

Simulasi Algoritme Hector SLAM untuk Pemetaan 2D pada *Quadcopter* berbasis ROS

Selina Kusmiawati¹, Eko Setiawan², Edita Rosana Widasari³

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹selinaks@student.ub.ac.id, ²ekosetiawan@ub.ac.id, ³editarosanaw@ub.ac.id

Abstrak

Quadcopter adalah salah satu *Unmanned Aerial Vehicle* (UAV) yang dikerahkan untuk beroperasi pada area yang tidak dapat diakses secara maksimal oleh *Unmanned Ground Vehicle* (UGV) pada struktur geografis yang telah terdistorsi akibat bencana alam. *Quadcopter* memerlukan kemampuan untuk mengenali lingkungan sekitarnya dengan menggunakan peta. Peta adalah seperangkat fitur yang menggambarkan lingkungan seperti dinding, rintangan, landmark, dan lain sebagainya. Peta relatif mudah dibuat pada lingkungan yang statis, akan tetapi pada lingkungan yang rusak akibat bencana, peta akan lebih sulit dibuat karena lingkungan yang sudah berubah. Solusi dari permasalahan ini adalah *quadcopter* harus bisa membangun peta lingkungannya sendiri. Untuk membangun peta, diperlukan proses pemetaan yang bisa dilakukan dengan *Simultaneous Localization and Mapping* (SLAM). Hector SLAM adalah salah satu algoritme SLAM yang bekerja berdasarkan teknik pencocokan pemindaian dan tanpa odometer. Simulasi dilakukan untuk menguji peta hasil pemetaan 2D algoritme Hector SLAM. Pemetaan dilakukan dengan sensor LiDAR yang disematkan pada *quadcopter* dan diujikan pada 3 environment berbeda. Simulasi dilakukan dengan simulator 3D Gazebo dan Rviz yang berbasis *Robot Operating System* (ROS). Terdapat 36 skenario pengujian yang dilakukan dengan 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%.

Kata kunci: *Quadcopter, Peta, Hector SLAM, Gazebo, Rviz, ROS*

Abstract

Quadcopter is an Unmanned Aerial Vehicle (UAV) that is deployed to operate in areas that are not maximally accessible by Unmanned Ground Vehicles (UGV) in geographic structures that have been distorted due to natural disasters. Quadcopter requires the ability to recognize the surrounding environment by using a map. A map is a set of features that describe the environment such as walls, obstacles, landmarks, etc. Maps are relatively easy to make in a static environment, but in a disaster-damaged environment, maps will be more difficult to create because the environment has changed. The solution to this problem is that the quadcopter must be able to build its own environmental map. To build a map, a mapping process is needed that can be done using Simultaneous Localization and Mapping (SLAM). Hector SLAM is one of the SLAM algorithms which works based on scan matching technique and without odometer. Simulations were carried out to test the 2D mapping results from the Hector SLAM algorithm. The mapping was carried out with a LiDAR sensor embedded in the quadcopter and tested in 3 different environments. Simulations were carried out with 3D Gazebo and Rviz simulators based on Robot Operating System (ROS). There are 36 test scenarios carried out with the best map accuracy obtained with a Structural Similarity Index (SSIM) value of 0.78, Mean Squared Error (MSE) value of 5344.1, and Pixel Matching percentage of 89.59%.

Keywords: *Quadcopter, Map, Hector SLAM, Gazebo, Rviz, ROS*

1. PENDAHULUAN

Indonesia sebagai negara kepulauan, memiliki karakteristik geografis secara tatanan

tektonik, dinamika meteorologis, maupun klimatologis yang rawan terhadap bencana alam (Murdiyanto, Gutomo, 2015). Badan Nasional Penanggulangan Bencana (BNPB) melaporkan akumulasi peristiwa bencana alam dari awal

Januari hingga akhir Agustus 2021 mencapai 1.805 kejadian (Firdaus A. 2021). Dampak dari bencana yang tersebut mengakibatkan korban meninggal dunia sebanyak 508 jiwa, korban hilang sebanyak 69 jiwa, korban luka-luka sebanyak 12.881 jiwa, dan korban mengungsi atau menderita sebanyak 5.8 juta jiwa. Untuk mengurangi dampak dari bencana alam, operasi pencarian dan penyelamatan perlu dilakukan dengan maksimal. Dalam kondisi mengancam nyawa, operasi pencarian dan penyelamatan manual untuk membantu korban yang terperangkap di bawah puing-puing reruntuhan ataupun terjebak genangan air mungkin tidak berjalan lancar sesuai dengan yang diharapkan yang menyebabkan penyelamatan tidak dapat dilakukan dengan tepat waktu.

