

Literature Review Sistem Navigasi Autonomous Mobile Robot Berbasis ROS (Robot Operating System)

Fadil Muhammad¹, Cakra Adipura Wicaksana¹

¹Jurusan Teknik Elektro, Fakultas Teknik, Universitas Sultan Ageng Tirtayasa, Cilegon, Banten.

Informasi Artikel

Naskah Diterima : 21 Juni 2021

Direvisi : 28 Juni 2021

Disetujui : 29 Juni 2021

doi: 10.36055/setrum.v10i1.11862

*Korespondensi Penulis:
fadil.muhammad@untirta.ac.id

Abstract

An autonomous mobile robot is one type of robots that is currently developing. To be able to move automatically to the desired place, the robot needs to know the conditions of the surrounding both locally and globally. This study is conducted to give insights into literature review in robot navigation system.

In carrying out the navigation processes, there are several things to consider. Those are the robot sensor used, the localization system for the surrounding, local route planning and global path planning. From this navigation process, of course, there are many challenges that need further research, namely sensors optimization, optimization in localization and mapping systems, collision avoidance, and finding the shortest path.

Keywords: Literature Review, Autonomous Mobile Robot, Robot Operating System, Navigation

Abstrak

Autonomous mobile robot merupakan salah satu jenis robot yang berkembang saat ini. Untuk dapat bergerak dan berpindah otomatis ke tempat yang diinginkan robot-robot tersebut perlu mengetahui kondisi lingkungan sekitar baik dalam cakupan lokal maupun global. Penelitian ini dibuat karena masih kurangnya literatur review dalam penelitian navigasi robot.

Dalam melakukan proses navigasi terdapat beberapa hal yang diperhatikan yaitu sensor robot yang digunakan, sistem lokalisasi terhadap lingkungan sekitar, perencanaan jalur lokal dan perencanaan jalur global.

Dari proses navigasi ini tentunya banyak tantangan yang dapat menjadi penelitian kedepannya yaitu optimasi pada sensor, optimasi pada sistem lokalisasi dan mapping, penghindaran tabrakan dan pencarian jalur terpendek.

Kata kunci: Literature Review, Autonomous Mobile Robot, Robot Operating System, Navigasi

© 2021 Penerbit Jurusan Teknik Elektro UNTIRTA Press. All rights reserved

1. PENDAHULUAN

Hingga saat ini robot telah banyak membantu pekerjaan manusia dari mulai sektor industri, militer, medis, logistik, pertanian hingga transportasi. Dalam sektor industri tercatat terdapat 422.000 unit robot di seluruh dunia dan diperkirakan akan terus bertambah sebanyak 12% per tahun hingga tahun 2022 [1]. *Autonomous mobile robot* merupakan salah satu jenis robot yang berkembang saat ini. Terdapat beberapa aplikasi dalam *autonomous mobile robot* ini seperti *autonomous vehicle*[2], *emergency assistance robot*, robot penjelajah[3] dan robot pembantu logistik[4] yang dapat berpindah dari satu tempat ke tempat lainnya secara otomatis. (Illah, 2004). *Autonomous mobile robot* juga dapat digunakan sebagai alat yang membantu manusia dalam melakukan pekerjaan yang tidak dapat dilakukan oleh manusia seperti evakuasi tempat-tempat berbahaya, melakukan eksplorasi suatu tempat yang berbahaya dan pekerjaan lainnya. Untuk dapat bergerak dan berpindah otomatis ke

tempat yang diinginkan robot-robot tersebut perlu mengetahui kondisi lingkungan sekitar baik dalam cakupan lokal maupun global [5]. penelitian tentang *autonomous mobile robot* ini biasanya berkaitan dengan tingkat ketahanan suatu robot terhadap lingkungannya misalnya *underwater mobile robot* tentunya akan tahan terhadap tekanan air dan robot penjelajah *outdoor* juga akan didesain untuk tahan terhadap lingkungan ekstrim. Selain itu penelitian yang berkaitan dengan *autonomous mobile robot* biasanya berkaitan dengan sistem navigasi robot baik itu navigasi lokal maupun navigasi global.

Dalam melakukan proses navigasi terdapat beberapa hal yang diperhatikan yaitu sensor robot yang digunakan, sistem lokalisasi terhadap lingkungan sekitar, perencanaan jalur lokal dan perencanaan jalur global. Dengan proses perencanaan jalur navigasi yang tepat maka dapat menghemat sumber daya yang terdapat pada robot tersebut. Selain itu terdapat beberapa tantangan dalam proses mengolah data inputan yang masuk. Pada *autonomous mobile robot* biasanya terdapat beberapa inputan misalnya inputan visual seperti kamera stereo[6], kamera RGB-D[7] dan Lidar[8] yang tidak hanya dapat menangkap gambar namun juga dapat melihat kedalaman suatu objek. Lidar biasa digunakan oleh *autonomous vehicle* sebagai sensor visualnya[9]. Selain itu juga terdapat inputan seperti IMU (*Inertial Measurement Unit*) yang terdiri dari *accelerometer*, *gyroscope* serta kompas yang berfungsi mengetahui kondisi lingkungan yang sedang dilalui oleh robot. Dari proses pengambilan data inputan ini pula akan dilakukan proses yang nantinya menentukan aksi dari robot tersebut. Proses pengolahan data yang dilakukan harus dapat berjalan maksimal agar *autonomous mobile robot* dapat melakukan navigasi dengan baik.

