cmd}^{pp} = [v_{cmd}^{pp} \ \omega_{cmd}^{pp}]^T$. HVO menghasilkan keluaran dalam bentuk kecepatan $V_{cmd}^{oa} = [v_{cmd}^{oa} \ \omega_{cmd}^{oa}]^T$ untuk tujuan penghindaran rintangan. Kecepatan ini menghasilkan implementasi dari HVO dengan diberikan masukan yang terdiri dari pose rintangan P_o dan pose robot P_r .

Tentukan kecepatan mobile robot v_r dalam kecepatan ruang V yang dapat menyebabkan tabrakan antara mobile robot MR dan rintangan O dalam setiap titik waktu t antara t_0 dan Time Horizon (TH)

$$VO = \bigcup_{t=t_0}^{TH} \{v_r \in V | \exists t \in [t_0, TH], MR(t) \cap O(t) \neq \emptyset\} \quad (2.40)$$

Dalam kasus penghindaran rintangan O memiliki karakteristik pengindraan yang sama dengan mobile robot MR , RVO diformulasikan sebagai berikut

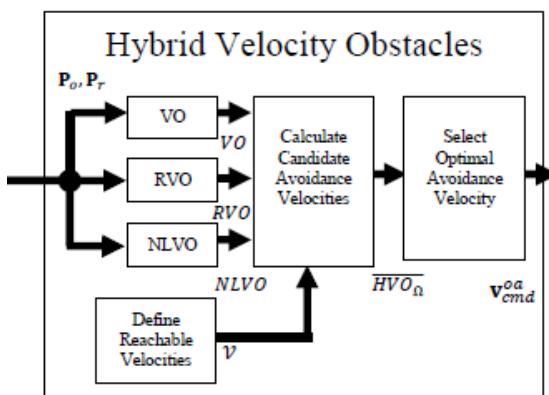
$$RVO_o^r(v_o, v_r, a_o^r) = \bigcup_{t=t_0}^{TH} \left\{ v_r' \left| \frac{1}{a_o^r} v_r' + \left(1 \frac{1}{a_o^r}\right) v_r \in VO_o^r(v_o) \right. \right\} \quad (2.41)$$

Pergerakan manusia merupakan rintangan yang kompleks dengan arah dan kecepatan yang tidak pasti. Penelitian ini menggantikan HRVO dengan NLVO yang dapat digunakan untuk non-linear objek bergerak. NLVO adalah persamaan umum dari VO dalam tujuan dan magnitude kecepatan rintangan v_o .

$$NLVO = VO \Leftrightarrow \exists (t_1, t_2) \in [t_0, TH]^2 | t_1 \neq t_2 \text{ dan } v_o(t_1) \neq v_o(t_2) \quad (2.42)$$

$$HVO = \bigcup_{o \in O} VO_{r|o} \cup \bigcup_{ro \in O} RVO_{r|ro} \cup \bigcup_{h \in P} NLVO_{r|h} \quad (2.43)$$

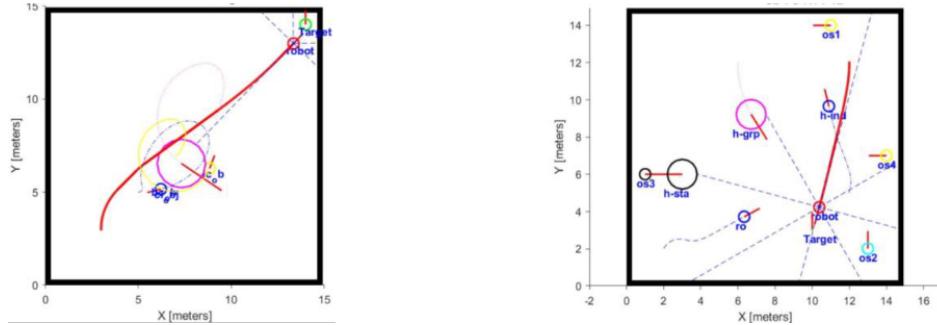
Kecepatan untuk mobile robot harus dipilih dari ruang kecepatan V dengan batasan dinamika robot untuk rintangan tunggal dan untuk banyak rintangan sebagai berikut, dan diagram alir HVO dengan penghindaran rintangan diilustrasikan pada Gambar 2.12.



Gambar 2.12. HVO berdasarkan penghindaran rintangan

Diameter body mobile robot adalah 0.6 m . Radius roda mobile robot adalah 0.15 m . Kecepatan linier robot yang diinginkan adalah $1.0 \frac{\text{m}}{\text{s}}$. Orang pertama

dengan diameter 0.6 m . Kecepatan orang berjalan $0.5 \frac{\text{m}}{\text{s}}$. Kelompok orang terdiri dari dua orang berdiameter 1.8 m , kecepatannya adalah $0.3 \frac{\text{m}}{\text{s}}$. Mobile robot agen kedua berdiameter 0.6 m , dengan kecepatan $0.15 \frac{\text{m}}{\text{s}}$.



Gambar 2.13. Skenario 1 menggunakan algoritma VO trajektori robot dan agen-agen lain

Skenario pertama dapat di lihat pada Gambar 2.13 bertujuan untuk mengkaji kemampuan mobile robot dalam menghindari objek bergerak multi modal dengan gerakan loop. Skenario kedua diimplementasikan untuk mengevaluasi kinerja mobile robot untuk menghindari objek bergerak multi modal dengan lintasan yang tidak diketahui. Hasilnya dari dua skenario dapat dilihat pada Tabel 2.1.

Kelebihan dari penelitian tersebut adalah penelitian tersebut berhasil mengaplikasikan metode VO dan HVO dengan dua skenario yang berbeda dan model objek rintangan yang berbeda, di mana lintasan objek rintangan diketahui tanpa memberikan informasi lintasan ke robot. Penelitian tersebut digunakan pada lingkungan padat sehingga pada penelitian ini dapat diusulkan metode prediksi arah yang tak tentu dari objek rintangan bergerak, diusulkan untuk diatas dengan estimasi *state* dan prediksi pergerakan menggunakan *Ensemble Kalman Filter* (EnKF) berdampingan dengan objek *tracking* dengan menggunakan PDAF, sehingga penelitian selanjutnya dengan ditambahkan estimasi dan prediksi dapat menguji keberhasilan algoritma penghindaran rintangan. Dari kekurangan dan kelebihan yang dijelaskan di atas, bisa diambil kesimpulan bahwa penelitian untuk menguji *obstacle avoidance* pada mobile robot pada lingkungan dinamis masih sangat menarik untuk dikembangkan.

Tabel 2.1. Performa metode HVO untuk penghindaran rintangan komparasi APF, VFH, VO dan HRVO

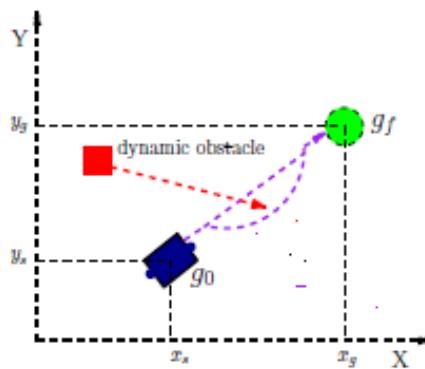
Simulation	Number of Obstacles	Modality of Moving Objects	Proximity Index (PI)				
			APF	VFH	VO	HRVO	HVO
Scenario 1	3	<ul style="list-style-type: none"> • 1st Moving object with cloverleaf path • 2nd Moving object with little circle path • 3rd Moving object with big circle path 	1.14	0.63	0.37	0.44	0.14
Scenario 2	3	<ul style="list-style-type: none"> • 1st Moving object represents single walking human • 2nd Moving object represents group of walking human • 3rd Moving object represents another mobile robot 	0.51	0.33	0.4	0.33	0.2
Average of PI			0.83	0.48	0.39	0.39	0.17

2.1.3 Real Time Obstacle Prediction Using Neural Network Based on Extended Kalman Filter for Robot Path Planning [18]

Penelitian melatar belakangi tentang manfaat dari autonomous mobile robot dalam melayani dalam beberapa pekerjaan membantu manusia di lingkungan tertentu seperti militer, rumah sakit, pertanian dan pengawasan. Pekerjaan ini tidaklah mudah jika diaplikasikan pada mobile robot, karena pada lingkungan yang padat dengan objek dinamis. Lingkungan dengan objek dinamis, mobile robot dapat di operasikan dalam situasi yang berbahaya dan berubah secara tidak terduga. Tinjauan literatur juga mengungkapkan bahwa, algoritma perencanaan jalur dinamis yang menggunakan sensor untuk prediksi gerakan

mungkin gagal untuk menghasilkan jalur bebas tabrakan yang tepat karena kesalahan prediksi jalur hambatan yang disebabkan oleh *noise* data. Dalam studi ini, kami menawarkan sebuah pendekatan untuk memperkirakan gerakan rintangan dalam kondisi dinamis, yang membantu robot menghindari rintangan, dapat diterapkan pada berbagai kecepatan, dan membutuhkan waktu komputasi yang lebih sedikit dengan akurasi prediksi yang lebih tinggi. Kalman filter adalah pilihan yang sangat baik untuk memprediksi jalur rintangan. Untuk prediksi yang akurat, bagaimanapun, pengetahuan tentang matriks kovarians kesalahan *noise* sangat penting. Selain itu, pemrosesan langsung dari matriks ini sering kali diperlukan untuk sistem nonlinier yang bervariasi waktu, seperti robot bergerak. penelitian ini berkontribusi mengembangkan pendekatan untuk menentukan gerakan rintangan di lingkungan dinamis menggunakan jaringan saraf multi layer yang cocok untuk berbagai kecepatan dan membutuhkan waktu komputasi yang lebih sedikit dengan akurasi prediksi yang ditingkatkan. Struktur jaringan saraf berbasis pembelajaran mendalam yang diusulkan dalam pekerjaan ini sangat andal dan kuat terhadap kebisingan. Filter Kalman diperpanjang berbasis autoencoder denoising yang dikembangkan mampu memprediksi keadaan hambatan dengan antara Gaussian dan non-Gaussian.

Dalam paper ini mendeskripsikan robot bergerak dalam ruang 2 dimensi, dengan objektif menemukan perencanaan jalur dari start sampai dengan posisi yang diinginkan.



Gambar 2.14. Definisi permasalahan

Tentukan g representasi dat jalur perencanaan gerak robot yang didefinisikan

$$g = [g_0, g_1, \dots, g_{n-2}, g_{n-1}, g_f] \quad (2.44)$$

di mana g_1, g_{n-2}, g_{n-1} adalah way point jalur robot.

Untuk memastikan jalur bebas dari tabrakan, harus ada zona aman antara robot dengan objek dinamis maupun statis dalam waktu bergerak.

$$\text{for } i = 1, 2, \dots, N, O_{pi}(t) \notin P(x(t)) \quad (2.45)$$

di mana N adalah banyaknya objek, $P(x(t))$ daerah aman robot yang sesuai dari robot dan o_p adalah posisi dari rintangan. Persamaan state j^{th} objek adalah

$$o_j(t) = \begin{bmatrix} o_{pj}(t) \\ o_{vj}(t) \end{bmatrix} \quad (2.46)$$

$o_{vj}(t)$ adalah kecepatan dari objek. Berdasarkan pada objek dengan kecepatan tidak berubah, hubungan antara posisi dan kecepatan dari persamaan state j^{th} objek menggunakan dasar dari persamaan kinetik didapatkan persamaan.

$$o_{pj}(t) = o_{pj}(t_0) + o_{vj}(t_0) * (t - t_0) \quad (2.46)$$

Persamaan model state space dari robot adalah

$$\dot{r}(t) = f(r(t), u(t)) \quad (2.47)$$

$r(t)$ adalah state robot dan $u(t)$ berdasarkan vektor kontrol. Di samping itu kondisi bebas tabrakan harus ditentukan perencanaan jalur terpendek dengan

$$g *= \arg\min_g \int_g dq \quad (2.48)$$

dq adalah perbedaan dari panjang arc pada jalur.

Algoritma perencanaan gerak, di mana objek bergerak dengan kecepatan konstan dan robot mengetahui posisinya. Dalam sistem navigasi, penggunaan extended kalman filter adalah metode yang umum digunakan untuk memprediksi jalur rintangan. Pilihan matriks kovarians noise pengukuran R sangat memengaruhi keakuratan prediksi kalman filter.

Kalman filter dapat digunakan secara *real time* dalam prosesnya dan dapat digunakan juga estimasi dari kecepatan dan posisi ketika robot bergerak.

Langkah-langkah *kalman filter*

1. Prediksi – berdasarkan data sensor sebelumnya, prediksi nilai selanjutnya
2. Pembaharuan – Untuk mendapatkan nilai yang mendekati nilai aktual, nilai prediksi disempurnakan menggunakan nilai terukur.

Ketika robot bergerak, sistem sensor sampling dengan observasi area setiap T_s ms. Ketika sensor menangkap objek, selanjutnya sensor mulai metrack objek untuk estimasi posisi dan kecepatan. Ketika track objek rintangan yang muncul dalam posisi yang acak di suatu lingkungan area, menggunakan extended kalman filter. Input observasi sensor dan outputnya adalah state estimasi sistem. Prediksi pertama berdasarkan persamaan state, dan kedua adalah pembetulan berdasarkan persamaan observasi.

Posisi dan kecepatan dari objek bergerak secara rekursif di estimasi dari data sensor tersedia, dinotasikan dengan estimasi state vektor $\hat{x}(k)$. Ketidakpastian dalam rintangan bergerak dan error pengukuran adalah variabel acak dengan asumsi gaussian white *noise* dengan zero mean dan variance dan secara rekursif diperbaharui dengan data sensor yang baru.

Jalur objek bergerak adalah non-linear dan *extended kalman filter* digunakan untuk estimasi objek bergerak. *Extended kalman filter*, persamaan non-linear yang di linearisasi dengan matriks jacobian. Robot bergerak memiliki persamaan state

$$r_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \quad (2.49)$$

model state space setelah dilakukan linearisasi diberikan

$$\begin{aligned} \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} &= A \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + B \begin{bmatrix} v_{k-1} \\ \omega_{k-1} \end{bmatrix} + v_{k-1} \\ A &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} \cos \theta_{k-1} * dk & 0 \\ \sin \theta_{k-1} * dk & 0 \\ 0 & dk \end{bmatrix}, v_{k-1} = \begin{bmatrix} noise_{k-1} \\ noise_{k-1} \\ noise_{k-1} \end{bmatrix} \end{aligned} \quad (2.50)$$

State pada langkah waktu k adalah perhitungan menggunakan model state space, estimasi state, dan input kontrol vektor pada langkah waktu sebelumnya $k - 1$

$$\hat{r}_k = f(r_{k-1}, u_{k-1}) \quad (2.50)$$

model observasi didapatkan

$$z_k = Hr_k + \omega_k \quad (2.51)$$

ω_k adalah *noise* sensor dan H adalah matriks dengan jumlah dari baris adalah pengukuran sensor dan jumlah dari kolom adalah state. Matriks H adalah

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.52)$$

Pembaharuan nilai state \hat{r}_k'

$$\hat{r}'_k = \hat{r}_k + K(z_k - H_k \hat{r}_k) \quad (2.53)$$

K adalah gain kalman

$$K = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \quad (2.54)$$

R_k adalah covariance dari *noise* sensor. P_k di sini adalah error covariance matriks dan prediksi pertama menggunakan

$$P_k = F_k P_{k-1} F_k^T + Q_k \quad (2.55)$$

Q_k adalah proses *noise* covariance, F_k adalah equivalent terhadap matriks A dan pembaharuan

$$P'_k = P_k - K H_k P_k \quad (2.56)$$

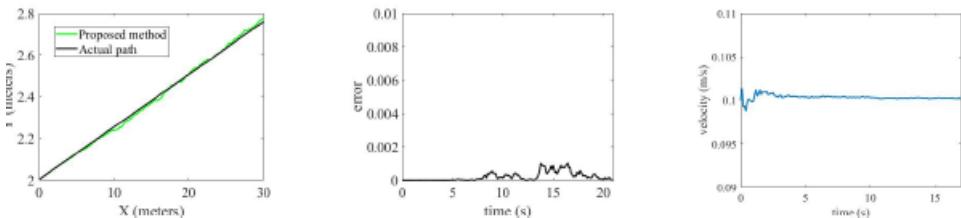
dari persamaan di atas, jelas bahwa *noise* sensor covariance R dan *noise* proses covariance Q adalah faktor penting yang ditentukan pada performa extended kalman filter. Dalam banyak kasus R diasumsikan konstan atau dapat diatur secara manual dengan trial and error. Bagaimanapun ini dapat mengakibatkan pada hasil akhir pada performa extended kalman filter dan ketidakakuratan estimasi dari objek bergerak.

Lintasan halangan diprediksi untuk waktu tertentu t yang merupakan waktu yang diperlukan untuk menghitung lintasan awal. Sekarang periksa apakah persimpangan jalur robot yang awalnya direncanakan dan jalur rintangan yang diperkirakan ada atau tidak. Jika terjadi perpotongan dua jalur maka jalur robot direncanakan ulang. Jalur baru tersebut sekarang menjadi jalur robot saat ini dan proses pengecekan jalur rintangan dan jalur robot dilanjutkan dan perencanaan ulang dilakukan saat kedua jalur berpotongan hingga posisi tujuan tercapai.

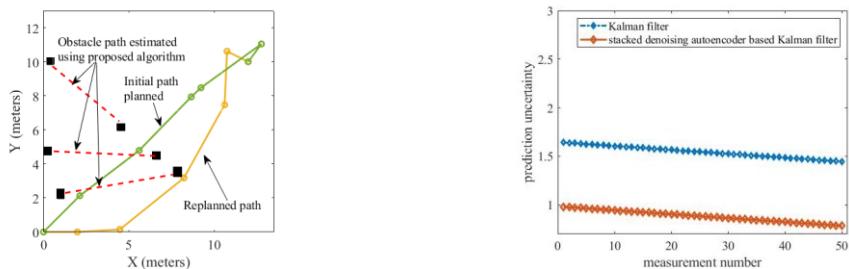
Hasil dari penelitian ini, Akurasi dari prediksi menggunakan *extended kalman filter* tergantung pada gain kalman yang selanjutnya tergantung pada matriks *noise* kovarians pengukuran.

Hambatan bergerak dideteksi menggunakan sensor ultrasonik. Setelah rintangan dinamis terdeteksi, jalur rintangan harus diperkirakan untuk memastikan navigasi bebas tabrakan. Jalur rintangan diprediksi menggunakan filter Kalman di

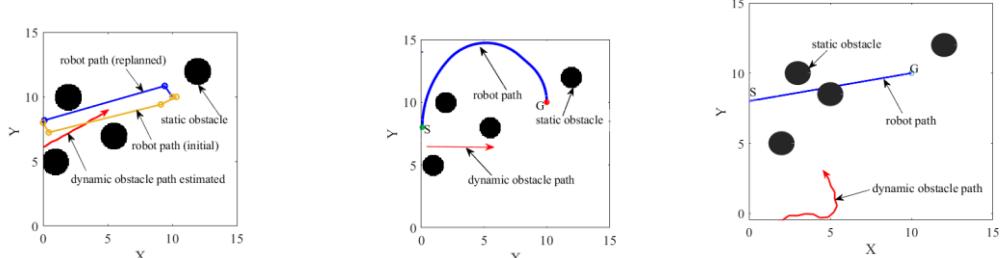
mana gain Kalman dihitung menggunakan Persamaan (12) yang akan ditentukan matriks kovarian *noise* pengukuran. Matriks kovarians *noise* pengukuran dihitung menggunakan Persamaan (15) dan diperoleh $R = [0.3 \ 0 \ 0; 0 \ 0.25 \ 0; 0 \ 0 \ 0.4]$. Perkiraan jalur rintangan dibandingkan dengan jalur robot yang direncanakan pada awalnya.



