

Simultaneous Localization and Mapping pada Smart Automated Guided Vehicle menggunakan Iterative Closest Point berbasis K-Means Clustering

NI PUTU DEVIRA AYU MARTINI, BAMBANG SUMANTRI, BIMA SENA BAYU DEWANTARA

Politeknik Elektronika Negeri Surabaya, Indonesia
Email : niputudevira@pasca.student.pens.ac.id

Received 25 April 2022 | *Revised* 21 Mei 2022 | *Accepted* 15 Juni 2022

ABSTRAK

Automated Guided Vehicle (AGV) merupakan salah satu jenis mobile robot yang digunakan untuk mengangkut barang menuju tempat tujuan. AGV mampu bekerja pada lingkungan yang dinamis tanpa menggunakan garis pemandu. Namun sebelumnya harus mempunyai informasi yang cukup terhadap lingkungan kerjanya. Teknik ini dikenal dengan Simultaneous Localization and Mapping (SLAM) yang digunakan robot untuk menggambar peta sekaligus mengetahui posisi robot di dalam peta. Pada penelitian ini, metode yang digunakan yaitu SLAM berbasis Iterative Closest Point (ICP) dengan algoritma K-Means yang menggunakan kumpulan titik dari sensor laser range finder (LRF) untuk membangun peta lingkungan. Pemetaan SLAM menggunakan algoritma K-Means memiliki error hasil scan jarak 77,69% lebih kecil dan waktu eksekusi 0,18% lebih cepat dibandingkan dengan KD-Tree. Peta yang dihasilkan dengan algoritma K-Means pada ICP-SLAM memberikan hasil yang lebih baik & mendekati keadaan ruangan sebenarnya dibandingkan menggunakan algoritma KD-Tree.

Kata kunci: ICP-SLAM, K-Means, Laser Range Finder.

ABSTRACT

Automated Guided Vehicle (AGV) is a type of mobile robot that is used to transport goods to destination. AGV is able to work in a dynamic environment without guidelines. However, it must have sufficient information about its working environment beforehand. This technique is known as Simultaneous Localization and Mapping (SLAM) which is used by a robot to be able to draw a map as well as to determine its position on the map. In this research, the method used is SLAM based on Iterative Closest Point (ICP) with the K-Means algorithm that uses a collection of points from the Laser Range Finder (LRF) sensor to build an environmental map. SLAM using the K-Means algorithm has 77,69% smaller distance error and 0,18% faster execution time than KD-Tree. The map generated by the K-Means algorithm on an ICP-SLAM gives better results & closer to the actual state than using the KD-Tree.

Keywords: ICP-SLAM, K-Means, Laser Range Finder.

1. PENDAHULUAN

Saat ini, penggunaan sistem *mobile robot* jenis AGV (*Automated Guided Vehicle*) sudah mulai banyak digunakan di industri. Pada umumnya AGV yang digunakan masih menggunakan pemandu berupa garis, di mana memiliki kelemahan keterbatasan daerah kerjanya. Robot AGV dengan kemampuan kecerdasan (*Smart AGV*) untuk beradaptasi di lingkungan yang dinamis dan tidak menggunakan garis sebagai pemandu pergerakan masih jarang ditemui. Pada penggunaan *Smart AGV*, aspek keamanan menjadi penting karena AGV sendiri merupakan kendaraan yang dapat bergerak bebas. Ini berbeda dengan *automated conveyor* yang pergerakannya pada jalur yang telah ditetapkan dan terbatas. AGV dapat melakukan tugasnya dalam pengangkutan barang menuju tujuan apabila telah mengetahui kondisi lingkungan di sekitarnya. Oleh karena itu, robot AGV perlu melakukan pemetaan dan lokalisasi pada area yang ditentukan sehingga robot dapat bernavigasi ke titik tujuan yang diberikan (Quan & Chen, 2019) (Rufus, dkk, 2020).

Pemetaan dan lokalisasi bertujuan untuk membangun sebuah peta lingkungan dan mengetahui lokasi dari suatu area atau teknik yang sering digunakan yaitu teknik *Simultaneous Localization and Mapping* (SLAM). Penelitian mengenai SLAM sebelumnya menggunakan berbagai macam metode yaitu *Probabilistic SLAM* (An, dkk, 2021), EKF-SLAM (Wang, dkk, 2013), *Graph-SLAM* (Aldibaja, dkk, 2021), *Gmapping-SLAM* (Revanth, dkk, 2020), GP-SLAM (Ruan, dkk, 2020), ICP-SLAM (Kusumo, dkk, 2019) dan sebagainya. Penelitian tentang pemetaan dan lokalisasi menggunakan SLAM berbasis ICP telah dilakukan sebelumnya, dimana salah satu prosesnya yaitu pencarian jarak terdekat antara kumpulan titik scannya dicari menggunakan metode KD-Tree (Yu, dkk, 2021). Namun metode tersebut menyebabkan proses komputasi menjadi berat, dikarenakan karakteristik dari algoritma KD-Tree itu sendiri.

