

Implementasi Algoritma Hector Slam dalam Melakukan Pemetaan Indoor Menggunakan Robot *Automated Guided Vehicle*

Implementation of Hector Slam Algorithm for Indoor Mapping Using Automated Guided Vehicle Robot

Muhamad Khosyi Izzatullah, Muhammad Aria Rajasa Pohan*

Program Studi Teknik Elektro, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112, Bandung, Indonesia 40132
Email* : muhammad.aria@email.unikom.ac.id

Abstrak – Robot otonom semakin berkembang dan digunakan dalam dunia industri dan menjadi penting yang dimana digunakan secara luas di berbagai sektor industri. Robot otonom bergerak menggunakan sistem navigasi untuk mengenali lingkungan, navigasi memiliki peran penting dalam robot otonom. Selain itu untuk menerapkan navigasi yang baik diperlukannya peta yang akurat dengan lingkungan sekitar agar robot dapat bergerak sesuai dengan lingkungan tersebut. Pada penelitian ini akan dilakukan analisis akurasi hasil pemetaan menggunakan metode algoritma Hector SLAM untuk melakukan pemetaan lingkungan sekitar dengan cara simulasi dan *real-time* (robot). Dalam penelitian ini memiliki beberapa komponen penting, yakni sistem operasi robot (*Robot Operating System*), filter data seperti *hector scan matching* dimana untuk mengkonstruksi data yang diperoleh, serta jenis sensor yang sangat berpengaruh terhadap kualitas peta yang dihasilkan. Penelitian ini menguji apakah algoritma *hector scan matching* dapat melakukan pemetaan lingkungan dengan akurat. Hasil dari penelitian ini dapat disimpulkan bahwa peta yang dihasilkan dari simulasi dan robot (*real*) ini memiliki akurasi yang cukup baik. Hal ini didasarkan dari pencocokan hasil pemetaan simulasi dan robot (*real*) pada skenario ke-1 yang memiliki akurasi sebesar SSIM: 77.26%, kesalahan MSE: 3.96%, Pixel Match: 55.46% dari hasil pemetaan robot (*real*). Sedangkan, dari hasil simulasi didapatkan pada skenario ke-2 yang digunakan untuk membandingkan dengan robot didapatkan akurasi sebesar SSIM: 87.29%, Kesalahan MSE: 1.65% dan Pixel Match : 81.11%.

Kata kunci : Robot Operating System, Hector-SLAM, Hector Scan matching, Mean Square Error, Pixel Match, Structural Similarity Image Metric, Peak Signal-to-Noise Ratio, LiDAR.

Abstract - Autonomous robots are increasingly advancing and being used in the industrial world, playing a significant role and widely employed across various industrial sectors. Autonomous robots navigate using navigation systems to perceive their environment, with navigation being crucial in autonomous robots. Additionally, to implement effective navigation, it requires an accurate map of the surrounding environment so that the robot can move according to its surroundings. This research analyzes the accuracy of mapping using the Hector SLAM algorithm method to map the surrounding environment through simulation and *real-time* (robot). This study comprises several important components, namely the robot operating system (*Robot Operating System*), data filters such as *Hector scan matching* used to construct acquired data, and sensor types significantly influencing the quality of the resulting maps. The research examines whether the Hector scan matching algorithm can accurately map the environment. The results of this study can be concluded that the maps generated from simulations and real robots have relatively good accuracy. This is based on the matching of simulation and real robot mapping results in scenario 1, which achieved an accuracy of SSIM: 77.26%, MSE error: 3.96%, Pixel Match: 55.46% of the real robot mapping results. Meanwhile, from the simulation results obtained in scenario 2 used for comparison with the robot, an accuracy of SSIM: 87.29%, MSE Error: 1.65%, and Pixel Match: 81.11% was obtained.

Keywords : Robot Operating System, Hector-SLAM, Hector Scan matching, Mean Square Error, Pixel Match, Structural Similarity Image Metric, Peak Signal-to-Noise Ratio, LiDAR

I. PENDAHULUAN

Saat ini, perkembangan dalam dunia industri semakin pesat seiring dengan perkembangan teknologi. Robot menjadi salah satu teknologi yang banyak digunakan dalam dunia industri, robot otonom digunakan secara luas di berbagai industri. Lokalisasi dan navigasi adalah teknologi yang menjadi inti dari robot industri untuk bergerak otonom. Umumnya robot otonom menggunakan pemindaian laser untuk bergerak otonom karena harga dari sensor laser yang tergolong murah.

Agar robot dapat bergerak secara otonom dibutuhkan proses navigasi robot otonom yang dibagi menjadi empat langkah, yakni pemetaan lingkungan, kontrol gerak, bentuk lintasan, dan fleksibilitas terhadap hambatan (penghindaran terhadap rintangan dalam lintasan). Dalam navigasi robot otonom yang dinamis terbagi menjadi dua variabel penting, yakni peta yang akurat dan navigasi jalur yang dapat menghindari rintangan dengan aman. Algoritma *Simultaneous Localization and Mapping* (SLAM) dapat digunakan untuk membangun peta lingkungan dinamis saat bergerak (real-time) dan memposisikan lokasi robot secara bersamaan. Algoritma SLAM terbagi menjadi dua yakni berbasis filter seperti Fast SLAM, Extended Kalman Filter-Simultaneous Localization and Mapping (EKF-SLAM), Unscented Kalman Filter-Simultaneous Localization and Mapping (UKF-SLAM), dll dan berbasis optimasi, salah satu contoh dari SLAM berbasis optimasi adalah SLAM dengan basis grafik/visualisasi seperti Graph-Based SLAM[1-4].

Pada penelitian sebelumnya yang dilakukan oleh Aria, penelitian tersebut bertujuan untuk melakukan pendekatan lokalisasi 2D dan pemetaan realtime pada robot dengan mengkombinasikan algoritma Particle Filter, Extended Kalman Filter (EKF), dan Iterative Closest Point (ICP). Hasil pengujian menunjukkan bahwa peta yang akurat dari jarak lantai yang besar dapat dihasilkan. Pemetaan waktu nyata dapat mencapai resolusi 5 cm. Telah ditunjukkan bahwa algoritma yang diusulkan memiliki kesalahan rata-rata jarak ke posisi sebenarnya yang lebih kecil dibandingkan dengan algoritma Particle Filter dan EKF-SLAM. Namun, penelitian ini belum membandingkan waktu komputasi dan kompleksitas memori antara keduanya algoritma[16][17][18].

Menurut penelitian yang dilakukan Kusmiawati dkk. Pada penelitian tersebut

dilakukan pemetaan menggunakan Lidar pada Unmanned Aerial Vehicle (UAV) yang menggunakan metode Hector SLAM untuk melakukan pemetaan 2D yang diuji menggunakan simulasi ROS (*Robot Operating System*), hasil yang diperoleh dari 36 skenario pengujian menunjukkan, akurasi peta terbaik didapatkan dengan nilai *Structural Similarity Index* (SSIM) sebesar 0.78, nilai *Mean Squared Error* (MSE) sebesar 5344.1, dan persentase *Pixel Matching* sebesar 89.59%. Waktu dan kecepatan quadcopter berpengaruh pada hasil pemetaan dimana nilai akurasi tertinggi, yaitu nilai *Structural Similarity Image Metric* (SSIM) 0.69, rata-rata nilai *Mean Square Error* (MSE) 8269.3, dan rata-rata nilai persentase *Pixel Matching* sebesar 84.98%. Hasil tersebut merupakan akurasi yang lebih baik dari skenario no hold, yaitu rata – rata nilai *Structural Similarity Image Metric* (SSIM) 0.68, rata-rata nilai *Mean Square Error* (MSE) 8641.3, dan rata-rata nilai persentase *Pixel Matching* sebesar 84.08%[19].