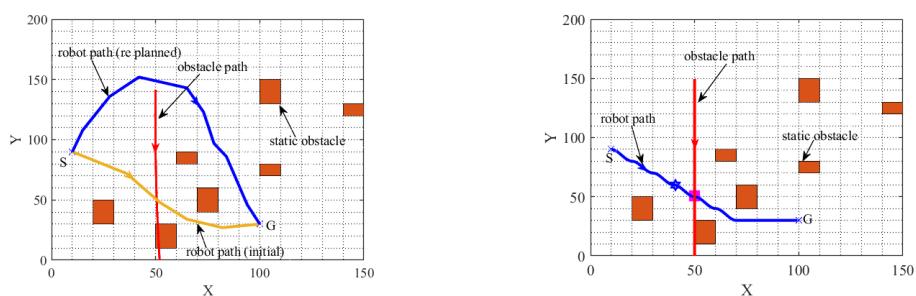
Gambar 2.15. Performa kalman filter dalam penelitian tersebut



Gambar 2.16. Gambar kanan path planing dan gambar kiri ketidakpastian prediksi



Gambar 2.17. Uji coba dengan algoritma penghindaran rintangan



Gambar 2.18. Perbandingan performa prediksi dengan penghindaran rintangan skenario
3

Dicoba dengan beberapa algoritma pada skenario 2, Posisi awal robot adalah (8,0) dan posisi tujuan adalah (10,10). gerak hambatan diasumsikan pada kecepatan konstan. Penelitian ini memiliki kelebihan robot dapat memprediksi gerak dari objek, setelah itu dapat melakukan perencanaan ulang jalur agar terhindar dari rintangan, namun dalam penelitian ini memiliki kekurangan robot bergerak secara linear atau garis lurus, sedangkan kenyataannya pada lingkungan robot berjalan dengan gerak non-linear, dan penggunaan EKF perlu dilakukan linearisasi sehingga perhitungan menjadi besar. Penelitian selanjutnya ditentukan jalur rintangan dengan gerak non-linear, menggunakan EnKF secara non-linear sekaligus melakukan *tracking* terhadap objek bergerak, dan dapat memprediksi bentuk objek serta posisi objek ketika di deteksi.

2.2 Teori Dasar

Pada sub-bab ini terdapat teori dasar yang menunjang dalam merumuskan dan menyelesaikan masalah yang akan dihadapi dalam menyusun tesis.

2.2.1 Ensemble Kalman Filter

Persamaan – persamaan umum untuk sistem non-linear *Ensemble Kalman Filter* [19] adalah sebagai berikut.

$$\begin{aligned} x_{k+1} &= Fx_k + Gv_k \\ y_k &= Hx_k + e_k \end{aligned} \quad (2.66)$$

Persamaan di atas adalah bentuk model linear dari state space yang digunakan dalam proses kalman filter, sedangkan untuk model non-linear adalah sebagai berikut

$$\begin{aligned} x_{k+1} &= f(x_k, v_k) \\ y_k &= h(x_k, e_k) \end{aligned} \quad (2.67)$$

Setelah menentukan model pengukuran, langkah berikutnya adalah menentukan ensemble

$$X_k^f \triangleq \left(x_k^{f_i}, \dots, x_k^{f_q} \right) \quad (2.68)$$

Ensemble dinotasikan dengan $X_k^f \in \mathbb{R}^{n \times q}$. Di mana f_i referensi ke i -th pada kelompok forecast ensemble. Rata-rata ensemble $\bar{x}_k^f \in \mathbb{R}^n$

$$\bar{x}_k^f \triangleq \frac{1}{q} \sum_{i=1}^q X_k^{f_i} \quad (2.69)$$

Ditentukan ensemble error matriks $E_k^f \in \mathbb{R}^{n \times q}$ dalam rata-rata ensemble dan memperkirakan P_k^f dengan \hat{P}_k^f , $p_{XY_k}^f$ dengan $\hat{P}_{XY_k}^f$ dan $P_{yy_k}^f$ dengan $\hat{P}_{yy_k}^f$.

$$\begin{aligned} E_k^f &\triangleq \left[x_k^{f_1} - \bar{x}_k^f \dots x_k^{f_q} - \bar{x}_k^f \right] \\ E_{yk}^u &\triangleq \left[y_k^{f_1} - \bar{y}_k^f \dots y_k^{f_q} - \bar{y}_k^f \right] \\ \hat{P}_k^f &\triangleq \frac{1}{q-1} E_k^f (E_k^f)^T \\ \hat{P}_{XY_k}^f &\triangleq \frac{1}{q-1} E_k^f (E_{yk}^u)^T \\ \hat{P}_{yy_k}^f &\triangleq \frac{1}{q-1} E_k^f (E_{yk}^u)^T \end{aligned} \quad (2.70)$$

Langkah kedua analisa langkah estimasi state, EnKF melakukan ensemble dengan data asimilasi secara pararel, untuk $i = 1, \dots, q$

$$\begin{aligned} x_k^{a_i} &= x_k^{f_i} + \hat{K}_k \left(y_k^i + v_k^i - h(x_k^{f_i}) \right) \\ y_k^i &= y_k + v_k^i \end{aligned} \quad (2.71)$$

di mana v_k^i adalah variabel acak zero-mean dengan distribusi normal dan matriks covariance R_k . Sampel matriks kesalahan covariance dihitung dari v_k^i konvergen terhadap R_k sebagai $q \rightarrow \infty$. Untuk analisa kesalahan covariance dari P_k^a dengan \hat{P}_k^a di mana

$$\hat{P}_k^a \triangleq \frac{1}{q-1} E_k^a E_k^{aT} \quad (2.72)$$

E_k^f yang didefinisikan pada (2.68a) dengan $x_k^{f_i}$ diganti dengan $x_k^{a_i}$ dan \bar{x}_k^f diganti dengan rata-rata dari hasil analisa pada estimasi kelompok ensemble. Dengan menggunakan gain kalman filter dan perkiraan error kovarian untuk menentukan gain filter \hat{K}_k

$$\hat{K}_k = \hat{P}_{xyk}^f (\hat{P}_{yyk}^f)^{-1} \quad (2.73)$$

Langkah terakhir adalah prediksi dari error statistik pada langkah forecast

$$x_{k+1}^{f_i} = f(x_k^{a_i}, u_k) + \omega_k^i \quad (2.74)$$

di mana ω_k^i adalah sampel dari distribusi normal dengan rata-rata nol dan kovarians Q_k . Matriks sampel error kovarinias dihitung dari ω_k^i konvergen ke Q_k dengan $q \rightarrow \infty$.

2.2.2 PDAF

2.2.2.1 Gambaran Umum PDAF

Dalam studi yang dirujuk sebagai [11] *tracking* objek dinamis berhasil dilakukan dengan menggunakan teknologi radar. Penelitian ini berfokus pada *tracking* kendaraan udara, khususnya pesawat dan helikopter. Data radar diproses secara rinci untuk memfasilitasi pelacakan efektif terhadap objek-objek tersebut.

Penelitian ini melibatkan berbagai skenario gerakan, termasuk situasi lepas landas, mendarat, dan rotasi searah serta berlawanan arah jarum jam. Setiap skenario menyajikan tantangan dan kebutuhan unik bagi sistem pelacakan, sehingga memberikan evaluasi komprehensif terhadap performanya. Traking objek diperlukan beberapa langkah-langkah yang perlu dilakukan agar dapat melakukan *tracking* trajectori dengan tepat, pada Gambar 2.19 menunjukkan flowchart dari algoritma PDAF untuk *tracking* objek. Pengukuran yang divalidasi, algoritma PDAF menghitung kemungkinan asosiasi ke target yang sedang dilacak. Algoritma pelacakan PDAF akan menggunakan data probabilistik atau bayesian untuk memperhitungkan ketidakpastian asal pengukuran. Tergantung pada asumsi persamaan state dan pengukurannya, algoritma PDAF didasarkan pada filter Kalman atau *Extended Kalman Filter*. Didasarkan pada filter Kalman jika persamaan state dan pengukuran diasumsikan linier. Sebaliknya, jika persamaan state dan pengukuran tidak linier, *Extended Kalman Filter*.

2.2.2.2 Asumsi PDAF

Asumsi yang digunakan pada PDAF sebagai berikut:

- 1) Di wilayah validasi, hanya ada satu target yang diinginkan (*target of interest*), yang statenya $x \in R^n$ berkembang dalam waktu sesuai dengan (2.75) dan persamaan pengukuran target $z(k) \in R^n$ (2.76)

$$x(k) = F(k-1)x(k-1) + v(k-1) \quad (2.75)$$

$$z(k) = H(k)x(k) + w(k) \quad (2.76)$$

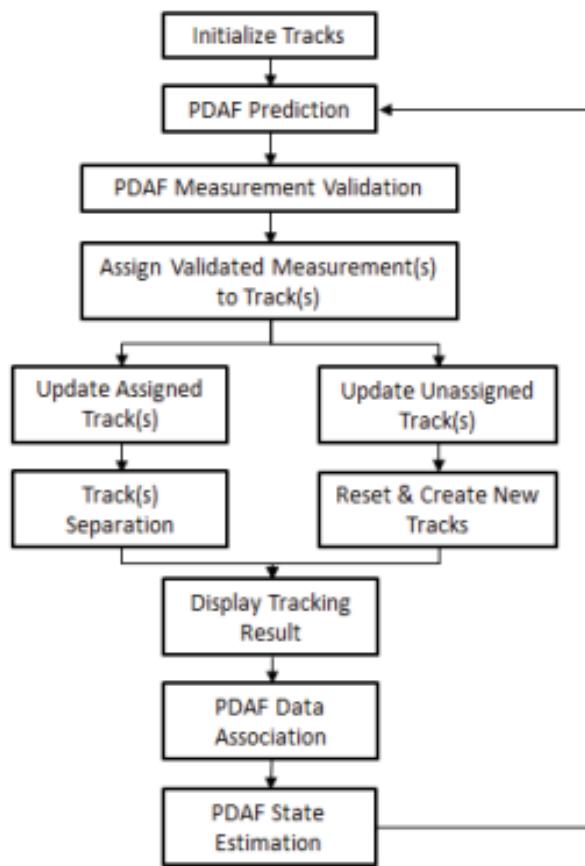
dengan $F(k - 1)$ adalah model dinamis, $H(k)$ adalah matriks pengukuran, serta $v(k - 1)$ dan $w(k)$ adalah *zero-mean mutually independent, white gaussian noise* dengan matriks kovarians yang diketahui, masing-masing $Q(k - 1)$ dan $R(k)$.

- 2) Target sudah memiliki jalur awal.
- 3) Informasi tentang masa lalu target melalui waktu $k - 1$ dapat didekati dengan persamaan dalam bentuk gaussian posterior

$$p[x(k - 1)|z(k - 1)] = N[x(k - 1); \hat{x}(k|k - 1), P(k|k - 1)] \quad (2.77)$$

dengan $N[x(k - 1); \hat{x}(k - 1)]$ adalah fungsi kepadatan probabilitas normal (PDF) dengan argumen $x(k)$, $\hat{x}(k - 1)$, dan matriks kovarians $P(k|k - 1)$.

- 4) Pada setiap waktu, wilayah validasi disiapkan di sekitar pengukuran yang terprediksi untuk memilih calon pengukuran untuk dikaitkan dengan target yang diinginkan.
- 5) Jika target terdeteksi dan pengukuran yang sesuai berada di dalam wilayah validasi, maka mengacu pada (2.76), terdapat satu target yang diinginkan.
- 6) Pengukuran yang tersisa dapat diabaikan karena dianggap sebagai *false alarm* atau *clutter*. *False alarm* dan *clutter* ini dimodelkan secara independen dan terdistribusi secara identik dengan distribusi spasial yang seragam.
- 7) *False alarm* atau *clutter* dapat diasumsikan terdistribusi Poisson, yang merupakan proses Poisson spasial dengan kerapatan spasial yang diketahui λ , atau menyebar sebelumnya.
- 8) Deteksi target dapat dimodelkan secara independen dari waktu ke waktu dengan probabilitas deteksi P_D yang diketahui.



Gambar 2.19. Flowchart sistem tracking dengan PDAF

BAB 3

DETEKSI DAN TRACKING RINTANGAN MENGGUNAKAN PDAF

3.1 Model Objek Rintangan Dinamis

Penelitian menggunakan beberapa tipe rintangan. Rintangan tipe pertama berupa objek statis, Dinding, pintu, meja dan semua objek yang tidak bergerak dalam ruang kantor. Tipe kedua dapat dijumpai dalam situasi saat manusia berinteraksi dengan ojek statis, Tipe ketiga berupa mobil robot lain yang beroperasi di lingkungan kerja yang sama dengan mobil robot *Differential Drive*. Tipe rintangan yang terakhir berupa individu manusia atau kelompok manusia yang bergerak dengan kecepatan berubah-ubah dan dengan lintasan non-linear. Individu manusia dapat dinyatakan menggunakan informasi pose dalam koordinat $X - Y$ $[x_{hi}, y_{hi}, \psi_{hi}]^T$, kecepatan v_{hi} , percepatan a_{hi} dan radius tubuh dari individu manusia ke- i^{th} dinotasikan dengan r_{hi} yang dapat dipengaruhi oleh jangkauan tangan atau alat bantu jalan dapat diformulasikan sebagai berikut

$$\mathbf{S}_{hi} = [x_{hi}, y_{hi}, \psi_{hi}, v_{hi}, a_{hi}, r_{hi}]^T \quad (3.1)$$

Objek bergerak multi modalitas merupakan rintangan yang bergerak dengan lintasan non-linear. Rintangan dalam lingkungan perkantoran, rumah sakit atau kampus dapat terdiri dari manusia yang bergerak dengan beberapa modalitas seperti berjalan, berlari atau dengan bantuan alat seperti tongkat atau kursi roda. Interaksi ini membentuk state sebagai berikut

$$\mathbf{S}_{h-so_m} = [x_{h-so_m}, y_{h-so_m}, r_{h-so_m}]^T \quad (3.2)$$

dengan (x_{h-so_m}, y_{h-so_m}) merepresentasikan titik pusat ke- m dari area HOI. Sedangkan radius dari area ini dinotasikan sebagai r_{h-so_m} . Titik pusat dan radius dari area HOI dapat dihitung sebagai berikut

$$(x_{h-so_m}, y_{h-so_m}) = (x_{so}, y_{so}) \quad (3.3)$$

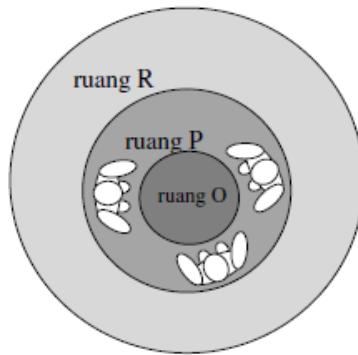
$$r_{h-so_m} = \sqrt{(x_{so} - x_h)^2 + (y_{so} - y_h)^2} \quad (3.4)$$

Interaksi sosial antar manusia dalam kondisi tidak bergerak membentuk formasi tatap muka (*face-to-face formation*) yang dikenal dengan F-Formation diilustrasikan pada Gambar 3.1. Geometri dari pergerakan manusia didekati

dengan model non-holonomic sebagaimana dapat dinyatakan dalam persamaan sebagai berikut

$$\begin{bmatrix} \dot{x}_h \\ \dot{y}_h \\ \dot{\psi}_h \end{bmatrix} = \begin{bmatrix} \cos \psi_h & 0 \\ \sin \psi_h & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_h \\ \omega_h \end{bmatrix} \quad (3.5)$$

dengan kecepatan gerak manusia dinyatakan sebagai $\mathbf{v}_h = [v_h \ \omega_h]^T$. posisi dan orientasi (pose) dari titik tengah bahu manusia dilambangkan dengan $[x_h, y_h, \psi_h]^T$ sebagai representasi dari manusia dalam koordinat global. Persamaan ini dapat menyusun transisi konfigurasi manusia dari koordinat tubuh ke global.

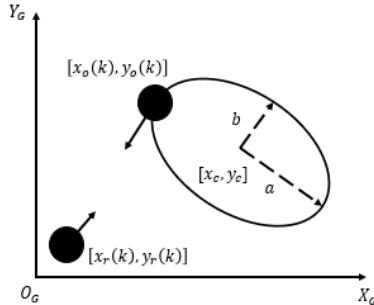


Gambar 3.1. Ruang interaksi antar manusia dalam F-Formation

Gerakan kelompok manusia dapat dimodelkan dengan diasumsikan kelompok manusia model lintasan non-linear objek rintangan bergerak multi modalitas. Penelitian ini melibatkan gerakan dengan perulangan yang dirancang secara khusus untuk membentuk lintasan-lintasan yang bervariasi. Tujuan dari gerakan ini adalah menciptakan lintasan-lintasan yang berbagai macam bentuk, yang mencakup lintasan oval (ellipse), daun semanggi (cloverleaf), dan Lissajous. Dalam penelitian ini, lintasan-lintasan ini dihasilkan dengan mengatur gerakan objek secara spesifik sehingga membentuk bentuk-bentuk yang diinginkan. Penggunaan lintasan oval (ellipse) memberikan gerakan melingkar yang halus, sedangkan lintasan daun semanggi (cloverleaf) memberikan pola perulangan yang rumit dan simetris. Sementara itu, lintasan Lissajous menghasilkan pola yang khas dan menarik dengan pergerakan yang harmonis. Lintasan oval diekspresikan sebagai berikut

$$\begin{aligned} x_o(k) &= x_c + a \cos(k\omega_0 T) \cos(k\omega_0 T) - b \sin(k\omega_0 T) \sin(k\omega_0 T) \\ y_o(k) &= y_c + a \cos(k\omega_0 T) \sin(k\omega_0 T) + b \sin(k\omega_0 T) \cos(k\omega_0 T) \end{aligned} \quad (3.6)$$

Titik pusat dari lintasan oval dinyatakan dengan x_c dan y_c dari titik pusat oval, $k = 0, 1, 2, \dots$, sedangkan ω_0 adalah kecepatan angular dari objek, dan T waktu sampling. Gambar 3.2 representasi dari sebuah objek dengan lintasan oval.

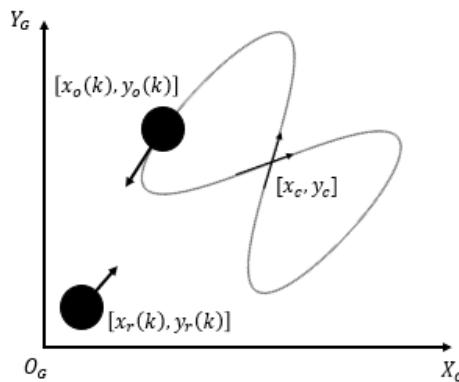


Gambar 3.2. Objek bergerak dengan lintasan oval

Objek yang bergerak dengan pola lintasan berulang dapat pada objek bergerak dengan lintasan *Lissajous* sebagaimana dapat dilihat pada Gambar 3.3. Lintasan *Lissajous* dapat diformulasikan sebagai berikut.

$$\begin{aligned} x_o(k) &= x_c + a \sin(k\omega_0 T) \\ y_o(k) &= y_c + b \cos(k\omega_0 T) \end{aligned} \quad (3.7)$$

titik pusat dari lintasan oval dinyatakan dengan x_c dan y_c dengan amplitudo a , $k = 0, 1, 2, \dots$, sedangkan ω_0 adalah kecepatan angular dari objek, dan T waktu sampling. Gerakan tanpa perulangan terdiri dari beberapa lintasan non-linear.