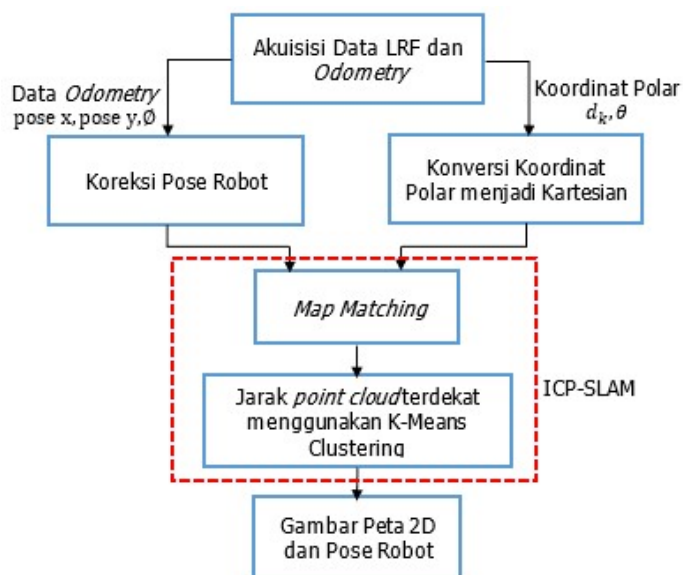
Untuk itu pada penelitian ini dilakukan perancangan dan implementasi pemetaan dan lokalisasi menggunakan teknik SLAM berbasis ICP dengan algoritma *K-Means*. Di mana algoritma *K-Means* digunakan untuk mencari jarak terdekat dari *source point cloud* menuju *reference point cloud*. Dengan sistem pemetaan ICP-SLAM menggunakan algoritma *K-Means* juga memiliki beberapa pengujian jumlah *cluster* yang mampu untuk menyesuaikan dengan kebutuhan dari segi hasil visualisasi peta yang baik atau waktu eksekusi yang cepat. Sementara itu, sistem ini disusun dengan membandingkan hasil pencocokan algoritma *K-Means* dan KD-Tree *clustering* dengan memperhatikan kinerja dan efisiensi komputasi, hasil visualisasi dan waktu eksekusi sistem pemetaan yang nantinya akan menjadi kontribusi pada penelitian ini.

2. METODOLOGI

Proses perancangan sistem ditunjukkan melalui diagram pada Gambar 1. Berdasarkan Gambar 1, proses perancangan secara umum dibagi menjadi beberapa tahapan yaitu:

1. Akuisisi data LRF dan *odometry* merupakan proses mendapatkan data dari sensor LRF berupa jarak dan sudut serta data *odometry* yang berupa posisi x , y dan orientasi robot. Data yang diterima dari sensor LRF dan *odometry* nantinya akan digunakan sebagai acuan dalam menerapkan SLAM.
2. Konversi koordinat polar menjadi kartesian merupakan proses yang berfungsi untuk mengubah data masukan yang diterima dari sensor LRF dalam format koordinat polar (d_k, θ) menjadi data dalam format koordinat kartesian (pose x , pose y, \emptyset).
3. Koreksi pose robot adalah sebuah proses yang berfungsi untuk melakukan koreksi pose robot (x, y, \emptyset) dengan melakukan estimasi posisi robot selanjutnya. Sehingga dapat mengantisipasi apabila terdapat *error* pada data sensor *odometry*.

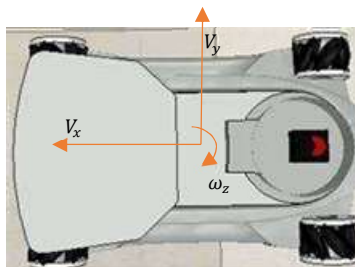
4. SLAM merupakan proses dalam mengolah data berupa jarak dan pose yang didapatkan dari sensor LRF dan *odometry*. Data dari sensor LRF dan *odometry* yang diterima oleh komputer akan digunakan untuk proses *mapping* dan *localization*. Proses *mapping*/pemetaan dilakukan oleh robot dengan cara menggabungkan beberapa data yang didapatkan dari *scanning* yang dilakukan oleh sensor LRF. Sedangkan data dari sensor *odometry* digunakan dalam menentukan pose robot pada peta.
5. ICP *map matching* merupakan sebuah proses yang berfungsi untuk menggabungkan gambar bagian-bagian dari peta menjadi satu gambar peta seutuhnya dengan menggunakan metode *map matching* dari algoritma ICP.
6. Gambar peta 2D dan pose robot adalah keluaran yang dihasilkan oleh sistem setelah melakukan koreksi pose robot dan ICP *map matching* pada proses sebelumnya.



Gambar 1. Diagram Umum Perancangan Sistem

2.1 Kinematika Gerak Robot

Pada penelitian ini, robot AGV didesain menggunakan empat roda mecanum yang memungkinkan AGV dapat bergerak secara *holonomic*, seperti yang ditunjukkan pada Gambar 2. Dalam melakukan penjelajahan, robot mecanum *wheel* dapat bergerak secara *mobile* ke segala arah baik dengan tidak atau mengubah *heading* robot dikarenakan terdapat *roller* yang terpasang pada roda mecanum dengan sudut sebesar 45 derajat (Taheri, dkk, 2015).



Gambar 2. Konfigurasi Pose Robot

Kinematika dari robot mecanum *wheel* diperlukan untuk dapat mengetahui informasi mengenai posisi serta orientasi atau arah hadap robot. Di mana seperti ditunjukkan pada Persamaan (1) digunakan untuk mengetahui posisi serta orientasi dari robot mecanum *wheel* berdasarkan kecepatan pada roda pertama hingga keempat, rumus ini disebut dengan *forward kinematic*.

$$\begin{bmatrix} V_x \\ V_y \\ \emptyset \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{I_x+I_y} & \frac{1}{I_x+I_y} & -\frac{1}{I_x+I_y} & \frac{1}{I_x+I_y} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad (1)$$

Hasil dari *forward kinematic* pada Persamaan (1) berupa kecepatan yang nantinya akan diubah menjadi jarak tempuh robot yang ditunjukkan pada Persamaan (2) dan (3) serta digunakan sebagai koreksi pose untuk sistem pemetaan.

$$pose\ x = V_x \times t \quad (2)$$

$$pose\ y = V_y \times t \quad (3)$$

Sedangkan model matematis yang akan diimplementasikan untuk memperoleh kecepatan sudut setiap roda mecanum dapat dilihat pada Persamaan *invers kinematic* (4) dengan memasukkan komponen vektor V_x , V_y dan \emptyset .