Hector SLAM didasarkan pada perbandingan data yang dipindai, sedangkan data pencocokan pemindaian akan dibuat menjadi peta, algoritma Gauss- Newton memecahkan data perbandingan data dari pemindaian sensor laser atau LiDAR, ini mencoba untuk menemukan posisi titik akhir pemindaian LiDAR dengan peta yang dibuat. Keuntungan besar dari metode ini adalah odometri robot tidak diperlukan. Di sisi lain, kerugian dari Hector SLAM adalah dalam kasus kombinasi dari kecepatan rangefinder frekuensi yang lambat, kurangnya odometri dan pergerakan robot yang cepat, menyebabkan peta menjadi tidak akurat. Perbandingan Hector SLAM dengan algoritma hybrid (PF/FIR) untuk lingkungan pemosisian robot, Menunjukkan bahwa filter hybrid memberikan perkiraan posisi yang lebih akurat, kontinu, dan andal. Selain itu, Hector SLAM masih belum dapat mencapai akurasi yang memuaskan[5] [6].

Navigasi merupakan hal yang penting dalam pembuatan Automated Guided Vehicle (AGV), namun pemetaan dengan akurasi yang tinggi juga merupakan hal yang paling penting karena navigasi tidak akan berguna jika tidak memiliki peta yang akurat.

Berdasarkan penelitian terdahulu, penelitian ini bertujuan untuk mengukur akurasi algoritma *Hector Scan Matching* dalam melakukan pemetaan lingkungan dengan menggunakan simulasi dan

robot (AGV). Sistematika pembahasan metoda pada paper ini adalah sebagai berikut. Metodologi penelitian berisi prinsip kerja sistem, perancangan perangkat keras, dan perancangan perangkat lunak. Selanjutnya hasil pengujian, pembahasan, analisis, dan diskusi yang dilakukan pada penelitian ini. Hasil pengujian, pembahasan, analisis, dan diskusi tersebut menjadi dasar dalam penarikan kesimpulan.

II. METODOLOGI

Fokus pada penelitian yang dilakukan adalah penggunaan metode yang dapat memetakan lingkungan/ruangan indoor dengan akurat dengan kondisi statis dan dinamis (terdapat lalu lalang manusia), untuk mengetahui akurasi dari metode Hector SLAM

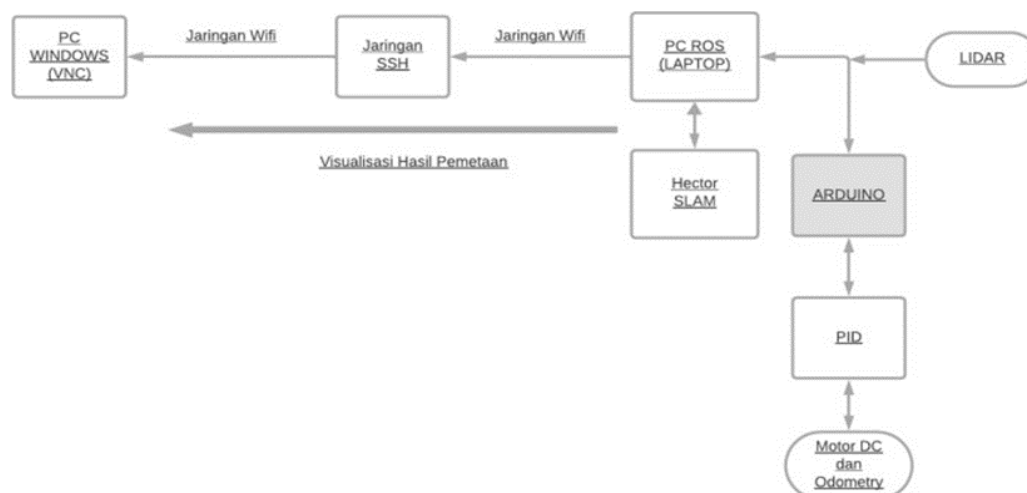
A. Perancangan Perangkat Keras

Robot pemetaan yang akan dibuat pada penelitian ini menggunakan dua buah komputer untuk melakukan pemetaan, yang mana fungsi dari masing-masing komputer tersebut adalah pengambilan data dan proses pembentukan peta. Rancangan sistem yang diusulkan untuk *mobile robot* ini adalah komputer dan jaringan yang merupakan komponen utama, dimana komputer ini merupakan inti yang menjalankan ROS untuk menyediakan interaksi antara robot dengan manusia. Selain itu karena ROS dirancang dengan komputasi yang terdistribusi maka ROS (Laptop/komputer) tunggal dapat menjalankan banyak *ROS-slave* dalam jaringan yang sama. Jaringan di sini bertanggung jawab atas ketersediaan koneksi selama program berjalan. Jaringan yang cepat dengan latensi yang rendah sangat diperlukan. Robot ini dilengkapi dengan komputer untuk menjalankan ROS (laptop) yang berisi dua paket yakni Hector SLAM untuk

pemetaan 2D menggunakan sensor LiDAR (RPLiDAR A1M8) dan Arduino untuk kontrol robot. Interaksi dua arah antara manusia dan robot menggunakan *tele-operated*, yakni operator yang menentukan posisi tujuan berdasarkan peta yang diperoleh dari remote area dan robot mengirimkan kembali peta yang telah dicapai robot ke manusia sebagai hasil dari eksplorasi. pada **Gambar 1** merupakan rancangan komponen sistem secara keseluruhan untuk alat yang digunakan dalam penelitian ini.

B. Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan bagian penting dalam pengembangan sistem, karena nentukan bagaimana sistem akan berfungsi dan memproses data. Dalam perancangan perancangan perangkat lunak ini akan dibuat program untuk memanggil library yang kemudian memproses data menggunakan metode Hector SLAM, lalu hasil proses data akan divisualisasikan menggunakan aplikasi Rviz. Program akan dibuat menggunakan format Python, XML, XACRO dan dibantu oleh perangkat lunak Visual Studio Code. Pada proses pembuatan topik robot, program dikhususkan untuk memvisualisasikan robot dan sinkronisasi dengan robot asli serta pembuatan kontrol dan konfigurasi robot. Dimana visualisasi robot dalam aplikasi Rviz menggunakan Robot core, yang dalam pembuatannya menggunakan sebuah library yang menghubungkan code dengan code lainnya. Dalam program ini terdapat base link dan base_footprint yang berfungsi menyatukan sasis, roda, dan sensor sebagai suatu kesatuan dalam visualisasi pada aplikasi Rviz, kemudian dalam program di atas juga dapat diatur besar roda serta besar rangka robot yang dapat disesuaikan dengan kebutuhan penelitian.