Robot yang telah digunakan pada operasi pencarian dan penyelamatan antara lain adalah Inuktun dan iRobot (Gosset, 2021). Akan tetapi, robot beroda otonom yang telah dikembangkan tidak dapat beroperasi dengan maksimal pada struktur geografis yang telah terdistorsi akibat bencana. *Unmanned Aerial Vehicle (UAV)* telah dikerahkan di area yang dianggap berbahaya bagi operasi manusia dan memberikan informasi penting mengenai lingkungan dalam aplikasi seperti pencarian dan penyelamatan, inspeksi lokasi, pencarian korban dalam situasi bencana, dan pemantauan (Ravankar, 2018). Salah satu *Unmanned Aerial Vehicle (UAV)* yang banyak digunakan adalah *quadcopter*. Sebuah *quadcopter* memiliki empat rotor dengan sudut tetap dan mereka membuat *quadcopter* memiliki empat gaya input, yang pada dasarnya adalah daya dorong yang diberikan oleh masing-masing baling-baling (Thua, Gavrilova, 2017).

Quadcopter memerlukan kemampuan untuk mengenali lingkungan sekitarnya. Untuk mengenali lingkungan sekitarnya *quadcopter* menggunakan peta. Peta adalah seperangkat fitur yang menggambarkan lingkungan seperti dinding, rintangan, landmark, dan lain sebagainya. *Quadcopter* menggunakan peta untuk melokalisasi posisinya. Peta relatif mudah dibuat pada lingkungan yang statis. Akan tetapi pada lingkungan yang rusak akibat bencana, peta akan lebih sulit dibuat karena lingkungan yang sudah berubah. Solusi dari permasalahan ini adalah *quadcopter* harus bisa membangun peta lingkungannya sendiri. Untuk membangun peta, diperlukan lokalisasi untuk mengetahui posisi *quadcopter*, akan tetapi lokalisasi membutuhkan peta, untuk mengatasi permasalahan ini algoritme *Simultaneous Localization and*

Mapping (SLAM) memungkinkan robot melakukan dua tugas ini bersama-sama, menggunakan data dari pelokalan dan sensornya untuk memperluas peta dan data dari peta untuk mencapai pelokalan yang akurat (Mordechai, Fransesco, 2017).

Salah satu algoritma SLAM yang disediakan oleh Robot Operating System (ROS) adalah Hector SLAM. Hector SLAM menggabungkan sistem SLAM 2D berdasarkan teknik pencocokan pemindaian yang kuat. Algoritma SLAM ini dapat digunakan tanpa odometer serta pada platform yang menunjukkan gerakan *roll* atau *pitch* dari sensor (Saat et al, 2020).

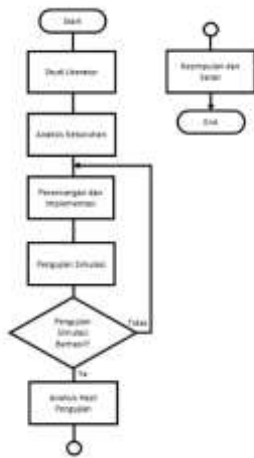
Bekerja dengan robot membutuhkan banyak sensor dan setiap proses harus ditangani secara *real-time* (Megalingsam, et al., 2018). ROS menyediakan layanan seperti pengiriman pesan antara proses dan paket, kontrol perangkat tingkat rendah, dan implementasi fungsi umum. Dalam ROS proses ditampilkan dalam format grafik dan pemrosesan terjadi di node.

Penelitian ini akan berfokus pada simulasi pemetaan yang dilakukan oleh *quadcopter* berbasis ROS pada lingkungan dalam ruangan untuk menyelidiki dan mencari kombinasi pengaturan parameter ukuran *grid cell*, jumlah *multi map resolution grid level*, kecepatan *quadcopter*, dan waktu pemindaian untuk menghasilkan peta dengan kualitas terbaik. Simulasi akan dilakukan pada simulator Gazebo dan Rviz.

2. METODOLOGI PENELITIAN

Pada Gambar 1 disajikan diagram alir dari penelitian ini. Dimulai dari studi literatur sebagai dasar teori dan pendukung penelitian ini. Selanjutnya dilakukan analisis kebutuhan untuk mengetahui kebutuhan apa saja yang diperlukan untuk simulasi. Setelah diketahui kebutuhan apa saja yang diperlukan, penelitian dilanjutkan dengan perancangan dan implementasi. Perancangan dan implementasi terdiri dari *environment* simulasi, *quadcopter*, dan

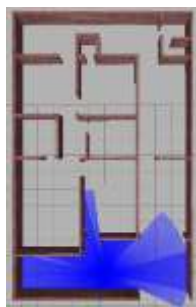
perangkat lunak.



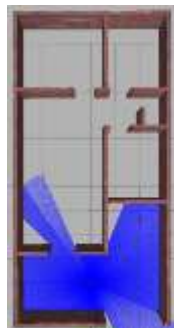
Gambar 2.1 Diagram Alir Penelitian

2.1. Perancangan dan Implementasi Environment

Perancangan *environment* dilakukan pada simulator Gazebo. Terdapat tiga *environment* dengan tiga ukuran dan desain yang berbeda.