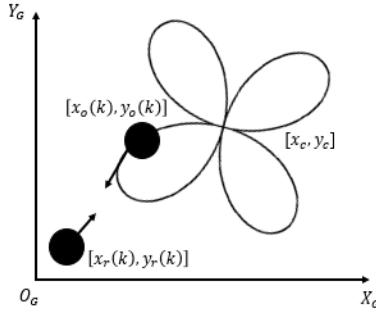
Gambar 3.3. Objek bergerak dengan lintasan Lissajous

Lintasan terakhir adalah lintasan *cloverleaf* sama seperti *Lissajous* menghasilkan lintasan berulang pada objek bergerak yang diilustrasikan pada Gambar 3.4, sehingga dapat diformulasikan sebagai berikut

$$x_o(k) = x_c + a \sin(k\omega_0 T) \cos(2k\omega_0 T) \quad (3.8)$$

$$y_o(k) = y_c + a \cos(k\omega_0 T) \sin(2k\omega_0 T)$$

titik pusat dari lintasan oval dinyatakan dengan x_c dan y_c dengan amplitudo a , $k = 0, 1, 2, \dots$, sedangkan ω_0 adalah kecepatan angular dari objek, dan T waktu sampling.



Gambar 3.4. Objek bergerak dengan lintasan Cloverleaf [21]

3.2 Trajectory Tracking Mobile Robot

Penelitian ini mengusulkan beberapa skenario untuk gerakan dari rintangan, seperti gerakan berulang yang ditunjukkan pada Gambar 3.2, Gambar 3.3, dan Gambar 3.4, sedangkan untuk gerakan tanpa perulangan terdiri dari beberapa lintasan non-linear. Lintasan ini dihasilkan dengan menerapkan *Probabilistic Road Map* (PRM), pada *global path planner*, perencanaan lintasan PRM digunakan untuk membangkitkan lintasan acak dari suatu titik awal menuju ke suatu titik tujuan atau target tertentu sebagai referensi bagi setiap objek dalam lingkungan pengujian. Tugas PRM sebagai perencana lintasan dalam tingkat global dalam sistem navigasi mobile robot ini memiliki nilai strategis untuk menghemat waktu komputasi. Hal ini dapat terjadi karena proses perencanaan lintasan hanya dilakukan sekali sehingga dapat menghindari *replanning* yang membutuhkan waktu komputasi lebih banyak. Beberapa data perlu disiapkan untuk menghasilkan lintasan non-linear dari gerakan tanpa perulangan untuk mobile robot ataupun objek rintangan. Peta lingkungan dari area kerja adalah input awal untuk PRM. Dari peta ini, titik-titik simpul (*node*) dibangkitkan secara acak. *Node* ini akan dihubungkan untuk menjadi jalur bagi mobile robot ataupun objek rintangan dari titik awal menuju titik tujuan dengan $[x_o(k), y_o(k)]$ dalam koordinat kartesian, sehingga jalur ini adalah jalur referensi bagi objek yang bekerja dalam lingkungan.

Dalam penelitian ini, *local path planner* menggunakan algoritma *Pure Pursuit* (PP) di pasang bersama metode penghindaran rintangan pada penelitian [16] dan [17] bertujuan untuk mengawasi dan melacak pergerakan mobile robot atau objek rintangan dari posisi awal hingga mencapai tujuan yang ditentukan. Algoritma PP memastikan bahwa robot atau objek rintangan mengikuti jalur yang diinginkan dengan presisi, dan dalam hal ini, *local path planner* berfungsi sebagai pemandu yang mengarahkan pergerakan robot atau objek rintangan. Dengan menggunakan algoritma PP, robot atau objek rintangan dapat mencapai posisi tujuan dengan akurasi tinggi dan meminimalkan potensi deviasi dari jalur yang diinginkan. Tujuan utama penggunaan metode ini adalah untuk mengembangkan sistem navigasi yang andal dan efektif dalam lingkungan yang kompleks, sehingga memungkinkan robot atau objek rintangan untuk mencapai tujuan dengan keamanan dan efisiensi yang maksimal.

3.3 Simultaneous Localization and Mapping (SLAM)

Teori dasar dari SLAM berputar di sekitar dua masalah fundamental [22]:

Lokalisasi: Mengestimasi posisi dan orientasi (pose) agen dalam lingkungan.

Pemetaan: Membuat gambaran lingkungan sekitar, biasanya dalam peta 2D atau 3D. Posisi robot dan orientasi terhadap lingkungannya membentuk x_k . Ini diambil dari waktu $k = 0$ dan didefinisikan sebagai jalur robot x_k .

$$X_k = \{x_0, x_1, x_2, x_3, \dots, x_k\} \quad (3.9)$$

Untuk pergerakan relatif U_k antara dua langkah waktu, $k - 1$ dan k , ini didefinisikan sebagai

$$U_k = \{u_0, u_1, u_2, u_3, \dots, u_k\} \quad (3.10)$$

Bergantung hanya pada odometri U_k dari satu robot saja untuk menentukan posisinya dalam suatu bidang tidak cocok, karena dalam aplikasi dunia nyata, hal tersebut kekurangan presisi yang diperlukan untuk lokalisasi yang akurat. Serangkaian pengukuran sensor Z_k di setiap langkah waktu didefinisikan sebagai berikut:

$$Z_k = \{z_0, z_1, z_2, z_3, \dots, z_k\} \quad (3.11)$$

Langkah-langkah berikut melibatkan pembuatan representasi lingkungan dan memprediksi lokasi setelah mengumpulkan dan mendefinisikan semua data

yang penting. Dengan menggabungkan distribusi probabilitas untuk mengestimasi posisi robot dan penanda berdasarkan peta yang dibuat, metodologi SLAM menggunakan pendekatan probabilistik. Diagram pada Gambar 3.5 menggambarkan distribusi probabilitas ini. Untuk menyediakan lokalisasi dan pemetaan yang kuat dan fleksibel, ini menangkap ketidakpastian dan variasi dalam posisi yang diprediksi. Sebagaimana yang didefinisikan oleh P .

$$P(x_k, m | Z_k, U_k) \quad (3.12)$$

Mempertimbangkan pengukuran historis dan data odometri, kemungkinan posisi pada waktu k dan peta dapat dipahami. ini juga membutuhkan hubungan kedua yang dikenal sebagai model pengamatan. Model pengamatan menentukan hubungan berikut antara posisi robot x_k dan odometrinya u_k .

$$P(x_k | x_{k-1}, u_k) \quad (3.13)$$

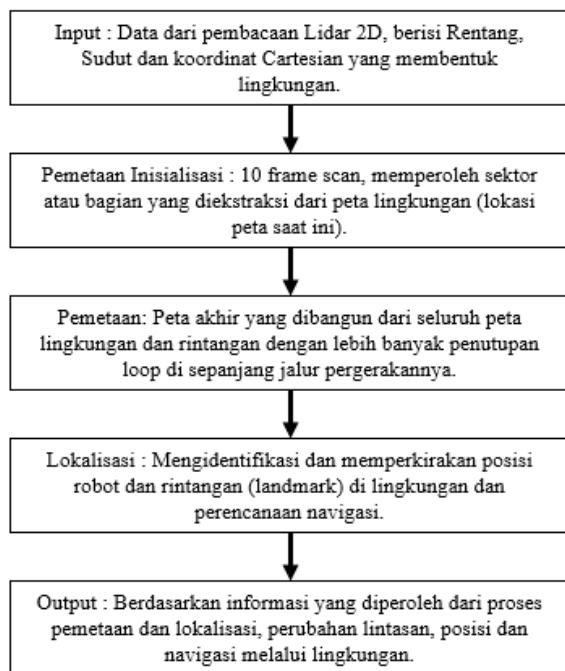
Robot dapat memahami pengidentifikasi berbeda, orientasi relatif, dan jarak ke penanda dalam sebagian besar aplikasi. Pemahaman ini menjadi dasar untuk membangun model pengukuran. Ini dapat dijelaskan sebagai berikut dengan menggunakan distribusi probabilitas dalam model pengukuran:

$$P(z_k | x_k, m) \sim N(h(x_k, m), Q_k) \quad (3.14)$$

Di mana N adalah distribusi normal dua dimensi, Q_k adalah kovariansi noise dua dimensi, dan $h(x_k, m)$ adalah fungsi sembarang yang mewakili operasi peralatan sensor. Variabel berikut didefinisikan di bawah ini:

- k : Time instant.
- x_k : Lokasi robot.
- X_k : Urutan lokasi atau jalur robot.
- u_k : Odometri antara waktu $k - 1$ dan k .
- U_k : Urutan odometri robot atau gerakan relatif.
- Z_k : Urutan pengukuran antara robot dan lingkungan.

Proses SLAM pada penelitian ini dirancang dengan maximum range lidar 20 meter, sedangkan resolusi map adalah 18. Parameter yang digunakan dalam menjalankan algoritma SLAM adalah *loop threshold* dan *loop search radius* masing-masing nilainya adalah 320 dan 8, dan dimodifikasi dengan beberapa nilai-nilai lain untuk setiap skenario.



Gambar 3.5. Langkah umum SLAM dilakukan pada differential drive mobile robot

3.4 Euclidean Clustering

Jarak Euclidean menghitung jarak linier antara dua titik dalam ruang Euclidean. Diberikan dua titik, $P(x_1, y_1)$ dan $Q(x_2, y_2)$, jarak Euclidean di antara mereka dihitung sebagai berikut:

$$d(P, Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.15)$$

Setelah hambatan dan lingkungan terdeteksi, gradien untuk setiap titik dalam kluster harus dihitung. Misalkan xy adalah *Point Cloud* dari data LIDAR. Gradien dihitung sebagai perbedaan antara titik-titik berturut-turut dalam xy . Jika G adalah matriks gradien berukuran $(m - 1) \times 2$:

$$G_i = xy_{\{i+1\}} - xy_i, \text{ untuk } i = 1, 2, \dots, m - 1 \quad (3.16)$$

Lalu normalisasi vektor gradien. Normalisasi setiap baris dari matriks G dengan membaginya dengan norma Euclidean-nya. Jika N adalah matriks gradien yang di-normalisasi berukuran $(m - 1) \times 2$:

$$N_i = \frac{G_i}{\|G_i\|_2}, \text{ untuk } i = 1, 2, \dots, m - 1 \quad (3.17)$$

Hitung sudut antara vektor gradien berturut-turut. Hitung sudut antara baris berturut-turut dari matriks gradien yang dinormalisasi N . Jika A adalah vektor berukuran $(m - 2) \times 1$, yang berisi sudut-sudut:

$$A_i = \arccos(N_i \cdot N_{\{i+1\}}), \text{ untuk } i = 1, 2, \dots, m-2 \quad (3.18)$$

Algoritma untuk deteksi lingkungan ditunjukkan dalam Algoritma 1.

ALGORITHM 1: Classify Clusters

Input: xy - matrix of cluster points, c - cluster index vector, $centroids$ - matrix of cluster centroids.
Output: classification - list of cluster classifications, dimensions - matrix of cluster dimensions.

```

1 Initialization: Initialize empty lists 'classification' and 'dimensions.'
2 For each cluster  $i$  in the input data:
3   | If the number of points in cluster  $i > 1$ 
4     | Compute gradient and angles between consecutive points in  $xy$ 
5     | Calculate total_angle and point_count
6     | If abs(total_angle - pi/2) < 0.1 or point_count >= 10
7       | Classify cluster as 'Wall.'
8       | Compute wall dimensions
9     | Else
10    |   | Classify cluster as 'Square.'
11    |   | Compute wall dimensions
12    | Else
13    |   | Classify cluster as 'Circle.'
14    |   | Compute mean circle radius
15    | Else
16    |   | Classify cluster as 'Empty.'
17 Return 'classification' and 'dimensions.'
```

Klasifikasikan kluster sebagai dinding atau lingkaran berdasarkan jumlah sudut dan hitungan titik. Hitung total sudut dengan menjumlahkan elemen-elemen dari vektor A , dan tentukan jumlah titik dalam kluster, yang sama dengan jumlah elemen dalam vektor indeks kluster c .

$$total_{angle} = \sum A_i, \text{ untuk } i = 1, 2, \dots, m-2 \quad (3.19)$$

$$point_{count} = n, \text{ dimana } n \text{ adalah nilai dari elemen dalam } c \quad (3.20)$$

Setelah berhasil melakukan deteksi objek sekitar lingkungan, selanjutnya mendapatkan titik X dan Y sebagai referensi PDAF untuk melakukan trajectory tracking atau bisa dikatakan titik *centroid*. Untuk mendapatkan titik tersebut dengan menentukan titik *centroid cluster* sesuai dengan diameter yang telah didapatkan. Semua titik pada *cluster* rintangan yang bergerak dalam satu iterasi ketika objek berhasil di deteksi. Ilustrasi dari titik *centroid* ditunjukkan pada Gambar 3.6.

Jarak ke *centroid* objek d_{xy} dengan mencari titik jarak yang memiliki nilai minimum dari jarak setiap titik pada *cluster*, kemudian tambahkan jari-jari r yang digunakan pada persamaan 3.21.

$$d_{xy} = \min(d_i) + r \quad (3.21)$$

Hitung minimum diameter objek ω_{min} yang ditentukan oleh titik jarak pertama dan terakhir dari *cluster* (dengan kata lain, sinar laser pertama dan terakhir mengenai objek) dan maksimum untuk diameter objek ω_{max} yang ditentukan oleh sinar laser sebelum sinar laser pertama mengenai objek dan sinar laser setelah sinar laser terakhir mengenai objek. Diameter lidar dianggap k

$$\omega_{min} \leq k < \omega_{max} \quad (3.22)$$

Perhitungan ω_{min} dan ω_{max} tergantung pada jarak antara robot dan pusat objek yang diketahui pada persamaan 3.26. Selain itu, itu tergantung pada sudut yang tepat di antara sinar. Bahwa jumlah sinar yang mengenai objek dari masing-masing sisi mungkin berbeda. Minimum diameter benda ω_{min} akan menjadi hasil dari menggabungkan bagian kiri ω_{minl} dan kanan ω_{minr} dari ω_{min}

$$\begin{aligned} \omega_{minl} &= d_{xy} \times \tan(a_{minl}) \\ \omega_{minr} &= d_{xy} \times \tan(a_{minr}) \\ \omega_{min} &= \omega_{minl} + \omega_{minr} \end{aligned} \quad (3.23)$$

sama seperti minimum, maksimum diameter objek ω_{max} akan menghasilkan penggabungan antara kiri ω_{maxl} dan kanan ω_{maxr} . Setalah nilai d_{xy} didapatkan akan dikonversi menjadi nilai x dan y sebagai *centroid* objek untuk dilakukan tracking objek lintangan bergerak.

$$\begin{aligned} \omega_{maxl} &= d_{xy} \times \tan(a_{maxl}) \\ \omega_{maxr} &= d_{xy} \times \tan(a_{maxr}) \\ \omega_{max} &= \omega_{maxl} + \omega_{maxr} \end{aligned} \quad (3.24)$$

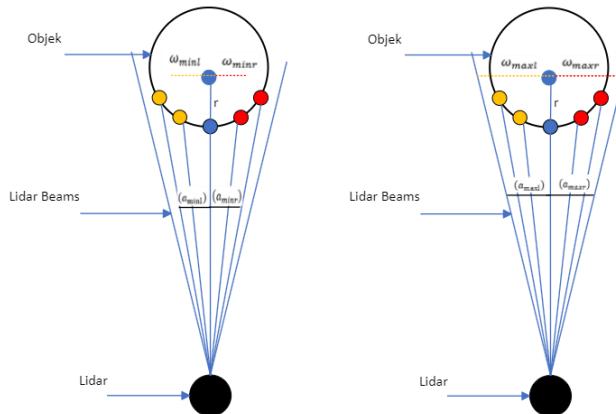
3.5 Filtering Data PDAF Dengan Ensemble Kalman Filter

Prediksi dan estimasi state untuk setiap trajectory lintangan sekaligus juga dilakukan *tracking* pada lintangan tersebut dapat menggunakan PDAF, di mana dalam algoritma PDAF mencakup *filtering* untuk penelitian ini *filtering* data menggunakan EnKF, sehingga sekaligus dapat melakukan prediksi dan estimasi dengan memodifikasi algoritma PDAF dengan EnKF pada data filter. Flowchart

sistem *tracking* dengan PDAF pada Gambar 2.19 dimodifikasi dengan EnKF ditunjukkan pada Gambar 3.7. Hasil Euclidean *Clustering* dengan menentukan *centroid* objek sesuai dengan persamaan (3.21) dijadikan titik referensi untuk melakukan objek *tracking*.

Algoritma secara keseluruhan PDAF termodifikasi data filter dengan EnKF dapat dilihat pada Gambar 3.8 yang mana merangkum satu siklus dari PDAF yang serupa dengan kalman filter, namun dengan tambahan beberapa fitur sebagai berikut:

- 1) PDAF memiliki prosedur seleksi untuk pengukuran tervalidasi
- 2) Untuk setiap pengukuran, probabilitas asosiasi dihitung sebagai pembobotan pengukuran dalam gabungan inovasi. Kemudian, gabungan inovasi digunakan dalam pembaruan estimasi state.
- 3) Kovarians state terakhir yang diperbarui menjelaskan untuk ketidakpastian asal pengukuran.