Gambar 1 Rancangan Sistem Robot

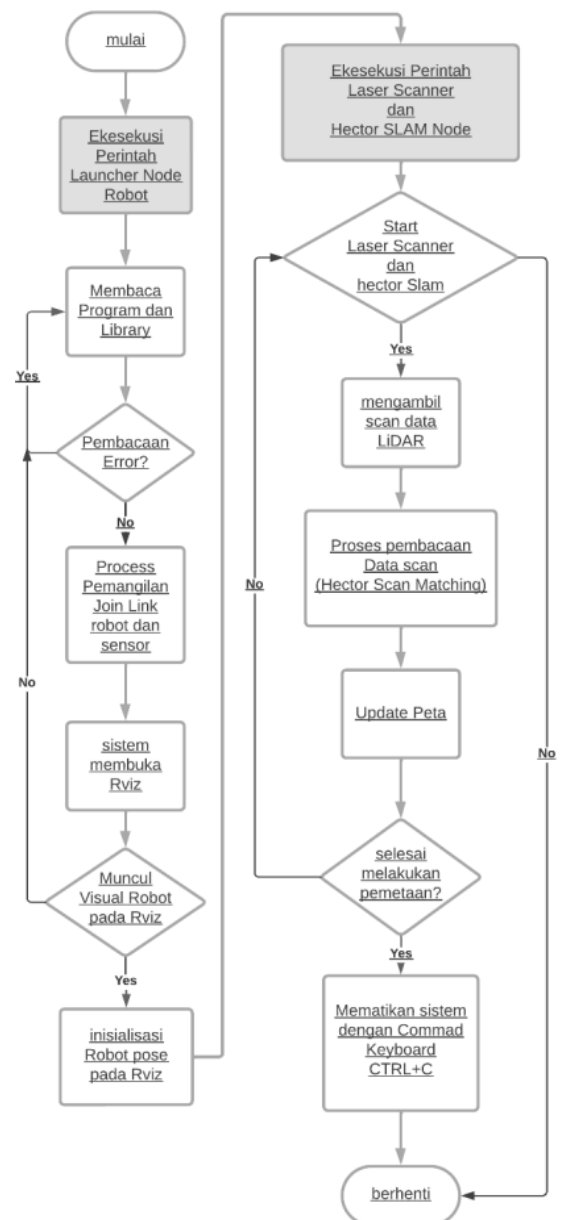
Setelah itu pembuatan *Inertial Macros*, dimana berperan sebagai massa dan menghitung inersia robot secara virtual agar robot tidak terjadi kesalahan jangka pendek (*glitch*) saat divisualisasikan dalam Rviz. Sedangkan proses pembacaan Sensor LiDAR merupakan langkah awal pengambilan data dari `sensor_msgs / laserscan` / topik atau bisa juga disebut dengan `frame grabber` dan yang terhubung dengan library dari Hector SLAM serta membuat data dari sensor divisualisasikan dalam Rviz.

Untuk pembuatan Kontrol Robot, yang harus dilakukan adalah membuat file konfigurasi untuk visualisasi maupun robot nyata yang menggerakkan kedua roda. Selanjutnya, dilakukan pembuatan pengontrol sesuai dengan konfigurasi yang dimana membaca mikrokontroler Arduino Nano untuk eksekusi dalam robot visual dan robot asli. Setelah itu dilanjutkan pada proses pembuatan Robot URDF yang bertujuan untuk dapat menyatukan seluruh program yang telah dibuat, seperti pengontrol dan *robot core*. agar tidak terjadi *error* akibat proses yang dilakukan secara bersamaan. Program ini juga berfungsi sebagai sendi yang menghubungkan berbagai program pengontrol dan *robot core*. Kemudian dilanjutkan dengan pembuatan *Launcher Node* yang berfungsi memanggil file robot URDF untuk melakukan proses pembacaan perangkat keras robot dan memulai visualisasi dalam aplikasi Rviz dan memanggil *library* pada paket ROS.

Setelah pembuatan semua sistem yang dibutuhkan dalam perangkat lunak selesai maka sistem untuk membaca semua sistem, digunakan terminal dalam OS (*Operating System*) Ubuntu untuk melakukan eksekusi *launcher node* sehingga robot dapat di kontrol, untuk proses pembacaan sistem seperti pada **Gambar 2**.

Sistem berjalan dengan mengeksekusi perintah *launcher node* yang bertujuan untuk memanggil dan membaca *library* pada sistem ROS (*Robot Operating System*), jika terdapat *error* dalam pembacaan *library* khusus seperti visual dapat diabaikan jika hanya dalam visualisasi bentuk virtual robot dalam Rviz. Proses setelah pemanggilan *library* dilakukan pembacaan perangkat keras seperti Arduino dan LiDAR untuk divisualisasikan dalam Rviz. Setelah proses visualisasi robot dilakukan proses inialisasi pose robot pada peta kisi yang masih kosong, kemudian dilakukan eksekusi perintah untuk memulai pemetaan dimana LiDAR dan Hector SLAM akan mulai mengambil data pemindaian lingkungan

yang kemudian dilakukan proses pembacaan data pemindaian oleh algoritma Hector SLAM.



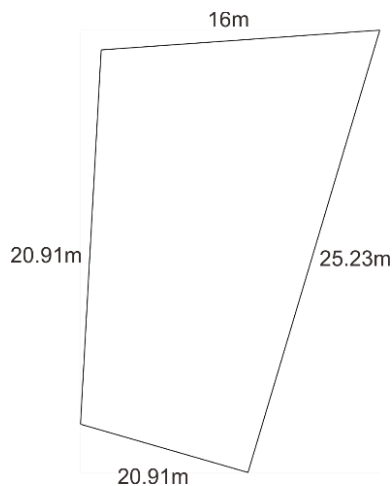
Gambar 2. Flowchart Sistem

III. HASIL DAN PEMBAHASAN

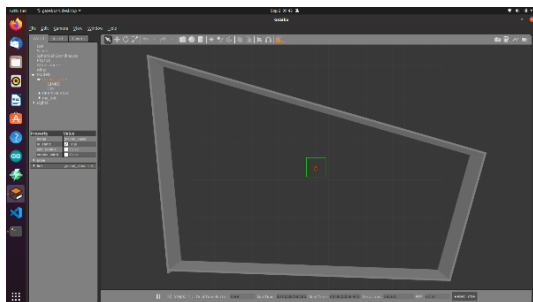
Berikut adalah hasil pengujian dan analisis data dari peta yang dihasilkan robot secara simulasi dengan kondisi lingkungan statis (tidak ada lalu-lalang manusia) dan langsung (perangkat keras) dengan kondisi lingkungan yang statis serta dinamis (terdapat lalu-lalang manusia) menggunakan metode Hector Scan Matcing. Hasil dari akurasi antara peta yang dihasilkan robot dengan peta asli yang telah diukur seperti pada **Tabel 1** sehingga gambar dalam gazebo akan berbentuk seperti pada **Gambar 3**, **Gambar 4**, dan **Gambar 5**.

