

Purwarupa *Autonomous Mobile Robot* Dengan *Hoverboard* dan Sensor RPLIDAR Menggunakan Algoritme Hector SLAM Dan Navfn

Bambang Gunawan Tanjung¹, Rizal Maulana², Rakhmadhany Primananda³

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹bambanggunawan@student.ub.ac.id, ²rizalmaulana@ub.ac.id, ³rakhmadhany@ub.ac.id

Abstrak

Jumlah permintaan pengiriman barang jasa ekspedisi mengalami peningkatan setiap tahunnya. Hal ini dipicu oleh eskalasi tren berbelanja *online*. Akibatnya banyak paket yang perlu dikelola di dalam gudang. Atas hal ini, diperlukan sistem distribusi otonom di dalam gudang agar menjadi lebih efisien dalam hal waktu dan tenaga kerja. Sistem terdiri dari sensor *encoder*, GY-521, RPLIDAR, motor BLDC, dan *Robot Operating System* (ROS). Sistem dengan ciri-ciri robot seperti ini dapat digolongkan sebagai jenis *autonomous mobile robot* (AMR). Keistimewaan penelitian terdapat pada penerapan *hoverboard* sebagai sistem penggerak robot. Proses pembuatan peta ruangan lokal dilakukan oleh algoritme Hector SLAM. Untuk navigasi menggunakan algoritme Navfn dan *Dynamic Window Approach* (DWA) dalam mencari perencanaan global dan lokal. Algoritme *Extended Kalman Filter* (EKF) dan *Adaptive Monte Carlo Localization* (AMCL) digunakan untuk lokalisasi robot. Hasil uji menunjukkan bahwa *hoverboard* dapat dipakai dalam sistem sebagai sistem penggerak dengan kemampuan berat angkut maksimal 40Kg. Kinerja sensor *odometry*, IMU, dan RPLIDAR menunjukkan hasil akurasi sensor masing-masing sebesar 99,31%, 96,22%, dan 99,84%. Untuk pembuatan peta, didapatkan hasil akurasi keberhasilan 80% menggunakan algoritme Hector SLAM. Pengujian navigasi tanpa halangan mencapai akurasi 80%. Sedangkan navigasi dengan halangan bergerak mencapai akurasi 80%.

Kata kunci: pemetaan, navigasi, ekspedisi, gudang, *autonomous mobile robot*, *hoverboard*, ROS.

Abstract

Every year, the number of requests for delivery of goods for expedition services has increased. The escalation of online shopping trends triggers this. As a result, many packages need to be managed in the warehouse. This autonomous distribution system is needed in the warehouse to be more efficient in terms of time and labor. The system consists of a sensor encoder, GY-521, RPLIDAR, BLDC motor, and Robot Operating System (ROS). Systems with mechanical characteristics like this can be classified as a type of autonomous mobile robot (AMR). The research's specialty is applying the *hoverboard* as a robot propulsion system. The Hector SLAM algorithm carries out the local spatial map creation process. For navigation using the Navfn algorithm and the Dynamic Window Approach (DWA) in searching for global and local plans. The Extended Kalman Filter (EKF) and Adaptive Monte Carlo Localization (AMCL) algorithms are used for robot localization. The test results show that the *hoverboard* can be used as a propulsion system with a maximum load capacity of 40Kg. The performance of the *odometry*, IMU, and RPLIDAR sensors showed sensor accuracy results of 99.31%, 96.22%, and 99.84%, respectively. The results obtained an accuracy of 80% success for map making using the Hector SLAM algorithm. Free navigation tests yield 80% accuracy. In comparison, navigation with moving obstacles gets 80% accuracy.

Keywords: mapping, navigation, expedition, warehouse, autonomous mobile robot, *hoverboard*, ROS

1. PENDAHULUAN

Setiap tahunnya, jumlah permintaan pengiriman barang jasa ekspedisi mengalami peningkatan (BPS 2017). Salah satu penyebabnya adalah meningkatnya frekuensi

belanja *online* di platform *e-commerce* sejak tahun 2017 (Databooks 2019) yang akan terus meningkat selama beberapa tahun ke depan (Statista 2019). Jika peningkatan ini tidak dibarengi dengan sistem distribusi yang lebih modern, maka akan menjadi masalah di tahun-

tahun mendatang. Rata-rata waktu pengiriman di Indonesia adalah 3,8 hari. Data ini didapat dari survei yang dilakukan oleh (Parcel Perform 2019). Survei ini juga menunjukkan bahwa 36% pelanggan tidak puas dengan layanan kurir. Dari 36% pelanggan yang melaporkan ketidakpuasan, 90% tidak puas dengan keterlambatan pengiriman, waktu pengiriman, dan kurangnya komunikasi mengenai status pengiriman.