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(I_x + I_y) \\ 1 & 1 & (I_x + I_y) \\ 1 & 1 & -(I_x + I_y) \\ 1 & -1 & (I_x + I_y) \end{bmatrix} \quad (4)$$

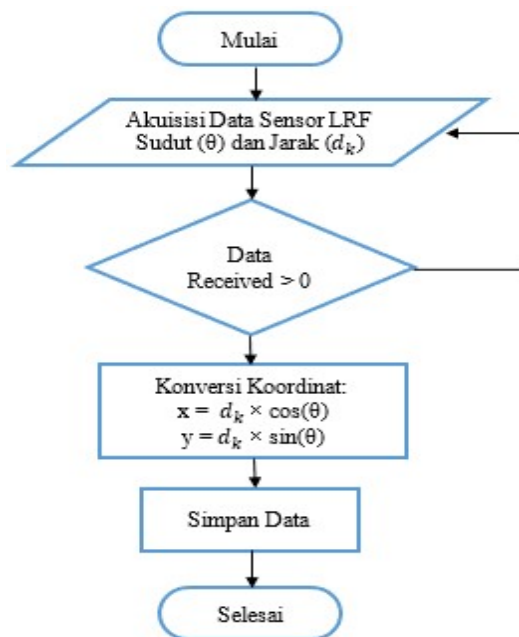
di mana:

V_x	= Longitudinal velocity
V_y	= Transversal velocity
\emptyset	= Angular velocity
r	= Jari-jari roda mecanum
$\omega_1, \omega_2, \omega_3, \omega_4$	= Wheels angular velocity
I_x	= Panjang titik tengah robot sampai titik terluar robot pada sumbu X
I_y	= Panjang titik tengah robot sampai titik terluar robot pada sumbu Y
$pose\ x$	= Posisi robot X
$pose\ y$	= Posisi robot Y

2.2 Sensor LRF dan Odometri

Proses pemetaan yang dilakukan oleh robot memerlukan data masukan berupa kondisi objek yang ada pada lingkungan di sekitar robot. Agar robot dapat memperoleh data masukan berupa kondisi objek pada lingkungan di sekitar robot, terdapat berbagai perangkat yang dapat digunakan untuk mengumpulkan data lingkungan seperti *Laser Range Finder* (Her, dkk, 2012), *Infrared Sensor* (Eom & Jeon, 2014), Kamera (Hui & Shiwei, 2013) & Sonar (Fallon, dkk, 2013). Berbagai perangkat tersebut memiliki keunggulan dan kelemahannya masing-masing yang dapat digunakan sesuai kebutuhan. Pada penelitian ini menggunakan sensor *Laser Range Finder* (LRF) yang dapat mengetahui kondisi lingkungan di tempat dengan maupun tanpa cahaya. Pada umumnya sensor tersebut bekerja dengan cara menembakkan cahaya laser secara simultan ke segala arah dengan gerakan berputar. Sensor LRF dapat mengetahui jarak dari objek yang terdapat di sekitar sensor dengan menghitung waktu yang diperlukan oleh cahaya laser mulai dari saat ditembakkan hingga cahaya laser memantul

kembali menuju *receiver* dari cahaya laser. Pada riset ini penerapan SLAM dilakukan dengan simulator V-Rep.



Gambar 3. Diagram Akuisisi Data Sensor LRF

Sensor LRF yang digunakan pada simulator ini adalah sensor LRF dengan tipe Hokuyo URG-04LX. Sensor LRF Hokuyo-04LX pada V-Rep diatur memiliki radius 270 derajat, dengan *angle step* sebesar 0,25 derajat, dan jumlah data jarak yang diterima sebanyak 1080 titik. Diagram akses sensor LRF ditunjukkan oleh Gambar 3. Data yang dibutuhkan untuk pemetaan SLAM, bukan hanya dari data Sensor LRF saja, namun memerlukan data berupa pose robot (posisi dan orientasi robot) juga. Untuk memperoleh data dari pose robot di sini menggunakan koordinat global dalam simulator V-Rep. Sebelum masuk ke pembuatan peta SLAM berbasis ICP, perlu mengolah data dari sensor LRF dan *odometry* terlebih dahulu sebagai acuan dalam menerapkan SLAM. Ketika robot melakukan pergerakan pergeseran dan berotasi, maka rumus penggabungan kedua data tersebut ditunjukkan pada Persamaan (5) dan (6).

$$\begin{bmatrix} x \text{ trans} \\ y \text{ trans} \end{bmatrix} = \begin{bmatrix} d_k \times \cos(\theta) \\ d_k \times \sin(\theta) \end{bmatrix} + \begin{bmatrix} \text{pose } x \\ \text{pose } y \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} x \text{ rotasi} \\ y \text{ rotasi} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \text{ trans} - \text{pose } x \\ y \text{ trans} - \text{pose } y \end{bmatrix} + \begin{bmatrix} \text{pose } x \\ \text{pose } y \end{bmatrix} \quad (6)$$

di mana:

$x \text{ trans}$	= Hasil translasi sumbu X
$y \text{ trans}$	= Hasil translasi sumbu Y
d_k	= Jarak tembakan dari sensor LRF
θ	= Sudut yang terbentuk dari tembakan sensor LRF
$\text{pose } x$	= Posisi robot X
$\text{pose } y$	= Posisi robot Y
$x \text{ rotasi}$	= Hasil rotasi sumbu X
$y \text{ rotasi}$	= Hasil rotasi sumbu Y

2.3. ICP-SLAM berbasis *K-Means Clustering*

Data gabungan dari sensor LRF dan *odometry* yang sebelumnya telah dihitung menggunakan Persamaan (5) dan (6) sudah bisa disebut dengan hasil dari pemetaan SLAM. Namun pemetaan menggunakan cara tersebut saja tidak cukup baik hasilnya. Maka dari itu riset ini membahas tentang pembuatan peta SLAM berbasis *ICP-Map matching*. Di mana *map matching* merupakan proses yang digunakan untuk membangun sebuah peta dengan menggunakan data berupa rumpun titik/*point cloud* yang didapatkan dari hasil setiap pemindaian yang dilakukan oleh sensor LRF. Berikut ini adalah algoritma *ICP-Map matching* yang digunakan pada penelitian ini:

Algoritma ICP Map Matching

Masukan: Jumlah iterasi N_{iter} , scan sebelumnya $S_{ref} : \{X_{ref}^i, Y_{ref}^i\}$ dan baru $S_{new} : \{X_{new}^j, Y_{new}^j\}$

Keluaran: Gambar peta S_{result} berdasarkan hasil map matching antara S_{ref} dan S_{new}

Deskripsi:

- 1: START
 - 2: FOR $i < N_{iter}$
 - 3: FOR $j < N_{new}$
 - 4: HITUNG jarak antar titik dengan euclidean distance

$$d(j) = \sqrt{d_x^2 + d_y^2}$$
 - 5: ENDFOR
 - 6: HITUNG nilai rata-rata dari S_{ref} dan S_{new}

$$\bar{S}_{ref} = \frac{1}{N_{ref}} \sum_{j=1}^{N_{ref}} S_{ref}^j \quad \bar{S}_{new} = \frac{1}{N_{new}} \sum_{j=1}^{N_{new}} S_{new}^j$$
 - 7: HITUNG matriks kovarian

$$H = \sum_j (S_{ref}^j - \bar{S}_{ref})(S_{new}^j - \bar{S}_{new})^T$$
 - 8: CARI matriks U dan V dengan SVD

$$[U, \Sigma, V] = SVD(H)$$
 - 9: HITUNG matriks rotasi dan translasi

$$R = VU^T \quad T = -R \bar{S}_{ref} + \bar{S}_{new}$$
 - 10: $S_{result} = (R S_{new}) + T$
 - 11: IF $(d(j) < d_{min})$ Keluar dari for loop
 - 12: ENDFOR
 - 13: CETAK S_{result}
 - 14: END
-

Langkah awal yang dilakukan pada saat menerapkan metode ICP dalam membangun peta yaitu dengan mencari jarak terdekat dari dua *point cloud*. *Point cloud* tersebut berupa rumpun titik yang didapatkan dari hasil *scan* sebelumnya dengan hasil *scan* yang baru. Di mana dalam ICP, data *scan* sebelumnya dianggap sebagai *reference point cloud* dan data *scan* yang baru disebut sebagai *source point cloud*. Setiap titik pada *source point cloud* diproyeksikan pada *reference point cloud* untuk menghitung jarak dari setiap titik. Untuk dapat mencari jarak dari dua titik yang masing-masing merupakan anggota dari *source point cloud* dan *reference point cloud* maka salah satu cara yang digunakan yaitu dengan mencari *euclidean distance*. Persamaan dari *euclidean distance* seperti pada Persamaan (7) berikut.

$$d(p, q) = \sqrt{\sum_{i=1}^n (S_{ref} - S_{new})^2} \quad (7)$$

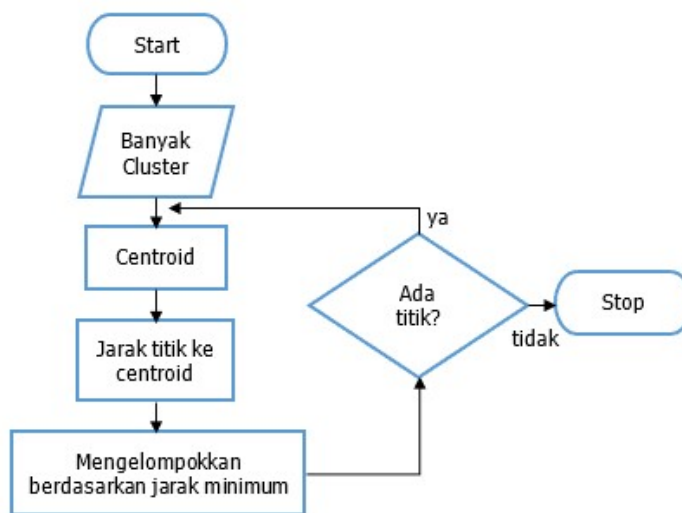
di mana:

$$d(p, q) \quad = \text{Jarak titik } reference \text{ dan } source$$

$$\frac{S_{ref}}{S_{new}}$$

= Titik *reference*

= Titik *source*



Gambar 4. Proses K-Means Clustering

Kemudian memilih jarak terdekat dari setiap *source* menuju *reference point cloud* menggunakan algoritma *K-Means*. Di mana proses dari pencarian jarak terdekat dengan *K-Means Clustering* dapat ditinjau pada Gambar 4. Algoritma *K-Means* adalah suatu metode penganalisaan data atau metode *data mining* yang melakukan proses pemodelan tanpa supervisi (*unsupervised*) dan merupakan salah satu metode yang melakukan pengelompokan data dengan sistem partisi (Kim, 2020). Dipilihnya algoritma ini untuk mencari jarak terdekat dari *source* menuju *reference point cloud* karena metode yang umum dipakai untuk mengurutkan data jarak terdekat, seperti metode *bubble sort*, *selection sort* dan sejenisnya memiliki performa yang sangat lambat, sedangkan dalam menjalankan sistem ICP-SLAM ini salah satu karakteristiknya adalah waktu *sampling* yang cepat.

3. HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan hasil pengujian dari beberapa skenario pengujian yang telah dilakukan. Pengujian sistem dilakukan dengan menjalankan program pada sebuah komputer dengan spesifikasi tertentu. Spesifikasi komputer, sistem operasi, dan perangkat pemrograman untuk proses pengujian pada riset ini ditunjukkan pada Tabel 1.

Tabel 1. Spesifikasi Perangkat Pengujian

No.	Deskripsi	Spesifikasi
1.	Sistem Operasi	Ubuntu 18.04 LTS
2.	V-Rep	V-Rep 3.5.0
3.	Sensor LRF	Hokuyo URG 04LX
4.	Library	V-Rep Remote API, OpenCV 3.2
5.	Processor	Intel Core I5-5200U
6.	Graphic Card	Dual VGA NVIDIA GT 930M 2GB
7.	RAM	4GB DDR3
8.	ROS	ROS Melodic

Pengujian pada riset ini dilakukan untuk mengetahui apakah sistem yang telah dirancang dapat bekerja dengan baik dan sesuai dengan yang diinginkan atau tidak. Pada pengujian simulasi, V-Rep menyediakan sensor LRF dengan tipe Hokuyo URG 04LX. Hokuyo URG 04LX memiliki spesifikasi berupa sudut maksimal yang dapat dijangkau sebesar 270 derajat dengan *angle step* 0,25 derajat. Sehingga, data jarak yang dihasilkan oleh sensor Hokuyo URG 04LX sebanyak 1080 data.