Tabel I. Pengukuran Ruangan Perpustakaan Lama Gedung *Miracle* UNIKOM

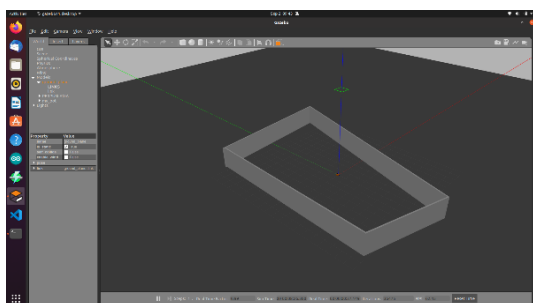
Sisi Kiri	16 meter
Sisi Kanan	9.3 meter
Atas	25.23 meter
Bawah	20.91 meter



Gambar 3. Data Lebar dan Panjang Perpustakaan Lama Gedung *Miracle* UNIKOM

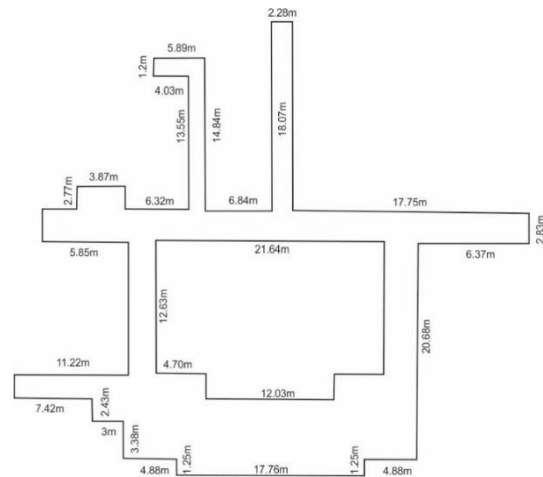


Gambar 4. Ruangan Perpustakaan Lama Gedung *Miracle* UNIKOM Tampak Atas

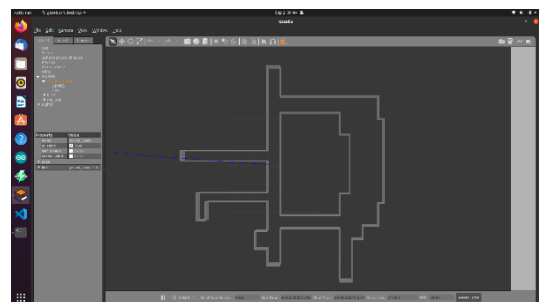


Gambar 5. Ruangan Perpustakaan Lama Gedung *Miracle* UNIKOM Tampak Samping

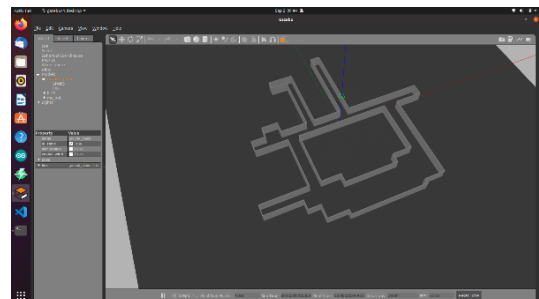
Dari data pada **Gambar 6** direkonstruksi menjadi peta yang dapat digunakan sebagai bahan uji untuk pemetaan secara simulasi seperti pada **Gambar 7** yang menunjukkan bentuk dan tampak peta dari hasil pengukuran lorong di gedung *Smart Building* lantai 12 UNIKOM.



Gambar 6. Data Lebar dan Panjang *Smart Building* UNIKOM Lantai 12



Gambar 7. Bentuk Lorong Lantai 12 *Smart Building* UNIKOM Tampak Atas



Gambar 8. Bentuk Lorong Lantai 12 *Smart Building* UNIKOM Tampak Samping

A. Pengujian dan Pengambilan Data

Pertama, peneliti menampilkan hasil dari pemetaan simulasi dengan kondisi statis dimana medan/lingkungan sekitar yang telah diukur sebelumnya dapat dilihat pada **Gambar 3**. Hasil yang ditunjukkan selama proses pemetaan ini dilakukan pengukuran waktu secara *real-time* dimana dilakukan pengambilan data dengan kondisi ruangan kosong, hasil pengambilan data tersebut merupakan hasil akhir pemetaan. Pengambilan data dilakukan sebanyak 5 kali dimana setiap percobaan dilakukan perhitungan waktu robot untuk melakukan pemetaan seperti yang ditunjukkan pada **Tabel II**.

Tabel II. Data Robot Simulasi Dalam Melakukan Pemetaan *Smart Building* UNIKOM Lantai 12

No	Waktu	Keterangan	Ground Truth	Hasil Pemetaan
1	5 menit 21 detik	Berhasil		
2	6 menit 5 detik	Berhasil		
3	6 menit 30 detik	Berhasil		
4	5 menit 15 detik	Berhasil		
5	5 menit 17 detik	Berhasil		

Selanjutnya, peneliti menampilkan hasil dari pemetaan simulasi dengan kondisi statis dimana medan/lingkungan sekitar yang telah diukur sebelumnya dapat dilihat pada **Gambar 6**. Hasil yang ditunjukkan selama proses pemetaan ini dilakukan pengukuran waktu secara real-time dimana dilakukan pengambilan data dengan kondisi ruangan kosong, hasil pengambilan data tersebut merupakan hasil akhir pemetaan. Pengambilan data dilakukan sebanyak 5 kali dimana setiap percobaan dilakukan perhitungan waktu robot untuk melakukan pemetaan seperti yang ditunjukkan pada **Tabel III**.

Terakhir, pengujian robot dilakukan pada ruangan yang sama dengan simulasi yakni Perpustakaan Lama UNIKOM dan dilakukan juga pemetaan pada Smart Building UNIKOM Lantai 12 secara keseluruhan. Pada **Tabel IV**, hasil tersebut merupakan hasil pemetaan robot dengan ruangan yang sama pada simulasi, kondisi ruangan ini kosong akan tetapi tidak statis dimana

banyak lalu-lalang orang melintas dalam ruangan tersebut. Pengambilan data dilakukan sebanyak 5 kali dimana setiap percobaan dilakukan perhitungan waktu robot untuk melakukan pemetaan.

B. Analisis Hasil Simulasi

Dari hasil percobaan yang dilakukan dapat diketahui seperti pada diagram batang yang ditunjukkan oleh **Gambar 9**, **Gambar 10**, **Gambar 11**, dan **Gambar 12**. Dari diagram tersebut, ditunjukkan bahwa robot dapat melakukan pemetaan dengan waktu yang relatif sama. Hal ini dipengaruhi oleh lebar ruangan dalam skenario 1 (pemetaan Smart Building Lantai 12). Nilai terendah dari MSE skenario ini terdapat pada percobaan ke-4, dimana presentase nilai MSE ini sebesar 0.5% dan nilai tertinggi sebesar 2.13% yang terjadi pada percobaan ke-1 dan ke-2. Metode MSE mengisyakakan bahwa

semakin kecil nilai/persentasenya maka semakin baik suatu sistem dalam memproses data menjadi gambar, hal tersebut berlaku juga untuk sebaliknya. Metode lain seperti SSIM merupakan kebalikan dari MSE, dimana semakin tinggi nilai/persentasenya maka semakin baik suatu sistem dalam memproses data menjadi gambar. Hal ini ditunjukkan pada percobaan ke-4 dengan metode SSIM ini memiliki nilai sebesar 100% yang dimana hasil pemetaan berhasil sama dengan peta referensi. Metode Pixel Match memiliki kesamaan