Untuk dapat bergerak ke suatu tempat yang diinginkan maka robot harus mengetahui keadaan robot tersebut terhadap lingkungan sekitarnya. Proses pengambilan dan pengolahan data di lingkungan sekitar ini disebut dengan *localization*. *Localization* merupakan langkah awal dalam suatu sistem navigasi. Terdapat beberapa contoh metode *localization* salah satunya adalah *Simultaneous localization and mapping* atau yang biasa disebut dengan SLAM. SLAM sendiri sebenarnya bukanlah suatu metode khusus. Dengan begitu SLAM sendiri merupakan jenis dari metode atau algoritma yang memanfaatkan sensor atau gabungan sensor pada robot untuk dapat melakukan *localization* serta melakukan pemetaan lingkungan yang berada di sekitar robot tersebut[10].

Setelah melakukan *localization*, suatu *autonomous mobile robot* akan melakukan perencanaan jalur navigasi yang akan ditempuh menuju ke titik tujuan. Perencanaan jalur merupakan merupakan proses penting karena robot diharapkan dapat mencari jalur terpendek dan dapat menghindari halang rintang baik yang statis maupun halang rintang yang dinamis. Proses perencanaan jalur navigasi dapat dilakukan apabila robot telah memiliki pemetaan jalur tersebut. Dalam melakukan perencanaan jalur model lingkungan peta yang berkelanjutan diubah menjadi model diskrit. Perencanaan jalur terbagi menjadi dua jenis yaitu perencanaan jalur lokal dan perencanaan jalur global.

Dalam membuat suatu sistem yang begitu kompleks tentunya peneliti tidak dapat membuat sistem tersebut dari awal dengan begitu diperlukan suatu *framework* dalam pembuatan robot. Salah satu *framework* yang saat ini banyak digunakan adalah ROS (*Robot Operating System*)[11]. ROS diciptakan karena setiap grup riset robotika biasanya terfokus pada topik-topik riset robot tertentu untuk itu ROS hadir demi memudahkan dalam pengembangan suatu topik riset tertentu tanpa harus memikirkan topik riset lainnya. ROS mencakup *tools*, *libraries* dan *convention* yang bertujuan untuk menciptakan serta mengembangkan robot yang kompleks. Dengan begitu grup riset yang mengembangkan sistem navigasi tidak perlu harus memikirkan pula sistem biomekanik dari robot tersebut dan peneliti juga dapat hanya mengembangkan salah satu bagian dari sistem navigasi tersebut tanpa harus membuat sistem navigasi dari dasar. Hingga saat ini ROS sudah memiliki 13 versi dimulai dari versi ROS Box Turtle yang rilis pada 2 maret 2010 hingga ROS Noetic Ninjemys yang rilis pada tanggal 23 Mei 2020 yang terlihat pada Tabel 1. Selain karena perkembangan library dan toolsnya ROS populer karena adanya komunitas yang aktif [12]. Terdapat beberapa fitur yang memudahkan peneliti yang ingin menggunakan *framework* ini seperti adanya dokumen wiki untuk tutorial, forum tanya jawab serta support yang berkelanjutan. Dengan banyaknya fitur-fitur yang terdapat pada ROS ini membuat ROS menjadi salah satu *framework* dalam bidang robotika yang banyak digunakan dan dikembangkan oleh peneliti dan pengembang di bidang robotika di seluruh dunia.

Sebagai salah satu penelitian yang aktif serta berkembang saat ini, literature review mengenai navigasi pada robot khususnya *autonomous mobile robot* masih sedikit, untuk itu perlu dibuat studi

