



Pengembangan Algoritma Pereduksi Noise Pada Point Cloud Data LiDAR Dua Dimensi Untuk Aplikasi Kendaraan Listrik Otonom Sederhana

(Algorithm of Noise Remover Development on LIDAR's Point Cloud Data 2D for autonomous electrical vehicle application)

Mas'ud Abdur Rosyid^{1*}, Yusuf Suhaimi Daulay², Denden Mohamad Arifin³,
Ardian Infantono⁴, Arief Suryadi Satyawan⁵, Ema⁶, Raden Aditya Satria
Nugraha⁷

^{1,2,5,6,7} Universitas Nurtanio Bandung /Badan Riset Dan Inovasi Nasional

E-mail: rasyidmasud.te17@student.unnur.ac.id, yusuf.te17@student.unnur.ac.id,
arief.suryadi@akane.waseda.jp, ema@unnur.ac.id, radityasn.te16@student.unnur.ac.id

^{3,4} Prodi Teknik Aeronautika Pertahanan, Akademi Angkatan Udara, Yogyakarta
E-mail: dendens8552@aau.ac.id, ardian.infantono@aau.ac.id

Abstract— *The application of 2-dimensional LiDAR (Light Detection And Ranging) technology is sometimes constrained by the presence of data anomalies or noise that affects the accuracy in detecting real objects. If it is not handled properly, it can interfere with its work operations, especially if it is applied to autonomous electric vehicles. Therefore, efforts are needed to reduce noise which is implemented in LiDAR data processing software. In this study, the development of noise reduction technology that appears in the two-dimensional LiDAR data point cloud is carried out. The concept applied is the development of a systematic LiDAR data processing algorithm. The design of this algorithm contains visualization of object detection, storage of LiDAR data point cloud as detected object information, as well as noise reduction methods on the two-dimensional LiDAR data point cloud. This algorithm is realized in software form on Raspberry Pi 4 hardware, using the Python programming language. There are six algorithms used to reduce or eliminate noise, namely Algorithm 1, Algorithm 2, Algorithm 3, Algorithm 4, Algorithm 5, Algorithm 6. The experimental results show that the six algorithms created are able to display data visualization based on a 2-dimensional mapping system that is corrected for noise. The six algorithms succeeded in selecting noise up to 100%, although approximately 80% of the data that were considered correct could not be presented. Even if only 20% of the data is correct, the object structure is still recognizable.*

Keywords— *Autonomous electric vehicles, LiDAR 2D, Noise removal*

Abstrak— *Penerapan teknologi LiDAR 2 dimensi (Light Detection And Ranging) terkadang terkendala oleh adanya anomaly data atau noise sehingga mempengaruhi keakuratan dalam mendeteksi objek yang sesungguhnya. Jika tidak diatasi dengan baik maka dapat mengganggu operasional kerjanya, terlebih lagi jika diterapkan pada kendaraan listrik otonom. Oleh sebab itu perlu upaya untuk mereduksi noise yang diimplementasikan pada software pemroses data LiDAR. Pada penelitian ini dilakukan pengembangan teknologi pereduksi noise yang muncul pada point cloud data LiDAR dua dimensi. Adapun konsep yang diterapkan adalah pengembangan algoritma pengolahan data LiDAR secara sistematis. Desain algoritma ini berisikan visualiasi dari pendeteksian objek, penyimpanan point cloud data LiDAR sebagai informasi objek yang terdeteksi, serta metode pengurangan noise pada point cloud data LiDAR dua dimensi tersebut. Algoritma ini di realisasikan dalam bentuk software pada perangkat keras Raspberry Pi 4, dengan menggunakan bahasa pemrograman Python. Terdapat enam Algoritma yang digunakan untuk mereduksi atau*

*Penulis Korespondensi (Mas'ud Abdur Rosyid)
E-mail: rasyidmasud.te17@student.unnur.ac.id

menghilangkan noise, yaitu Algoritma 1, Algoritma 2, Algoritma 3, Algoritma 4, Algoritma 5, Algoritma 6. Hasil percobaan memperlihatkan bahwa dari keenam Algoritma yang di buat mampu menampilkan visualisasi data berdasarkan sistem pemetaan 2 dimensi yang terkoreksi dari noise. Keenam Algoritma tersebut berhasil menyeleksi noise hingga 100%, meskipun kurang lebih 80% data yang dianggap benar tidak dapat disajikan. Meskipun hanya dengan 20% data benar, namun struktur objek masih dapat dikenali.