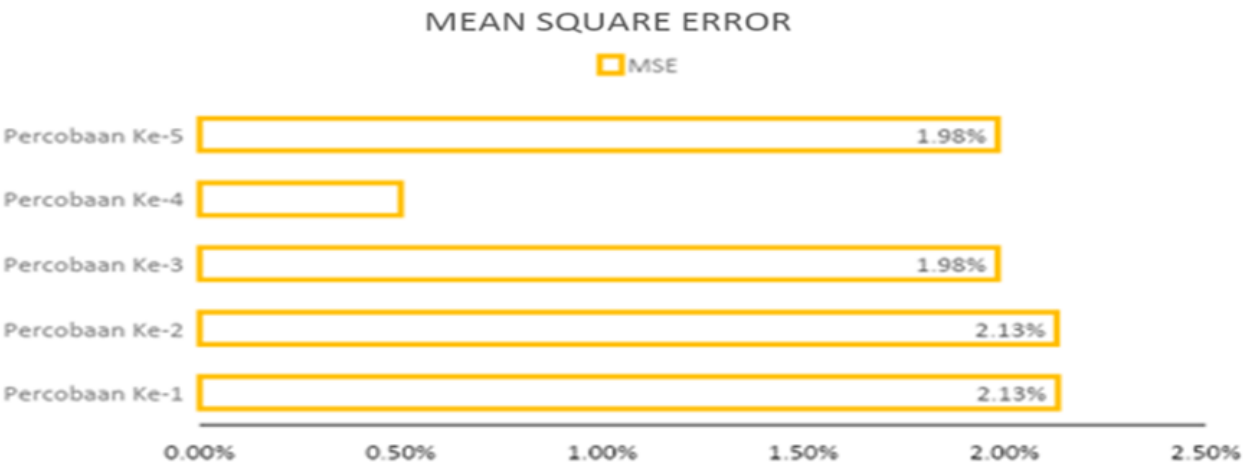
seperti SSIM dimana semakin besar nilai/persentasenya maka semakin baik pula algoritma/sistem dalam mengolah data menjadi gambar [20]. Nilai terbesar dari pixel match ini terdapat pada percobaan ke-4 dimana hasilnya 100%. berlaku pula pada PSNR (Peak Signal-to-Noise Ratio) dimana nilai tertinggi diperoleh pada percobaan ke-4 dengan besar nilai 65.42% , akan tetapi dalam PSNR nilai tinggi tidak selalu baik.

Tabel III. Data Robot Simulasi Dalam Melakukan Pemetaan Perpustakaan Lama Gedung *Miracle* UNIKOM

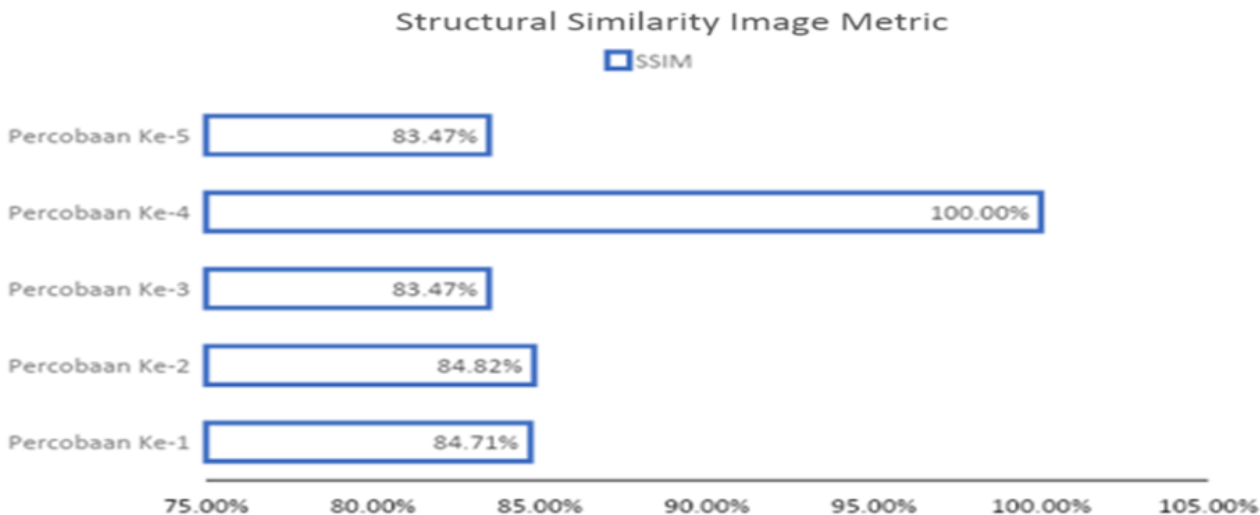
No	Waktu	Keterangan	Ground Truth	Hasil Pemetaan
1	2 menit 9 detik	berhasil		
2	2 menit 18 detik	berhasil		
3	3 menit 30 detik	berhasil		
4	3 menit 15 detik	berhasil		
5	3 menit 20 detik	Berhasil		

Tabel IV. Data Robot *Real* Dalam Melakukan Pemetaan Perpustakaan Lama Gedung *Miracle* UNIKOM

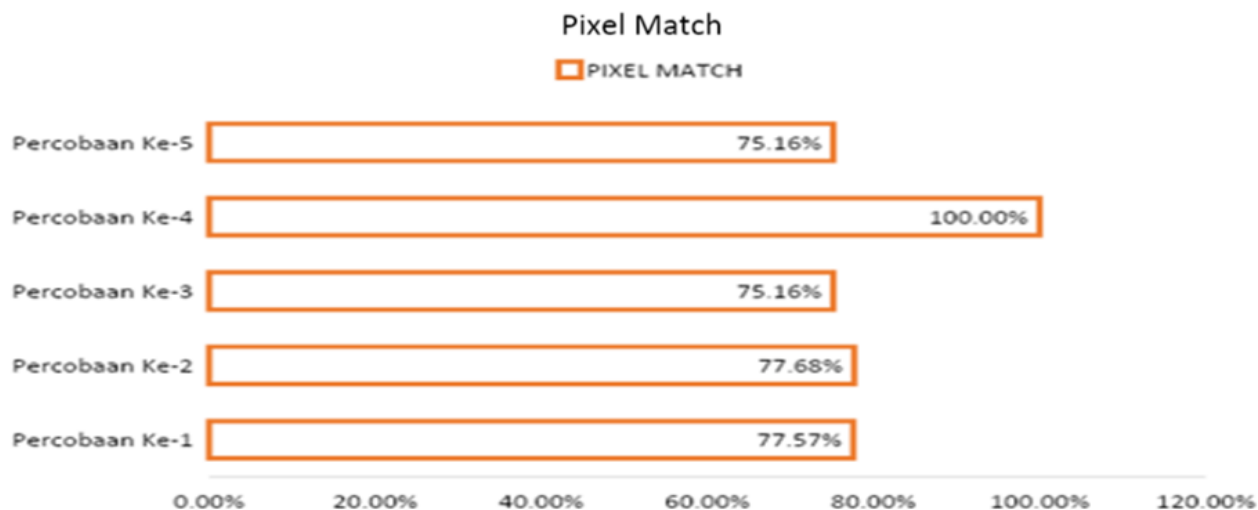
No	Waktu	Keterangan	Ground Truth	Hasil Pemetaan
1	5 menit 23 detik	berhasil		
2	6 menit 16 detik	berhasil		
3	5 menit 18 detik	berhasil		
4	5 menit 43 detik	berhasil		
5	5 menit 30 detik	berhasil		