3.1 Lingkungan Pengujian

Tahap pengujian ini bertujuan untuk menguji SLAM berbasis ICP dengan algoritma *K-Means* dalam membangun sebuah peta. Untuk melakukan pengujian algoritma ICP dilakukan pada *simulator* V-Rep sehingga dapat mengetahui apakah algoritma ICP dapat digunakan dalam membangun sebuah peta. ICP berfungsi untuk menggabungkan beberapa gambar yang berupa gambar potongan-potongan dari ruangan menjadi sebuah gambar ruangan yang utuh. Sehingga untuk melakukan pengujian algoritma ICP, sensor LRF perlu diletakkan pada sebuah *mobile robot* agar dapat mengelilingi atau menjelajahi seluruh bagian ruangan yang digunakan pada *simulator* V-Rep.



Gambar 5. (a) Lingkungan Pengujian 1 (b) Lingkungan Pengujian 2

Robot yang digunakan dalam pengujian ini yaitu robot mecanum empat roda. Robot mecanum-*wheel* yang telah dilengkapi dengan sensor LRF berjalan mengelilingi ruangan sesuai dengan jalur yang telah ditetapkan pada simulator V-Rep. Ruangan yang digunakan dalam pengujian memiliki dimensi 6,4 x 9,1 meter. Lingkungan Pengujian yang digunakan oleh robot sebagai acuan dalam mengelilingi ruangan ditunjukkan pada Gambar 5. Di mana gambar (a) merupakan lingkungan yang berisi tiang kecil untuk mendapatkan nilai *error* sistem pemetaan SLAM berbasis ICP ini. Sementara itu gambar (b) memiliki bentuk lingkungan seperti ruangan yang berisi barang-barang berupa kursi, meja, lemari, dinding dan lain sebagainya.

3.2 Pengujian ICP-SLAM berbasis *K-Means Clustering*

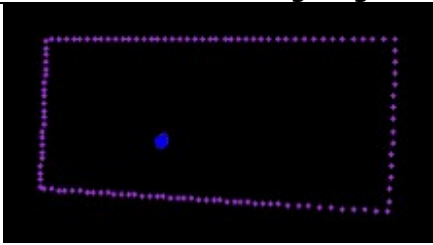
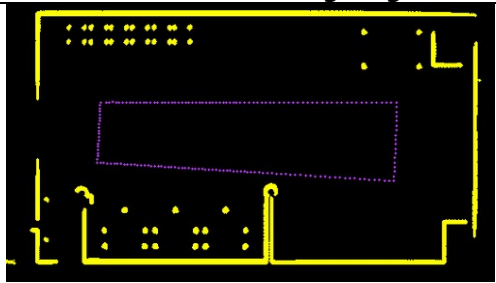
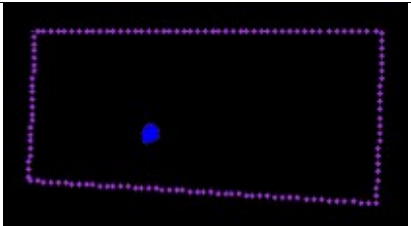
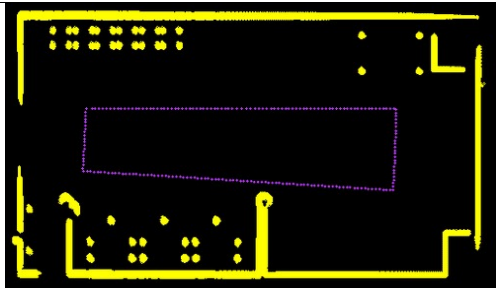
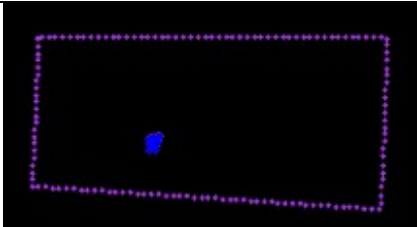
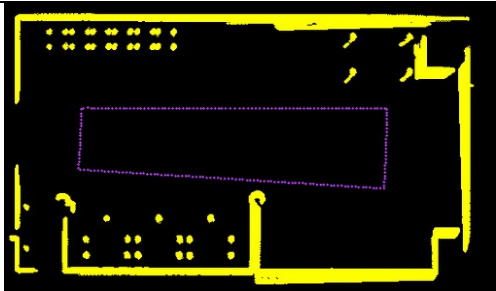
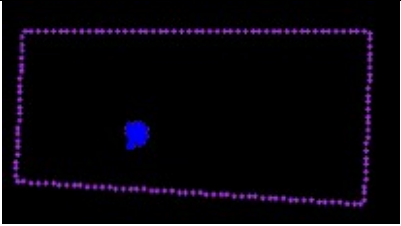
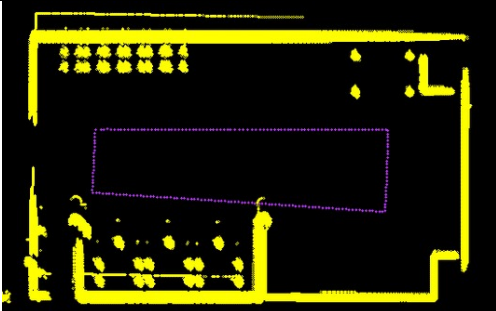
Terdapat empat percobaan yang dilakukan pada sistem pemetaan dengan metode ICP-SLAM dengan algoritma *K-Means*. Pada riset ini, pembuatan peta menggunakan metode ICP diberikan iterasi yang sama yaitu sebesar 200 iterasi dengan peng-*cluster*-an data sebanyak 3, 4, 5 dan 8 data per-*cluster* untuk algoritma *K-Means* yang digunakan.