Kata Kunci— kendaraan listrik otonom, LiDAR 2D, Pereduksi noise

I. PENDAHULUAN

Untuk dapat bergerak secara otonom dengan sedikit campur tangan manusia dalam pengendaliannya, maka kendaraan listrik otonom harus memiliki fasilitas yang memungkinkan kendaraan tersebut dapat bergerak secara aman dan benar. Salah satu fasilitasnya adalah teknologi LiDAR (*Light Detection And Ranging*) dua dimensi yang dapat mendeteksi keberadaan objek sehingga kendaraan dapat merespon dalam bentuk pengereman atau manuver untuk menghindari objek tersebut. Namun demikian penerapan teknologi LiDAR terkadang terkendala oleh adanya *anomaly* data atau *noise* sehingga mempengaruhi keakuratan dalam mendeteksi objek yang sesungguhnya. Permasalahan penting yang di hadapi oleh pengembangan kendaraan listrik otomatis yang aman dan benar adalah kemampuan teknologi LiDAR (*Light Detection And Ranging*) dua Dimensi untuk mendeteksi dan mengetahui jarak kendaraan terhadap lingkungan sekitar saat beroperasi dengan akurat, ini di karenakan adanya *anomaly* yang tidak di inginkan berupa *noise* atau gangguan lain yang membuat kesalahan dalam pembacaan jarak kendaraan terhadap objek yang sebenarnya, sehingga jika tidak diatasi dengan baik maka dapat mengganggu operasional kendaraan listrik otonom. Untuk mengatasi hal tersebut, maka pada penelitian ini akan di usulkan pengembangan teknologi pereduksi *noise* pada *point cloud data* LiDAR dua dimensi. Adapun teknologi yang di terapkan adalah pengembangan algoritma pengolahan data LiDAR secara sistematis.

II. LANDASAN TEORI

A. YDLIDAR G4

YDLIDAR G4 adalah salah satu jenis dari LiDAR 2D (*light detection and ranging*), sebuah alat pemindai jarak jauh yang menggunakan properti cahaya yang terpantulkan untuk menemukan jarak dan informasi dari objek yang dituju, dan pemindaian jarak jauh ini berputar bebas 360⁰ terhadap sumbu horizontal. YDLIDAR G4 menggunakan laser scan dengan standar FDA laser safety class I Produk ini didasarkan pada prinsip pengukuran jarak triangulasi dengan menggunakan optik, kelistrikan, dan desain algoritma yang relevan, untuk mencapai pengukuran jarak dengan presisi tinggi.

Pengguna bisa mendapatkan *point cloud data* berdasarkan pemindaian jangkauan. Yang didapat dari prinsip kerja sensor LiDAR yang memancarkan pulsa laser terhadap objek yang kemudian dipantulkan kembali kepada sensor, sinar yang dipantulkan kemudian diproses oleh LiDAR. Hasil dari proses LiDAR adalah berupa *point cloud data*, yang kemudian dimanfaatkan sesuai dengan kebutuhan pengguna, melalui *communication interface* (*Port serial / USB*) [1].

B. Raspberry Pi 4

Raspberry Pi merupakan sebuah *Single Board Computer* (SBC), seukuran kartu kredit yang dapat dihubungkan dengan mouse, keyboard, dan monitor untuk difungsikan sebagai PC desktop, Raspberry Pi bisa digunakan sebagai *development board* dengan menggunakan sistem operasi berbasis Linux, yang mana dengan sistem operasi ini memungkinkan ununtut dapat

bebas bereksperimen atau membuat project yang lebih kompleks seperti *database*, *image processing*, *AI*, dan lain-lain dengan keahlian sistem operasi Linux.