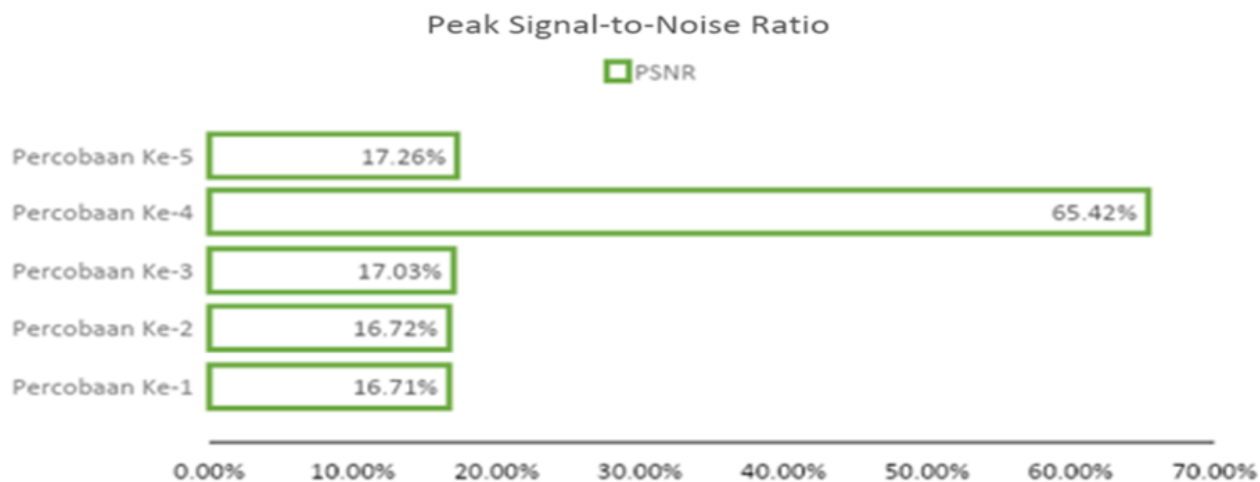
Gambar 9. Diagram Mean Square Error pada Sekenario 1



Gambar 10. Diagram Structural Similiarity Index Metric pada Skenario 1



Gambar 11. Diagram Pixel Match pada Skenario 1



Gambar 12. Diagram Peak Signal-to-Noise Ratio pada Skenario 1

Dalam hal memproses data menjadi gambar, pada percobaan ke-4 lah yang memiliki nilai

akurasi tertinggi berdasarkan dari nilai SSIM dan Pixel Match. Dari hasil percobaan skenario 2

(pemetaan perpustakaan lama Gedung *Miracle* UNIKOM) yang dilakukan dapat diketahui seperti pada diagram batang yang ditunjukkan oleh **Gambar 13**, **Gambar 14**, **Gambar 15**, dan **Gambar 16**. Nilai terendah dari metode MSE pada skenario ini terdapat pada percobaan ke-1 dimana presentase nilai MSE ini sebesar 0.74% dan nilai tertinggi sebesar 1.53% yang terjadi pada percobaan ke-3 dan ke-4. Pada percobaan ke-1 dengan metode SSIM skenario ini memiliki nilai sebesar 95% yang menunjukkan bahwa hasil pemetaan yang dilakukan hampir sama dengan peta referensi. Sama halnya dengan metode SSIM, dengan metode *Pixel Match* pada skenario ini diperoleh nilai tertinggi pada percobaan ke-1 dengan hasil sebesar 91.98%. Nilai tersebut menunjukkan bahwa hasil pemetaan yang dilakukan hampir sama dengan peta referensi. berlaku pula pada PSNR dimana nilai tertingginya diperoleh pada percobaan ke-1 dengan besar nilai 21.31%, akan tetapi dalam PSNR nilai tinggi tidak selalu baik. Dalam hal memproses data menjadi gambar percobaan ke-1 memiliki nilai akurasi tertinggi berdasarkan dari nilai SSIM dan *Pixel Match*. Berdasarkan data tersebut, diketahui bahwa algoritma Hector SLAM memiliki tingkat akurasi yang tinggi dimana rata rata dari SSIM, MSE dan *Pixel Match* memiliki persentase akurasi lebih dari 90.28%, 1.34%, dan 88.41% pada skenario 1 sedangkan pada skenario 2 87.29%, 1.65% dan 81.11%. Dari hasil tersebut diketahui bahwa selisih hasil antara skenario 1 dan skenario 2 tidak terlalu jauh. Akan tetapi, perlu diingat pada skenario 2 memiliki kompleksitas peta lebih lebih dibandingkan dengan skenario 1. Pada skenario 1 hanya terdapat ruangan kosong tanpa percabangan (kompleksitas).

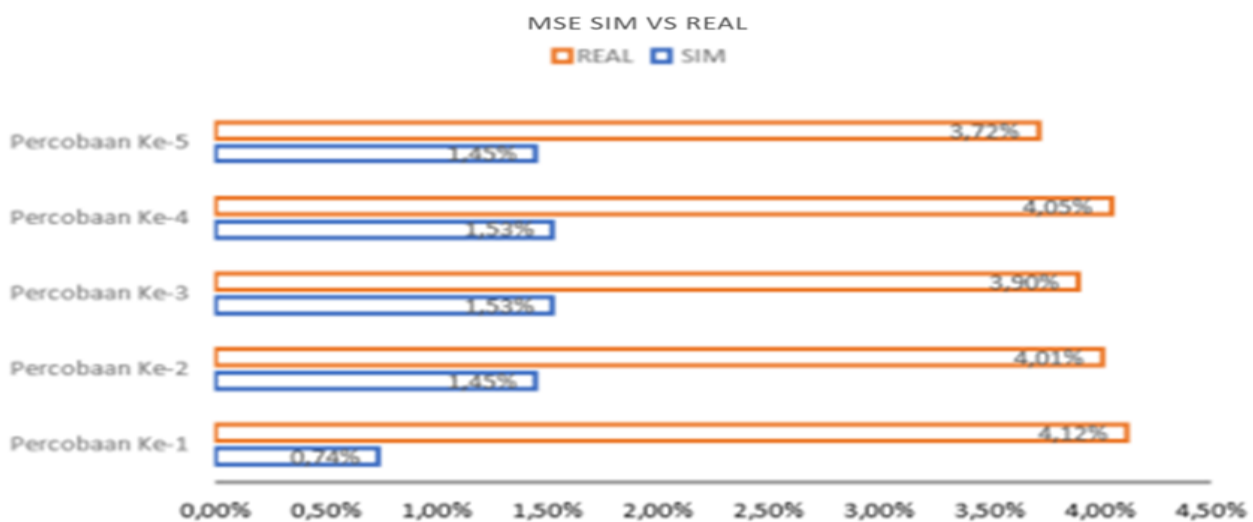
C. Analisis Hasil Robot

Pada analisis hasil pemetaan robot akan dibandingkan dengan hasil pemetaan simulasi sekenario 2 (perpustakaan lama Gedung *Miracle* UNIKOM) seperti pada diagram yang ditunjukkan oleh **Gambar 13**. Berdasarkan diagram tersebut, selisih nilai metode MSE dari hasil pemetaan simulasi dan real (robot) ini terbilang sangat besar. Akan tetapi nilai error yang diperoleh masih dibawah 5% yang dimana tergolong cukup baik untuk pemetaan secara langsung. Error yang terjadi pada pemetaan langsung dipengaruhi oleh keadaan lingkungan. Error tersebut tidak terjadi pada simulasi karena dalam simulasi semua kondisi berada dalam keadaan yang ideal.