Gambar 3.6 Perhitungan untuk mencari titik centroid dari objek setiap objek

A. Prediksi

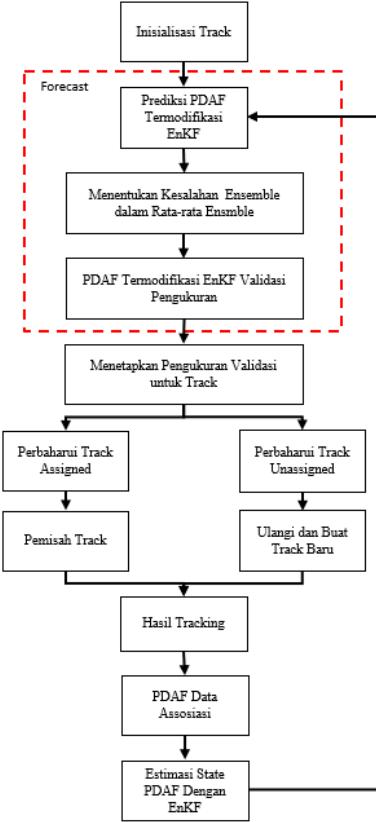
Persamaan matriks state yang diprediksi, pengukuran, prediksi rata-rata, pengukuran rata-rata, dan kovarian state dari PDAF dengan inisialisasi $x^a = [x^{a_1}, \dots, x^{a_q}]$ dengan nilai q adalah jumlah ensemble state yang digunakan adalah

$$\hat{x}^{a_q}(k|k-1) = F(k-1)\hat{x}^{a_q}(k-1) + v(k-1) \quad (3.25)$$

$$\hat{z}^{a_q}(k|k-1) = H(k)\hat{x}^{a_q}(k|k-1) + w(k) \quad (3.26)$$

$$\bar{x}^{a_q}(k|k-1) = \left(\frac{1}{q}\right) \sum_{i=1}^q \hat{x}^{a_q}(k|k-1) \quad (3.27)$$

$$\bar{z}^{aq}(k|k-1) = \left(\frac{1}{q}\right) \sum_{i=1}^q \hat{z}^{aq}(k|k-1). \quad (3.28)$$



Gambar 3.7. Flowchart tracking sistem PDAF termodifikasi EnKF

Selisih dari setiap nilai prediksi dan pengukuran dengan rata-rata prediksi dan pengukuran masing-masing disimpan dalam matriks ensemble error prediksi E_k^f dan matriks ensemble error pengukuran E_{xyk}^a .

$$E_k^a \triangleq [\hat{x}^{a_1}(k|k-1) - \bar{x}^{a_1}(k|k-1) \dots \hat{x}^{a_q}(k|k-1) - \bar{x}^{a_q}(k|k-1)] \quad (3.29)$$

$$E_{xyk}^a \triangleq [\hat{z}^{a_1}(k|k-1) - \hat{z}^{a_1}(k|k-1) \dots \hat{z}^{a_q}(k|k-1) - \hat{z}^{a_q}(k|k-1)] \quad (3.30)$$

$$P_k(k|k-1) = \left(\frac{1}{q}\right) E_k^a (E_k^a)' + Q(k-1) \quad (3.31)$$

$$P_{xyk}(k|k-1) = \left(\frac{1}{q}\right) E_{xyk}^a (E_{xyk}^a)' + Q(k-1) \quad (3.32)$$

Dengan $P(k-1|k-1)$ tersedia dari (2.77). Kovarian inovasi yang sesuai dengan pengukuran yang benar adalah

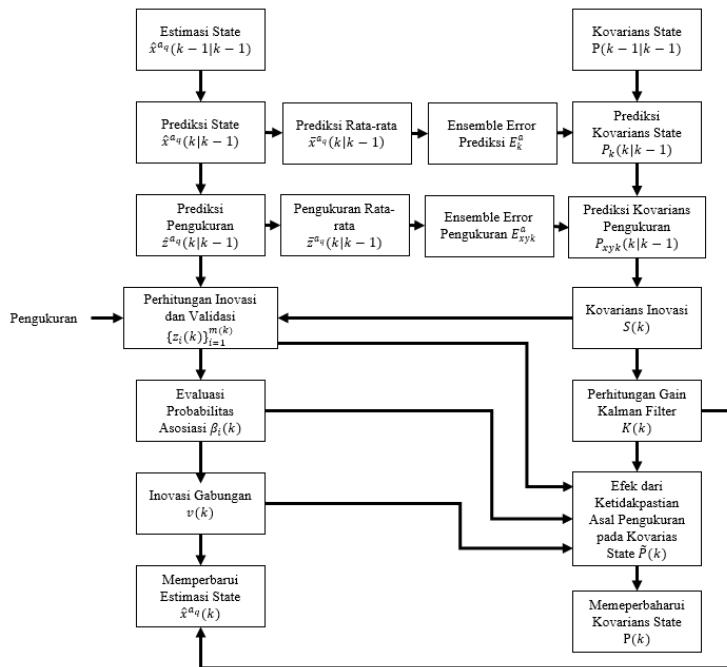
$$S(k) = H(k)P_k(k|k-1)P_{xyk}(k|k-1)P(k-1|k-1)H(k)' + R(k) \quad (3.33)$$

B. Validasi Pengukuran

Dari asumsi gaussian (2.77), daerah validasi berbentuk elips

$$V(k, \gamma) = \{z: [z - \bar{z}^{aq}(k|k-1)]' S(k)^{-1} [z - \bar{z}^{aq}(k|k-1)] \leq \gamma\} \quad (3.34)$$

dengan γ adalah ambang gerbang (*Gate Threshold*) yang sesuai dengan probabilitas gerbang P_G , yang merupakan probabilitas bahwa gerbang berisi pengukuran yang benar. Untuk volume dari wilayah validasi adalah



Gambar 3.8. Algoritma EnKF pada PDAF

$$V(k) = c_{n_z} |\gamma S(k)|^{\frac{1}{2}} = c_{n_z} \gamma^{\frac{n_z}{2}} |S(k)|^{\frac{1}{2}} \quad (3.35)$$

dengan koefisien c_{n_z} bergantung pada dimensi n_z dari pengukuran. Contoh, $c_1 = 2$, $c_2 = \pi$, dan $c_3 = 4\pi/3$. Kumpulan data pengukuran yang tervalidasi menurut (3.34) adalah

$$z(k) = \{z_i(k)\}_{i=1}^{m(k)} \quad (3.36)$$

dengan $m(k)$ adalah jumlah pengukuran yang tervalidasi saat pemindaian k .

C. Assosiasi Data

Untuk mengevaluasi probabilitas asosiasi, sampel data dibagi menjadi data masa lalu Z^{k-1} dan data masa sekarang $Z(k)$. Sebuah inferensi probabilistik dapat didefinisikan sebagai

$$\beta_i(k) = P\{\theta_i(k)|Z^k\} = P\{\theta_i(k)|Z^k, m(k), Z^{k-1}\}, \quad (3.37)$$

menggunakan rumus bayesian, persamaan di atas dapat ditulis menjadi

$$\beta_i(k) = \frac{1}{c} p[Z(k)|\theta_i(k), m(k), Z^{k-1}] p\{\theta_i(k)|m(k), Z^{k-1}\}, \quad (3.38)$$

Dengan $i = 0, \dots, m(k)$. Kepadatan gabungan (*joint density*) dari pengukuran yang divalidasi pada $\theta_i(k), i \neq 0$, adalah hasil dari :

- PDF gaussian dari pengukuran yang benar
- PDF dari pengukuran yang salah, yang diasumsikan seragam di wilayah validasi

PDF dari pengukuran yang benar adalah

$$\begin{aligned} p[Z(k)|\theta_i(k), m(k), Z^{k-1}] &= p_G^{-1} N[z_i(k); \bar{z}^{a_q}(k|k-1), S(k)] \\ &= p_G^{-1} N[v_i(k); 0, S(k)]. \end{aligned} \quad (3.39)$$

Dari persamaan di atas, probabilitas kejadian asosiasi hanya pada sejumlah pengukuran tervalidasi adalah

$$\begin{aligned} p[Z(k)|\theta_i(k), m(k), Z^{k-1}] \\ = \begin{cases} V(k)^{-m(k)+1} p_G^{-1} N[v_i(k); 0, S(k)], & i = 1, \dots, m(k) \\ V(k)^{-m(k)}, & i = 0 \end{cases} \end{aligned} \quad (3.40)$$

Dengan menggunakan model poisson (*parametric*) dalam (3.40) menghasilkan.

$\gamma_i[m(k)]$

$$= \begin{cases} P_D P_G [P_D P_G m(k) + (1 - P_D P_G) \lambda V(k)]^{-1}, & i = 1, \dots, m(k) \\ (1 - P_D P_G) \lambda V(k) [P_D P_G m(k) + (1 - P_D P_G) \lambda V(k)]^{-1}, & i = 0 \end{cases} \quad (3.41)$$

Model persebaran (*non-parametric*) juga dapat diperoleh dari (3.40) dengan mendefinisikan.

$$\lambda = \frac{m(k)}{V(k)} \quad (3.42)$$

Persamaan akhir dari *parametric* PDAF dengan model *clutter* poisson dan densitas spasial λ yang memodelkan *clutter* menghasilkan probabilitas asosiasi $z_i(k)$ merupakan pengukuran yang benar,

$$\beta_i(k) = \begin{cases} \frac{L_i(k)}{1 - P_D P_G + \sum_j^{m(k)} L_i(k)}, & i = 1, \dots, m(k) \\ \frac{1 - P_D P_G}{1 - P_D P_G + \sum_j^{m(k)} L_i(k)}, & i = 0 \end{cases} \quad (3.43)$$

dengan $i = 0$ menyatakan bahwa tidak ada pengukuran yang benar,

$$L_i(k) \triangleq \frac{N[z_i(k); \bar{z}^{a_q}(k|k-1), S(k)]P_D}{\lambda} \quad (3.44)$$

merupakan rasio *likelihood* dari pengukuran $z_i(k)$ yang berasal dari target daripada *clutter*, dan persamaan akhir untuk *non-parametric* PDA sama dengan (3.43), namun mengganti λ pada (3.44) seperti (3.42).

D. Estimasi State

Estimasi state atau pembaruan state pada EnKF, didapatkan dengan persamaan dari pembaruan state PDAF adalah

$$\hat{x}^{a_q}(k) = \hat{x}^{a_q}(k|k-1) + K(k)[\hat{z}^{a_q}(k) - H(k)\hat{x}^{a_q}(k|k-1)] \quad (3.45)$$

$$\hat{z}^{a_q}(k) = \hat{z}^{a_q}(k|k-1) + v(k) \quad (3.46)$$

dengan *gain* $K(k)$ adalah

$$K(k) = P_k(k|k-1)P_{xyk}(k|k-1)H(k)'S(k)^{-1}, \quad (3.47)$$

dan inovasi gabungan $v(k)$ adalah

$$v(k) = \sum_{i=1}^{m(k)} \beta_i(k)v_i(k), \quad (3.48)$$

Kovarians dari estimasi state dalam (3.47) adalah

$$P(k) = \beta_0(k)P_k(k|k-1)P_{xyk}(k|k-1) + [1 - \beta_0(k)]P^c(k) + \tilde{P}(k), \quad (3.49)$$

dengan kovarians dari estimasi state dengan pengukuran yang benar $P^c(k|k)$ adalah

$$P^c(k) = P_k(k|k-1)P_{xyk}(k|k-1) - K(k)S(k)K(k)', \quad (3.50)$$

dan penyebaran inovasi gabungan $\tilde{P}(k)$ di (3.50)

$$\tilde{P}(k) \triangleq K(k) \left[\sum_{i=1}^{m(k)} \beta_i(k)v_i(k)v_i(k)' - v(k)v(k)' \right] K(k)'. \quad (3.51)$$

Jika asal pengukuran tidak pasti, maka istilah $\tilde{P}(k)$ akan meningkatkan kovarians dari estimasi state. Kenaikan ini terjadi karena ketergantungan akurasi estimasi pada (3.51), yang menunjukkan bahwa PDAF merupakan estimator non-linear.

Hasil dari proses estimasi, kemudian digunakan sebagai referensi dasar dalam menghasilkan prediksi trajectory $E(k + 1)$. Dalam konteks ini, trajectory $E(k + 1)$ mengacu pada perkiraan trajektori atau lintasan pergerakan objek berikutnya. Persamaan dalam konteks prediksi trajectory berikutnya menggunakan persamaan ensemble pada persamaan (3.51), data hasil persamaan tersebut yang digunakan untuk prediksi langkah berikutnya.

3.6 Analisa Data

Setelah data simulasi diproses secara keseluruhan, langkah berikutnya adalah analisis data. Dalam proses ini, beberapa hal yang akan dianalisis, yaitu:

1. Grafik perbandingan data posisi objek asli, prediksi, dan tervalidasi dalam koordinat kartesian.
2. Perhitungan error menggunakan RMSRE terhadap data posisi objek asli dan prediksi pada koordinat kartesian, serta menggunakan MAE terhadap data posisi objek asli dan prediksi pada koordinat kartesian.
3. Perhitungan error filter yaitu perhitungan error terhadap data posisi objek asli dan prediksi pada koordinat kartesian dan ditampilkan dalam plot grafik diskrit.

Perbedaan data posisi akan ditampilkan dalam grafik pada poin 1. Grafik akan dimulai dari frame pertama hingga frame terakhir ketika robot sudah mendapatkan titik pada koordinat kartesian untuk setiap objek yang memiliki state pelacakan.

Perhitungan error akan dievaluasi pada poin 2. Perhitungan RMSRE dari data posisi objek asli dan prediksi pada koordinat kertesian dan perhitungan MAE dari data posisi objek asli terhadap data prediksi pada koordinat kartesian adalah perhitungan error yang dimaksud. Perhitungan MAE ditunjukkan pada (3.52) dan perhitungan RMSRE ditunjukkan pada (3.53).

$$mae = \frac{1}{m} \sum_{i=1}^m |\hat{z}_i - z_i| \quad (3.52)$$

$$rmsre = \sqrt{\frac{1}{m} \sum_{i=1}^n \left(\frac{\hat{z}_i - z_i}{z_i} \right)^2} \quad (3.53)$$

dengan m adalah indeks sapuan pada pelacak yang hanya mengandung state pelacak selama proses pelacakan objek menggunakan algoritma PDAF. Pada poin 3, perhitungan error filter dari data posisi 2D objek asli dan data terprediksi akan ditampilkan dalam bentuk grafik diskrit.

3.7 Identifikasi Lidar

Pembacaan LIDAR 2D menghasilkan keluaran berupa *range*, *angles*, dan kartesian, di mana pada nilai kartesian akan di ubah ke dalam *point cloud* digunakan untuk identifikasi objek mulai dari deteksi objek hingga *tracking* objek. Diperlukan beberapa langkah untuk melakukan *tracking* dengan menggunakan metode yang PDAF sebagai berikut:

1. Pemrosesan Data LIDAR: Data dari sensor LIDAR harus diproses terlebih dahulu untuk menghasilkan informasi yang berguna untuk pelacakan objek.
2. Inisialisasi Objek: Langkah pertama dalam PDAF adalah menginisialisasi objek yang akan dilacak, tahap ini dilakukan dengan menentukan posisi awal objek dan membangun model prediksi gerak untuk objek tersebut.
3. Prediksi Objek: Berdasarkan model prediksi gerak yang telah dibangun, langkah selanjutnya adalah memperbarui posisi prediksi objek pada waktu berikutnya.
4. Pengukuran: Data LIDAR selanjutnya digunakan untuk mengukur objek yang sedang dilacak. Ini melibatkan pencocokan titik-titik LIDAR dengan objek yang di prediksi pada langkah sebelumnya.
5. Asosiasi Data: Dalam langkah ini, PDAF menggunakan metode probabilistik untuk mengaitkan setiap pengukuran dengan objek yang sesuai.
6. Pembaharuan: Setelah asosiasi data dilakukan, langkah terakhir adalah memperbarui estimasi posisi objek yang sedang dilacak berdasarkan pengukuran yang diasosiasikan.

Berdasarkan persamaan (2.75) dan (2.76) penelitian ini menggunakan $v(k - 1)$ dan $w(k)$ adalah *zero-mean mutually independent, white gaussian noise* dengan matriks kovarian diketahui. Pada data pengukuran dalam data LIDAR ditambahkan noise dengan variance noise 10^{-4} .

3.8 Skema Sistem Keseluruhan

Proses SLAM membuat peta lingkungan dan memperbarui posisi robot. Ini membantu robot tahu dengan pasti di mana ia berada. Metode SLAM mudah digunakan di MATLAB. Algoritma clustering geometris mengelompokkan titik LIDAR yang dekat menjadi objek. Ini membantu mengidentifikasi objek yang penting untuk menghindari rintangan, seperti dinding atau objek bergerak. Clustering membantu mengatur data LIDAR yang rumit menjadi objek yang jelas. Pelacakan objek menggunakan PDAF dengan EnKF untuk filter data. PDAF membantu memperbarui perkiraan posisi dan kecepatan objek setelah objek diidentifikasi dan diprediksi sebelumnya. Estimasi objek dapat disesuaikan berdasarkan pengukuran LIDAR baru melalui asosiasi data dan persamaan EnKF.

Secara keseluruhan, sistem *tracking* dan prediksi dengan *Differential drive* Mobile Robot melibatkan langkah-langkah seperti simulasi dan pengambilan data LIDAR, SLAM, *Clustering* dengan *Euclidean Clustering*, dan *Tracking Objek* dengan PDAF dimodifikasi EnKF. Setiap langkah ini memiliki peran penting dalam memproses data LIDAR, memperoleh informasi tentang lingkungan sekitar, dan menghindari rintangan dengan sukses. Sistem secara keseluruhan dapat di lihat pada Gambar 3.9. Pada penelitian ini juga menentukan keadaan aman pada lingkungan kerja robot yang dapat dilihat pada Tabel 3.1.

3.9 Hipotesa Penelitian

Hipotesa dari penelitian yang akan dilakukan adalah:

1. Dengan menggunakan metode SLAM, robot bergerak dapat membuat peta lingkungan yang bagus dan memperbarui perkiraan posisinya saat bergerak. Ini membantu robot menghindari rintangan dengan lebih baik..
2. Dengan menggunakan algoritma Euclidean Clustering pada data LIDAR, robot dapat mengenali objek dengan mendapatkan posisi objek dan bentuknya.

- Dengan menggunakan PDAF dengan EnKF pada data filter, robot dapat memprediksi dan mengestimasi posisi objek berikutnya dengan lebih baik.

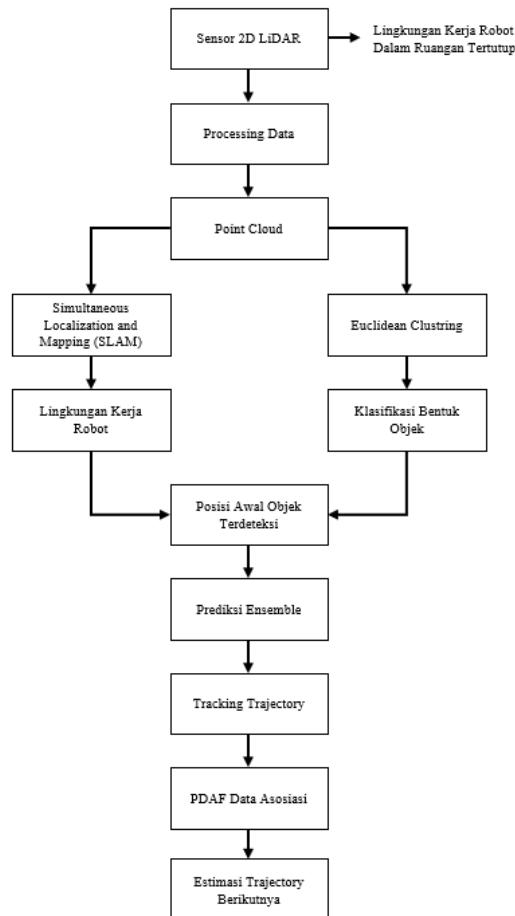
3.10 Kriteria Pengujian

Kriteria pengujian pada penelitian yang dilakukan adalah:

- Memastikan dari hasil beberapa metode bahwa robot *differential drive* dapat berhasil mendeteksi objek dengan menggunakan LIDAR 2D.
- Memastikan dari hasil simulasi bahwa robot dapat *tracking* dan prediksi trejectory objek bergerak dengan tepat saat pengujian dengan evaluasi dari kesalahan RMSRE dan MAE.