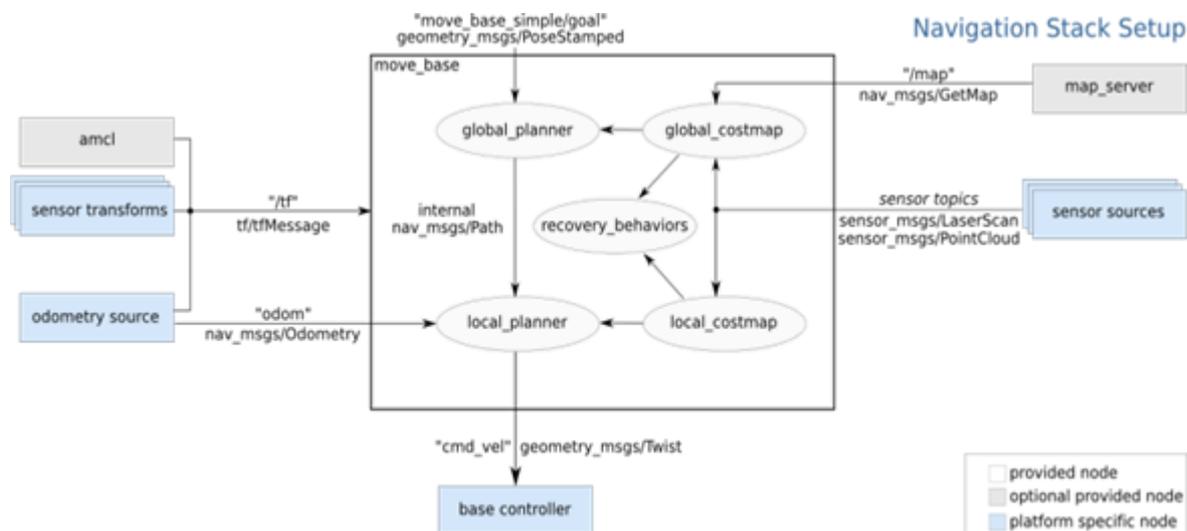
literatur khususnya dalam bahasa Indonesia yang diharapkan dapat mempermudah dalam menentukan roadmap penelitian yang berhubungan dengan sistem navigasi pada *autonomous mobile robot*.

Tabel 1 Daftar Distro ROS

No	Nama Distro	Tanggal Rilis	Tanggal EOL (stop support)
1	ROS Noetic Ninjemys	23 Mei 2020	Mei 2025
2	ROS Melodic Morenia	23 Mei 2018	Mei 2023
3	ROS Lunar Loggerhead	23 Mei 2017	Mei 2019
4	ROS Kinetic Kame	23 Mei 2016	April 2021
5	ROS Jade Turtle	23 Mei 2015	Mei 2017
6	ROS Indigo Igloo	22 Juli 2014	April 2019
7	ROS Hydro Medusa	4 September 2013	Mei 2015
8	ROS Groovy Galapagos	31 Desember 2012	Juli 2014
9	ROS Fuerte Turtle	23 April 2012	-
10	ROS Electric Emys	30 Agustus 2011	-
11	ROS Diamondback	2 Maret 2011	-
12	ROS C Turtle	2 Agustus 2010	-
13	ROS Box Turtle	2 Maret 2010	-

2. NAVIGASI PADA ROBOT OPERATING SYSTEM (ROS)

Pada bab ini akan dijelaskan tahapan-tahapan apa saja yang berperan dalam proses navigasi pada *autonomous mobile robot*. Terdapat beberapa hal yang perlu diperhatikan dalam proses navigasi mulai dari perencanaan sensor, SLAM, perencanaan jalur lokal dan perencanaan jalur global. salah satu contoh sistem navigasi pada ROS dapat dilihat pada Gambar 1[13].



Gambar 1. Sistem Navigasi pada ROS

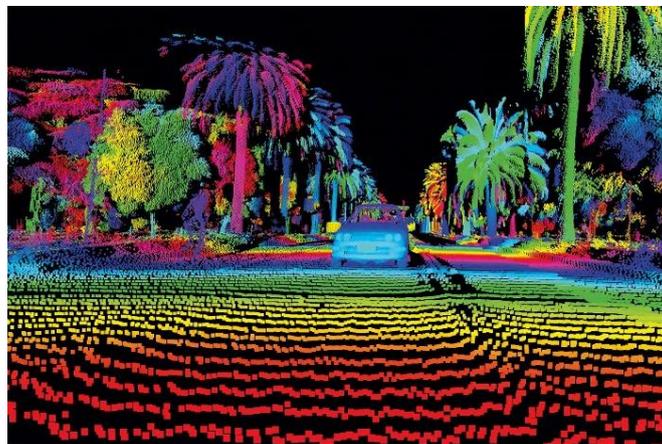
Dari Gambar 1 terlihat bahwa *amcl*, *sensor transform* serta *odometry source* digunakan sebagai lokalisasi dari posisi robot terhadap lingkungan dan informasi dari *odometry* digunakan saat melakukan pencarian jalur lokal. *Global_planner* didapat dari informasi *global_costmap* yang merupakan proses dari *map_server* yang telah dibuat sebelumnya oleh proses *mapping*. Selain itu dalam mencari *global_costmap* dan *local_costmap* diperlukan informasi dari sensor-sensor yang terdapat pada robot. Informasi dari *global_planner* ini akan dipecah-pecah dan menjadi informasi pada *local_planner* yang nantinya akan menentukan gerak dari *controller* pada *autonomous mobile robot*. Keterangan dari setiap tahapan ini dapat dilihat pada Tabel 2.