3.2.1 Pengujian Visualisasi Pemetaan ICP-SLAM berbasis *K-Means Clustering*

Hasil berupa gambar yang didapatkan dari keluaran algoritma ICP yang ditunjukkan pada Tabel 2, yang didapatkan dari penggabungan seluruh potongan gambar ruangan yang didapatkan

oleh sensor LRF selama robot dalam perjalanan mengelilingi ruangan. Pada peta visualisasi lingkungan satu terlihat bahwa tiang yang menjadi objek dapat tergambar dengan baik pada pengujian tiga data per-*cluster*. Dan semakin besar nilai pengujian data per-*cluster*, maka visualisasi petanya semakin tidak rapat. Selanjutnya pada visualisasi peta lingkungan dua terlihat adanya kumpulan titik yang tersebar di pojok peta yang menunjukkan objek dari kursi yang ada pada area pengujian dan objek-objek lain di sekitarnya. Empat macam peta didapatkan dari pemilihan jumlah data *cluster* yang berbeda.

Tabel 2. Visualisasi Pemetaan SLAM-ICP menggunakan Algoritma *K-Means*

No.	Hasil Scan ICP-SLAM dengan <i>K-Means</i>	
	Visualisasi Gambar Lingkungan-1	Visualisasi Gambar Lingkungan-2
1.	 <p>3 Data</p>	
2.	 <p>4 Data</p>	
3.	 <p>5 Data</p>	
4.	 <p>8 Data</p>	

Pada pengujian ini, jumlah *cluster* yang dijadikan uji coba pada algoritma ICP yaitu 3, 4, 5 dan 8 *cluster*. Jumlah *cluster* yang digunakan pada pengujian ditentukan secara acak. Untuk jumlah *cluster* di atas delapan tidak disarankan karena dapat membuat hasil gambar pemetaan menjadi tidak bagus, walaupun proses komputasinya cenderung lebih cepat daripada *cluster* di bawahnya. Hasil ini juga dibuktikan dengan pengukuran nilai *error* yang dihasilkan oleh algoritma ICP pada data yang disajikan pada Tabel 3. Di mana nilai *error* rata-rata *cluster* 3, 4, 5 dan 8 yaitu 0,01828895; 0,03230042; 0,03935900 dan 0,07722872 meter.

Tabel 3. Nilai *Error* ICP menggunakan *Algoritma K-Means*

<i>Cluster</i>	Error Rata-Rata (m)
3 <i>Cluster</i>	0,01828895
4 <i>Cluster</i>	0,03230042
5 <i>Cluster</i>	0,03935900
8 <i>Cluster</i>	0,07722872

Dari data berupa nilai *error* yang dimiliki oleh setiap titik *scan*, didapatkan bahwa nilai rata-rata *error* terendah dimiliki oleh jumlah data per-*cluster* sebanyak tiga data. Pada data tersebut menunjukkan nilai *error* pemetaan berbasis ICP menggunakan algoritma *K-Means* dengan beberapa pengujian data per-*cluster*. Terlihat bahwa semakin sedikit jumlah data per-*cluster*, maka semakin bagus juga hasil pemetaannya, dibuktikan juga kecilnya nilai *error* rata-rata yang terukur.

3.2.2 Pengujian Waktu Eksekusi ICP-SLAM berbasis *K-Means Clustering*

Selain itu waktu eksekusi pada pembuatan peta SLAM menggunakan *K-Means Clustering* untuk menghitung jarak terdekat antar titik pada setiap *point-cloud* dapat ditinjau pada Tabel 4. Di mana waktu eksekusi ini diambil saat setiap sistem selesai bekerja membuat peta dengan metode ICP di setiap *cluster*-nya. Pengukuran ICP-SLAM menggunakan algoritma *K-Means* memiliki waktu eksekusi bervariasi dari 0,421212; 0,250967; 0,174257 & 0,077581 detik untuk 3, 4, 5 & 8 data per-*cluster*.

Tabel 4. Waktu Eksekusi ICP-SLAM berbasis *K-Means Clustering*

No.	Data	Waktu Eksekusi (detik)
1.	3 data per <i>cluster</i>	0,421212
2.	4 data per <i>cluster</i>	0,250967
3.	5 data per <i>cluster</i>	0,174257
4.	8 data per <i>cluster</i>	0,077581

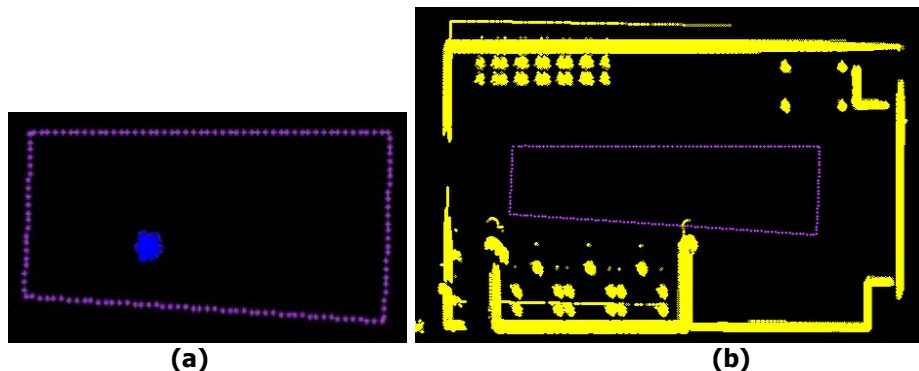
Dari pengukuran ini terlihat ketika pembuatan peta SLAM menggunakan algoritma *K-Means* bahwa semakin besar jumlah data per-*cluster* maka semakin cepat pemrosesan datanya, begitu juga sebaliknya.