Tabel 3.1. Parameter lingkungan kerja robot

Kecepatan Robot	Kecepatan Objek Bergerak	Batas Kecepatan Maximal
0.5 m/s	3 m/s	5 m/s



Gambar 3.9. Flowchart penghindaran rintangan dengan prediksi dan estimasi state.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Estimasi, Prediksi dan *Tracking* Terhadap Rintangan Bergerak

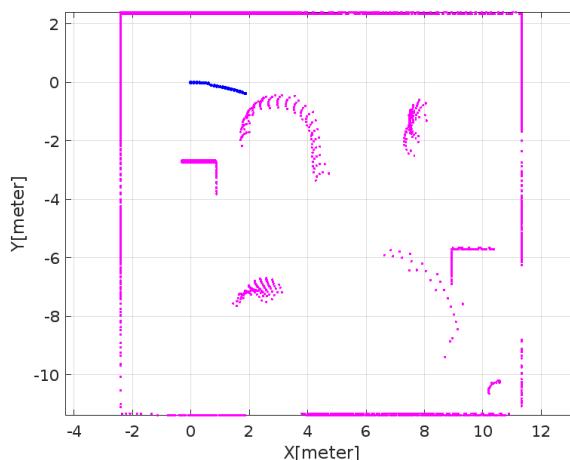
Pada simulasi ini menggunakan dua objek dinamis dan dua objek statis. Objek dinamis adalah objek lingkaran yang diasumsikan sebagai manusia tunggal dan objek statis adalah objek persegi dengan menyesuaikan pada lingkungan tertutup yang kebanyakan objeknya adalah pejalan kaki. Objek persegi memiliki ukuran 0.6 meter dan objek lingkaran memiliki ukuran radius 0.3 meter. Pembatas lingkungan diasumsikan sebagai dinding.

4.1.1 Hasil Pengujian Algoritma SLAM

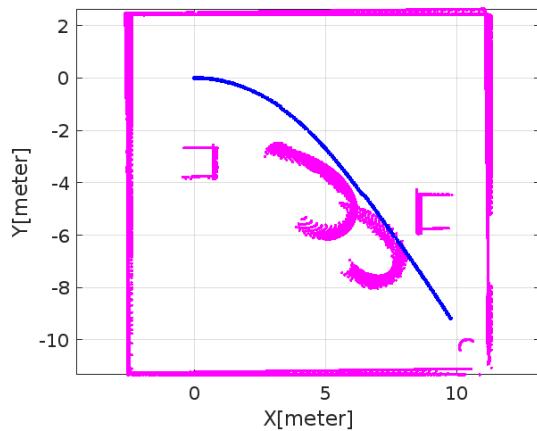
Pengujian pertama ini melakukan lokalisasi dan mapping yang menghasilkan lingkungan kerja *differential drive* mobile robot. Parameter-parameter yang digunakan dalam SLAM dapat dilihat pada Tabel 4. 1. Namun, hasil simulasi juga menunjukkan bahwa robot belum mampu mengenali lingkungan sekitarnya dengan baik.

Tabel 4. 1 Parameter SLAM skenario 1

Parameter	Nilai
Max jangkauan LIDAR	18 meter
Radius peta	18 piksel
Threshold (loop closure)	320
Radius (loop closure)	8 meter



Gambar 4.1 SLAM dengan 10 scan dalam frame data



Gambar 4.2 SLAM secara keseluruhan scan dalam frame data

Simulasi SLAM ini dilakukan dengan menghasilkan beberapa langkah, pertama mencoba dengan 10 scan dalam frame data LIDAR yang diilustrasikan pada Gambar 4.1. Dilihat bahwa data yang digunakan berupa titik sehingga dari titik tersebut membentuk sebuah lingkungan dengan berwarna magenta, sedangkan untuk warna biru adalah hasil pose robot selama 10 scan. Selanjutnya untuk lokalisasi dan mapping untuk seluruh lingkungan diilustrasikan pada Gambar 4.2. Hasil scan untuk seluruh data mengilustrasikan hasil lingkungan kerja robot secara keseluruhan, sehingga terlihat beberapa objek yang ada pada lingkungan secara utuh, dan pose robot terlihat jelas dimulai dari titik awal sampai dengan titik tujuannya.

4.1.2 Hasil Pengujian Algoritma *Euclidean Clustering*

Dalam proses penggunaan Euclidean Clustering untuk mengklasifikasikan objek, algoritma ini digunakan untuk membedakan tiga jenis objek: dinding, manusia (lingkaran), dan objek persegi (objek statis). Pengelompokan dilakukan berdasarkan jarak antara titik-titik data, dengan menggunakan jarak Euclidean sejauh 1.3 meter.

Frame awal dalam pengujian *Clustering* dapat dilihat pada Gambar 4.3, frame ke-8 ini menunjukkan bahwa robot dapat mengklasifikasikan setiap objek dengan baik, di mana terdapat dua objek persegi dengan masing-masing ukurannya 1.11 meter dan 0.87 meter dan dua objek manusia dengan ukuran masing-masing radius 0.58 meter dan 0.14

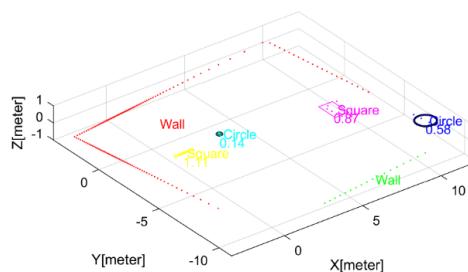
meter. Frame selanjutnya untuk frame ke-30 diilustrasikan pada Gambar 4.4, dapat dilihat ada perubahan berupa penambahan objek lingkaran, dan ukuran dari setiap objek juga berubah untuk perubahannya pada objek persegi masing-masing berubah dengan ukuran 1.54 meter dan 0.83 meter, sedangkan untuk objek lingkaran masing-masing memiliki radius 0.13, 0.13, dan 0.55 meter. Untuk pengujian berikutnya frame ke-48 pada Gambar 4.5. Pada pengujian frame ke-48 ini tidak ada perubahan yang signifikan dalam deteksi objek, tetapi ukuran setiap objek berubah dengan objek persegi masing-masing berubah 1.31 meter dan 0.9 meter, sedangkan untuk objek lingkaran dengan masing-masing radiusnya 0.15, 0.16 dan 0.32 meter. Pengujian pada frame akhir simulasi pada frame ke-130 yang diilustrasikan pada Gambar 4.6 dapat dilihat banyak perubahan objek persegi berkurang sedangkan objek lingkaran bertambah. Kesalahan deteksi dan ukuran yang dihitung menggunakan MAE terhadap objek lingkaran dan persegi pada Gambar 4.7, Gambar 4.8, dan Tabel 4.2.

Tabel 4.2 Hasil perhitungan kesalahan skenario rintangan bergerak

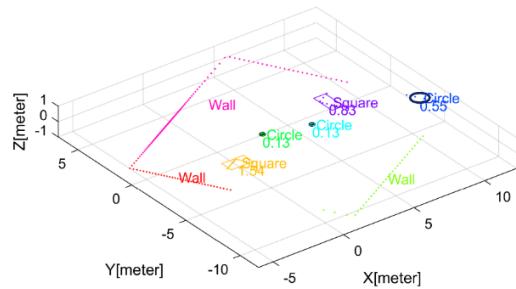
Objek	MAE
Lingkaran	0.17
Persegi	25.13

Tabel 4.3 Posisi awal objek terdeteksi skenario rintangan bergerak

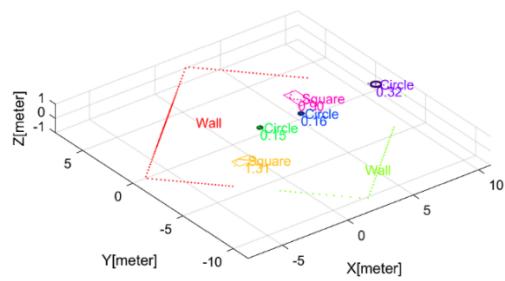
Koordinat	Objek 1	Objek 2	Objek 3
X	5.950	8	13.500
Y	6	8.050	13.500



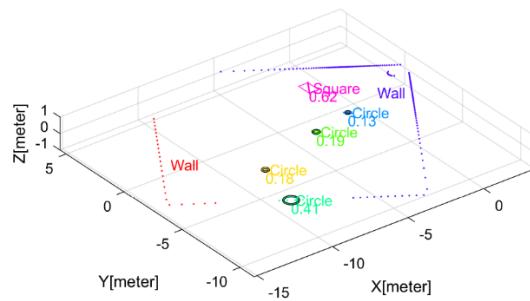
Gambar 4.3 Pengujian frame ke-8 skenario rintangan bergerak



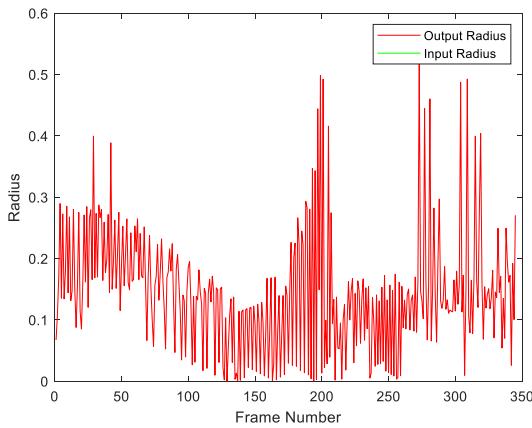
Gambar 4.4 Pengujian frame ke-30 skenario rintangan bergerak



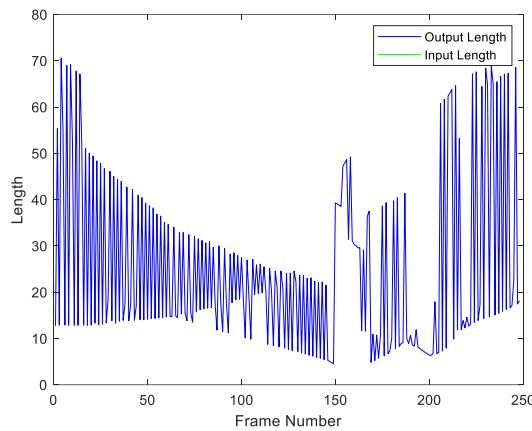
Gambar 4.5 Pengujian frame ke-48 sekanario rintangan bergerak



Gambar 4.6 Pengujian frame ke-130 skenario rintangan bergerak



Gambar 4.7 Kesalahan ukuran pada objek lingkaran skenario rintangan bergerak



Gambar 4.8 Kesalahan ukuran pada objek persegi skenario rintangan bergerak

4.1.3 Hasil Pengujian Estimasi Menggunakan PDAF

Setelah tahap pengujian *Clustering* selesai, didapatkan nilai koordinat X dan Y dari objek-objek bergerak dan statis. Nilai koordinat *centroid* yang didapatkan dari hasil *Clustering* dapat dilihat pada Tabel 4.3. Pertama, nilai-nilai koordinat digunakan untuk memperkirakan bagaimana objek akan bergerak ke depan. Kedua, nilai koordinat juga digunakan untuk melacak pergerakan objek yang tidak bergerak yang membantu melacak posisi objek seiring waktu. Meskipun objek tidak bergerak, melacak posisi mereka masih penting. Informasi ini membantu robot memahami lingkungan dan merencanakan rute yang aman dan efisien.

Trajectory tracking menggunakan PDAF dihasilkan pada Gambar 4.9. Dari hasil analisis, tampak bahwa trajektori objek bergerak bisa diprediksi dan di estimasi dengan efektif. Hal ini menandakan bahwa robot dapat mengantisipasi dan menyiapkan diri untuk pergerakan objek tersebut di masa mendatang, memfasilitasi navigasi yang lebih aman dan efisien. Prediksi dan estimasi trajektori melibatkan memperkirakan posisi berikutnya dari objek, baik yang bergerak maupun yang tidak bergerak. Dengan mengetahui posisi objek saat ini.

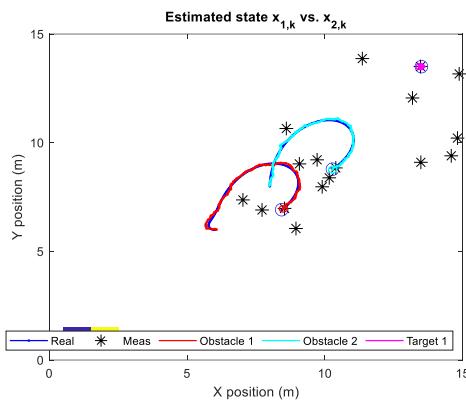
Estimasi posisi berikutnya ini membantu robot beradaptasi dengan perubahan lingkungan dan merespons dengan tepat. Misalnya, jika objek bergerak diperkirakan akan masuk ke jalur robot, robot dapat merencanakan rute alternatif untuk menghindari tabrakan. Begitu juga, jika objek yang tidak bergerak diperkirakan akan berada di jalur robot, robot dapat merencanakan jalan yang berpindah sebelum mencapai objek tersebut. Selanjutnya adalah perhitungan kesalahan filter posisi setiap objek bergerak terhadap trajektori yang ditampilkan pada Gambar 4.9. Pada kesalahan filter. Kesalahan filter dari pembacaan posisi terhadap sumbu X dan Y untuk setiap objek diilustrasikan pada Gambar 4.10, sedangkan ilustrasi pada Gambar 4.11 adalah grafik rata-rata kesalahan setiap objek bergerak terhadap sumbu X dan Y. Pada grafik, titik-titik merepresentasikan ID objek yang menunjukkan estimasi posisinya.

Hasil pengujian menunjukkan nilai RMSRE pada objek 1, objek 2, dan objek 3. Nilai RMSRE yang lebih rendah menunjukkan tingkat akurasi yang lebih tinggi, nilai kesalahan dapat dilihat pada Tabel 4.4. Pada objek 1, RMSRE untuk sumbu X adalah 0.0468 dan untuk sumbu Y adalah 0.0472. Ini menunjukkan bahwa model cukup akurat dalam memprediksi posisi objek 1 di kedua sumbu. Sementara itu, objek 2 memiliki RMSRE sumbu X sebesar 0.0335 dan sumbu Y sebesar 0.0449. Nilai RMSRE yang lebih rendah pada sumbu X menunjukkan bahwa model lebih akurat dalam memprediksi posisi objek 2 pada sumbu X dibandingkan dengan sumbu Y untuk target 1, RMSRE

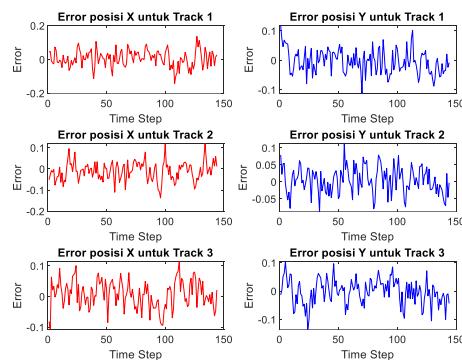
sumbu X adalah 0.0375 dan sumbu Y adalah 0.0324. Di sini, model tampak lebih akurat dalam memprediksi posisi target 1 pada sumbu Y dibandingkan dengan sumbu X, sebagaimana ditunjukkan oleh nilai RMSRE yang lebih rendah. Dalam konteks analisis, Gambar 4.12 menunjukkan hasil prediksi trajectory langkah berikutnya untuk objek pertama. Secara visual, ini menampilkan perbandingan antara posisi sumbu X dan sumbu Y yang dihasilkan dari proses *tracking* dan prediksi langkah berikutnya. Untuk objek lain yang diamati dalam skenario rintangan bergerak ini, detail serupa dapat ditemukan dalam Lampiran 1.

Tabel 4.4 RMSRE skenario rintangan bergerak

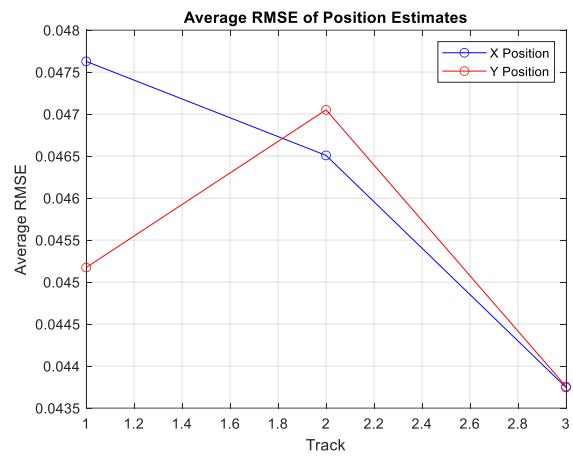
Koordinat	Objek 1	Objek 2	Objek 3
X	0.0468	0.0335	0.0375
Y	0.0472	0.0449	0.0324



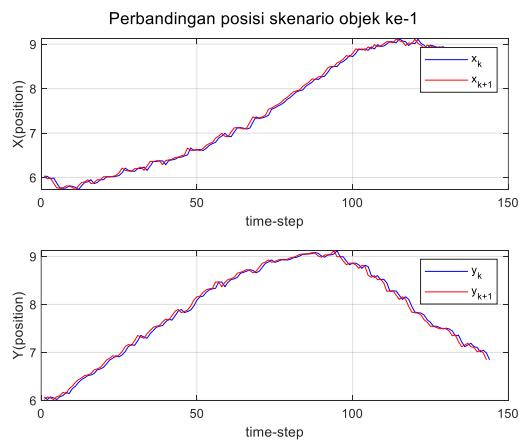
Gambar 4.9 Prediksi, estimasi dan tracking objek dinamis sekanrio rintangan bergerak



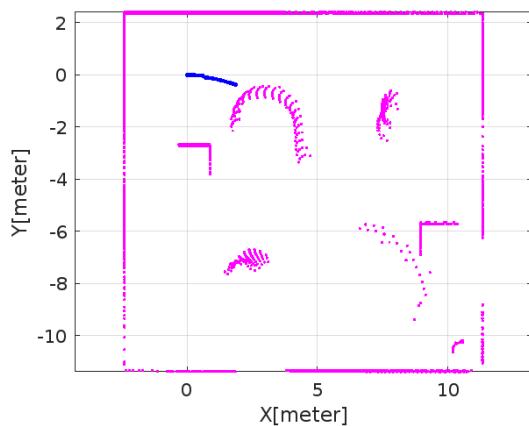
Gambar 4.10 Error posisi dalam sumbu x dan y pada skenario rintangan bergerak



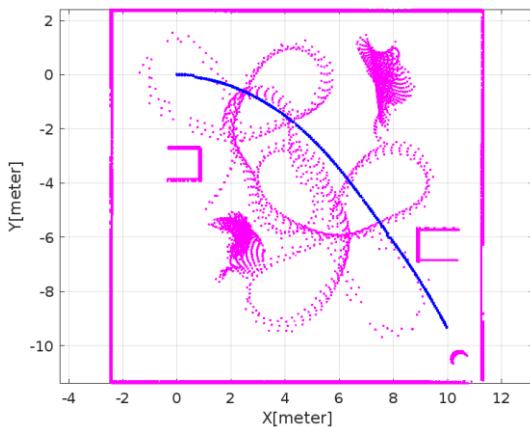
Gambar 4.11 Rata-rata error estimasi posisi skenario rintangan bergerak



Gambar 4.12 Prediksi posisi objek ke-1 skenario rintangan bergerak



Gambar 4.13 SLAM dengan 10 scan dalam frame data



Gambar 4.14 SLAM secara keseluruhan scan dalam frame data

Tabel 4.5 Parameter SLAM skenario2

Parameter	Nilai
Max jangkauan LIDAR	18 meter
Radius peta	18 piksel
Threshold (loop closure)	340
Radius (loop closure)	7 meter

4.2 Estimasi, prediksi dan *Tracking* Terhadap Rintangan Bergerak dan Berinteraksi Dengan Benda Statis

Simulasi ini menggunakan tujuh objek, masing-masing lima dinamis dan dua statis. Objek dinamis adalah lingkaran yang dianggap sebagai pejalan kaki tunggal, dan objek statis, yang disesuaikan dengan lingkungan di dalam ruangan, di mana sebagian besar objeknya adalah pejalan kaki. Objek persegi memiliki ukuran 0,6 meter dan objek lingkaran memiliki dua variasi ukuran pertama dengan radius 0,3 meter, kedua dengan radius 0,7 meter. Batasan lingkungan dianggap sebagai dinding.