Dari kumpulan fakta tersebut dapat disimpulkan bahwa jumlah permintaan pengiriman barang dari tahun ke tahun semakin meningkat, namun seiring berjalannya waktu sistem distribusi di gudang menjadi bermasalah. Proses pengiriman yang cepat ini berdampak pada waktu, tenaga, dan biaya yang terkait. Karena banyaknya masalah jasa ekspedisi yang ada, penulis hanya membahas sebagian kecil dari masalah pembagian *camp*. Solusi sistem robot otomatis diyakini dapat mengatasi masalah distribusi produk di gudang. Hal ini bertujuan untuk mempercepat proses penyortiran dan pemindahan barang di gudang. Pada saat yang sama, kebutuhan transportasi penumpang berkurang, yang secara signifikan dapat mengurangi biaya operasional.

Berdasarkan hal tersebut, diperlukan suatu sistem yang mendistribusikan paket secara mandiri sehingga prosesnya dapat dilakukan secara otomatis. Sistem ini menggunakan sensor *encoder*, GY-521, RPLIDAR, aktuator motor BLDC, dan sistem operasi robot (ROS) sebagai *middleware*. Sistem dapat menavigasi secara mandiri dan memahami kondisi lingkungan berdasarkan karakteristik ini. Robot bergerak otonom (AMR) adalah jenis robot terbaik untuk sistem ini. AMR berkewajiban mengeluarkan barang dari proses distribusi, proses penyortiran, dan mengangkutnya ke gudang.

Dalam implementasi tersebut, (Utomo 2015) mengusulkan untuk menggunakan sensor laser jarak jauh dengan jangkauan dengan sudut 360 derajat, seperti sensor RPLIDAR. Navigasi robot menggunakan perencanaan jalur global Navfn untuk mengoptimalkan waktu dan sumber daya. Selain itu, perencanaan jalur lokal dengan algoritme DWA digunakan untuk proses menghindari halangan bergerak. Hector SLAM bertindak sebagai algoritma untuk proses pemetaan karena robot membutuhkan peta lokal sebelum melakukan navigasi. Selama proses pemetaan dan navigasi, robot perlu mengetahui lokasinya saat ini secara bersamaan melalui proses lokalisasi. Lokalisasi yang digunakan menggunakan kombinasi algoritma AMCL dan

EKF. Kombinasi algoritme ini menyediakan sistem yang dapat bermanuver secara mandiri, menghindari rintangan, dan melacak lokasi..

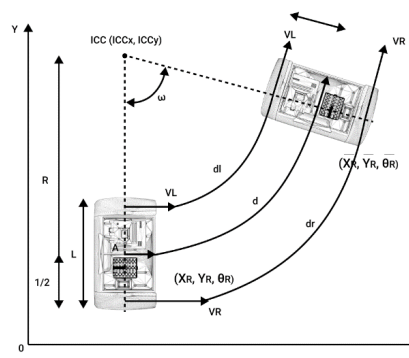
Hasil akhir dari penelitian ini adalah *prototype* robot otonom yang digunakan di gudang ekspedisi. Implementasi proses navigasi menggunakan peta global dan lokal. Harapan dari penelitian ini adalah agar perusahaan jasa ekspedisi Indonesia dapat memanfaatkan pengembangan robot mobil otonom untuk meningkatkan efisiensi waktu dan tenaga kerja serta menekan biaya produksi dalam proses pendistribusian barang.

2. DASAR TEORI

2.1 Hoverboard

Hoverboard menggunakan dua motor BLDC yang berputar pada kecepatan tinggi dan menghasilkan torsi yang besar. *Hoverboard* memiliki *encoder* internal yang digunakan untuk memperoleh data *odometry*, sehingga dapat digunakan sebagai penggerak motor untuk robot beroda. Dalam penelitian ini, *hoverboard* digunakan sebagai sistem penggerak menggunakan motor BLDC dan *motherboard*.

2.2 Differential Drive



Gambar 1. Kinematics Differential Drive

Penggerak diferensial adalah sistem penggerak dua roda dengan aktuator independen untuk setiap roda (Dudek dan Jenkin 2001). Agar robot dapat berputar, ia harus berputar di sekitar titik di sepanjang sumbu kiri dan kanan. Pada Gambar 1 menunjukkan diagram gerak penggerak diferensial yang memberikan rumus:

$$R = \left(\frac{L}{2} \frac{vr + vl}{vr - vl} \right) \quad (1)$$

$$\omega = \left(\frac{vr - vl}{L} \right) \quad (2)$$

Dari persamaan di atas, dapat diturunkan beberapa hal:

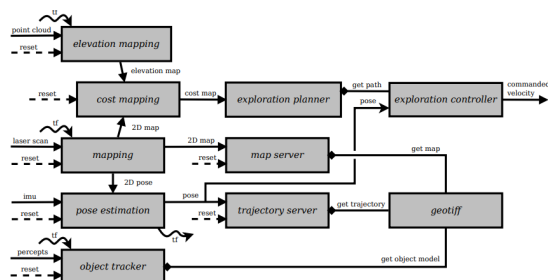
1. Jika $vl = vr$, maka didapatkan gerak maju linier..
2. Jika $vl = -vr$, maka didapatkan rotasi pada titik $l/2$ atau rotasi di tempat.
3. Jika $vl = 0$, maka didapatkan gerak poros ke kiri karena $R = l/2$, kebalikannya berlaku untuk roda kanan ketika $vr = 0$.

vl dan vr adalah nilai kecepatan pada roda yang dapat dikontrol untuk mendapatkan gerak rotasi yang dibutuhkan. l adalah jarak pusat antara kedua roda. Di sisi lain, R adalah nilai $l/2$ yang merupakan jarak antara pusat.

2.2.1 Simultaneous Localization and Mapping (SLAM)

SLAM (*Simultaneous Localization And Mapping*) adalah cara terbaru untuk memetakan lingkungan ke data di peta. SLAM melakukan eksplorasi dan pemetaan lingkungan yang tidak diketahui dan menggunakan sensor yang terpasang pada robot untuk memperkirakan sikap robot itu sendiri (Pyo, dkk. 2017). Untuk melakukan pemetaan, posisi robot terbaru harus diperbarui secara bersamaan. Sementara itu, Anda akan membutuhkan data peta untuk hasil pemetaan untuk memperbarui lokasi. Istilah SLAM digunakan untuk masalah pelokalan saat membuat peta.

2.2.1.1 Hector SLAM



Gambar 2. Skema Sistem Hector SLAM
Sumber: (Kohlbrecher, dkk. 2014)

Hector SLAM berisi satu set paket ROS yang digunakan untuk melakukan perhitungan SLAM dengan menghubungkan peta dengan perkiraan posisi robot di lingkungan yang tidak terstruktur (Kohlbrecher, dkk. 2014). Hector SLAM memiliki proses utama untuk membuat peta pemetaan. Pemetaan Hector menggunakan

laser lidar frekuensi tinggi tanpa informasi *odometry*. Pemetaan Hector menggunakan algoritma pencocokan pemindaian yang dikembangkan dengan menyelaraskan data dari sensor untuk membuat peta. Pada Gambar 2 menunjukkan proses pemetaan Hector, Geotiff Hector, dan bentuk lain dari skema SLAM Hector dan alur proses terkaitnya.

2.2.1.2 Hector Mapping

Pemetaan Hector adalah algoritma SLAM yang membuat peta *grid* yang ditempati tanpa perlu menggunakan *odometry* dalam metode pencocokan pemindaian. Metode pencocokan pemindaian untuk pemetaan Hector didasarkan pada pendekatan *Gauss-Newton* (Kohlbrecher, dkk. 2014). Menggabungkan sistem estimasi sikap dengan opsi unit *pitch / roll* yang menstabilkan pemindai laser, sistem dapat memetakan lingkungan bahkan di tanah yang tidak rata. Ada dua persamaan yang menjadi persamaan utama untuk proses pemetaan menggunakan pemetaan Hector. Pergeseran posisi robot dihitung dengan Persamaan 3 dan nilai H dihitung dengan Persamaan 4.

$$\Delta\xi = H^{-1} \sum_{i=1}^r \left[\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right]^T [1 - M(S_i(\xi))] \quad (3)$$

$$H = \sum_{i=1}^r \left[\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right]^T \left[\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right] \quad (4)$$

$\Delta\xi$ merupakan nilai perpindahan posisi robot dalam koordinat *cartesian* dan $S_i(\xi)$ merupakan koordinat hasil pemindaian lidar. Nilai M merupakan probabilitas adanya halangan pada koordinat hasil pemindaian.

2.2.2 Perencanaan Jalur Global Algoritme Navfn

Navfn merupakan algoritma pengoptimalan algoritma *Dijkstra*. Paket Navfn menyediakan implementasi cepat dari fitur navigasi interpolasi yang digunakan untuk merencanakan basis robot bergerak (Pyo, dkk. 2017). Algoritma optimasi *Dijkstra*, Navfn, memiliki kesamaan dalam hal pencarian tetangga, kecuali bahwa tidak seperti *Dijkstra* yang mengunjungi semua tetangga, Navfn hanya mengunjungi tetangga yang paling dekat dengan target.

2.2.3 Perencanaan Jalur Lokal Algoritme Dynamic Windows Approach (DWA)

Menurut (Pyo, dkk. 2017) pendekatan *dynamic window approach* adalah metode umum perencanaan penghindaran rintangan. Perencana jalur digunakan untuk perencanaan jalur lokal, tetapi DWA diganti karena kinerjanya yang sangat baik. Pertama, robot berada di ruang pencarian kecepatan yang berpusat pada kecepatan translasi v dan kecepatan rotasi, bukan pada koordinat x dan y .