Tabel 2 Keterangan pada sistem navigasi ROS

Tahap	Keterangan
amcl	Salah satu metode lokalisasi probabilitas 2 dimensi yang menggunakan pendekatan <i>adaptive monte carlo</i>
sensor transform	Bentuk transformasi dari sensor yang ada pada robot
odometry source	Penggunaan data yang berasal dari <i>motion sensor</i> untuk mengetahui perubahan robot dari waktu ke waktu
global_planner	Proses perencanaan jalur global
local_planner	Proses perencanaan jalur lokal
global_costmap	Nilai yang dihasilkan peta jalur global
local_costmap	Nilai yang dihasilkan peta jalur lokal
recovery behaviors	Perilaku pemulihan saat robot dalam posisi macet/stuck
map_server	Server tempat informasi dari peta yang dibuat
sensor source	Sumber data lingkungan yang diambil oleh sensor
base controller	Luaran dari proses yang akan digunakan sebagai basis gerak robot

2.1. Sensor

Agar *autonomous mobile robot* dapat melakukan proses lokalisasi dan proses *mapping* maka diperlukan suatu perangkat yang dapat mengambil data lingkungan. Sensor yang dapat digunakan dalam proses lokalisasi berupa sensor visual berupa RGB-D Camera[14], LIDAR[15] dan sensor pergerakan seperti IMU[16]. RGB-D Camera merupakan kamera yang memiliki perangkat *infra red* sehingga tidak hanya mengambil data dua dimensi juga dapat mengambil data kedalaman suatu citra[7]. Sedangkan LIDAR adalah suatu sensor visual yang menggunakan laser untuk membuat *point cloud* rekonstruksi tiga dimensi pada lingkungan sekitar[8]. Contoh visual yang ditangkap oleh LIDAR dapat dilihat pada Gambar 2. Selain itu juga terdapat sensor gabungan seperti gabungan LIDAR dan IMU[17], gabungan RGB-D camera dan IMU[18] dalam melakukan lokalisasi dan pemetaan yang simultan.



Gambar 1. Visual yang ditangkap oleh LIDAR

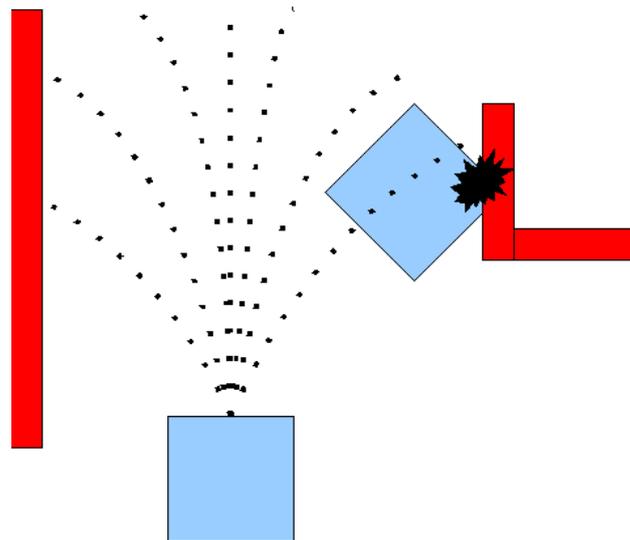
2.2. SLAM

SLAM merupakan metode ataupun algoritma yang melakukan proses lokalisasi serta mapping lingkungan pada *autonomous mobile robot*. Proses lokalisasi bawaan yang digunakan oleh ROS saat ini menggunakan metode lokalisasi AMCL (Adaptive Monte Carlo Localization)[19]. AMCL sendiri adalah suatu sistem lokalisasi probabilitas 2 dimensi yang menggunakan pendekatan *adaptive monte carlo*. Dengan bantuan *odometry*, pendekatan ini menggunakan filter partikel untuk membuat robot mengetahui posisinya di dalam peta yang akan diciptakan. Pembuatan peta dari robot sendiri dengan cara membuat robot berjalan-jalan di sekitar lingkungannya dan sensor pada robot akan mengambil informasi dimana saja terdapat halangan-halangan yang nantinya akan menghalangi pergerakan dari Turtlebot.