4.2.1 Hasil Pengujian Algoritma SLAM

Pengujian pertama ini melakukan lokalisasi dan mapping yang menghasilkan lingkungan kerja *differential drive* mobile robot. Parameter-parameter yang digunakan dalam SLAM dapat dilihat pada Tabel 4.5. Hasil simulasi juga menunjukkan bahwa robot masih memiliki keterbatasan dalam memahami lingkungan sekitarnya. Meski robot mampu memetakan lingkungan, robot masih menghadapi

tantangan dalam mendefinisikan dan membedakan berbagai elemen atau objek dalam lingkungan tersebut. Untuk mengatasinya, metode *Clustering* dapat diterapkan pada tahap berikutnya.

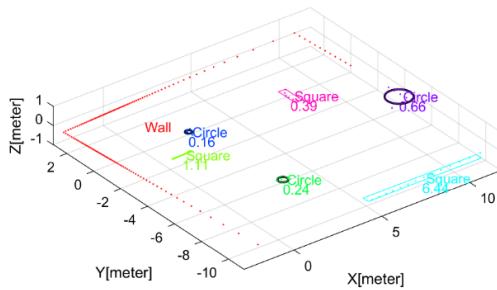
Simulasi SLAM ini dilakukan dengan menghasilkan beberapa langkah, pertama mencoba dengan 10 scan dalam frame data LIDAR yang diilustrasikan pada Gambar 4.13. Selanjutnya untuk lokalisasi dan mapping untuk seluruh lingkungan diilustrasikan pada Gambar 4.14. Hasil scan untuk seluruh data mengilustrasikan hasil lingkungan kerja robot secara keseluruhan, sehingga terlihat beberapa objek yang ada pada lingkungan secara utuh, dan pose robot terlihat jelas dimulai dari titik awal sampai dengan titik tujuannya.

4.2.2 Hasil Pengujian Algoritma *Euclidean Clustering*

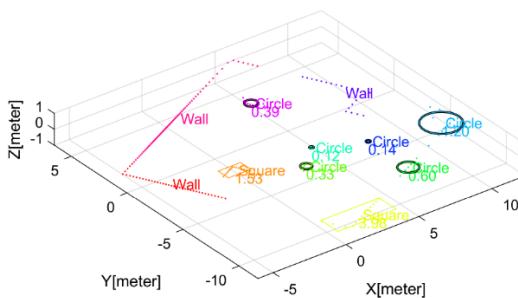
Dalam proses penggunaan Euclidean Clustering untuk mengklasifikasikan objek, algoritma ini digunakan untuk membedakan tiga jenis objek: dinding, manusia (lingkaran), dan objek persegi (objek statis). Pengelompokan dilakukan berdasarkan jarak antara titik-titik data, dengan menggunakan jarak Euclidean sejauh 1.3 meter. Dengan menggunakan metode ini.

Frame awal dalam pengujian *Clustering* dapat di lihat pada Gambar 4.15, frame ke-4 ini menunjukkan bahwa robot dapat mengklasifikasikan setiap objek dengan baik, di mana terdapat tiga objek persegi dengan masing-masing ukurannya 1.11 m, 0.39 m, dan 6.44 m. Tiga objek manusia dengan ukuran masing-masing radius 0.16, 0.66, dan 0.24 m. Frame selanjutnya untuk frame ke-30 diilustrasikan pada Gambar 4.16, dapat di lihat banyak perubahan berupa penambahan objek lingkaran, sedangkan untuk persegi berkurang. Pengurangan pada objek persegi menjadi dua yang masing-masing ukurannya 1.53 m dan 3.98 m, sedangkan untuk objek lingkaran mengalami penambahan sebanyak enam yang masing-masing memiliki radius 0.33, 0.12, 0.14, 0.39, 1.2 dan 0.6 m. Untuk pengujian berikutnya frame ke-66 pada Gambar 4.17. Pada pengujian frame ke-66 ini mengalami kembali

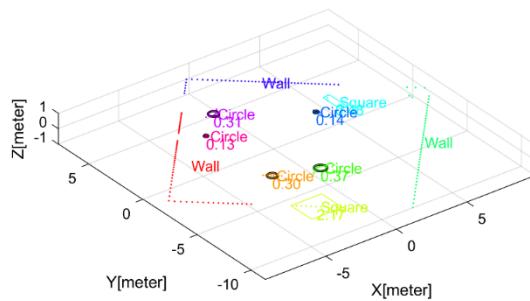
perubahan yang signifikan dalam deteksi objek, dan ukuran setiap objek berubah dengan objek persegi masing-masing berubah 2.17 m dan 0.48 m, sedangkan untuk objek lingkaran mengalami pengurangan menjadi lima yang masing-masing memiliki radius 0.3, 0.37, 0.14, 0.31 dan 0.13 m. Pengujian pada frame akhir simulasi pada frame ke-126 yang dilustrasikan pada Gambar 4.18 dapat dilihat bahwa objek berubah kembali seperti objek persegi berkurang dan objek lingkaran berkurang juga. Kesalahan prediksi ukuran dan bentuk ini terjadi karena hasil pancaran LIDAR terhadap objek kurang sehingga klasifikasinya mengalami kesalahan. MAE terhadap objek lingkaran dan persegi dapat dilihat pada Gambar 4.19, Gambar 4.20, dan Tabel 4.6.



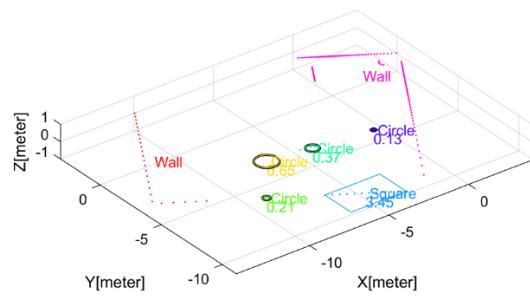
Gambar 4.15 Pengujian frame ke-4 skenario rintangan multiobjek



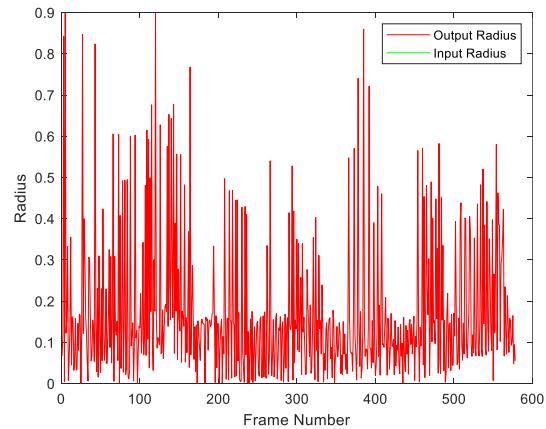
Gambar 4.16 Pengujian frame ke-30 skenario rintangan multiobjek



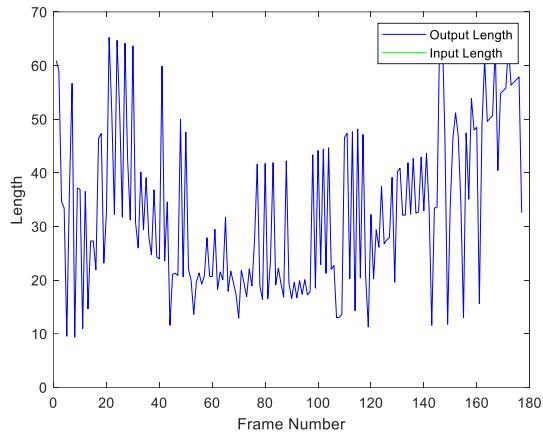
Gambar 4.17 Pengujian frame ke-66 sekanario rintangan multiobjek



Gambar 4.18 Pengujian frame ke-126 skenario rintangan multiobjek



Gambar 4.19 Kesalahan ukuran pada objek lingkaran skenario rintangan multiobjek



Gambar 4.20 Kesalahan ukuran pada objek persegi skenario rintangan multiobjek

Tabel 4.6 Hasil perhitungan kesalahan skenario rintangan hybrid

Objek	MAE
Lingkaran	0.19
Persegi	32.85

Tabel 4.7 Posisi awal objek terdeteksi skenario rintangan multi objek

Koordinat	Objek 1	Objek 2	Objek 3	Objek 4	Objek 5
X	4.7000	9	10.950	4.9500	13.500
Y	5	9.3000	5	11	13.500

4.2.3 Hasil Pengujian Estimasi Menggunakan PDAF

Setelah tahap pengujian *Clustering* selesai, didapatkan nilai koordinat X dan Y dari objek-objek bergerak dan statis. Nilai koordinat *centroid* yang didapatkan dari hasil *Clustering* dapat dilihat pada Tabel 4.7. Pertama, nilai-nilai koordinat digunakan untuk memperkirakan bagaimana objek akan bergerak ke depan. Kedua, nilai koordinat juga digunakan untuk melacak pergerakan objek yang tidak bergerak yang membantu melacak posisi objek seiring waktu. Meskipun objek tidak bergerak, melacak posisi mereka masih penting. Informasi ini membantu robot memahami lingkungan dan merencanakan rute yang aman dan efisien.

Trajectory *tracking* menggunakan PDAF dihasilkan pada Gambar 4.21. hasil ini, bahwa trajektori objek bergerak bisa di prediksi dan di estimasi dengan efektif. Hal ini menandakan bahwa robot dapat

mengantisipasi dan menyiapkan diri untuk pergerakan objek tersebut di masa mendatang, memfasilitasi navigasi yang lebih aman dan efisien.

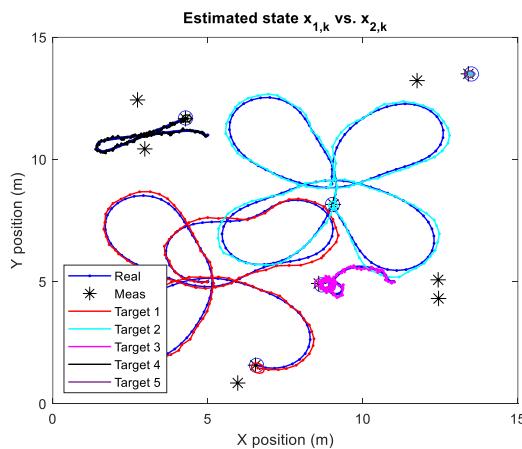
Prediksi dan estimasi trajektori tidak hanya berlaku untuk objek yang bergerak; setelah mengetahui posisi objek saat ini, langkah selanjutnya adalah memperkirakan posisi objek berikutnya. Dengan kata lain, dengan menggunakan koordinat X dan Y, Estimasi posisi berikutnya ini membantu robot beradaptasi dan bertindak sesuai dengan perubahan lingkungan. Misalnya, robot dapat merencanakan rute alternatif untuk menghindari tabrakan jika objek bergerak diperkirakan akan masuk ke jalur robot. Jika objek statis diperkirakan akan berada di jalur robot, robot juga dapat merencanakan rute yang berpindah sebelum mencapai objek tersebut.

Selanjutnya, menghitung kesalahan dari filter posisi untuk setiap objek bergerak dibandingkan dengan trajektori yang ditampilkan di Gambar 4.21. Dalam konteks ini, kesalahan filter pada perbedaan posisi actual dan pengukuran pada sumbu X maupun Y untuk setiap objek, yang ditunjukkan di Gambar 4.22. Pada Gambar 4.23 menampilkan grafik kesalahan rata-rata untuk setiap objek bergerak terkait sumbu X dan Y. Dalam grafik tersebut, setiap titik mewakili ID objek, yang menunjukkan perkiraan posisinya.

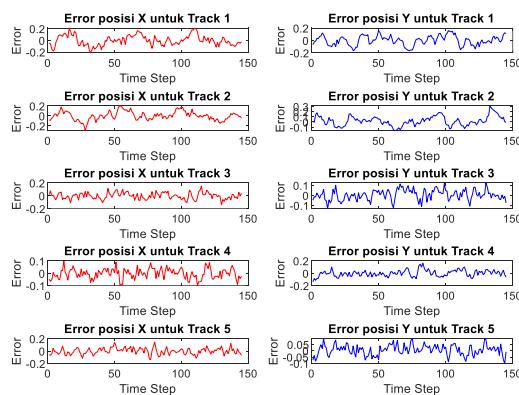
Hasil pengujian menunjukkan nilai RMSRE pada objek 1 sampai objek 5. Nilai RMSRE yang lebih rendah menunjukkan tingkat akurasi yang lebih tinggi, nilai kesalahan dapat dilihat pada Tabel 4.8. Pada objek 1, RMSRE untuk sumbu X adalah 0.0876 dan untuk sumbu Y adalah 0.0831. Ini menunjukkan bahwa model cukup akurat dalam memprediksi posisi objek 1 di kedua sumbu. Sementara itu, objek 2 memiliki RMSRE sumbu X sebesar 0.0854 dan sumbu Y sebesar 0.0920. Nilai RMSRE yang lebih rendah pada sumbu X menunjukkan bahwa model lebih akurat dalam memprediksi posisi objek 2 pada sumbu X dibandingkan dengan sumbu Y. Objek 3, RMSRE sumbu X adalah 0.0467 dan sumbu Y adalah 0.0468. Di sini, model tampak lebih akurat dalam memprediksi posisi target 1 pada sumbu Y dibandingkan

dengan sumbu X, sebagaimana ditunjukkan oleh nilai RMSRE yang lebih rendah. Objek 4, RMSRE sumbu X adalah 0.0472 dan sumbu Y adalah 0.0482. Terakhir Objek 5, RMSRE sumbu X adalah 0.0458 dan sumbu Y adalah 0.0510. Hasil RMSRE menunjukkan jika objek 4 memiliki nilai yang akurat dengan memiliki kesalahan RMSRE yang lebih kecil dibandingkan dengan objek lain.

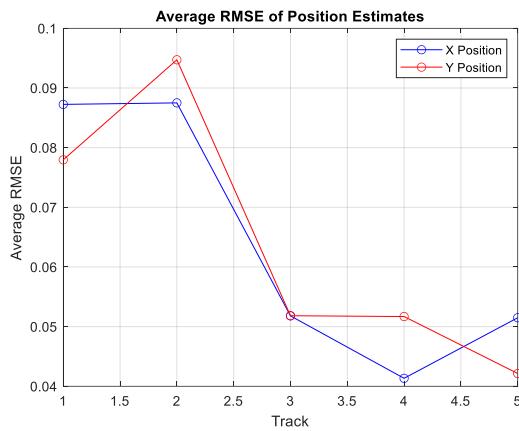
Merujuk ke Gambar 4.24 adalah hasil proyeksi pergerakan berikutnya, atau trajectory, dari objek pertama. Ini memvisualisasikan perbandingan posisi di sepanjang sumbu X dan Y, yang didapat dari proses pelacakan dan prediksi langkah berikutnya. Untuk data objek lainnya dapat ditemukan di Lampiran 2.



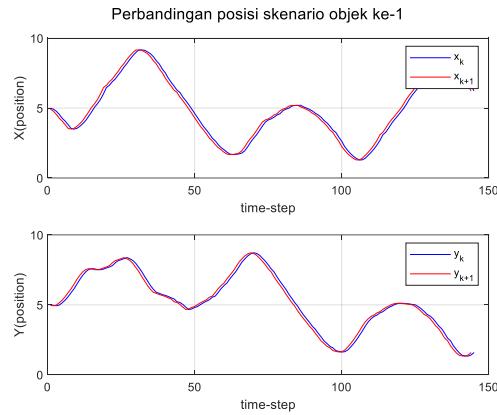
Gambar 4.21 Prediksi, estimasi dan tracking objek dinamis sekanrio rintangan multiobjek



Gambar 4.22 Error posisi dalam sumbu x dan y pada skenario rintangan multiobjek



Gambar 4.23 Rata-rata error estimasi posisi skenario rintangan multiobjek



Gambar 4.24 Prediksi posisi objek ke-1 skenario rintangan multiobjek

Tabel 4.8 RMSRE skenario rintangan multiobjek

Koordinat	Objek 1	Objek 2	Objek 3	Objek 4	Objek 5
X	0.0876	0.0854	0.0467	0.0472	0.0458
Y	0.0831	0.0920	0.0468	0.0482	0.0510

Tabel 4.9 Parameter SLAM skenario 3

Parameter	Nilai
Max jangkauan LIDAR	18 meter
Radius peta	18 piksel
Threshold (loop closure)	350
Radius (loop closure)	9 meter

4.3 Estimasi, Prediksi dan *Tracking* Terhadap Rintangan Bergerak Multi Modalitas Serta Kelompok Pedestrian

Pada simulasi ini menggunakan 8 objek dinamis dan 3 objek statis. Objek dinamis adalah objek lingkaran yang diasumsikan sebagai manusia tunggal dan beberapa kelompok manusia berjalan. Objek statis berupa lingkaran dengan

interaksi dengan objek lingkaran dinamis. Objek lingkaran memiliki ukuran radius 0.3 m, sedangkan objek statis memiliki radius 0.4 m yang mana lebih besar sedikit sebagai pembeda dengan objek dinamis. Pembatas lingkungan dalam simulasi diasumsikan sebagai dinding dalam ruangan tertutup.

4.3.1 Hasil Pengujian Algoritma SLAM

Pengujian ketiga ini melakukan lokalisasi dan mapping yang menghasilkan lingkungan kerja *differential drive* mobile robot yang berbeda dengan kedua simulasi sebelumnya. Parameter-parameter yang digunakan dalam SLAM dapat dilihat pada Tabel 4.9.

Simulasi SLAM ini dilakukan dengan menghasilkan beberapa langkah, pertama mencoba dengan 10 scan dalam frame data LIDAR yang diilustrasikan pada Gambar 4.25. Dilihat bahwa data yang digunakan berupa titik sehingga dari titik tersebut membentuk sebuah lingkungan dengan berwarna magenta, sedangkan untuk warna biru adalah hasil pose robot selama 10 scan. Selanjutnya untuk lokalisasi dan mapping untuk seluruh lingkungan diilustrasikan pada Gambar 4.26. Hasil scan untuk seluruh data mengilustrasikan hasil lingkungan kerja robot secara keseluruhan, sehingga terlihat beberapa objek yang ada pada lingkungan secara utuh, dan pose robot terlihat jelas di mulai dari titik awal sampai dengan titik tujuannya.

Dalam ilustrasi lokalisasi secara utuh memperlihatkan pose objek bergerak dengan jelas, tetapi dalam proses SLAM ini tidak menampilkan objek apa itu, sesuai dengan penjelasan di atas perlu ada langkah berikutnya untuk mengklasifikasikan objeknya.

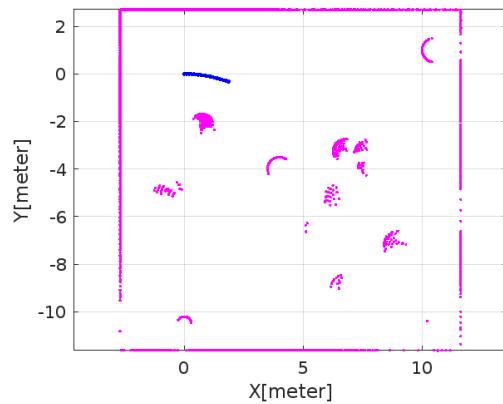
4.3.2 Hasil Pengujian Algoritma *Euclidean Clustering*

Dalam proses penggunaan Euclidean Clustering untuk mengklasifikasikan objek, algoritma ini digunakan untuk membedakan tiga jenis objek: dinding, manusia (lingkaran), dan objek persegi (objek statis). Pengelompokan dilakukan berdasarkan jarak antara titik-titik data, dengan menggunakan jarak Euclidean sejauh 1.3 meter.