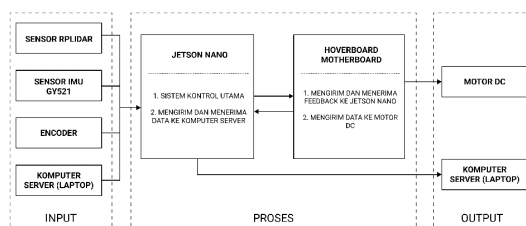
2.3 Robot Operating System (ROS)

ROS adalah sistem operasi meta yang menyediakan layanan seperti sistem operasi, abstraksi perangkat keras, kontrol perangkat tingkat rendah, implementasi fitur umum, pesan antar proses, dan manajemen paket. ROS mengembangkan, mengelola, dan mengirimkan aplikasi paket untuk berbagai tujuan. ROS dapat menjalankan beberapa paket dan sensor secara bersamaan, sehingga sistem terlihat paralel. ROS mengkhususkan diri dalam mendukung perpustakaan, aplikasi pihak ketiga, dan komunitas yang sangat aktif. Inilah alasan utama mengapa penelitian menggunakan ROS sebagai kerangka sistem utama.

3. PERANCANGAN DAN IMPLEMENTASI

3.1 Gambaran Umum Sistem

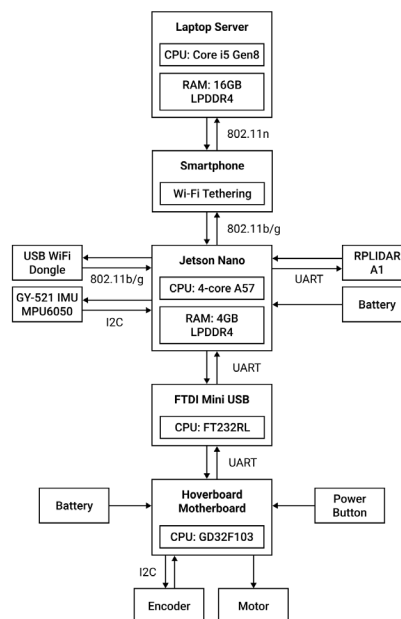
Sistem yang dibuat dilengkapi dengan fungsi pengangkutan barang dari titik awal ke tujuan, bergerak dengan roda, dan penginderaan lingkungan.



Gambar 3. Blok Diagram Sistem

Dari Gambar 3 menunjukkan diagram blok sistem yang terdiri dari tiga bagian: *input*, *proses*, dan *output*. Bagian *input* dari sistem akan menggunakan sensor RPLIDAR, IMU dan Odometer. Ketiga sensor ini digabungkan menjadi satu kesatuan untuk melakukan proses lokalisasi. Hasil sensor RPLIDAR digunakan untuk lokalisasi menggunakan metode AMCL, yang menggabungkan data laser dan peta untuk membuat estimasi posisi.

3.2 Perancangan Perangkat Keras

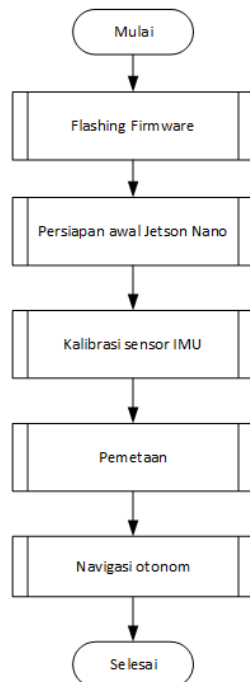


Gambar 4. Rancangan Umum Perangkat Keras

Gambar 4 adalah diagram tata letak khas perangkat keras sistem. Sistem ini menggunakan perangkat utama yang terintegrasi sebagai penghubung satu sama lain melalui beberapa protokol komunikasi. Sistem ini terdiri dari *motherboard hoverboard*, motor BLDC, baterai, sakelar *on/off*, dan Jetson Nano yang terhubung ke FTDI melalui protokol UART. *Motherboard hoverboard* bertindak sebagai *driver* motor utama yang mengatur arah dan kecepatan putaran motor BLDC.

3.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak dibagi menjadi lima proses utama yaitu *flashing firmware*, persiapan Jetson Nano, kalibrasi sensor IMU, pemetaan, dan navigasi otonom. Gambar 7 menunjukkan diagram alir proses dari program utama. Seluruh rangkaian program ini sudah berisi persyaratan dan kriteria untuk membuat robot dalam sistem robot bergerak otonom: pemetaan dan navigasi. Hal pertama yang perlu Anda lakukan dengan program ini adalah mengkalibrasi sensor IMU. Proses kalibrasi hanya dilakukan pada sensor IMU, karena sensor IMU membutuhkan nilai kalibrasi untuk setiap kondisi perakitan. Oleh karena itu, jika Anda memindahkan sensor IMU ke lokasi lain, Anda perlu mengkalibrasi ulang. Data mentah konsisten dan perangkat dilengkapi dengan *microcontroller*, sehingga tidak memerlukan sensor seperti *encoder* dan IMU.