Raspberry Pi menggunakan SoC berarsitektur ARM 64-bit yang dapat melakukan tugas yang lebih kompleks. Clockspeed yang dimiliki pun jauh lebih tinggi dibanding mikrokontroler, Raspberry Pi dilengkapi memori RAM berkapasitas 1 GB (Raspberry Pi 3 2B/3B/3B+). Untuk menginstal OS dan menyimpan file, Anda harus menambahkan micro SD secara terpisah[2].

C. Python 3

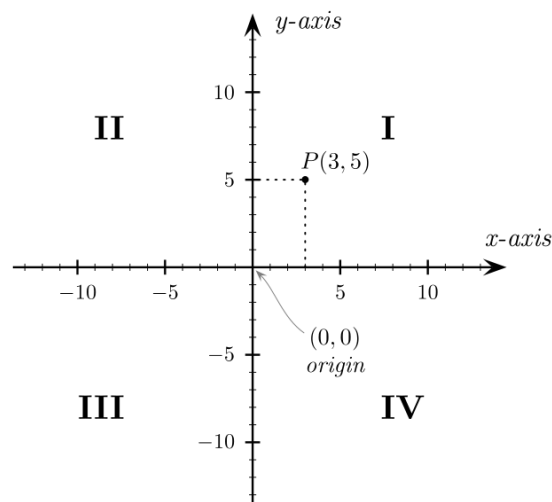
Python adalah bahasa pemrograman yang luar biasa dan kuat, mudah digunakan (mudah dibaca dan ditulis). Dengan pengoperasian melalui Raspberry Pi, memungkinkan pengguna menghubungkan proyek pengguna ke prototipe. Python memiliki struktur data yang sederhana serta *keyword* yang sedikit, serta mudah diaplikasikan karena penulisan *sintaks* yang lebih sederhana dibandingkan dengan bahasa pemrograman yang lain sehingga sangat cocok untuk pengembangan *software* dalam bentuk yang sederhana. Serta kode python dapat di-*embed* ke bahasa pemrograman lain seperti C dan Java[3].

Python memiliki *library* yang beragam, *Library* merupakan istilah untuk kode tambahan yang disediakan python dengan fungsi yang bermacam-macam, sehingga sangat mempermudah dalam membuat program dengan fungsi tertentu tanpa harus menulis kode mulai dari dasar.

D. Koordinat dan Pemetaan 2D

Sistem koordinat kartesian adalah sistem koordinat yang menetapkan setiap titik secara unik dalam bidang dengan serangkaian koordinat numerik yang merupakan jarak yang bertanda titik dari dua garis berorientasi tegak lurus tetap, dan diukur dalam satuan panjang yang sama^[4].

Koordinat juga dapat didefinisikan sebagai posisi proyeksi tegak lurus dari titik ke dua sumbu, yang dinyatakan sebagai jarak yang ditandatangani dari titik asal. Ilustrasi koordinat kartesian dapat dilihat pada Gambar 1.



Gambar 1. Sistem koordinat kartesian

Pemetaan merupakan salah satu kemampuan untuk mendapatkan posisi dari sebuah sensor dalam koordinat kartesian. Maka untuk plotting data point cloud LiDAR ini di perlukan data pada koordinat x, dan data koordinat y. Untuk dapat mendapatkan data x, dan data y, maka

digunakan persamaan 1 untuk mendapatkan data koordinat x dan koordinat y dari posisi awal lokasi sensor[4]:

$$\begin{aligned} X &= S * \cos \Theta \\ Y &= S * \sin \Theta \end{aligned} \quad (1)$$

dengan :

X = posisi sensor dalam koordinat x

Y = posisi sensor dalam koordinat y

S = jarak tempuh sensor

Θ = arah hadap pada sensor

Dan persamaan 2 untuk mendapatkan data koordinat x dan koordinat y dari posisi terkini sensor terhadap koordinat sebelumnya :

$$\begin{aligned} X_a &= S_x * \cos \Theta_x + X_{x-1} \\ Y_a &= S_x * \sin \Theta_x + Y_{x-1} \end{aligned} \quad (2)$$

dengan :