Terdapat beberapa jenis metode yang digunakan untuk melakukan proses *mapping* pada *autonomous mobile robot* seperti misalnya *gmapping* [20], *hector*[21][22], *karto*[23], *cartographer*[24], dan *frontier exploration*[25][26]. Metode *mapping* tersebut adalah metode *mapping* yang menggunakan LIDAR sebagai sensornya. Selain metode tersebut terdapat beberapa metode *mapping* lainnya yang menggunakan kamera *monocular* yaitu *parallel tracking and mapping* (PTAM)[27], *LSD (Large Scale Direct monocular) SLAM* [28], *Direct Sparse Odometry* (DSO) [29] dan *ORB SLAM* [30] yang juga dapat menggunakan kamera stereo. Terdapat pula beberapa metode SLAM lainnya yang menggunakan kamera stereo seperti *stereo parallel tracking and mapping* (S-PTAM)[31], dan *real-time appearance-based mapping* (RTAB map)[32].

2.3. Perencanaan Jalur Lokal

Dalam melakukan perencanaan jalur terpendek dalam cakupan lokal terdapat beberapa metode yang telah ada di library ROS misalnya *DWA (Dynamic Window Approach)* yang membuat lintasan berdasarkan peta yang telah dibuat sebelumnya[33]. *DWA* akan membuat beberapa macam lintasan dan membuat suatu fungsi nilai untuk menghitung jarak terdekat serta menghindari adanya penghalang baik itu penghalang yang statis maupun yang dinamis. Nilai dari jarak yang dibuat berdasarkan grid yang dibentuk oleh *DWA*. Fungsi nilai ini berupa dx , dy dan $d\theta$ yang akan secara simultan dikirimkan ke robot. Ide dasar dari algoritma *DWA* dapat dilihat pada Gambar 3.



Gambar 2 Ide dasar dari algoritma DWA

Selain menggunakan *DWA* terdapat beberapa algoritma dalam perencanaan jalur lokal seperti *Base Local Planner*[34] yang menggabungkan *DWA* dan peluncuran lintasan. Selain menggunakan *DWA* sebagai penghitung nilai yang akan dilalui, pada *base local planner* juga melakukan simulasi untuk memprediksi serta mencari nilai-nilai terbaik dalam melakukan pencarian jarak terdekat. Terdapat pula beberapa karakteristik yang perlu diperhatikan seperti kedekatan dengan rintangan, kedekatan dengan tujuan kecepatan dan karakteristik lainnya. Setelah dilakukan simulasi maka nilai terbaik yang nantinya akan dipilih oleh robot dalam menentukan lintasannya.

Dan yang terakhir terdapat pula metode *TEB (Timed Elastic Band) local planner*[35][36]. Metode ini mengimplementasikan perencanaan jalur lokal online secara optimal. Pada perjalanannya terdapat beberapa kendala dalam pelaksanaan metode ini seperti adanya nilai optimal lokal dan tidak dapat melewati rintangan yang ada. Oleh karena itu terdapat pembaharuan dengan mencari nilai optimal global pada metode ini[37][38]. Terdapat pula library khusus untuk robot yang bertipe *autonomous vehicle*[39].

2.4. Perencanaan Jalur Global

Setelah melakukan lokalisasi, *autonomous mobile robot* akan melakukan perencanaan jalur navigasi yang akan ditempuh menuju ke titik tujuan. Perencanaan jalur merupakan proses penting karena robot diharapkan dapat mencari jalur terpendek dan dapat menghindari halang rintang

baik yang statis maupun halang rintang yang dinamis. Proses perencanaan jalur navigasi dapat dilakukan apabila robot telah memiliki pemetaan jalur tersebut. Dalam melakukan perencanaan jalur model lingkungan peta yang berkelanjutan diubah menjadi model diskrit. Terdapat tiga pendekatan dalam penentuan jalur navigasi ini yaitu pendekatan *road map*, pendekatan *cell decomposition* dan pendekatan *potential field*[5].

Pendekatan *roadmap* membuat suatu set kurva satu dimensi yang masing-masing akan menghubungkan suatu node. Hubungan antara node ini akan menjadi suatu jalur dan setiap jalur akan diberikan bobot-bobot sesuai dengan parameter-parameter misalnya seperti jarak, hambatan ataupun parameter lainnya. Untuk mencari jarak terpendek menggunakan pendekatan *road map* maka akan dicari solusi bobot paling minimum yang berarti memiliki jarak serta hambatan paling minimum. Jenis algoritma pencarian yang menggunakan pendekatan *road map* ini adalah algoritma pencarian yang menggunakan graph, beberapa dari algoritma tersebut yaitu *Breadth-First Search*, Algoritma Dijkstra's dan A*[40].

Pendekatan *Cell Decomposition* akan membedakan antara area geometri yang nantinya disebut dengan sel yang bebas dan yang terisi oleh objek. Algoritma dari pendekatan *cell decomposition* yaitu:

1. Langkah pertama adalah membagi peta menjadi daerah-daerah kecil dan terhubung yang disebut dengan sel.
2. Tentukan sel-sel yang bebas yang berdekatan dan buat konektivitas antara sel-sel yang bebas tersebut. Sel-sel yang tidak bebas disebut dengan sel yang terdekomposisi.
3. Tentukan sel konfigurasi awal dan tujuan dan buat grafik konektivitas antara kedua sel tersebut.
4. Dari urutan sel tersebut hitung jalur di setiap sel yang dilewati serta buat urutan gerakan untuk urutan terbaik.