Gambar 5. Diagram Alir Program Utama

3.4 Implementasi Keseluruhan Sistem

Pada Gambar 6 menunjukkan hasil implementasi alat dan sistem. Sensor RPLIDAR terletak di depan robot, sehingga memudahkan untuk membaca jika ada objek di tengah robot. Jika sensor RPLIDAR terletak di bagian belakang robot dan objek lebih tinggi dari sensor RPLIDAR, robot tidak akan dapat melihat lingkungan depan dan tidak akan dapat merencanakan jalan. Sebaliknya, pada posisi maju, setidaknya RPLIDAR dapat membaca lingkungan sebelum gerakan maju robot dan posisi rintangan di belakangnya, sehingga perencanaan rute tidak terhambat.



Gambar 6. Implementasi Keseluruhan Sistem

4. PENGUJIAN DAN ANALISIS

Pengujian terdiri dari pengujian sensor, pemetaan dan navigasi. Uji pemetaan dijalankan dalam berbagai kondisi awal. Uji navigasi, di sisi lain, dilakukan pada berbagai posisi dan jarak.

4.1 Pengujian Kinerja Sensor *Encoder*

Pengujian kinerja *odometry encoder* bertujuan untuk mengetahui unjuk kerja motor BLDC pada jumlah putaran yang ditentukan.

Tabel 1 Hasil Pembacaan Sensor Encoder

No	Input	Pembacaan Odometry		Data Seharusnya
	m/s	Rad/s	RPM	RPM
1	0,2	2,30384	22	22
2	0,3	3,76992	36	34
3	0,4	4,81712	46	45
4	0,5	6,07376	58	57
5	0,6	7,12896	68	68
6	0,7	8,48232	81	80
7	0,8	9,63424	92	93
8	0,9	10,89888	104	103
9	1	12,0428	115	115
10	1,1	13,19472	126	126
11	1,2	14,6688	140	138
12	1,3	15,788	151	149
13	1,4	16,7552	160	162
14	1,5	18,01184	172	174
15	1,6	19,47792	186	185

Tabel 2 Hasil Akurasi Pembacaan Odometry

Pengukuran Kesalahan	Total	
	Total RPM Odometry	Total RPM Observasi
	1557	1551
WAPE	0,39%	
Nilai akurasi	99,61%	

Dari hasil pengujian yang dilakukan pada Tabel 1 didapatkan hasil akurasi sebesar 99,61% diperoleh. Skor akurasi ini dihitung berdasarkan Dapat disimpulkan bahwa performa dan konfigurasi *odometry* motor BLDC yang digunakan pada sistem dinilai baik dan akurat.

4.2 Pengujian Kinerja Sensor IMU

Pengujian kemampuan data IMU sistem ini adalah untuk mengetahui bagaimana kinerja perangkat GY-521 digunakan untuk mendeteksi perubahan kecepatan atau percepatan berdasarkan pergerakan sistem. Hasil pengujian kemampuan alat GY-521 dijalankan sebanyak 20 kali pada percepatan statis dan gerak 0 m/s, 0,3 m/s, 0,5 m/s, dan 2 rad/s.

Tabel 3 Hasil Pembacaan Sensor GY-521

No	m/s	Sensor			Smartphone		
		Xi	Yi	Zi	Xj	Yj	Zi
1	0,0	0,097	0,056	0,980	0,2942	0,3334	9,669
2	0,0	0,108	0,029	9,790	0,2942	0,3236	9,660
3	0,0	0,103	0,032	9,803	0,2942	0,3040	9,650
4	0,0	0,053	0,014	9,804	0,3432	0,2746	9,679
5	0,0	0,038	0,022	9,782	0,3432	0,6865	9,532
6	0,3	0,337	0,165	9,612	0,5001	0,1177	9,444
7	0,3	0,046	0,178	9,677	0,4119	1,0493	9,522

8	0,3	0,087	0,007	9,739	0,3334	0,1961	9,601
9	0,3	0,188	0,114	9,909	0,4805	0,1373	10,140
10	0,3	0,296	0,018	9,987	0,2059	0,0883	12,454
11	0,5	0,030	0,108	9,895	0,2550	0,3432	11,474
12	0,5	0,089	0,007	10,167	1,0199	1,3435	10,336
13	0,5	0,048	0,048	10,435	0,9120	0,4021	11,032
14	0,5	0,341	0,077	9,659	1,1866	2,0888	9,003
15	0,5	0,078	0,007	10,002	0,1863	0,2550	9,522
16	2,0	0,244	1,099	11,410	5,7957	2,3830	14,622
17	2,0	0,171	1,418	10,629	0,5394	1,8731	10,424
18	2,0	0,284	1,500	10,822	1,3533	2,0202	13,092
19	2,0	0,672	1,617	10,309	1,2160	1,4122	10,287
20	2,0	0,120	1,492	8,809	5,0014	3,3539	8,247

Dari hasil pengujian pada *Tabel 3* yang dilakukan untuk mengetahui penentuan pembacaan accelerometer oleh sensor GY-521 pada percepatan aktual robot untuk mencapai akurasi positioning. Didapatkan akurasi sebesar 81,93% seperti terlihat pada *Tabel 4*.