Gambar 2 Environment 1 Gambar 3 Environment 2



Gambar 4 Environment 3

2.2. Perancangan Quadcopter

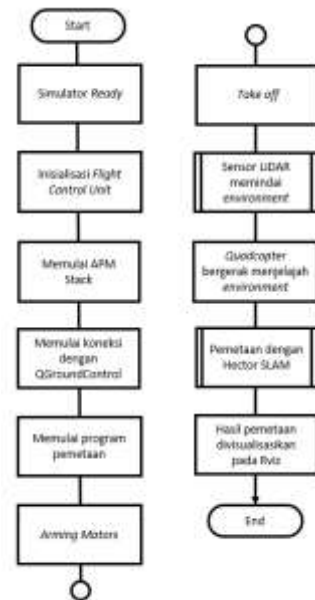
Quadcopter yang digunakan adalah

quadcopter dengan konfigurasi X. Semua baling – baling berguna untuk menghasilkan pergerakan seperti *roll*, *pitch*, dan *yaw*.



Gambar 5 *Quadcopter* Konfigurasi X

2.2. Perancangan dan Implementasi Perangkat Lunak



Gambar 6 Diagram Alir Utama Sistem

Simulasi ini dimulai dengan mempersiapkan simulator Gazebo dengan world yang berisi desain *environment* yang telah disiapkan sebelumnya. Selanjutnya dilakukan inisialisasi Flight Control Unit (FCU) dan APM Stack. Setelah itu menjalankan koneksi dengan Ground Control Station yaitu QGroundControl dan memulai program utama dari package Hector SLAM untuk melakukan pemetaan. Setelah program dimulai, dilakukan arming motors pada quadcopter melalui kontroler gamepad dan quadcopter diterbangkan untuk menjelajah *environment* yang telah disiapkan. Sensor LiDAR bekerja dengan memindai keseluruhan objek di dalam ruangan dan mengirimkan data pemindaian untuk diproses menjadi bentuk peta

yang divisualisasikan pada Rviz.

Perancangan perangkat lunak selanjutnya yaitu perancangan perancangan pengambilan data sensor LiDAR dan perancangan algoritme Hector SLAM.

3. PENGUJIAN

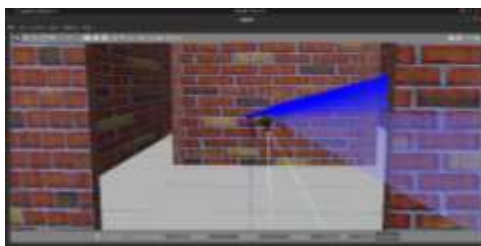
Pengujian dibagi menjadi 3 bagian yaitu pengujian gerak dasar *quadcopter*, pengujian sensor LiDAR, dan pengujian pemetaan 2D dengan algoritme Hector SLAM.

3.1. Pengujian Gerak Dasar *Quadcopter*

Pada pengujian ini komponen dari simulasi yang akan diuji adalah *quadcopter*. Dengan pengujian ini diharapkan *quadcopter* dapat bergerak sesuai dengan perintah yang diberikan untuk selanjutnya dapat digunakan untuk melakukan simulasi pemetaan. Prosedur pengujian adalah sebagai berikut:

1. Menjalankan simulator Gazebo dengan fail lidar.world
2. Menjalankan Flight Control Unit (FCU)
3. Memuat modul joystick pada Flight Control Unit (FCU)
4. Menjalankan APM Stack
5. Menjalankan ground station QGroundControl
6. Memberi perintah kepada quadcopter untuk bergerak melalui gamepad dengan kendali dari pilot.

Hasil dari pengujian pergerakan dasar *quadcopter* berhasil dilakukan. Gambar 7 menunjukkan hasil dari pergerakan *quadcopter*.



Gambar 7 Pergerakan *Quadcopter*

Hasil pengujian dilakukan dengan beberapa perintah disajikan pada Tabel 1 berikut.

Tabel 1 Hasil Pengujian Pergerakan *Quadcopter*

Perintah	Gerakan	Hasil
<i>Take off</i>	<i>Quadcopter</i> lepas landas	Berhasil

<i>Land</i>	<i>Quadcopter</i> mendarat	Berhasil
Maju	<i>Quadcopter</i> bergerak maju	Berhasil
Mundur	<i>Quadcopter</i> bergerak mundur	Berhasil
Ke Kiri	<i>Quadcopter</i> bergerak ke kiri	Berhasil
Ke Kanan	<i>Quadcopter</i> bergerak ke kanan	Berhasil
Naik	<i>Quadcopter</i> bergerak naik	Berhasil
Turun	<i>Quadcopter</i> bergerak turun	Berhasil
Berputar searah jarum jam	<i>Quadcopter</i> berputar searah jarum jam	Berhasil
Berputar berlawanan arah jarum jam	<i>Quadcopter</i> berputar berlawanan arah jarum jam	Berhasil
Diam di udara	<i>Quadcopter</i> diam di udara	Berhasil

Berdasarkan hasil tabel Tabel 1, pengujian pergerakan *quadcopter* berhasil dilakukan sesuai dengan perintah yang diberikan.