3.3. Pengujian ICP-SLAM berbasis KD-Tree

Untuk melihat seberapa baik hasil dari penerapan algoritma *K-Means* dalam pembuatan peta berbasis ICP *map matching*. Maka perlu membandingkan hasil visualisasi maupun hasil *error* pemetaan dengan algoritma lain yang sudah pernah dicoba atau diteliti yaitu menggunakan algoritma KD-Tree. KD-Tree umumnya diaplikasikan dalam pencarian yang memiliki banyak kunci pencarian seperti *range search* dan *nearest neighbor search*.

3.3.1 Pengujian Visualisasi Pemetaan ICP-SLAM berbasis KD-Tree

Terdapat satu pengujian yang dilakukan pada sistem pemetaan dengan metode ICP-SLAM dengan algoritma KD-Tree. Di mana diberikan iterasi yang sama dengan menggunakan *K-Means Clustering* yaitu 200 iterasi.

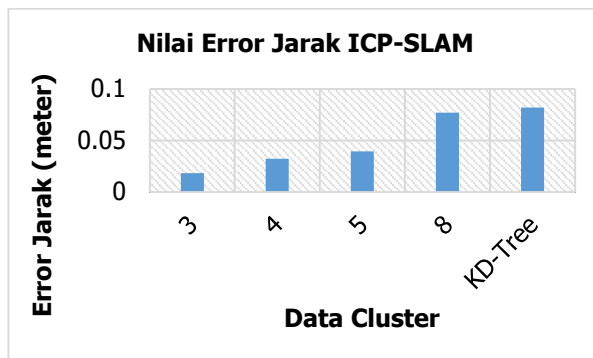


Gambar 6. Visualisasi Pemetaan ICP-SLAM menggunakan KD-Tree (a) Lingkungan Pengujian-1 (b) Lingkungan Pengujian-2

Hasil visualisasi pemetaan ICP-SLAM menggunakan algoritma KD-Tree dapat ditinjau pada Gambar 6 dengan peta lingkungan-1 adalah objek lingkungan berupa tiang, sedangkan peta lingkungan-2 memiliki bentuk lingkungan berupa ruangan yang terdapat dinding, kursi, tembok dan sebagainya. Dapat dilihat bahwa visualisasi peta yang dihasilkan oleh ICP-SLAM dengan algoritma KD-Tree dapat memetakan lingkungan seperti dinding, kaki kursi, kaki meja dan lingkungannya. Namun terlihat tidak cukup rapat seperti yang dihasilkan saat menggunakan algoritma *K-Means Clustering*. Hasil tersebut juga tercermin pada pengukuran nilai *error* rata-rata jarak yang memiliki nilai *error* rata-rata sebesar 0,081994725 meter. Nilai ini lebih besar daripada nilai *error* rata-rata saat menggunakan algoritma *K-Means*.

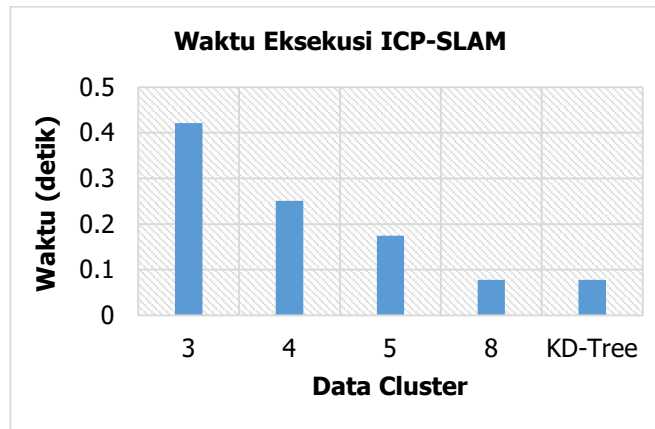
3.3.2 Pengujian Waktu Eksekusi ICP-SLAM berbasis KD-Tree

Selain itu waktu eksekusi pembuatan peta SLAM menggunakan algoritma KD-Tree diambil saat setiap sistem selesai bekerja membuat peta. Pengukuran ICP-SLAM menggunakan algoritma KD-Tree hanya memiliki satu data dengan jumlah iterasi 200 yaitu memiliki waktu eksekusi sebesar 0,077719 detik. Waktu eksekusi peta menggunakan algoritma KD-Tree ini lebih lama daripada saat menggunakan *K-Means Clustering* dengan *cluster* berapapun yang telah dicoba pada pengujian sebelumnya.



Gambar 7. Grafik *Error* Jarak ICP-SLAM

Untuk membandingkan seberapa baik hasil pemetaan ICP-SLAM berbasis *K-Means Clustering* dengan KD-Tree maka dibuatlah diagram seperti ditunjukkan pada Gambar 7. Dimana nilai *error* jarak *scan* lidar paling kecil didapat saat menggunakan algoritma *K-Means* dengan pengujian tiga data per-*cluster* dan nilai *error* hasil *scan* jarak pemetaan paling besar didapat saat menggunakan algoritma KD-Tree.



Gambar 8. Grafik Waktu Eksekusi ICP-SLAM

Begitu juga dengan waktu eksekusi tercepat didapat saat melakukan sistem pemetaan ICP-SLAM berbasis *K-Means Clustering* dengan pengujian delapan data per-*cluster* seperti yang ditunjukkan pada Gambar 8. Hal ini dikarenakan saat menggunakan algoritma *K-Means* tidak ada proses *update centroid* atau titik pusat *cluster*, artinya *centroid point cloud* bernilai tetap dalam setiap *scan reference*. Karena data *output* jarak dari lidar sudah pasti beraturan danurut dari 0 – 360 titik, tidak menyebar seperti data *input* yang diterapkan menggunakan algoritma *clustering* pada umumnya. Selain itu perhitungan nilai *centroid* cukup dilakukan dengan perhitungan nilai rata-rata data yang ada dalam *cluster* tersebut. Tidak perlu menghitung satu-persatu jarak setiap data dan memperoleh jarak terdekatnya. Pemilihan metode ini cukup tepat dan cepat diterapkan pada sistem pemetaan SLAM berbasis ICP ini karena *output scan* dari lidar berurutan dan sebaliknya jika menggunakan algoritma KD-Tree, walaupun waktu eksekusinya tidak jauh berbeda tetapi algoritma KD-Tree memiliki tingkat kesulitan yang lebih daripada menggunakan algoritma *K-Means* karena dalam KD-Tree terdapat penyimpanan data *array* yang cukup banyak dan pencarian jarak dalam metode ini adalah pendekatan dan bisa saja bukan jarak terdekat pertama yang digunakan, namun jarak terdekat kedua atau ketiga. Ini akan mempengaruhi perhitungan dalam proses ICP selanjutnya.