Tabel 4 Hasil Akurasi Pembacaan Data Sensor IMU

Pengukuran Kesalahan	Total	
	Sensor	Smartphone
	202,658	247,343
WAPE	18,07%	
Nilai akurasi	81,93%	

4.3 Pengujian Kinerja Sensor RPLIDAR

Pengujian kemampuan laser *point cloud* dari sistem ini adalah untuk melihat bagaimana kinerja perangkat RPLIDAR yang digunakan untuk memperoleh data jarak yang berubah dengan kondisi lingkungan.

Tabel 5 Hasil Pembacaan Sensor RPLIDAR

No.	Halangan	Pengukuran		RPLIDAR	
		Sudut	Jarak	Sudut	Jarak
1	Terdekat	180	9,7	180	1
2	Terjauh	0	6000	180	5988
3	Tembok	0	76	0	77
4	Tembok	90	65	90	66,3
5	Tembok	180	75	180	76,4
6	Tembok	270	65	270	66,5
7	Objek 1	180	43	180	43,7
8	Objek 2	0	60	0	61,3
9	Objek 3	135	50	141	51,5
10	Objek 4	315	42	330	43,5

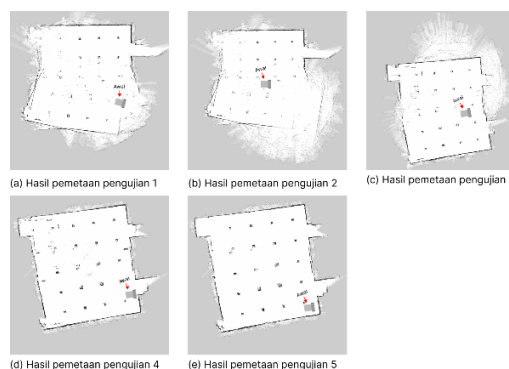
Tabel 6 Hasil Akurasi Pembacaan Sensor RPLIDAR

Pengukuran Kesalahan	Total	
	Total jarak Observasi	Total jarak RPLIDAR
	6485,7	6475,2
WAPE	0,16%	
Nilai akurasi	99,84%	

Dari hasil pengujian yang dilakukan pada

Tabel 5 didapatkan nilai akurasi sebesar 99,84% seperti yang terhitung pada *Tabel 6*. Berdasarkan hasil akurasi yang diperoleh maka dapat disimpulkan bahwa kinerja dan konfigurasi sensor RPLIDAR yang digunakan pada sistem dapat tergolong baik dan sangat akurat.

4.4 Pengujian Pemetaan



Gambar 7 Hasil Pemetaan

Pengujian ini dilakukan dengan mengaktifkan program pemetaan dan memanipulasi robot secara manual menggunakan *controller* robot *rqt*. Tujuan pengujian pemetaan menggunakan Hector SLAM. Hasil pengujian pemetaan SLAM Hector dijalankan sebanyak 5 kali pada posisi pemetaan awal yang berbeda. Berdasarkan bentuk denah basemen dapat disimpulkan bahwa 4 dari 5 pengujian yang dilakukan sesuai dengan bentuk peta denah *existing*. Kemudian akurasi dihitung 5 kali dan hasilnya akurasi 80% terdapat pada Persamaan 6.

$$\text{Akurasi} = \frac{12}{15} \times 100 = 80\% \quad (5)$$

Tabel 7 Hasil Pengukuran Peta Terhadap Denah

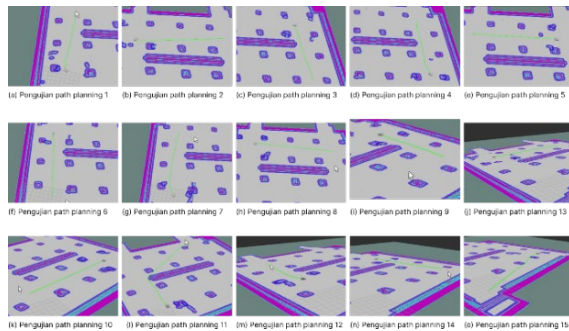
No	Ukuran Peta		Ukuran Denah Basemen	
	Panjang	Lebar	Panjang	Lebar
1	33.519	37.9165	31.32	37.1
2	36.2777	36.9291	31.32	37.1
3	34.842	37.971	31.32	37.1
4	33.1916	38.0782	31.32	37.1
5	33.6972	37.6029	31.32	37.1

Uji pengukuran peta pada *Tabel 7*, menangkap hasil panjang dan lebar dari peta yang dihasilkan oleh proses pemetaan. Hasil uji pengukuran peta menunjukkan akurasi 94,43% yang dihitung dengan WAPE. Hal ini dapat dilihat pada *Tabel 8*. Dari lima peta yang berhasil dibuat, peta kedua gagal karena ukuran tidak sesuai dengan pengukuran denah dan hasil *occupancy grid* menempati posisi yang salah.