3.2. Pengujian Sensor LiDAR

Pada pengujian ini komponen dari simulasi yang akan diuji adalah sensor LiDAR. Sensor ini berperan untuk mengambil data pemindaian sinar laser yang berupa data header, angle_min, angle_max, angle_increment, time_increment, scan_time, range_min, range_max, ranges, dan intensities. Pengujian dilakukan dengan menjalankan simulasi pada simulator Gazebo dan memeriksa hasil pemindaian pada rostopic /scan. Dengan pengujian ini diharapkan sensor dapat bekerja dengan baik untuk menghasilkan data pemindaian laser yang selanjutnya akan digunakan untuk proses pemetaan. Pengujian sensor LiDAR dilakukan melalui tahapan berikut:

1. Menjalankan simulator Gazebo dengan

- fail lidar.launch
- 2. Membuka terminal baru dan menjalankan command rostopic echo /scan



Gambar 8 Hasil Pembacaan Sensor LiDAR

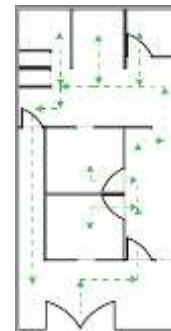
Hasil dari pemindaian sensor LiDAR disajikan pada Gambar 8 dimana dapat dilihat data hasil pemindaian berupa header, angle_min, angle_max, angle_increment, time_increment, scan_time, range_min, range_max, ranges, dan intensities.

3.2. Pengujian Pemetaan 2D dengan Algoritme Hector SLAM

Pengujian akan dilakukan pada 3 environment berbeda dengan masing – masing 12 skenario berbeda yang menghasilkan total 36 skenario pengujian. Pengujian ini dilakukan untuk mengetahui pengaruh ukuran *grid cell* dan jumlah *multi map resolution level* terhadap akurasi peta, pengaruh kecepatan quadcopter dan waktu pemetaan pada akurasi peta hasil pemetaan dengan menggunakan algoritme Hector SLAM. Pengujian algoritme Hector SLAM dilakukan melalui tahapan berikut:

1. Menjalankan simulator Gazebo dengan fail lidar.world
2. Menjalankan Flight Control Unit (FCU)
3. Memuat modul joystick pada Flight Control Unit (FCU)
4. Menjalankan APM Stack
5. Menjalankan ground station QGroundControl
6. Menjalankan Hector SLAM dengan fail tutorial.launch
7. Menjalankan quadcopter untuk menjelajah environment dengan kecepatan yang telah ditentukan dengan kendali dari pilot melalui gamepad
8. Melihat hasil simulasi secara keseluruhan.

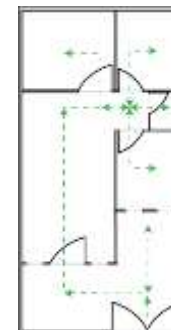
Berikut merupakan gambaran rute pergerakan quadcopter pada tiap environment



Gambar 9 Rute Pergerakan *Quadcopter* pada *Environment 1*



Gambar 10 Rute Pergerakan *Quadcopter* pada *Environment 2*



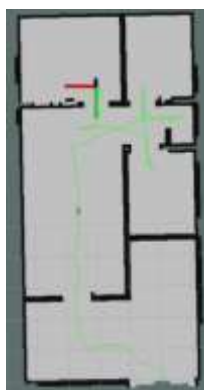
Gambar 11 Rute Pergerakan *Quadcopter* pada *Environment 3*



Gambar 12 Peta Hasil Pemetaan *Environment 1*



Gambar 13 Peta Hasil Pemetaan *Environment 2*



Gambar 14 Peta Hasil Pemetaan *Environment 3*

Gambar 12, Gambar 13, dan Gambar 14 menunjukkan peta hasil pemetaan dari masing – masing *environment* dengan berbagai skenario berbeda. Gambar 12 menunjukkan peta hasil pemetaan dengan skenario ukuran *grid cell* 0.025 m, jumlah multi map resolution level 3, dan kecepatan 0.0 – 2.5 mph. Gambar 13 menunjukkan peta hasil pemetaan dengan skenario ukuran *grid cell* 0.05 m, jumlah multi map resolution level 5, dan kecepatan 0.0 – 2.5 mph. Gambar 14 menunjukkan peta hasil pemetaan dengan skenario ukuran *grid cell* 0.1 m, jumlah multi map resolution level 7, dan kecepatan 0.0 – 2.5 mph.

Dari 36 skenario pengujian, didapatkan rata – rata waktu pemetaan *no hold* yaitu skenario pemetaan tanpa *hold quadcopter* di beberapa ruangan yaitu 7.12 menit. Sedangkan untuk skenario *hold* didapatkan rata – rata waktu pemetaan 8.07 menit.