Pendekatan berikutnya adalah pendekatan *potential field*. Pendekatan ini memodelkan robot menjadi suatu partikel yang bergerak dibawah pengaruh medan potensial yang ditentukan oleh tujuan serta beberapa rintangan [41]. Setiap pergerakan robot akan ditentukan oleh medan potensial yang berada di lokasinya. Kelebihan dari metode ini adalah komputasi yang dihasilkan tidak terlalu besar karena setiap pergerakan robot hanya tergantung pada pendekatan medan potensialnya. Selain itu pada pendekatan ini juga memiliki respon yang baik pada halang rintang yang dinamis karena setiap adanya halang rintang maka medan potensial akan selalu berubah.

Terdapat pula pendekatan RRT (*Rapidly-Exploring Random TREE*)[42] maupun turunannya seperti RRT*. RRT adalah suatu algoritma pencarian jarak terdekat dengan cara membuat suatu graf tree yang saling terhubung dan mencari jalur global berdasarkan graf yang telah dibuat. Graf pada RRT dibuat secara random dan terus menerus. Terdapat beberapa modifikasi seperti RRT* yang berupa algoritma RRT yang telah dioptimasi dengan cara memberikan jalur tersingkat ke tujuan saat jumlah node pada graf mendekati tak hingga. Sudah banyak pula contoh robot yang menggunakan RRT dan RRT* sebagai perencanaan jalur global pada robotnya [43][44]. Selain itu juga terdapat beberapa penelitian yang memodifikasi RRT dengan metode perencanaan jalur global lainnya seperti RRT dengan *potential field*[45].

3. TANTANGAN DAN PENELITIAN LANJUTAN

Dari literature review yang telah dilakukan di bab sebelumnya maka kami dapat membagi empat jenis tantangan serta penelitian lanjutan yang dapat dilakukan dalam sistem navigasi pada *autonomous mobile robot* yang dapat dilihat pada Tabel 3.

3.1 Tantangan Sensor

Teknologi sensor memiliki peran penting dalam perkembangan *autonomous mobile robot*. Suatu sistem *autonomous mobile robot* dapat terdiri dari satu atau lebih sensor sebagai pengenal lingkungan. Dalam pemilihan sensor suatu *autonomous robot* dapat dilihat dari jenis *autonomous mobile robot* tersebut dan pemilihan jenis robot tersebut dapat dilihat dari:

1. Kemampuan komputasi dari *autonomous mobile robot*
2. Area jelajah dari suatu *autonomous mobile robot*, apakah *indoor* atau *outdoor*
3. Tujuan dan fungsi dari *autonomous mobile robot*

Dari pemilihan jenis *autonomous mobile robot* tersebut dapat disimpulkan jenis-jenis sensor yang akan digunakan. Tolok ukur keberhasilan dari sensor tersebut adalah optimalnya penggunaan sensor baik dari fungsionalnya serta dari komputasi yang dilakukan.

3.2 Tantangan SLAM

Dalam pengenalan lingkungan robot, pengembangan SLAM terus dilakukan untuk proses lokalisasi dan mapping lingkungan secara akurat dan cepat karena lingkungan yang disekitar *autonomous mobile robot* tidak hanya lingkungan yang statis tetapi juga dinamis. Tantangan kedepannya, dengan memanfaatkan sensor yang ada diharapkan *autonomous mobile robot* dapat melakukan proses lokalisasi dan *mapping* secara real time dan akurat juga dapat meminimalisir sumber daya komputasi yang terdapat pada *autonomous mobile robot*.

3.3 Tantangan Perencanaan Jalur Lokal

Setelah dapat melakukan pengenalan lingkungan maka *autonomous mobile robot* akan melakukan perencanaan jalur lokal. Tantangan dari perencanaan jalur lokal adalah penghindaran tabrakan dengan rintangan baik rintangan statis maupun rintangan dinamis. Selain itu pada perencanaan jalur lokal ini *autonomous mobile robot* juga akan melakukan pencarian optimal global.

3.4 Tantangan Perencanaan Jalur Global

Tingkat kesuksesan dalam melaksanakan perencanaan jalur global adalah dapat mencari jalur terpendek dengan memanfaatkan sensor, pengenalan lingkungan oleh SLAM dan perencanaan dari jalur lokal. Terdapat beberapa jenis metode yang digunakan. Penelitian selanjutnya dapat mengoptimasi dengan cara memperbaharui metode-metode tersebut atau dapat pula menggabungkan metode-metode tersebut dan tidak menutup kemungkinan pula untuk menciptakan metode baru dalam melakukan perencanaan jalur global.