Tabel 8 Hasil Akurasi Pengukuran Peta Terhadap Ukuran Denah Basemen Gedung G

	Total			
	Panjang	Panjang	Lebar	Lebar
Total data	171.5275	156.6	188.4977	185.5
WAPE	0.0953224		0.0161601	
Rata-rata WAPE			5.57%	
Nilai akurasi			94.43%	

4.5 Pengujian Navigasi Pada Halangan Diam



Gambar 8 Hasil Pengujian Perencanaan Jalur Global

Pengujian dijalankan 15 kali di berbagai titik sasaran. Keberhasilan pengujian diwakili oleh kecepatan waktu komputasi algoritma Navfn dan akurasi operasi robot saat berjalan. Tujuan dari pengujian ini adalah untuk mengetahui seberapa akurat manuver pergerakan robot dan waktu komputasi sistem ketika pengguna yang hanya memasukkan *input* berupa titik target bernavigasi secara mandiri tanpa kemudi. Selain itu, pengujian ini juga menentukan apakah robot AMRN layak digunakan di area gudang kurir. Hal ini karena bertabrakan dengan benda lain akibat pergerakan yang tidak stabil. Hasil pengujian diwakili oleh gambar posisi awal dan akhir untuk melihat seberapa besar perubahan posisi robot dari posisi target.

Tabel 9 Hasil Uji Waktu Komputasi Navigasi Dengan Algoritme Navfn dan DWA

No	Waktu Komputasi Path Planner (detik)	Waktu Tempuh Manuver (detik)	Total Waktu Navigasi (detik)
1	0.37	41.53	41.9
2	0.99	56.11	57.1
3	0.84	40.29	41.13
4	0.41	59.24	59.65
5	0.88	52.08	52.96
6	1.53	52.43	53.96
7	0.63	67.06	67.69
8	1.23	46.61	47.84
9	1.32	37.53	38.85
10	0.92		
11	0.98		
12	0.67	42.35	43.02

13	1.31	54.38	55.69
14	0.45	40.72	41.17
15	1.4		
Rata-rata waktu tempuh (detik)			50.08

Pengujian akurasi dan waktu kalkulasi dilakukan sebanyak 15 kali pada posisi yang berbeda untuk mengetahui akurasi dan waktu kalkulasi. Dari data pada Tabel 6.12 kita dapat melihat bahwa jumlah total navigasi yang berhasil adalah 12 dari 15 upaya.. Navigasi dianggap berhasil jika Anda dapat menavigasi dari titik awal ke tujuan. Proses robot tidak pernah mencapai titik target karena ada tiga pengujian yang menunjukkan kesalahan lokalisasi dan perencanaan jalur. Akurasi diberikan oleh Persamaan 6.

$$Akurasi = 12/15 \times 100 = 80\% \quad (6)$$

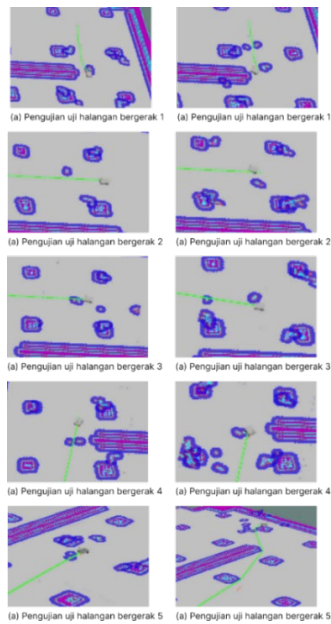
Saat menguji perhitungan navigasi dan waktu tempuh yang tercantum dalam Tabel 6.13 waktu tempuh rata-rata adalah 50,08 detik. Analisis ini dilakukan dengan menghitung waktu mulai dan titik akhir robot. Anda dapat menghitung selisih selama proses navigasi dengan menggunakan fungsi waktu dari aplikasi *Rviz*, atau lebih tepatnya jam dinding. Dari rata-rata waktu perhitungan ini, kita dapat melihat bahwa algoritma Navfn dan DWA memberikan waktu perhitungan yang cepat kurang dari 1 detik. Kecepatan pencarian jalur ini menjadikan algoritme Navfn sebagai perencana jalur terbaik untuk sistem di gudang.