X_a = posisi terkini pada koordinat X

Y_a = posisi terkini pada koordinat Y

S_x = jarak tempuh

Θ_x = arah hadap

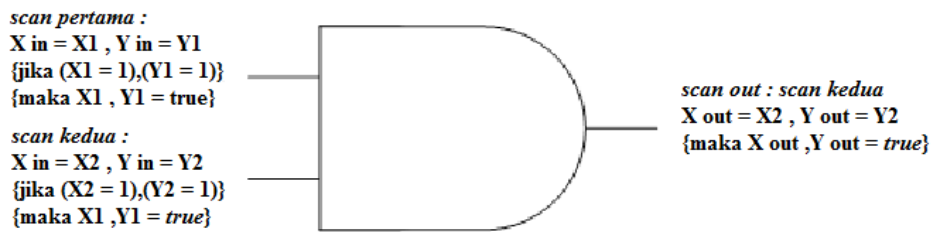
X_{x-1} = posisi koordinat x sebelumnya

Y_{x-1} = posisi koordinat Y sebelumnya

E. Metoda pengolahan data

Saat program didalam sebuah *software* terlalu banyak mengolah data dalam waktu singkat menyebabkan kesalahan acak atau variasi dalam variable terukur, ini disebut juga sebagai data *noise*. *Noise* dapat mempengaruhi kualitas visualisasi data, oleh karena itu data *noise* perlu dihilangkan atau di kurangi agar menghasilkan pemrosesan data yang optimal dan berkualitas, salah satu cara yang akan dilakukan untuk membersihkan *noise* adalah dengan menggunakan metoda pengolahan data. Metoda pengolahan data adalah suatu serangkaian algoritma untuk mereduksi atau mengurangi data yang salah atau noise yang dihasilkan oleh perangkat LiDAR.

Cara kerjanya dengan mengelompokan data dengan membagi beberapa kelompok data dalam hal ini adalah data *scan* dari pendeteksian LiDAR yang memiliki kesamaan dengan kelompok data *scan* lainnya. Mekanismenya sama seperti proses *logic gate* "AND", dimana suatu output akan bernilai (*true*) jika semua input memiliki nilai 1 atau tidak samadengan 0, nilai 1 atau tidak samadengan 0 merepresentasikan data jarak dari pembacaan pulsa laser LiDAR. Namun pada noise remover ini output yang akan ditampilkan adalah data jarak dari pembacaan LiDAR. Seperti yang terlihat Pada Gambar 2.12 diperlihatkan proses *logic gate* sistem *noise remover* yang menyerupai "AND" *gate* dengan *output* yang di tentukan adalah *scan* data keduana seperti yang terlihat pada Gambar 2.



Gambar 2. And Gate Pengolahan Data

III. MODEL YANG DIUSULKAN

A. Metoda noise removal (filtering)

Noise remover atau filter adalah suatu algoritma yang menggunakan metode pengolahan data untuk mereduksi atau mengurangi *anomaly data* yang dihasilkan oleh perangkat LiDAR. Pada Algoritma Laksman (*non-noise removal*), program yang dijalankan pada perangkat YDLIDAR G4 ini masi terdapat *noise* dalam pendeteksiannya, maka Pada penelitian ini terdapat enam Algoritma yang akan digunakan untuk mereduksi atau menghilangkan *noise* yang dikarenakan *anomaly*. Metode yang akan digunakan di antaranya:

1. Algoritma 1

Diperlihatkan pada Gambar 3 Setelah data scanning LiDAR pertama $\{X1=1, \text{jika } x \neq 0 \mid x_{\text{scan}} 1\}$, kedua $\{X2=1, \text{jika } x \neq 0 \mid x_{\text{scan}} 2\}$ dan ketiga $\{X3=1, \text{jika } x \neq 0 \mid x_{\text{scan}} 3\}$ didapatkan, selanjutnya dilakukan perbandingan dengan dua kondisi, kondisi pertama membandingkan antara data scan pertama dengan data scan kedua, dan kondisi kedua membandingkan antara data scan kedua dengan data scan ketiga. Kedua kondisi ini diproses dengan menggunakan operator “AND” $\{f(X1,X2=X1*X2)\}$, dimana $\{f(0,0)=0*0=0\}$, $\{f(0,1)=0*1=0\}$, $\{f(1,0)=1*0=0\}$, dan $\{f(1,1)=1*1=1\}$. Hal ini berarti bahwa, jika data pada scan pertama juga muncul pada scan kedua, maka data kedua tersebut dianggap benar $\{X(X1,X2)=X(1,1)=1*1=1\}$, tetapi jika data pada scan pertama tidak muncul pada data scan kedua maka berlaku kondisi kedua. Prosesnya menggunakan operator “AND” $\{f(X2,X3=X2*X3)\}$, dimana $\{f(0,0)=0*0=0\}$, $\{f(0,1)=0*1=0\}$, $\{f(1,0)=1*0=0\}$, dan $\{f(1,1)=1*1=1\}$. Hal ini berarti bahwa, jika data pada scan kedua juga muncul pada scan ketiga, maka data scan ketiga tersebut dianggap benar $\{X(X2,X3)=X(1,1)=1*1=1\}$, dan dapat dikeluarkan sebagai data luaran $\{X=X3\}$. Akan tetapi jika hal tersebut tidak terpenuhi $\{X(0,0)=0*0=0\}$, $\{X(0,1)=0*1=0\}$, $\{X(1,0)=1*0=0\}$, maka dianggap sebagai derau, sehingga tidak ada data yang dikeluarkan, atau luaran bernilai 0 $\{X=0\}$. Data final yang dihasilkan ditampilkan dan disimpan.

2. Algoritma 2

Diperlihatkan pada Gambar 4 Setelah data scanning LiDAR pertama $\{X1=1, \text{jika } x \neq 0 \mid x_{\text{scan}} 1\}$, kedua $\{X2=1, \text{jika } x \neq 0 \mid x_{\text{scan}} 2\}$ dan ketiga $\{X3=1, \text{jika } x \neq 0 \mid x_{\text{scan}} 3\}$ didapatkan, selanjutnya dilakukan perbandingan antara keduanya dengan menggunakan operator “AND” $\{f(X1,X2=X1*X2)\}$, dimana $\{f(0,0)=0*0=0\}$, $\{f(0,1)=0*1=0\}$, $\{f(1,0)=1*0=0\}$, dan $\{f(1,1)=1*1=1\}$. Hal ini berarti bahwa, jika data pada scan pertama juga muncul pada scan kedua, maka data pertama tersebut dianggap benar $\{X(X1,X2)=X(1,1)=1*1=1\}$, dan dapat dikeluarkan sebagai data luaran $\{X=X1\}$. Akan tetapi jika hal tersebut tidak terpenuhi $\{X(0,0)=0*0=0\}$, $\{X(0,1)=0*1=0\}$, $\{X(1,0)=1*0=0\}$, maka data yang dikeluarkan sebagai luaran adalah data scan ketiga $\{X=X3\}$.