4. ANALISIS HASIL PENGUJIAN

Parameter yang akan dianalisis adalah akurasi peta hasil pengujian, kecepatan pemetaan, dan waktu hasil pengujian. Analisis dibagi menjadi beberapa bagian yaitu analisis

akurasi peta dengan Structural Similarity Image Metric (SSIM). Analisis akurasi peta dengan Mean Squared Error (MSE). Analisis akurasi peta dengan Pixel Matching. Analisis pengaruh ukuran grid cell dan jumlah multi map resolution terhadap akurasi peta. Analisis pengaruh kecepatan quadcopter terhadap akurasi peta. Analisis pengaruh waktu pemetaan terhadap akurasi peta.

Analisis akurasi peta dengan Structural Similarity Image Metric (SSIM), Mean Squared Error (MSE), dan Pixel Matching dilakukan dengan bantuan perangkat lunak Matlab. Sebelum dilakukan proses analisis, perlu dilakukan beberapa pemrosesan citra yaitu:

1. Menghapus bagian dari citra hasil pemetaan yaitu garis trajectory yang dihasilkan dari quadcopter dan grid pada latar belakang peta pada Rviz.
2. Membuang bagian dari citra yang keluar dari garis peta hasil pemetaan agar proses pencarian nilai SSIM bisa dilakukan dan sesuai dengan peta asli.
3. Melakukan segmentasi citra peta hasil dari pemetaan dan citra peta asli dengan metode Otsu Thresholding untuk memperjelas *threshold* atau ambang batas garis – garis peta dan menghasilkan citra *grayscale*.

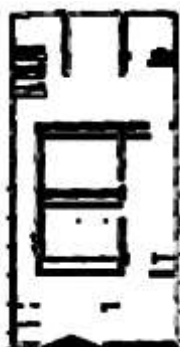


Gambar 15 Citra Peta Hasil Pemrosesan

Gambar 15 menunjukkan citra yang telah melewati beberapa pemrosesan. Setelah diproses, citra selanjutnya dapat dianalisis dengan menggunakan metode *Similarity Image Metric* (SSIM), *Mean Squared Error* (MSE), dan *Pixel Matching*.

4.1. Analisis Akurasi Peta dengan *Structural Similarity Image Metric* (SSIM)

Analisis ini dilakukan untuk mengetahui nilai kesamaan peta hasil pemetaan dengan peta asli dari desain *environment*.



Gambar 16 Citra Peta Hasil SSIM

Hasil analisis SSIM terendah memiliki nilai 0.48 yang dihasilkan dari skenario 7 yaitu pengujian dengan kecepatan antara 2.6 - 5.0 mph, ukuran *grid cell* 0.025 m dan jumlah *multi map resolution level* 3. Hasil tertinggi memiliki nilai 0.78 yang dihasilkan dari skenario 36 yaitu pengujian dengan kecepatan antara 2.6 - 5.0 mph, ukuran *grid cell* 0.1 m dan jumlah *multi map resolution level* 7 serta dilakukan *hold quadcopter* pada beberapa ruangan.

4.2. Analisis Akurasi Peta dengan Mean Squared Error (MSE)

Analisis ini dilakukan untuk mengetahui nilai *Mean Squared Error* (MSE) hasil pemetaan. Didapatkan nilai rata – rata MSE yaitu 8907.4. Nilai MSE terendah yang didapatkan yaitu 5344.1 dari pengujian skenario 12 dengan kecepatan antara 2.6 - 5.0 mph, ukuran *grid cell* 0.025 m dan jumlah *map resolution level* 3 serta dilakukan *hold quadcopter* pada beberapa ruangan. Nilai MSE tertinggi yang didapatkan yaitu 15893.1 dari pengujian skenario 7 dengan kecepatan antara 2.6 - 5.0 mph, ukuran *grid cell* 0.025 m dan jumlah *multi map resolution level* 3.

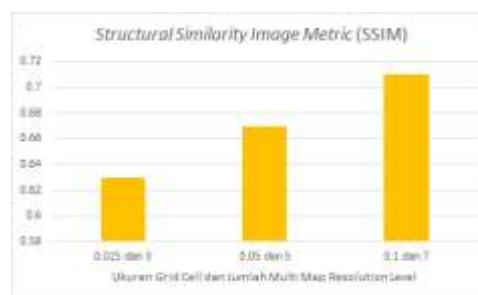
4.3. Analisis Akurasi Peta dengan Pixel Matching

Analisis ini dilakukan untuk mengetahui nilai kesamaan peta hasil pemetaan dengan peta asli dari desain *environment*. Didapatkan nilai persentase rata – rata yaitu 83.64%. Persentase tertinggi dengan nilai 89.59% didapatkan dari skenario 21 yaitu pengujian dengan kecepatan antara 0.0 - 2.5 mph, ukuran *grid cell* 0.05 m dan jumlah *multi map resolution level* 5. Persentase terendah dengan nilai 71.03% didapatkan dari skenario 7 yaitu pengujian dengan kecepatan antara 2.6 - 5.0 mph, ukuran *grid cell* 0.025 m

dan jumlah *multi map resolution level* 3.