4.6 Pengujian Navigasi Pada Halangan Bergerak

Pengujian akurasi dan waktu perhitungan ini dilakukan 5 kali pada setiap perbedaan posisi untuk menentukan akurasi dan waktu perhitungan yang dibutuhkan Navfn untuk menentukan jalur global. Jika Anda menghitung akurasi 5 kali, Anda mendapatkan akurasi Persamaan 7.

$$Akurasi = 12/15 \times 100 = 80\% \quad (7)$$

Kesimpulannya, sistem ini bagus dalam mendeteksi rintangan dan cenderung berhenti ketika rintangan di depan bergerak. Namun, jika rintangannya terlalu dekat atau terlalu cepat, Anda mungkin mendapatkan kesalahan perencanaan saat melacak ulang. Berkat *cloud* titik data RPLIDAR, sistem bahkan dapat menghindari pergerakan rintangan di depannya.



Gambar 9 Hasil Pengujian Menghindari Halangan Bergerak Dengan Perencanaan Jalur Lokal

5. KESIMPULAN

Pada pengujian sensor *encoder*, IMU, dan RPLIDAR didapatkan masing-masing nilai akurat sebesar 99,61%, 81,91%, dan 99,84%. Dari hasil ketiga akurasi tersebut dapat disimpulkan bahwa penggunaan sensor dan *software* sistem sangat berhasil dan dapat mendukung sistem robot bergerak otonom. Pada pengujian pemetaan menggunakan algoritme Hector SLAM menghasilkan nilai akurasi sebesar 80%. Ketika robot berada di luar area di sekitar tembok atau area sekitar tidak dapat terbaca oleh laser RPLIDAR, maka hasil peta cenderung buruk dikarenakan algoritme Hector SLAM tidak dapat memberikan data TF dengan akurat, sehingga posisi robot pada sistem tidak sinkron terhadap posisi sebenarnya. Pada pengujian ukuran hasil peta terhadap denah basemen menghasilkan akurasi sebesar 94.76%.

Pengujian navigasi tanpa halangan menggunakan algoritme Navfn dan DWA menghasilkan akurasi 80%. Hasil ini didapat dari pengujian sebanyak 15 kali/ Didapatkan 3 kegagalan navigasi dari 12 keberhasilan pada pengujian. Kegagalan navigasi ini dikarenakan pada percobaan tersebut posisi awal navigasi, dimulai dari tengah basemen yang mana RPLIDAR tidak dapat membaca area tembok terdekat, sehingga lokalisasi dengan AMCL mengalami *error*. Pada pengujian navigasi dengan halangan dinamis atau bergerak

menggunakan perencanaan global Navfn dan lokal DWA menghasilkan nilai akurasi sebesar 80%.

Dari keseluruhan pengujian dan implementasi, terdapat tiga hal yang dapat diperbaiki dan menjadi masukan untuk penelitian selanjutnya. Saran pertama menggunakan sistem kemudi *steering Ackerman* agar kemudi mudah dikendalikan. Kedua menggunakan variasi lebih banyak pengujian pemetaan dan navigasi. Ketiga menggunakan roda *omni wheel* dengan kombinasi 4WD motor BLDC agar robot dapat bergerak ke segala arah.

6. DAFTAR PUSTAKA

- BPS. 2017. *Laporan Hasil Survei Triwulanan Kegiatan Usaha Terintegrasi 2017*. Jakarta: Badan Pusat Statistik.
- Databooks. 2019. "Tren Pengguna E-Commerce Terus Tumbuh." *Dkatadata.co.id*.
- Dudek, Gregory, dan Michael Jenkin. 2001. "Computational Principles of Mobile Robotics." in *Computational Principles of Mobile Robotics*.
- Kohlbrecher, Stefan, Johannes Meyer, Thorsten Graber, Karen Petersen, Uwe Klingauf, dan Oskar Von Stryk. 2014. "Hector open source modules for autonomous mapping and navigation with rescue robots." *Lecture Notes in Computer Science*.
- Parcel Perform. 2019. "Consumers are still not happy with their e-commerce delivery experience, a new survey by Parcel Perform and iPrice Group reveals | Mini Me Insights." *Minime Insights*. Diambil 12 Maret 2022.
- Pyo, YoonSeok, HanCheol Cho, RyuWoon Jung, dan TaeHoon Lim. 2017. *Robot Programming From The Basic Concept To Practical Programming and Robot Application*. First Edit. Seoul, Republic of Korea: ROBOTIS Co., Ltd.
- Statista. 2019. *E-commerce in Indonesia - statistics & Facts*.
- Utomo, Eko Budi. 2015. "Autonomous Mobile Robot Berbasis Landmark Menggunakan Particle Filter Dan Occupancy Grid Maps Untuk Navigasi, Lokalisasi dan Mapping."