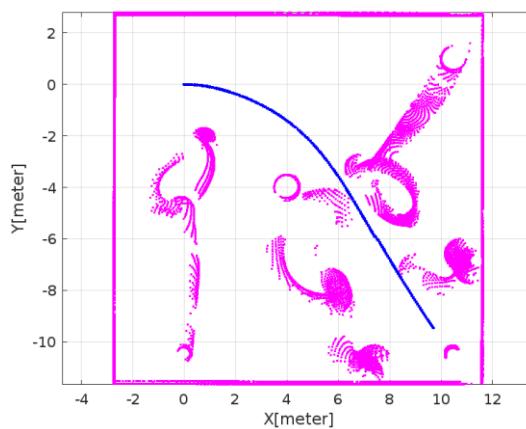
Frame awal dalam pengujian *Clustering* dapat dilihat pada Gambar 4.27, frame ke-4 menunjukkan bahwa robot dapat mengklasifikasikan setiap objek dengan baik, di mana terdapat dua objek persegi dengan masing-masing ukurannya 4.07 m dan 0.33 m dan sembilan objek manusia dengan ukuran masing-masing radius 0.12, 0.13, 0.26, 0.71, 0.35, 0.22, 0.71, 0.28 dan 0.24 m. Frame selanjutnya untuk frame ke-68 diilustrasikan pada Gambar 4.28, dapat di lihat ada perubahan berupa penambahan objek lingkaran, dan ukuran dari setiap objek juga berubah untuk perubahannya pada objek persegi berkurang menjadi satu objek dengan ukuran 0.84 m, sedangkan untuk objek lingkaran mengalami pengurangan juga menjadi enam objek masing-masing memiliki radius 0.1, 0.17, 1.22, 0.12, 0.28 dan 0.25 m. Untuk pengujian berikutnya frame ke-88 pada Gambar 4.29. Pada pengujian frame ke-88 ini tidak ada perubahan dalam deteksi objek, tetapi ukuran setiap objek berubah dengan objek persegi berubah menjadi 1.63 m, sedangkan untuk objek lingkaran bertambah menjadi tujuh objek dengan masing-masing radiusnya 0.8, 0.14, 0.11, 0.11, 0.19, 0.16 dan 0.25 m. Pengujian pada frame akhir simulasi pada frame ke-120 yang diilustrasikan pada Gambar 4.30 dapat dilihat banyak perubahan objek persegi berkurang sedangkan objek lingkaran bertambah. Kesalahan deteksi dan ukuran yang di hitung menggunakan MAE terhadap objek lingkaran dan persegi di lihat pada Gambar 4.31, Gambar 4.32, dan Tabel 4.10.

Tabel 4.10 Hasil perhitungan kesalahan skenario rintangan hybrid

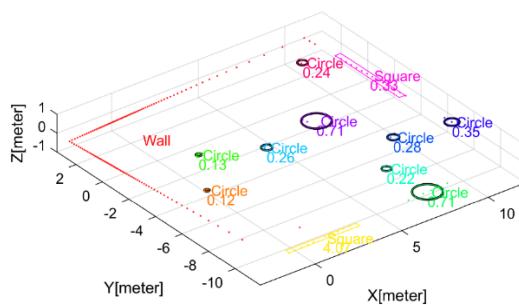
Objek	MAE
Lingkaran	0.19
Persegi	23.63



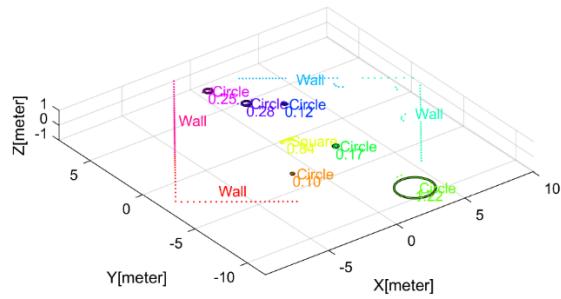
Gambar 4.25 SLAM dengan 10 scan dalam frame data



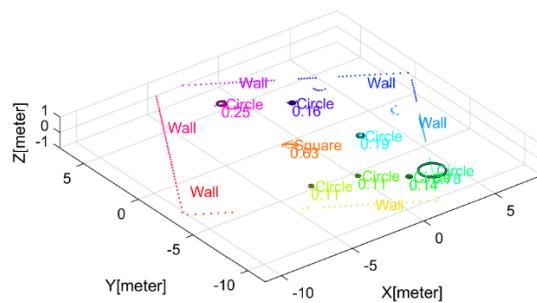
Gambar 4.26 SLAM secara keseluruhan scan dalam frame data



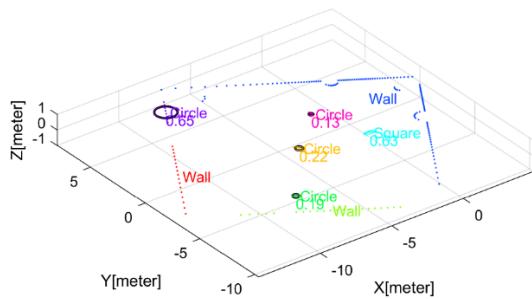
Gambar 4.27 Pengujian frame ke-4 skenario rintangan hybrid



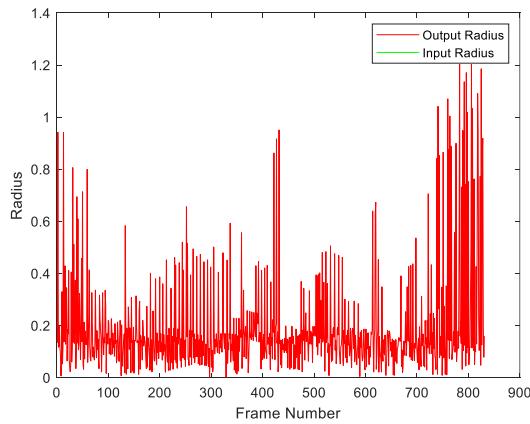
Gambar 4.28 Pengujian frame ke-68 skenario rintangan hybrid



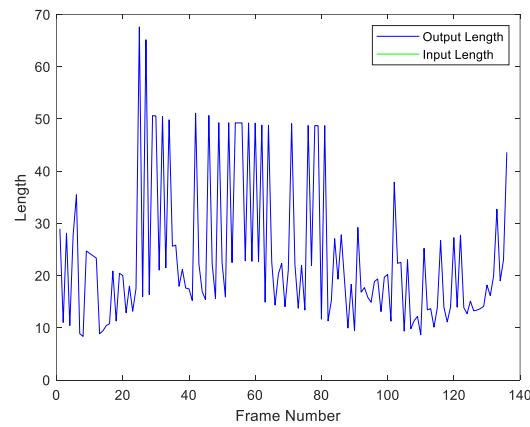
Gambar 4.29 Pengujian frame ke-88 sekanario rintangan hybrid



Gambar 4.30 Pengujian frame ke-120 skenario rintangan hybrid



Gambar 4.31 Kesalahan ukuran pada objek lingkaran skenario rintangan hybrid



Gambar 4.32 Kesalahan ukuran pada objek persegi skenario rintangan hybrid

4.3.3 Hasil Pengujian Estimasi Menggunakan PDAF

Setelah tahap pengujian *Clustering* selesai, didapatkan nilai koordinat X dan Y dari objek-objek bergerak dan statis. Nilai koordinat *centroid* yang didapatkan dari hasil *Clustering* dapat dilihat pada Tabel 4.11. Pertama, nilai-nilai koordinat digunakan untuk memperkirakan bagaimana objek akan bergerak ke depan. Kedua, nilai koordinat juga digunakan untuk melacak pergerakan objek yang tidak bergerak yang membantu melacak posisi objek seiring waktu. Meskipun objek tidak bergerak, melacak posisi mereka masih penting. Informasi ini membantu robot memahami lingkungan dan merencanakan rute yang aman dan efisien. Trajectory *tracking* menggunakan PDAF dihasilkan pada Gambar 4.33. Dari hasil analisis, tampak bahwa trajektori objek bergerak bisa diprediksi dan diestimasi dengan efektif. Hal ini

menandakan bahwa robot dapat mengantisipasi dan menyiapkan diri untuk pergerakan objek tersebut di masa mendatang, memfasilitasi navigasi yang lebih aman dan efisien.

Prediksi dan estimasi trajektori tidak hanya berlaku untuk objek yang bergerak; setelah mengetahui posisi objek saat ini, langkah selanjutnya adalah memperkirakan posisi objek berikutnya. Dengan kata lain, dengan menggunakan koordinat X dan Y, Estimasi posisi berikutnya ini membantu robot beradaptasi dan bertindak sesuai dengan perubahan lingkungan.

Tabel 4.11 Posisi awal objek terdeteksi skenario rintangan hybrid

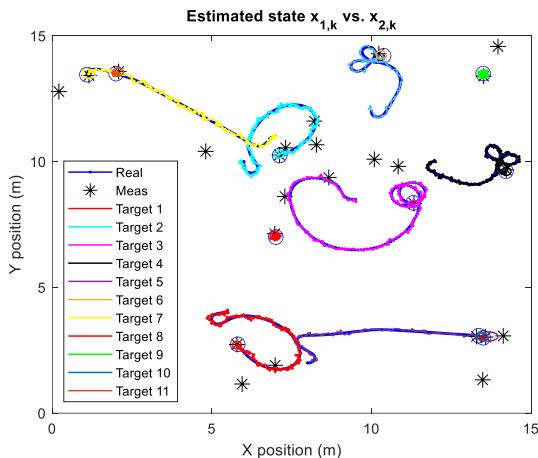
Koor	Objek 1	Objek 2	Objek 3	Objek 4	Objek 5	
X	5.4001	6.0016	9.3031	11.901	8.0984	
Y	3.9965	9.9000	8.5247	9.9880	2.0123	
Koor	Objek 6	Objek 7	Objek 8	Objek 9	Objek 10	Objek 11
X	10.011	6.9015	7	13.500	13.500	2
Y	11.901	10.987	7	13.500	3	13.500

Evaluasi kesalahan filter posisi yang dihasilkan oleh setiap objek bergerak dalam hubungannya dengan trajektori yang ditampilkan pada Gambar 4.33. Mengenai kesalahan filter, variasi dari pembacaan posisi berdasarkan sumbu X dan Y untuk masing-masing objek digambarkan secara visual pada Gambar 4.34. Lebih lanjut, Gambar 4.35 memberikan ilustrasi grafis dari kesalahan rata-rata setiap objek bergerak sehubungan dengan sumbu X dan Y. Dalam grafik tersebut, setiap titik mewakili ID objek, yang menunjukkan perkiraan posisi relatifnya. Hasil pengujian menunjukkan nilai RMSRE pada objek 1, objek 2, objek 3, objek 4 sampai dengan objek 11. Nilai RMSRE yang lebih rendah menunjukkan tingkat akurasi yang lebih tinggi, nilai kesalahan dapat dilihat pada Tabel 4.12, ditentukan dengan melihat semua nilainya kesalahan yang terbaik adalah pada objek ke-8 dengan kesalahan sumbu X 0.0441 dan pada sumbu Y 0.0377, hal ini disebabkan oleh pengaturan nilai proses *covariance* dan nilai pengukuran *covariance clutter rate* menggunakan 5 yang mengakibatkan sedikitnya gangguan dalam lingkungan.

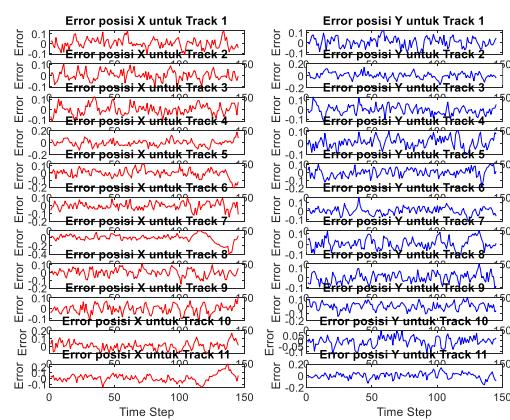
Gambar 4.36 dan Gambar 4.37 hasil pergerakan berikutnya dari objek pertama dan kedua dalam analisis ini. Hasil ini menunjukkan perbandingan posisi di sepanjang sumbu X dan Y, yang diperoleh dari proses pelacakan dan prediksi langkah berikutnya. Lihat Lampiran 3 untuk data tambahan untuk objek rintangan lain.

Tabel 4.12 RMSRE skenario rintangan hybrid

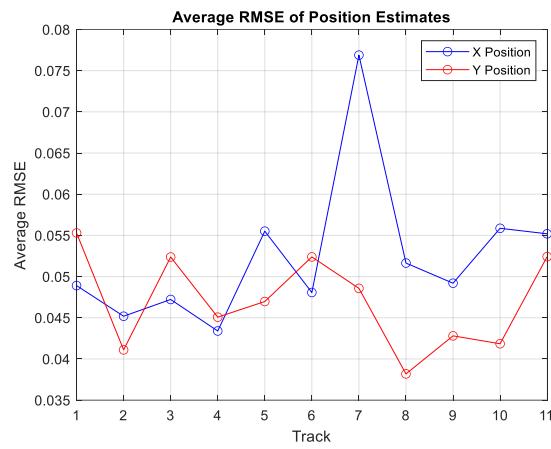
Koor	Objek 1	Objek 2	Objek 3	Objek 4	Objek 5
X	0.0452	0.0465	0.0484	0.0515	0.0504
Y	0.0543	0.0512	0.0445	0.0439	0.0409
Koor	Objek 6	Objek 7	Objek 8	Objek 9	Objek 10
X	0.0456	0.0787	0.0441	0.0431	0.0494
Y	0.0417	0.0499	0.0377	0.0457	0.0520
Objek 11					
					0.0589
					0.0472



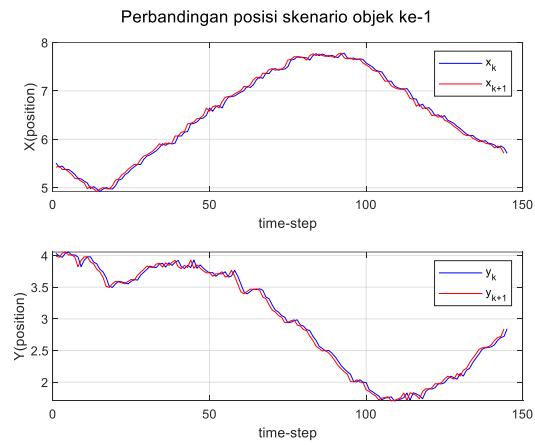
Gambar 4.33 Prediksi, estimasi dan tracking objek dinamis sekanrio rintangan hybrid



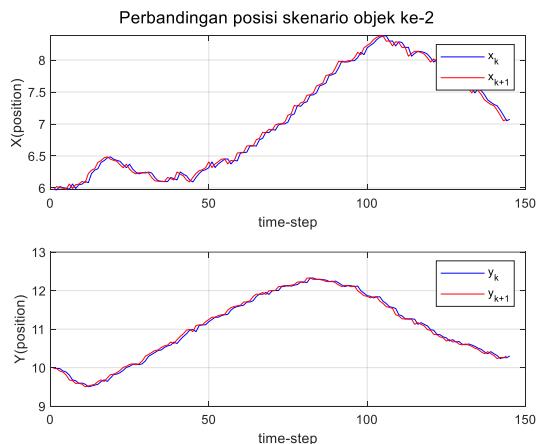
Gambar 4.34 Error posisi dalam sumbu x dan y pada skenario rintangan hybrid



Gambar 4.35 Rata-rata error estimasi posisi skenario rintangan hybrid



Gambar 4.36 Prediksi posisi objek ke-1 skenario rintangan hybrid



Gambar 4.37 Prediksi posisi objek ke-2 skenario rintangan hybrid

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Lokalisasi menggunakan metode SLAM diimplementasikan pada berbagai skenario dengan parameter yang bervariasi. Untuk rintangan bergerak, threshold ditetapkan pada 320 dengan radius 8 meter, sementara skenario rintangan multiobjek menggunakan threshold 340 dan radius 7 meter. Skenario rintangan hybrid melibatkan threshold 350 dan radius 9 meter. Semua skenario diatur dengan pembacaan maksimal LIDAR sebesar 18 meter dan radius peta sebesar 9 meter. Penerapan parameter yang berbeda untuk setiap skenario dalam SLAM menunjukkan variasi tingkat keberhasilan dan kejelasan dalam hasil pemetaan akhir.

Clustering menggunakan metode Euclidean yang melibatkan objek lingkaran dan persegi. Pada skenario rintangan bergerak, objek lingkaran memberikan MAE sebesar 0.17, sementara persegi menghasilkan MAE 26.04. Pada skenario rintangan multiobjek, objek lingkaran menghasilkan MAE 0.19, sedangkan persegi memberikan MAE 38.44. Skenario rintangan hybrid memperlihatkan bahwa objek lingkaran memberikan MAE 0.19, sementara persegi menghasilkan MAE 31.10.

PDAF digunakan untuk prediksi dan pelacakan objek bergerak. Dalam skenario rintangan bergerak, RMSRE untuk setiap objek di koordinat X adalah objek 1 = 0.0468, objek 2 = 0.0335, dan objek 3 = 0.0375 dan di koordinat Y adalah objek 1 = 0.0472, objek 2 = 0.0449, dan objek 3 = 0.0324. Untuk skenario rintangan multiobjek, RMSRE di koordinat X adalah objek 1 = 0.0876, objek 2 = 0.0854, objek 3 = 0.0467, objek 4 = 0.0472, dan objek 5 = 0.0458 dan di koordinat Y adalah objek 1 = 0.0831, objek 2 = 0.0920, objek 3 = 0.0468, objek 4 = 0.0482, dan objek 5 = 0.0510. Pada skenario rintangan hybrid, RMSRE dapat dilihat pada Tabel 4.12. Dengan demikian, dapat disimpulkan bahwa metode yang digunakan memberikan hasil yang berbeda tergantung pada parameter dan skenario yang digunakan. Metode ini menunjukkan potensi untuk ditingkatkan lebih lanjut dengan penyesuaian parameter dan metode untuk mencapai hasil yang

lebih baik. Metode PDAF dengan menggunakan EnKF sebagai data filter dilihat berhasil melakukan *tracking* dan prediksi satu langkah berikutnya setiap objek rintangan bergerak.

5.2 Saran

Penelitian ini memberikan beberapa saran untuk peningkatan di penelitian berikutnya. Pertama, metode *Euclidean Clustering* yang digunakan dalam penelitian ini baik, tetapi masih ada ruang untuk pengembangan algoritma yang dapat menangani lebih banyak jenis objek. Kedua, meski metode PDAF sukses dalam melacak objek bergerak, ada tantangan dalam dynamic *tracking* ketika objek datang dan pergi. Diperlukan peningkatan algoritma PDAF untuk menangani situasi ini. Terakhir, robot harus diatur untuk mengurangi kecepatan ketika bertemu dengan rintangan bergerak yang sudah diprediksi dan estimasi jalur berikutnya.