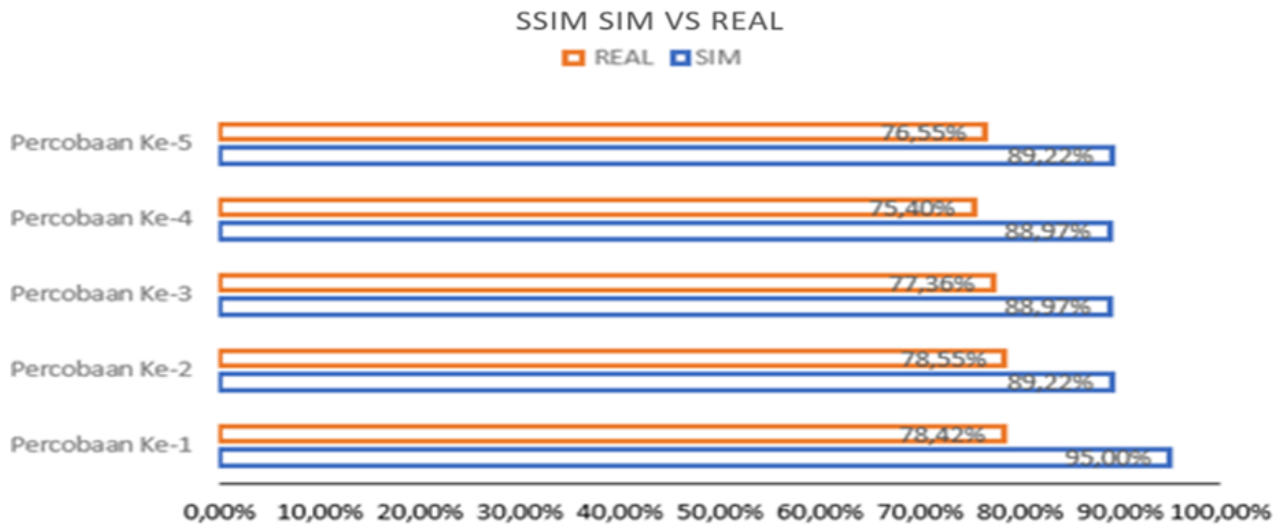
Berdasarkan diagram pada **Gambar 14**, selisih nilai metode SSIM dari hasil pemetaan simulasi dan real (robot) ini tidak terlalu besar jika dibandingkan dengan hasil metode MSE. Dengan nilai yang lebih dari 70% untuk hasil pemetaan real (robot) ini tergolong baik untuk pemetaan secara langsung yang dimana dipengaruhi oleh faktor-faktor lain seperti kondisi lingkungan dan jenis perangkat yang dipakai untuk membaca data.

Dalam hasil metode *pixel match* yang ditunjukkan oleh **Gambar 15** memiliki selisih yang cukup besar antara pemetaan secara simulasi maupun *real* (robot) dimana selisihnya kurang lebih sebesar 30%. Walaupun selisih tersebut besar nilai persentase dari *pixel match* ini sudah cukup baik.

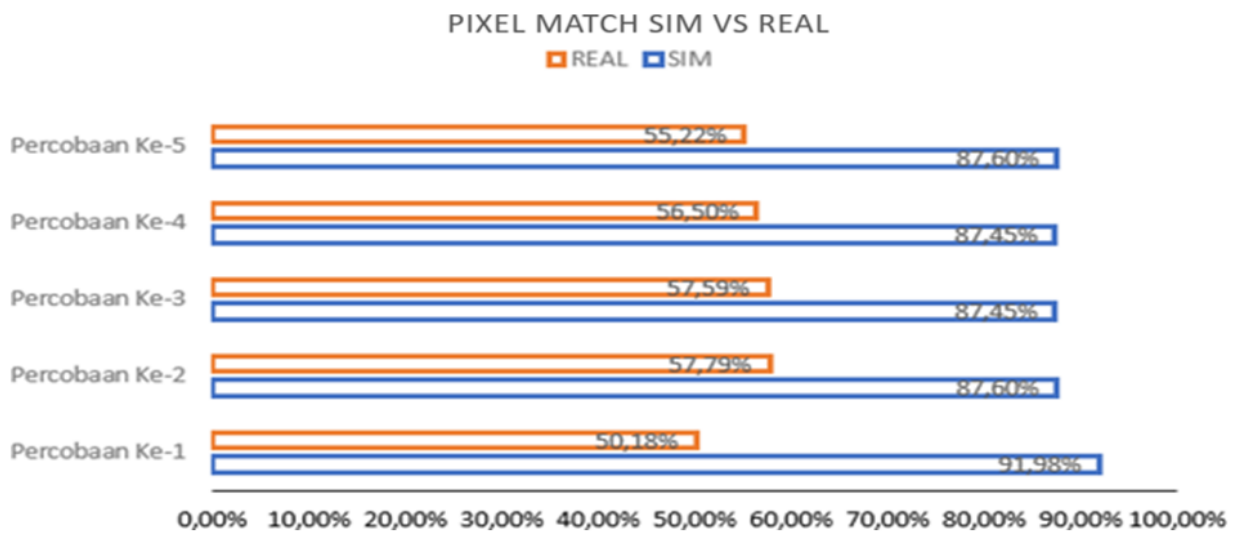
Dari data metode PSNR yang ditunjukkan oleh **Gambar 16** terbilang baik, karena hasil 4 dari 5 hasil percobaan dapat menghasilkan perbedaan yang kurang dari 6%.



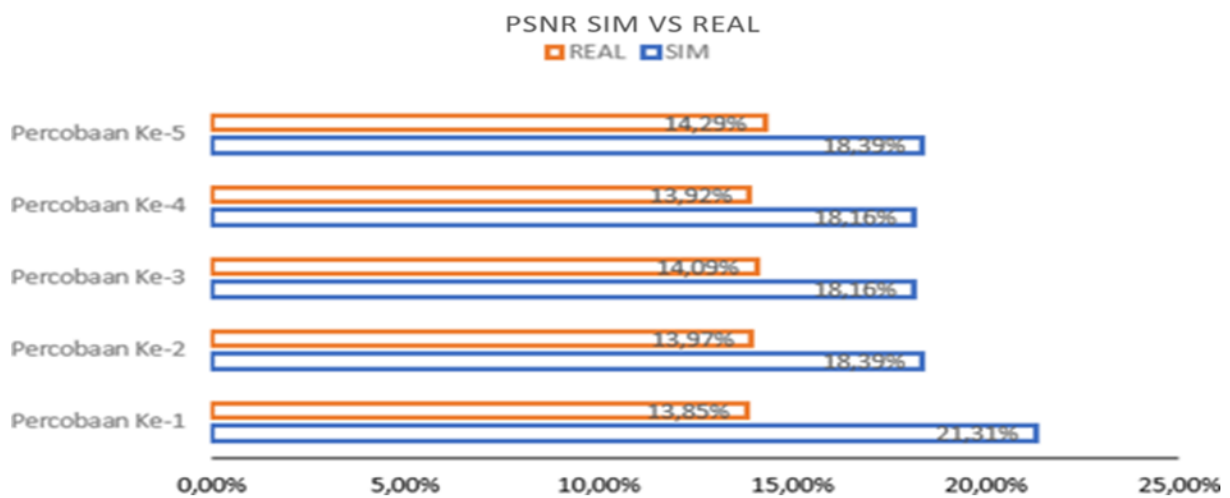
Gambar 13. MSE SIM VS REAL Pada Skenario 2