Tabel 3. Tantangan dan Penelitian Lanjutan

Jenis Penelitian	Tantangan dan Penelitian Lanjutan
Sensor	Optimasi sensor visual (LIDAR, Stereo Camera, RGB-D Camera, Monocular Camera), Optimasi sensor lainnya seperti IMU
SLAM	Lokalisasi dan Mapping
Perencanaan Jalur Lokal	Penghindaran tabrakan dan pencarian optimal global
Perencanaan Jalur Global	Pencarian jalur terpendek, Modifikasi dan penggabungan metode perencanaan

4. KESIMPULAN

4.1 Kesimpulan

Penelitian ini merupakan literatur review dari sistem navigasi pada *autonomous mobile robot*. Terdapat empat bagian dalam sistem navigasi robot ini yaitu:

- a) Bagian pertama adalah sensor yang berfungsi untuk mengambil data lingkungan di sekitar robot dan menjadi inputan yang akan mempengaruhi proses perencanaan jalur yang dilalui oleh *autonomous mobile robot*.
- b) Bagian kedua adalah SLAM yang merupakan gabungan dari lokalisasi serta mapping yang dilakukan oleh robot. SLAM tergantung pada sensor yang digunakan oleh robot tersebut.
- c) Bagian ketiga adalah perencanaan jalur lokal yang merupakan implementasi langsung dari hasil lokalisasi dari SLAM. Terdapat beberapa tantangan dalam melakukan perencanaan jalur lokal seperti penghindaran tabrakan.
- d) Bagian terakhir adalah perencanaan jalur global. Tantangan dari perencanaan jalur global adalah pencarian jalur terpendek yang akan dilewati oleh robot.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada LPPM UNTIRTA yang telah memberikan dana hibah penelitian dosen pemula Fakultas Teknik.

REFERENSI

- [1] N. America, "Executive Summary World Robotics 2019 Industrial Robots," pp. 13–16, 2019.
- [2] Todd Litman, "Autonomous Vehicle Implementation Predictions: Implications for Transport Planning," *Transp. Res. Board Annu. Meet.*, vol. 42, no. January 2014, pp. 36–42, 2019.
- [3] Z. Meng, H. Sun, H. Qin, Z. Chen, C. Zhou, and M. H. Ang, "Intelligent robotic system for autonomous exploration and active SLAM in unknown environments," *SII 2017 - 2017 IEEE/SICE Int. Symp. Syst. Integr.*, vol. 2018-Janua, pp. 651–656, 2018.
- [4] C. Wang and D. Du, "Research on logistics autonomous mobile robot system," *2016 IEEE Int. Conf. Mechatronics Autom. IEEE ICMA 2016*, pp. 275–280, 2016.
- [5] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. USA: Bradford Company, 2004.
- [6] N. Ayache and F. Lustman, "Trinocular stereo vision for robotics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 1, pp. 73–85, 1991.
- [7] K. Shimura, Y. Ando, T. Yoshimi, and M. Mizukawa, "Research on person following system based on RGB-D features by autonomous robot with multi-Kinect sensor," *2014 IEEE/SICE Int. Symp. Syst. Integr. SII 2014*, pp. 304–309, 2014.
- [8] M. A. Markom, A. H. Adom, E. S. M. M. Tan, S. A. A. Shukor, N. A. Rahim, and A. Y. M. Shakaff, "A mapping mobile robot using RP Lidar scanner," pp. 87–92, 2016.
- [9] E. Ackerman, "Lidar that will make self-driving cars affordable [News]," *IEEE Spectr.*, vol. 53, no. 10, pp. 14–14, 2016.
- [10] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [11] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009, vol. 3.
- [12] P. Estefo, J. Simmonds, R. Robbes, and J. Fabry, "The Robot Operating System: Package reuse and community dynamics," *J. Syst. Softw.*, vol. 151, no. April, pp. 226–242, 2019.
- [13] ROS, "Setup and Configuration of the Navigation Stack on a Robot," *ROS.org*, 2020. [Online]. Available: <http://wiki.ros.org/navigation/Tutorials/RobotSetup>.
- [14] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 573–580, 2012.
- [15] D. Droschel and S. Behnke, "Efficient continuous-time SLAM for 3D lidar-based online mapping," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 5000–5007, 2018.
- [16] F. Aghili, "3D SLAM using IMU and its observability analysis," *2010 IEEE Int. Conf. Mechatronics Autom. ICMA 2010*, pp. 377–383, 2010.
- [17] G. Ajay Kumar, A. K. Patil, R. Patil, S. S. Park, and Y. H. Chai, "A LiDAR and IMU integrated indoor navigation system for UAVs and its application in real-time pipeline classification," *Sensors (Switzerland)*, vol. 17, no. 6, 2017.
- [18] L. Cheng, Y. Dai, R. Peng, and X. Nong, "Positioning and navigation of mobile robot with asynchronous fusion of binocular vision system and inertial navigation system," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 6, pp. 1–16, 2017.
- [19] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo Localization: efficient position