3. Algoritma 3

Diperlihatkan pada Gambar 5 Setelah data scanning LiDAR pertama $\{X_1=1, \text{jika } x \neq 0 \mid x \in \text{scan } 1\}$, kedua $\{X_2=1, \text{jika } x \neq 0 \mid x \in \text{scan } 2\}$, ketiga $\{X_3=1, \text{jika } x \neq 0 \mid x \in \text{scan } 3\}$ dan keempat $\{X_4=1, \text{jika } x \neq 0 \mid x \in \text{scan } 4\}$ didapatkan (131), selanjutnya dilakukan perbandingan dengan dua kondisi, kondisi pertama membandingkan antara data scan pertama dengan data scan kedua, dan kondisi kedua membandingkan antara data scan kedua dengan data scan keempat. Kedua kondisi ini diproses dengan menggunakan operator “AND” $\{f(X_1, X_2) = X_1 * X_2\}$, dimana $\{f(0,0) = 0 * 0 = 0\}$, $\{f(0,1) = 0 * 1 = 0\}$, $\{f(1,0) = 1 * 0 = 0\}$, dan $\{f(1,1) = 1 * 1 = 1\}$. Hal ini berarti bahwa, jika data pada scan pertama juga muncul pada scan kedua, maka data scan kedua tersebut dianggap benar $\{X(X_1, X_2) = X(1,1) = 1 * 1 = 1\}$, tetapi jika data pada scan pertama tidak muncul pada data scan kedua maka berlaku kondisi kedua. Prosesnya menggunakan operator “AND” $\{f(X_2, X_4) = X_2 * X_4\}$, dimana $\{f(0,0) = 0 * 0 = 0\}$, $\{f(0,1) = 0 * 1 = 0\}$, $\{f(1,0) = 1 * 0 = 0\}$, dan $\{f(1,1) = 1 * 1 = 1\}$. Hal ini berarti bahwa, jika data pada scan kedua juga muncul pada scan keempat, maka data scan ketiga tersebut dianggap benar $\{X(X_2, X_4) = X(1,1) = 1 * 1 = 1\}$, dan dapat dikeluarkan sebagai data luaran $\{X = X_3\}$. Akan tetapi jika hal tersebut tidak terpenuhi $\{X(0,0) = 0 * 0 = 0\}$, $\{X(0,1) = 0 * 1 = 0\}$, $\{X(1,0) = 1 * 0 = 0\}$, maka data yang dikeluarkan sebagai luaran adalah data scan keempat $\{X = X_4\}$.

4. Algoritma 4

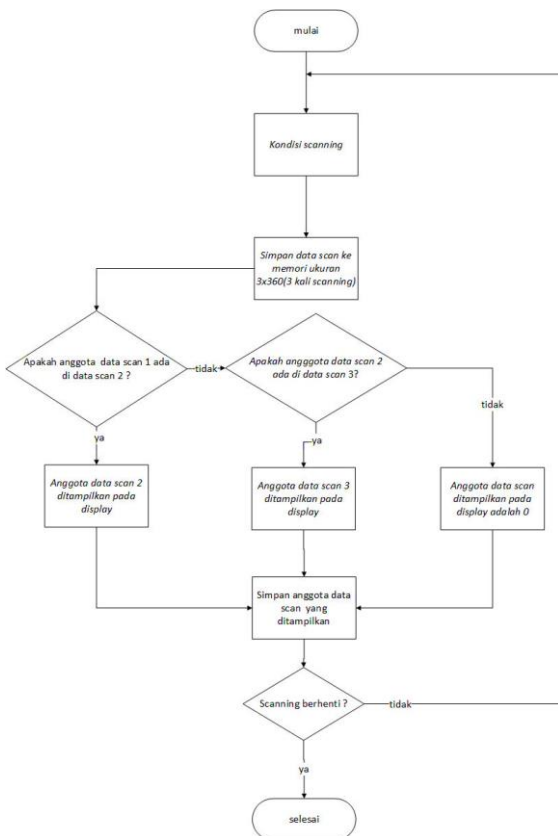
kelima dapat dilihat pada gambar 6 Setelah data scanning LiDAR pertama $\{X_1=1, \text{jika } x \neq 0 \mid x \in \text{scan } 1\}$, kedua $\{X_2=1, \text{jika } x \neq 0 \mid x \in \text{scan } 2\}$, ketiga $\{X_3=1, \text{jika } x \neq 0 \mid x \in \text{scan } 3\}$, dan keempat $\{X_4=1, \text{jika } x \neq 0 \mid x \in \text{scan } 4\}$ didapatkan, selanjutnya dilakukan perbandingan antara data scan ketiga dan keempat dengan menggunakan operator “AND” $\{f(X_3, X_4) = X_3 * X_4\}$, dimana $\{f(0,0) = 0 * 0 = 0\}$, $\{f(0,1) = 0 * 1 = 0\}$, $\{f(1,0) = 1 * 0 = 0\}$, dan $\{f(1,1) = 1 * 1 = 1\}$. Hal ini berarti bahwa, jika data pada scan ketiga juga muncul pada scan keempat, maka data ketiga tersebut dianggap benar $\{X(X_3, X_4) = X(1,1) = 1 * 1 = 1\}$, dan dapat dikeluarkan sebagai data luaran $\{X = X_3\}$. Akan tetapi jika hal tersebut tidak terpenuhi $\{X(0,0) = 0 * 0 = 0\}$, $\{X(0,1) = 0 * 1 = 0\}$, $\{X(1,0) = 1 * 0 = 0\}$, maka dianggap sebagai derau, sehingga tidak ada data yang dikeluarkan, atau luaran bernilai 0 $\{X = 0\}$.