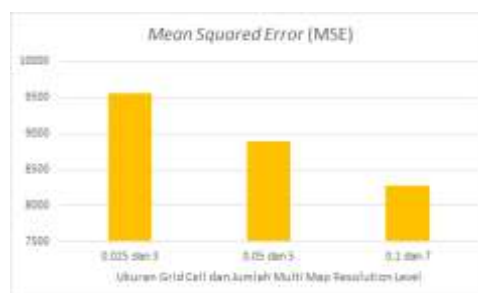
4.4. Pengaruh Ukuran Grid Cell dan Jumlah Multi Map Resolution Level Terhadap Akurasi Peta

Analisis ini dilakukan untuk mengetahui pengaruh ukuran *grid cell* dan jumlah *multi map resolution level* terhadap akurasi peta yang dihasilkan dari proses pemetaan. Analisis dilakukan berdasarkan nilai akurasi peta berdasarkan hasil analisis akurasi peta yang telah dilakukan dengan metode *Structural Similarity Image Metric* (SSIM), *Mean Squared Error* (MSE), dan *Pixel Matching*.



Gambar 17 Histogram Rata – rata Hasil SSIM Pengujian Ukuran *Grid Cell* dan Jumlah *Multi Map Resolution Level*

Dari Gambar 17 di atas, dapat disimpulkan pengujian dengan ukuran *grid cell* 0.1 dan jumlah *multi map resolution level* 7 menghasilkan peta dengan akurasi yang paling baik dari pengujian dengan ukuran *grid cell* 0.05 dan jumlah *multi map resolution level* 5 dan ukuran *grid cell* 0.025 dan jumlah *multi map resolution level* 3.



Gambar 18 Histogram Rata – rata Hasil MSE Pengujian Ukuran *Grid Cell* dan Jumlah *Multi Map Resolution Level*

Dari Gambar 18, bisa dilihat bahwa rata – rata akurasi peta yang paling baik didapatkan dari pengujian dengan ukuran *grid cell* 0.1 dan jumlah *multi map resolution level* 7 dengan nilai rata – rata MSE 8275.4.

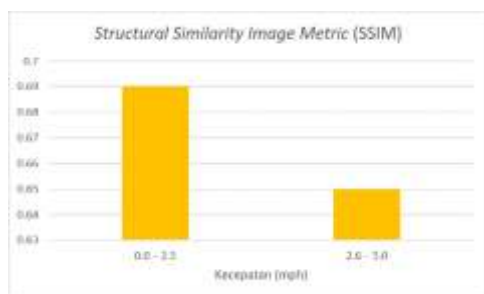


Gambar 19 Histogram Rata – rata Hasil *Pixel Matching* Pengujian Ukuran *Grid Cell* dan Jumlah *Multi Map Resolution Level*

Dari Gambar 19, rata – rata akurasi peta yang paling baik didapatkan dari pengujian dengan ukuran *grid cell* 0.1 dan jumlah *multi map resolution level* 7 dengan rata – rata persentase 85.31%.

4.5. Pengaruh Kecepatan *Quadcopter* Terhadap Akurasi Peta

Analisis ini dilakukan untuk mengetahui pengaruh kecepatan *quadcopter* terhadap akurasi peta yang dihasilkan dari proses pemetaan. Analisis dilakukan berdasarkan nilai akurasi peta berdasarkan hasil analisis akurasi peta yang telah dilakukan dengan metode *Structural Similarity Image Metric (SSIM)*, *Mean Squared Error (MSE)*, dan *Pixel Matching*.

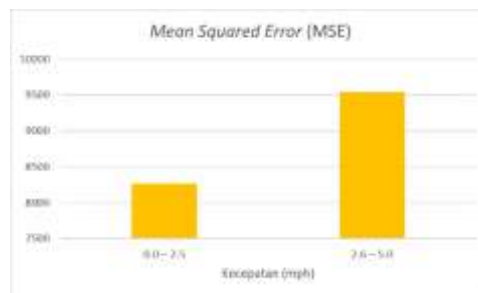


Gambar 20 Histogram Rata – rata Hasil *SSIM* Pengujian Kecepatan *Quadcopter*

Dari Gambar 20, dapat disimpulkan bahwa kecepatan 0.0 – 2.5 mph menghasilkan peta dengan akurasi yang lebih baik dari kecepatan 0.0 – 2.5. Dilihat dari nilai *Structural Similarity Image Metric (SSIM)* pada peta hasil pengujian dengan kecepatan 0.0 – 2.5 yang menghasilkan rata – rata nilai 0.69, lebih besar dari rata – rata nilai yang dihasilkan pengujian dengan kecepatan 2.6 – 5.0 yaitu 0.65.

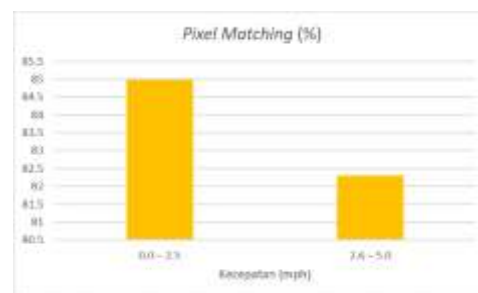
Selanjutnya nilai *Mean Squared Error (MSE)* bisa dilihat pada Gambar 21 yaitu untuk rata – rata akurasi peta yang lebih baik

didapatkan dari pengujian dengan kecepatan 0.0 – 2 dengan 8629.3.