- estimation for mobile robots,” *Proc. Natl. Conf. Artif. Intell.*, no. Handschin 1970, pp. 343–349, 1999.
- [20] G. G. S. C, and B. W, “Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters,” *IEEE Trans. Robot.*, pp. 23–34, 2007.
- [21] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, “SSRR 2015 - 2015 IEEE International Symposium on Safety, Security, and Rescue Robotics,” *SSRR 2015 - 2015 IEEE Int. Symp. Safety, Secur. Rescue Robot.*, pp. 0–5, 2016.
- [22] S. Saat, W. N. Abd Rashid, M. Z. M. Tumari, and M. S. Saecalal, “HECTORSLAM 2D MAPPING for SIMULTANEOUS LOCALIZATION and MAPPING (SLAM),” *J. Phys. Conf. Ser.*, vol. 1529, no. 4, 2020.
- [23] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, “Efficient sparse pose adjustment for 2D mapping,” *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 22–29, 2010.
- [24] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2D LIDAR SLAM,” *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2016-June, pp. 1271–1278, 2016.
- [25] A. Topiwala, P. Inani, and A. Kathpal, “Frontier based exploration for autonomous robot,” *arXiv*, 2018.
- [26] B. Yamauchi, “A Frontier-Based Approach for Autonomous Exploration,” 1997.
- [27] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” *2007 6th IEEE ACM Int. Symp. Mix. Augment. Reality, ISMAR*, pp. 225–234, 2007.
- [28] M. Filipenko and I. Afanasyev, “Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment,” *9th Int. Conf. Intell. Syst. 2018 Theory, Res. Innov. Appl. IS 2018 - Proc.*, pp. 400–407, 2018.
- [29] J. Engel, V. Koltun, and D. Cremers, “DSO Journal: Direct Sparse Odometry,” 2016.
- [30] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: A Versatile and Accurate Monocular SLAM System,” *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [31] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. Jacobo Berlles, “S-PTAM: Stereo Parallel Tracking and Mapping,” *Rob. Auton. Syst.*, vol. 93, pp. 27–42, 2017.
- [32] M. Labbé and F. Michaud, “Online global loop closure detection for large-scale multi-session graph-based SLAM,” *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 2661–2666, 2014.
- [33] F. Dieter, B. Wolfram, and T. Sebastian, “The Dynamic Window Approach to Collision Avoidance,” pp. 137–146, 1997.
- [34] B. P. Gerkey and K. Konolige, “Planning and Control in Unstructured Terrain,” *ICRA Work. Path Plan. Costmaps*, 2008.
- [35] C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, “Trajectory modification considering dynamic constraints of autonomous robots,” *7th Ger. Conf. Robot. Robot. 2012*, pp. 74–79, 2012.
- [36] C. Rosmann, W. Feiten, T. Wösch, F. Hoffmann, and T. Bertram, “Efficient trajectory optimization using a sparse model,” *2013 Eur. Conf. Mob. Robot. ECMR 2013 - Conf. Proc.*, pp. 138–143, 2013.
- [37] C. Rösmann, F. Hoffmann, and T. Bertram, “Planning of multiple robot trajectories in distinctive topologies,” *2015 Eur. Conf. Mob. Robot. ECMR 2015 - Proc.*, 2015.
- [38] C. Rösmann, F. Hoffmann, and T. Bertram, “Integrated online trajectory planning and optimization in distinctive topologies,” *Rob. Auton. Syst.*, vol. 88, pp. 142–153, 2017.
- [39] C. Rosmann, F. Hoffmann, and T. Bertram, “Kinodynamic trajectory optimization and control for car-like robots,” *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2017-September, pp. 5681–5686, 2017.

- [40] H. Reddy, "PATH FINDING - Dijkstra's and A* Algorithm's," *Int. J. IT Eng.*, pp. 1–15, 2013.
- [41] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots in cluttered environments," vol. 19, no. 5, pp. 572–577, 1990.
- [42] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," *Tech. Report. Comput. Sci. Dep. Iowa State Univ.*, 1998.
- [43] L. G. D. O. Veras, F. L. L. Medeiros, and L. N. F. Guimaraes, "Systematic Literature Review of Sampling Process in Rapidly-Exploring Random Trees," *IEEE Access*, vol. 7, pp. 50933–50953, 2019.
- [44] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT," *Proc. - IEEE Int. Conf. Robot. Autom.*, no. May, pp. 1478–1483, 2011.
- [45] A. H. Qureshi and Y. Ayaz, "Potential Functions based Sampling Heuristic For Optimal Path Planning *," 2017.