5. Algoritma 5

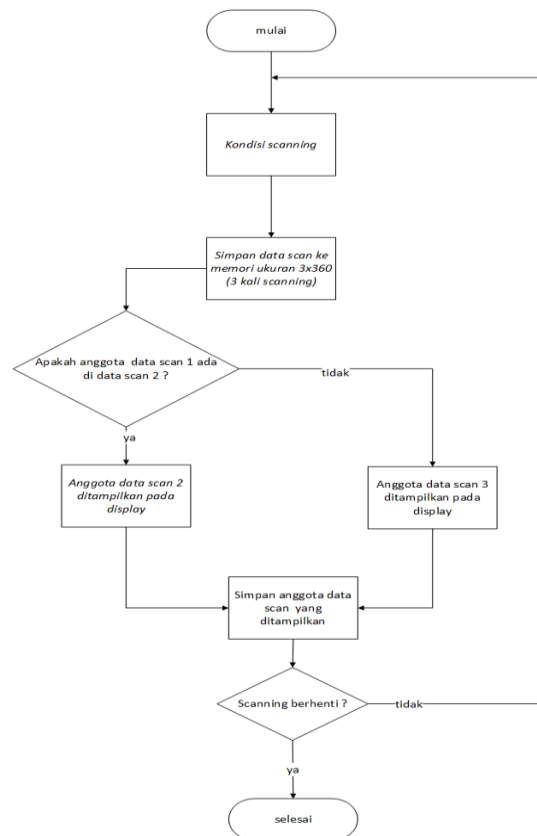
Algoritma 5 dapat dilihat pada gambar 7 Setelah data scanning LiDAR pertama $\{X_1=1, \text{jika } x \neq 0 \mid x \in \text{scan } 1\}$, kedua $\{X_2=1, \text{jika } x \neq 0 \mid x \in \text{scan } 2\}$, dan ketiga $\{X_3=1, \text{jika } x \neq 0 \mid x \in \text{scan } 3\}$ didapatkan, selanjutnya dilakukan perbandingan antara data scan kedua dan ketiga dengan menggunakan operator “AND” $\{f(X_2, X_3) = X_2 * X_3\}$, dimana $\{f(0,0) = 0 * 0 = 0\}$, $\{f(0,1) = 0 * 1 = 0\}$, $\{f(1,0) = 1 * 0 = 0\}$, dan $\{f(1,1) = 1 * 1 = 1\}$. Hal ini berarti bahwa, jika data pada scan kedua juga muncul pada scan ketiga, maka data kedua tersebut dianggap benar $\{X(X_2, X_3) = X(1,1) = 1 * 1 = 1\}$, dan dapat dikeluarkan sebagai data luaran $\{X = X_2\}$. Akan tetapi jika hal tersebut tidak terpenuhi $\{X(0,0) = 0 * 0 = 0\}$, $\{X(0,1) = 0 * 1 = 0\}$, $\{X(1,0) = 1 * 0 = 0\}$, maka dianggap sebagai derau, sehingga tidak ada data yang dikeluarkan, atau luaran bernilai 0 $\{X = 0\}$.