4. KESIMPULAN

Kami telah mengembangkan teknik pencocokan hasil *point-cloud* di lingkungan sekitar robot pada metode ICP-SLAM menggunakan *K-Means Clustering*. Kami menguji dan membandingkan hasil pencocokan *K-Means* dan KD-Tree *Clustering* dengan memperhatikan hasil visualisasi pemetaannya. Dari hasil percobaan didapatkan bahwa visualisasi pemetaan dengan *K-Means* memiliki *error* sebesar 0,01828895 meter. Hasil tersebut lebih baik 77,69% dibandingkan dengan pengujian algoritma KD-Tree yang memiliki *error* rata-rata sebesar 0,081994725 meter. Sementara itu, pembuatan ICP-SLAM menggunakan *K-Means clustering* dengan delapan data per-*cluster* memiliki waktu eksekusi *scan point-cloud* tercepat yaitu 0,077581 detik, di mana hasil tersebut 0,18% lebih cepat daripada pengujian menggunakan algoritma KD-Tree yang memiliki waktu eksekusi 0,077719 detik. Dari beberapa pengujian

pemetaan pada algoritma *K-Means*, jumlah data per-*cluster* dapat diatur sesuai kebutuhan dari segi waktu yang cepat atau hasil visualisasi pemetaan yang baik. Di mana semakin kecil jumlah data per-*cluster* maka semakin baik hasil visualisasi pemetaannya namun semakin lama proses komputasinya. Hasil pemetaan ICP-SLAM menggunakan algoritma *K-Means* secara keseluruhan lebih baik daripada saat menggunakan algoritma KD-Tree. Selanjutnya hasil dari pembuatan peta yang baik nantinya dapat digunakan untuk proses sistem navigasi pada *Automated Guided Vehicle*.

DAFTAR RUJUKAN

- Aldibaja, M., Yanase, R., Sukanuma, N., Furuya, T., & Oko, A. (2021). LIDAR Graph SLAM based Autonomous Vehicle Maps using XY and Yaw Dead-Reckoning Measurements. *IEEE International Conference on Mechatronics and Automation (ICMA)*, (pp. 1119-1125).
- An, J., Mou, H., Lu, R., & Zhou, L. (2021). A Study on SLAM Based on Probabilistic Motion Model of Mobile Robot. *International Conference on Robotics and Automation Sciences (ICRAS)*, (pp. 56-59).
- Eom, H. D., & Jeon, J. W. (2014). Environment map building using low-cost IR sensors and a servo motor for mobile robot. *International Symposium on Consumer Electronics*, (pp. 1-2).
- Fallon, M. F., Folkesson, J., McClelland, H., & Leonard, J. J. (2013). Relocating Underwater Features Autonomously Using Sonar-Based SLAM. *Journal of Oceanic Engineering*, *38*(3), 500-513.
- Her, K. W., Kim, D. H., & Ha, J. E. (2012). Localization of mobile robot using laser range finder and IR landmark. *International Conference on Control, Automation and Systems*, (pp. 459-461).
- Hui, C., & Shiwei, M. (2013). Visual SLAM based on EKF filtering algorithm from omnidirectional camera. *International Conference on Electronic Measurement & Instruments*, (pp. 660-663).
- Kim, H. (2020). Performance Analysis of K Means Clustering Algorithms for mMTC Systems. *International Conference on Information and Communication Technology Convergence*, (pp. 30-35).
- Kusumo, A. A., Marta, B. S., & Dewantara, B. S. (2019). 2D Mapping and Localization using Laser Range Finder for Omnidirectional Mobile Robot. *International Electronics Symposium*, (pp. 126-131).
- Quan, S., & Chen, J. (2019). AGV Localization Based on Odometry and LiDAR. *World Conference on Mechanical Engineering and Intelligent Manufacturing*, (pp. 483-486).

- Revanth, C. M., Saravanakumar, D., Jegadeeshwaran, R., & Sakthivel, G. (2020). Simultaneous Localization and Mapping of Mobile Robot using GMapping Algorithm. *International Symposium on Smart Electronic Systems*, (pp. 56-60).
- Ruan, J., Li, B., Wang, Y., & Fang, Z. (2020). GP-SLAM+: real-time 3D lidar SLAM based on improved regionalized Gaussian process map reconstruction. *International Conference on Intelligent Robots and Systems*, (pp. 5171-5178).
- Rufus, N., R Nair, U., Avula, B., Madiraju, V., & Krishna, M. (2020). SROM: Simple Real-time Odometry and Mapping using LiDAR data for Autonomous Vehicles. *Intelligent Vehicles Symposium*, (pp. 1867-1872).
- Taheri, H., Qiao, B., & Ghaeminezhad, N. (2015). Kinematic Model of a Four Mecanum Wheeled Mobile Robot. *International Journal of Computer Applications*, 113, 6-9.
- Wang, D., Liang, H., Mei, T., Zhu, H., Fu, J., & Tao, X. (908-912). Lidar Scan matching EKF-SLAM using the differential model of vehicle motion. *Intelligent Vehicles Symposium*, (p. 2013).
- Yu, J., Yu, C., Lin, C., & Wei, F. (2021). Improved Iterative Closest Point (ICP) Point Cloud Registration Algorithm based on Matching Point Pair Quadratic Filtering. *International Conference on Computer, Internet of Things and Control Engineering*, (pp. 1-5).