Gambar 21 Histogram Rata – rata Hasil *MSE* Pengujian Kecepatan *Quadcopter*

Rata– rata nilai persentase bisa dilihat pada Gambar 22 rata – rata akurasi peta yang lebih baik didapatkan dari pengujian dengan kecepatan 0.0 – 2.5 dengan persentase 84.98%.

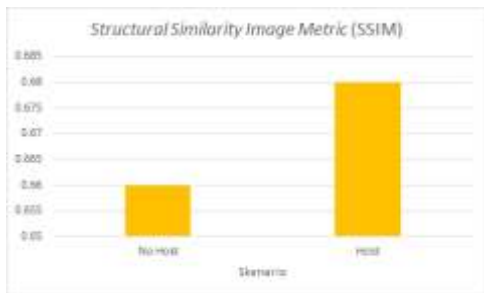


Gambar 22 Histogram Rata – rata Hasil *Pixel Matching* Pengujian Kecepatan *Quadcopter*

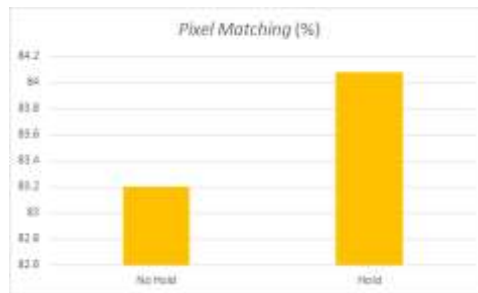
4.6. Pengaruh Waktu Pemetaan Terhadap Akurasi Peta

Analisis ini dilakukan untuk mengetahui pengaruh waktu pemetaan terhadap akurasi peta yang dihasilkan dari proses pemetaan. Analisis dilakukan berdasarkan nilai akurasi peta berdasarkan hasil analisis akurasi peta yang telah dilakukan dengan metode *Structural Similarity Image Metric (SSIM)*, *Mean Squared Error (MSE)*, dan *Pixel Matching*.

Bisa dilihat pada Gambar 23, skenario *hold* menghasilkan peta dengan akurasi yang lebih baik dari skenario *no hold*. Dilihat dari nilai *Structural Similarity Image Metric (SSIM)* pada peta hasil pengujian dengan skenario *hold* yang menghasilkan rata – rata nilai 0.68

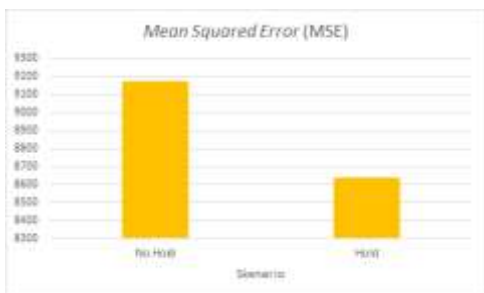


Gambar 23 Histogram Rata – rata Hasil SSIM Pengujian Waktu Pemetaan



Gambar 25 Histogram Rata – rata Hasil Pixel Matching Pengujian Waktu Pemetaan

Selanjutnya nilai *Mean Squared Error* (MSE) bisa dilihat pada Gambar 24 yaitu Peta hasil pengujian dengan skenario *hold* memiliki rata – rata nilai *Mean Squared Error* (MSE) yang lebih kecil yaitu 8641.3. Jadi rata – rata akurasi peta yang lebih baik didapatkan dari pengujian dengan skenario *hold*.



Gambar 24 Histogram Rata – rata Hasil MSE Pengujian Waktu Pemetaan

Rata– rata nilai persentase bisa dilihat pada Gambar 25 yaitu untuk peta hasil pengujian dengan skenario *no hold* berada di angka 83.20%. Untuk peta hasil pengujian dengan skenario *hold* menghasilkan rata – rata 84.08%. Peta hasil pengujian dengan skenario *hold* memiliki rata – rata persentase *Pixel Matching* yang lebih besar daripada peta hasil pengujian dengan skenario *no hold*. Jadi rata – rata akurasi peta yang lebih baik didapatkan dari pengujian dengan skenario *hold*.

5. KESIMPULAN

Berdasarkan hasil implementasi, pengujian, dan analisis yang telah dilakukan, dapat disimpulkan bahwa ukuran grid cell dan jumlah multi map resolution level berpengaruh pada peta hasil pemetaan seperti terdapat bagian yang tidak terpetakan sepenuhnya dan bagian yang tidak terpetakan secara akurat. Akurasi peta terbaik didapatkan dengan pengaturan ukuran grid cell 0.1 dan jumlah multi map resolution level 7. Pengaturan ini mendapatkan nilai akurasi tertinggi dengan rata – rata nilai Structural Similarity Image Metric (SSIM) 0.71, rata – rata nilai Mean Squared Error (MSE) 8275.4, dan rata – rata nilai persentase Pixel Matching sebesar 85.31%.