Gambar 14. SSIM SIM VS REAL Pada Skenario 2



Gambar 15. Pixel Match SIM VS REAL Pada Skenario 2



Gambar 16. PSNR SIM VS REAL Pada Skenario 2

Dari seluruh diagram diatas dapat diketahui bahwa hasil pemetaan simulasi lebih baik dibandingkan dengan hasil pemetaan robot (*real*), hal ini disebabkan karena dalam simulasi semua sistem dianggap absolut dan tidak memiliki hambatan baik dalam sensor maupun dari luar. Pada hasil pemetaan robot (*real*) diketahui pada implementasi nilai akan jauh berbeda dengan nilai pada saat simulasi, faktor dari luar maupun hardware ini akan mempengaruhi hasil pemetaan dimana lingkungan pemetaan yang memberikan pengaruh besar terhadap hasil pemetaan robot. Akan tetapi, dari percobaan yang dilakukan memiliki rata-rata sebesar SSIM: 77.26%, kesalahan MSE: 3.96%, persentase Pixel Cocok: 55.46%. Dari hasil perbandingan tersebut hasil pemetaan yang dilakukan robot masih dapat dikatakan akurat.

IV. KESIMPULAN

Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa peta yang dihasilkan dari simulasi dan robot (*real*) ini memiliki akurasi yang cukup baik. Hal ini berdasarkan dari pencocokan hasil pemetaan simulasi dan robot (*real*) pada skenario ke-1 yang memiliki akurasi sebesar SSIM: 77.26%, Kesalahan MSE: 3.96%, Pixel Match: 55.46% dari hasil pemetaan robot (*real*). Sedangkan, dari hasil simulasi didapatkan pada skenario ke-2 yang digunakan untuk membandingkan dengan robot didapatkan akurasi sebesar SSIM: 87.29%, Kesalahan MSE: 1.65% dan Pixel Match : 81.11%. Perbedaan hasil pemetaan simulasi dan robot (*real*) ini banyak disebabkan faktor dari luar dan spesifikasi sensor yang dimana dalam penelitian ini sensor yang digunakan merupakan grade hobby. Namun, bahkan dengan menggunakan sensor dengan kualitas tersebut, algoritma Hector SLAM dapat memberikan akurasi di atas 50% yang sulit untuk mendapatkan hasil ini jika digunakan pada robot. Pada beberapa penelitian sebelumnya, nilai SSIM adalah 78% dengan menggunakan drone.

DAFTAR PUSTAKA

- [1] P. Oksa, P. Loula, and E. Castrén, "Mapping, localization and navigation improvements by using manipulated floor plan and ROS-based mobile robot parameters," *Multidisciplinary Engineering Science and Technology*, vol. 5, issue 10, pp. 8910-8918, October 2018. K. Iagnemma and M. Buehler, eds., "Special issue: Special issue on the DARPA grand challenge, part 2," *Journal of Field Robotics*, vol. 23, no. 9, pp. 655-835, Sept. 2006.
- [2] S. Khan, D. Wollherr, and M. Buss, "Modeling laser intensities for simultaneous," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 692-699, July 2016
- [3] H. Peel, S. Luo, A. G. Cohn, and R. Fuentes, "An improved robot for bridge inspection," in *Proc International Symposium on Automation and Robotics in Construction*, 2017.
- [4] G. Grisetti et al., "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems*, vol. 2, no. 4, pp. 31-43, December 2010.
- [5] A. Bassiri, M. A. Oskoei, and A. Basiri, "Particle filter and finite impulse response filter fusion and hector SLAM to improve the performance of robot positioning," *Hindawi Journal of Robotics*, pp. 1-9, November 2018.
- [6] K. Sugiura and H. Matsutani, "A Universal LiDAR SLAM Accelerator System on Low-Cost FPGA," *IEEE Access*, vol. 10, pp. 26931-26947, 2022, doi: 10.1109/ACCESS.2022.3157822.
- [7] H. Achmad, M. R. Daud, S. Razali, and D. Pebrianti, "A ROS-based human-robot interaction for indoor exploration and mapping," *Int. J. Adv. Appl. Sci.*, 2016.
- [8] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an opensource Robot Operating System. In *ICRA workshop on open source software* (Vol. 3, No. 3.2, p. 5).
- [9] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46-57, Jun. 1989, doi: 10.1109/2.30720.
- [10] K. Wang, "Implementation of Odometry with EKF for Localization of Hector SLAM Method" (Doctoral dissertation, NSYSU). 2016.
- [11] M. Talukder, "Parameter variations in Hector Slam," thesis K. Sugiura and H. Matsutani, "A Universal LiDAR SLAM Accelerator System on Low-Cost FPGA," *IEEE Access*, vol. 10, pp. 26931-26947, 2022, doi: 10.1109/ACCESS.2022.3157822.
- [12] M. Sahuch et al., "Raspberry PI 3B + microcomputer as a central control unit in intelligent building automation management systems," *MATEC Web Conf.*, vol. 196, p. 04032, 2018, doi: 10.1051/mateconf/201819604032.
- [13] Kohlbrecher S, Von Stryk O, Meyer J and Klingauf U (2011). A flexible and scalable slam system with full 3d motion estimation. In *Safety, Security, and Rescue Robotics (SSRR)*, 2011 IEEE International Symposium on IEEE: 155-160.
- [14] M. Sahuch et al., "Raspberry PI 3B + microcomputer as a central control unit in intelligent building automation management systems," *MATEC Web Conf.*, vol. 196, p. 04032, 2018, doi: 10.1051/mateconf/201819604032.
- [15] M. Sahuch et al., "Raspberry PI 3B + microcomputer as a central control unit in intelligent building automation management systems," *MATEC Web Conf.*, vol. 196, p. 04032, 2018, doi: 10.1051/mateconf/201819604032.
- [16] M. Aria, "Real-Time 2D Mapping and Localization Algorithms for Mobile Robot Applications," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 662, no. 2, p. 022131, Nov. 2019, doi: 10.1088/1757-899X/662/2/022131.
- [17] M. A. R. Pohan and J. Utama, "Enhancing Passenger Comfort and Safeness in the Local Path Planning Strategy for Autonomous Road Vehicles Operating in New Capital City of Indonesia," *2023 9th International Conference on Signal Processing and Intelligent Systems (ICSPIS)*, Bali, Indonesia, 2023, pp. 1-5, doi: 10.1109/ICSPIS59665.2023.10402743
- [18] M. A. Rajasa Pohan and J. Utama, "Efficient Path Planning Algorithm for Unmanned Aerial Vehicle in Dynamic Environment," *2023 International Conference on Informatics Engineering, Science & Technology (INCITEST)*, Bandung, Indonesia, 2023, pp. 1-7, doi: 10.1109/INCITEST59455.2023.10397062.
- [19] S. Kusmiawati, E. Setiawan, and E. R. Widasari, "Simulasi Algoritme Hector SLAM untuk Pemetaan 2D pada Quadcopter berbasis ROS".
- [20] Sara, U. , Akter, M. and Uddin, M. (2019) Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *Journal of Computer and Communications*, 7, 8-18. doi: 10.4236/jcc.2019.73002.