DAFTAR PUSTAKA

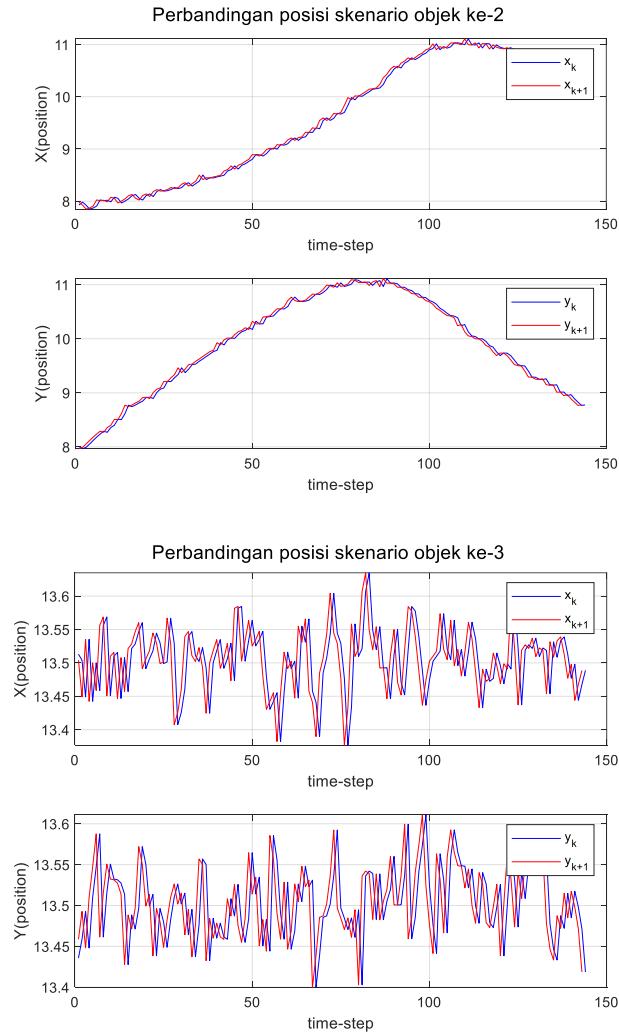
- [1] B. K. Jogeshwar and K. Lochan, ‘Algorithms for Path Planning on Mobile Robots’, *IFAC-PapersOnLine*, vol. 55, no. 1, pp. 94–100, 2022, doi: 10.1016/j.ifacol.2022.04.016.
- [2] D. Ghorpade, A. D. Thakare, and S. Doiphode, ‘Obstacle Detection and Avoidance Algorithm for Autonomous Mobile Robot using 2D LIDAR’, in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, IEEE, Aug. 2017, pp. 1–6. doi: 10.1109/ICCUBEA.2017.8463846.
- [3] A. Farley, J. Wang, and J. A. Marshall, ‘How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion’, *Simul Model Pract Theory*, vol. 120, p. 102629, Nov. 2022, doi: 10.1016/j.simpat.2022.102629.
- [4] C. Cadena *et al.*, ‘Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age’, *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016, doi: 10.1109/TRO.2016.2624754.
- [5] Boru Diriba Hirpo and Prof. Wang Zhongmin, ‘Design and Control for Differential drive Mobile Robot’, *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)*, vol. 06, no. 10, Oct. 2017.
- [6] S. Thrun, W. Burgard, and D. Fox, ‘Probabilistic Robotics’, MA: MIT Press., Cambridge, 2005.
- [7] R. B. Rusu and S. Cousins, ‘3D is here: Point Cloud Library (PCL)’, in *2011 IEEE International Conference on Robotics and Automation*, IEEE, May 2011, pp. 1–4. doi: 10.1109/ICRA.2011.5980567.
- [8] D. Zermas, I. Izzat, and N. Papanikolopoulos, ‘Fast segmentation of 3D point clouds: A paradigm on LIDAR data for autonomous vehicle applications’, in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2017, pp. 5067–5073. doi: 10.1109/ICRA.2017.7989591.
- [9] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, ‘Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction’, *Sci Rep*, vol. 12, no. 1, p. 6256, Apr. 2022, doi: 10.1038/s41598-022-10358-x.
- [10] Y. Bar-Shalom, F. Daum, and J. Huang, ‘The probabilistic data association filter’, *IEEE Control Syst*, vol. 29, no. 6, pp. 82–100, Dec. 2009, doi: 10.1109/MCS.2009.934469.
- [11] M. Sahal, Z. A. Said, R. E. A. Kadir, Z. Hidayat, Y. Bilfaqih, and A. Alkaff, ‘Tracking Position of Airborne Target on SPx-Radar-Simulator Using Probabilistic Data Association Filter’, in *Proceedings of 2021 13th International Conference on Information and Communication Technology*

and System, ICTS 2021, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 258–263. doi: 10.1109/ICTS52701.2021.9608370.

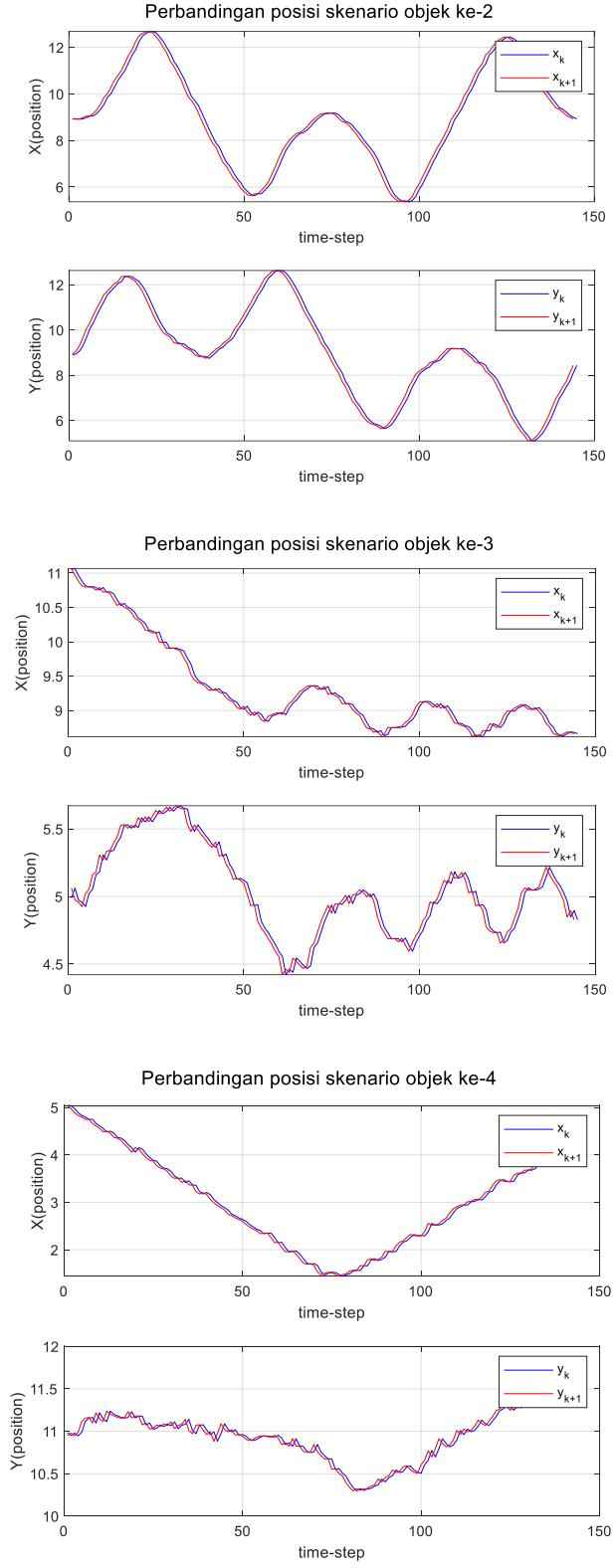
- [12] T. Yang and P. G. Mehta, ‘Probabilistic Data Association-Feedback Particle Filter for Multiple Target Tracking Applications’, *J Dyn Syst Meas Control*, vol. 140, no. 3, Mar. 2018, doi: 10.1115/1.4037781.
- [13] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. L. R. De Moor, D. S. Bernstein, and A. Ridley, ‘What is the Ensemble Kalman Filter and how well does it work?’, in *2006 American Control Conference*, IEEE, 2006, p. 6 pp. doi: 10.1109/ACC.2006.1657419.
- [14] A. M. Aziz, ‘A joint probabilistic data association technique for tracking multiple targets in a cluttered environment’, *Inf Sci (N Y)*, vol. 280, pp. 239–260, Oct. 2014, doi: 10.1016/j.ins.2014.04.055.
- [15] G. Evensen, ‘The Ensemble Kalman Filter for combined state and parameter estimation’, *IEEE Control Syst*, vol. 29, no. 3, pp. 83–104, Jun. 2009, doi: 10.1109/MCS.2009.932223.
- [16] M. Fuad, T. Agustinah, and D. Purwanto, ‘Modified Headed Social Force Model Based on Hybrid Velocity Obstacles for Mobile Robot to Avoid Disturbed Groups of Pedestrians’, *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 3, pp. 222–241, 2021, doi: 10.22266/ijies2021.0630.20.
- [17] M. Fuad, T. Agustinah, and D. Purwanto, ‘Collision avoidance of multi modal moving objects for mobile robot using hybrid velocity obstacles’, *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 3, pp. 407–421, 2020, doi: 10.22266/IJIES2020.0630.37.
- [18] Hasan Najva and Saleem Abdul, ‘Real time obstacle motion prediction using neural network based extended Kalman filter for robot path planning’, *Kuwait Journal of Science*, May 2022, Accessed: Nov. 13, 2022. [Online]. Available: <https://doi.org/10.48129/kjs.18361>
- [19] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. L. R. de Moor, D. S. Bernstein, and A. Ridley, ‘What is the Ensemble Kalman Filter and how well does it work?’, in *Proceedings of the American Control Conference*, 2006, pp. 4448–4453. doi: 10.1109/acc.2006.1657419.
- [20] F. A. Setiawan, T. Agustinah, and M. Fuad, ‘Modified Extremum Seeking Control for Target Tracking and Formation Control in Pursuit-Evasion Game’, 2022.
- [21] M. Fuad, T. Agustinah, and D. Purwanto, ‘Modified Headed Social Force Model Based on Hybrid Velocity Obstacles for Mobile Robot to Avoid Disturbed Groups of Pedestrians’, *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 3, pp. 222–241, Jun. 2021, doi: 10.22266/ijies2021.0630.20.
- [22] A. R. Khairuddin, M. S. Talib, and H. Haron, ‘Review on Simultaneous Localization and Mapping (SLAM)’, in *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, IEEE, Nov. 2015, pp. 85–90. doi: 10.1109/ICCSCE.2015.7482163.

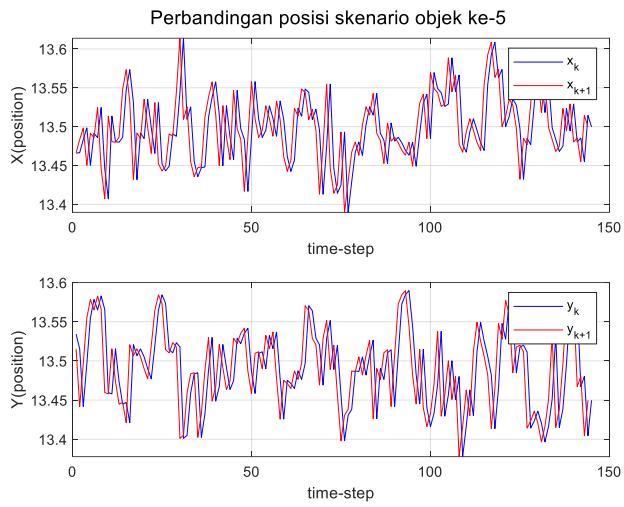
LAMPIRAN

Lampiran 1: Skenario rintangan bergerak untuk objek ke-2 dan ke-3

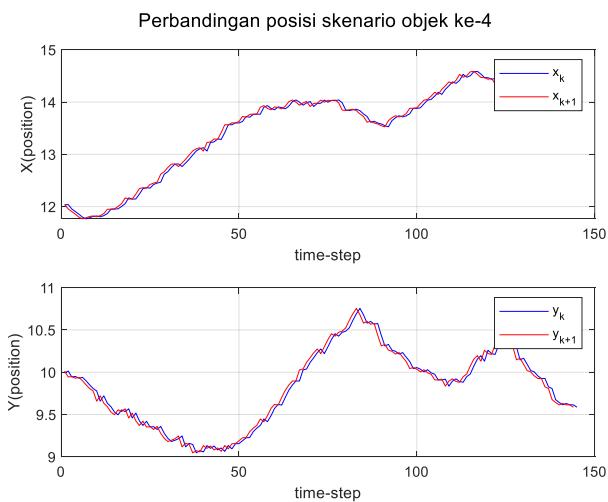
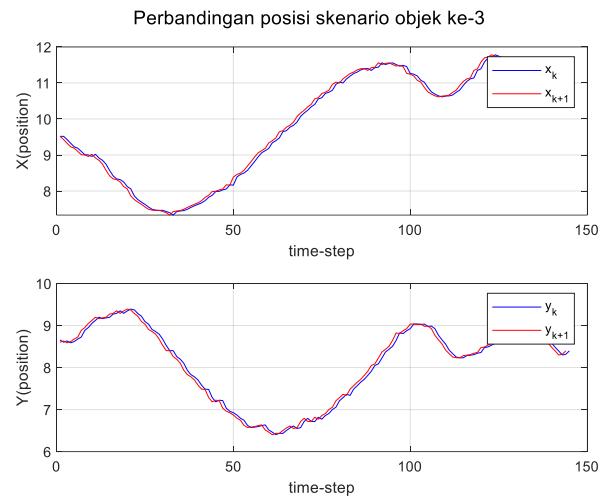


Lampiran 2: Skenario rintangan multiobjek untuk objek ke-2, 3, 4, dan 5

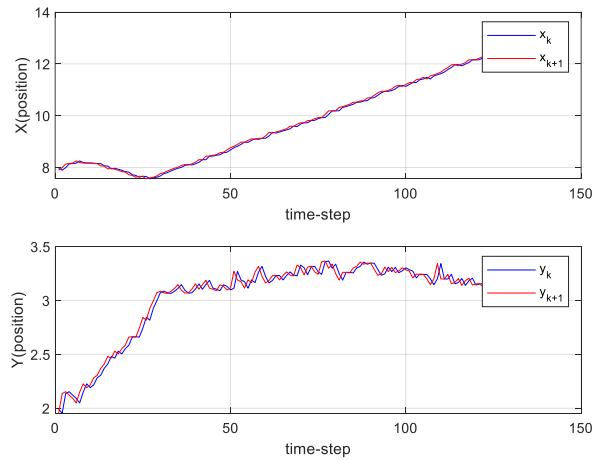




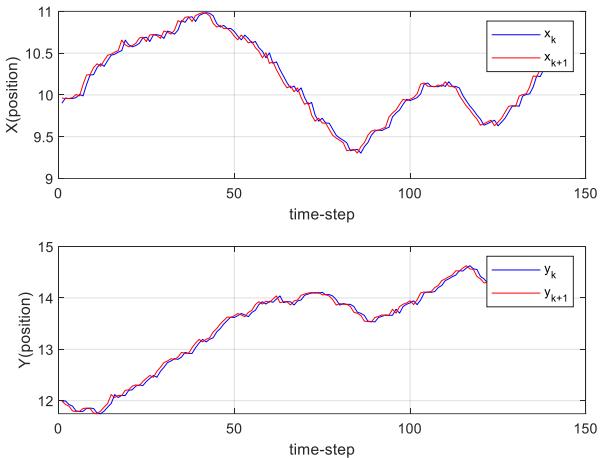
Lampiran 3: Skenario rintangan hybrid untuk objek ke-3, 4, 5, 6, 7, 8, 9, 10, dan 11



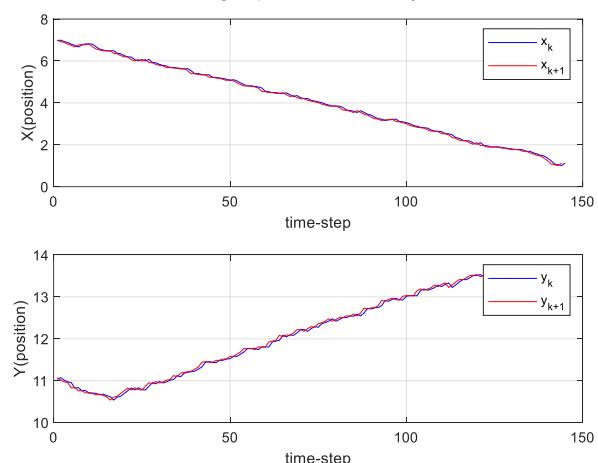
Perbandingan posisi skenario objek ke-5



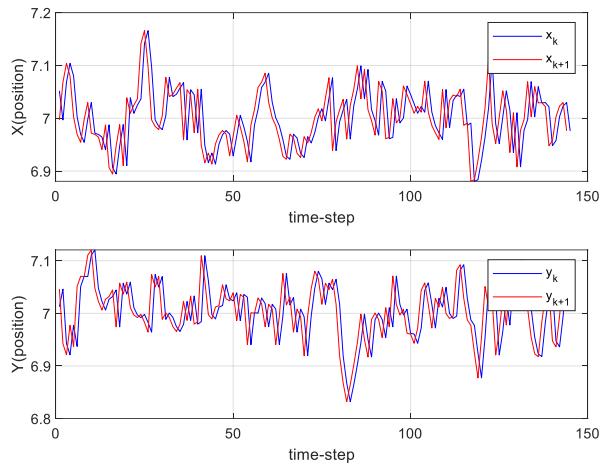
Perbandingan posisi skenario objek ke-6



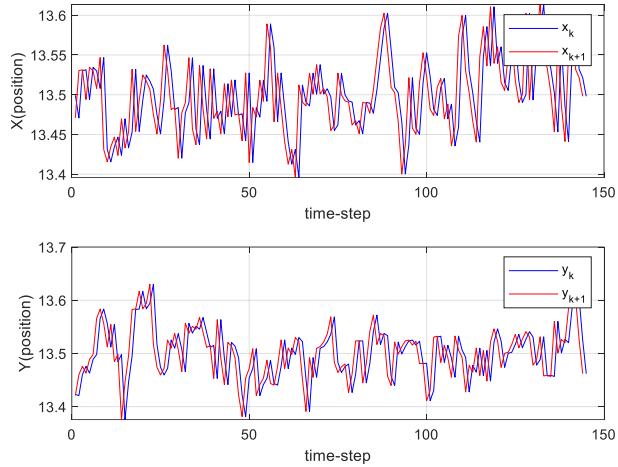
Perbandingan posisi skenario objek ke-7



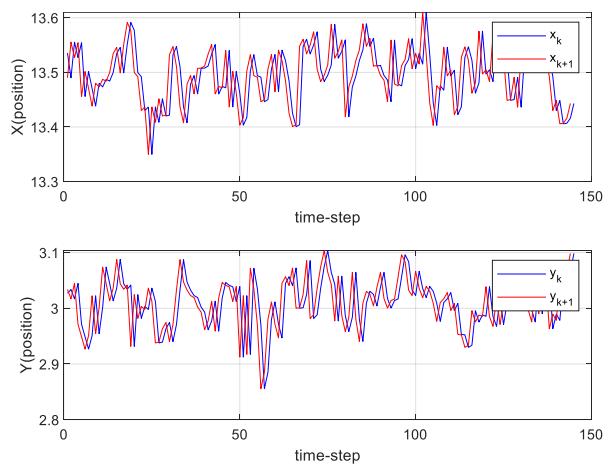
Perbandingan posisi skenario objek ke-8



Perbandingan posisi skenario objek ke-9



Perbandingan posisi skenario objek ke-10



Perbandingan posisi skenario objek ke-11