Kecepatan quadcopter berpengaruh pada peta hasil pemetaan seperti terdapat peta yang terdistorsi. Akurasi peta terbaik didapatkan dengan kecepatan quadcopter antara 0.0 – 2.5 mph. Kecepatan ini menghasilkan peta dengan nilai akurasi tertinggi yaitu nilai Structural Similarity Image Metric (SSIM) 0.69, rata – rata nilai Mean Squared Error (MSE) 8269.3, dan rata – rata nilai persentase Pixel Matching sebesar 84.98%.

Waktu pemetaan berpengaruh pada peta hasil pemetaan seperti terdapat bagian peta yang belum terpetakan sepenuhnya. Akurasi peta terbaik didapatkan dengan skenario hold. Skenario ini menghasilkan peta dengan nilai akurasi yang lebih baik dari skenario no hold yaitu rata – rata nilai Structural Similarity Image Metric (SSIM) 0.68, rata – rata nilai Mean Squared Error (MSE) 8641.3, dan rata – rata nilai persentase Pixel Matching sebesar 84.08%.

Meskipun penelitian ini berhasil dilakukan namun tetap ada beberapa bagian yang dapat diperbaiki ke depannya yaitu dapat menambahkan lagi parameter pengujian seperti ukuran grid cell dan jumlah multi map resolution

dengan beberapa variasi pengujian berbeda. Dapat juga ditambahkan pengaturan kecepatan *quadcopter* agar memiliki kecepatan yang konstan pada proses pemetaan.

6. DAFTAR PUSTAKA

- Firdaus, A. (2021, September 4). 1.805 bencana Alam melanda INDONESIA pada januari-agustus 2021. Antara News. <https://www.antaranews.com/berita/2371406/1805-bencana-alam-melanda-indonesia-pada-januari-agustus-2021> [Diakses 19 September 2021]
- Gossett, S. (2021). 12 Examples of Rescue Robots. <https://builtin.com/robotics/rescue-robots> [Diakses 27 November 2021]
- Kannan Megalingam, R., Ravi Teja, C., Sreekanth, S., & Raj, A. (2018). ROS based Autonomous Indoor Navigation Simulation Using SLAM Algorithm. *International Journal of Pure and Applied Mathematics*, 118(7), 199–205.
- Murdiyanto, & Gutomo, T. (2015). Bencana Alam Banjir dan Tanah Longsor dan Upaya Masyarakat dalam Penanggulangan. *Jurnal PKS*, 14(4), 437–452.
- Nagla, S. (2020). 2D Hector SLAM of Indoor Mobile Robot using 2D Lidar. *ICPECTS 2020 - IEEE 2nd International Conference on Power, Energy, Control and Transmission Systems, Proceedings*, 18–21.
- Olalekan, A. F., Sagor, J. A., Hasan, M. H., & Oluwatobi, A. S. (2021). Comparison of Two SLAM Algorithms Provided by ROS (Robot Operating System). May, 1–5.
- Pramod Thale, S., Mangesh Prabhu, M., Vinod Thakur, P., & Kadam, P. (2020). ROS based SLAM implementation for Autonomous navigation using Turtlebot. *ITM Web of Conferences*, 32, 01011.
- Qu, P., Su, C., Wu, H., Xu, X., Gao, S., & Zhao, X. (2021). Mapping performance comparison of 2D SLAM algorithms based on different sensor combinations. *Journal of Physics: Conference Series*, 2024(1).
- Ravankar, A., Ravankar, A., Kobayashi, Y., & Emaru, T. (2018). Autonomous Mapping and Exploration with Unmanned Aerial Vehicles Using LowCost Sensors. *Proceedings*, 4(1), 44.
- Riisgaard, M. R. B. and S. (2000). SLAM for Dummies. A Tutorial Approach to Simultaneous Localization and Mapping, 80(4), 127.
- Saat, S., Abd Rashid, W. N., Tumari, M. Z. M., & Saealal, M. S. (2020). HECTORSLAM 2D MAPPING for SIMULTANEOUS LOCALIZATION and MAPPING (SLAM). *Journal of Physics: Conference Series*, 1529(4).
- Sitorus, Tulus Irpan H. (2020). Corporate Social Responsibility (CSR) Sebagai Alternatif Sumber Dana Penanggulangan Bencana Alam. *Jurnal Dialog Penanggulangan Bencana*, 3(2), 103–116
- Taheri, H., & Xia, Z. C. (2021). SLAM; definition and evolution. *Engineering Applications of Artificial Intelligence*, 97(September 2020), 104032.
- Thu, K. M., & Gavrilov, A. I. (2017). Designing and Modeling of Quadcopter Control System Using L1 Adaptive Control. *Procedia Computer Science*, 103(October 2016), 528–535.
- Valsan, A., B., P., G. H., V. D., Unnikrishnan, R. S., Reddy, P. K., & A., V. (2020). Unmanned aerial vehicle for search and rescue mission. *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)* (48184).