6. Algoritma 6

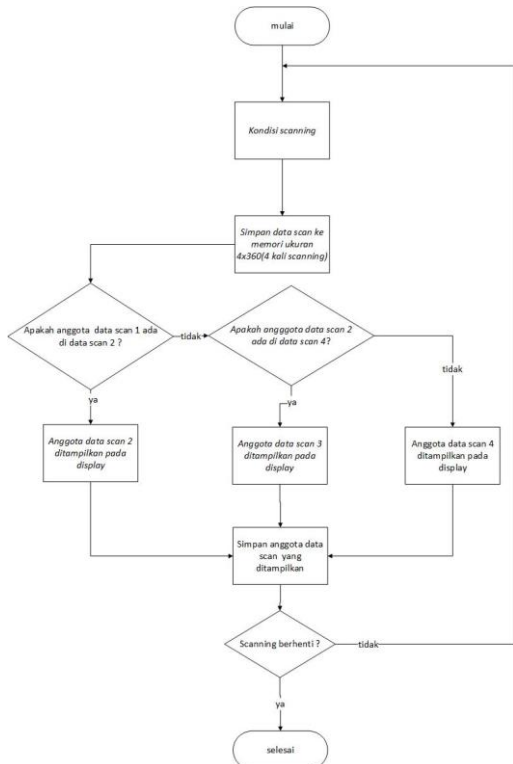
Algoritma 6 dapat dilihat pada gambar 8 Setelah data scanning LiDAR pertama $\{X_1=1, \text{jika } x \neq 0 \mid x \in \text{scan } 1\}$, kedua $\{X_2=1, \text{jika } x \neq 0 \mid x \in \text{scan } 2\}$, dan ketiga $\{X_3=1, \text{jika } x \neq 0 \mid x \in \text{scan } 3\}$ didapatkan, selanjutnya dilakukan perbandingan antara data scan pertama dan ketiga dengan menggunakan operator “AND” $\{f(X_1, X_3) = X_1 * X_3\}$, dimana $\{f(0,0) = 0 * 0 = 0\}$, $\{f(0,1) = 0 * 1 = 0\}$, $\{f(1,0) = 1 * 0 = 0\}$, dan $\{f(1,1) = 1 * 1 = 1\}$. Hal ini berarti bahwa, jika data pada scan pertama juga muncul pada scan ketiga, maka data pertama tersebut dianggap benar $\{X(X_1, X_3) = X(1,1) = 1 * 1 = 1\}$, dan dapat dikeluarkan sebagai data luaran $\{X = X_1\}$. Akan tetapi jika hal tersebut tidak terpenuhi $\{X(0,0) = 0 * 0 = 0\}$, $\{X(0,1) = 0 * 1 = 0\}$, $\{X(1,0) = 1 * 0 = 0\}$, maka dianggap sebagai derau, sehingga tidak ada data yang dikeluarkan, atau luaran bernilai 0 $\{X = 0\}$.



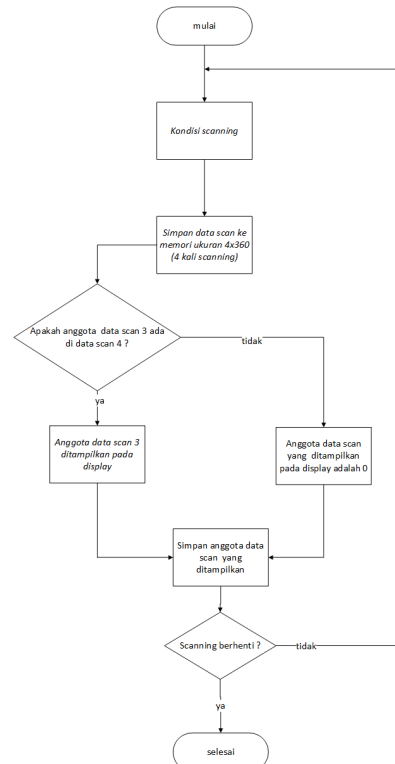
Gambar 3. Flowchart Algoritma 1



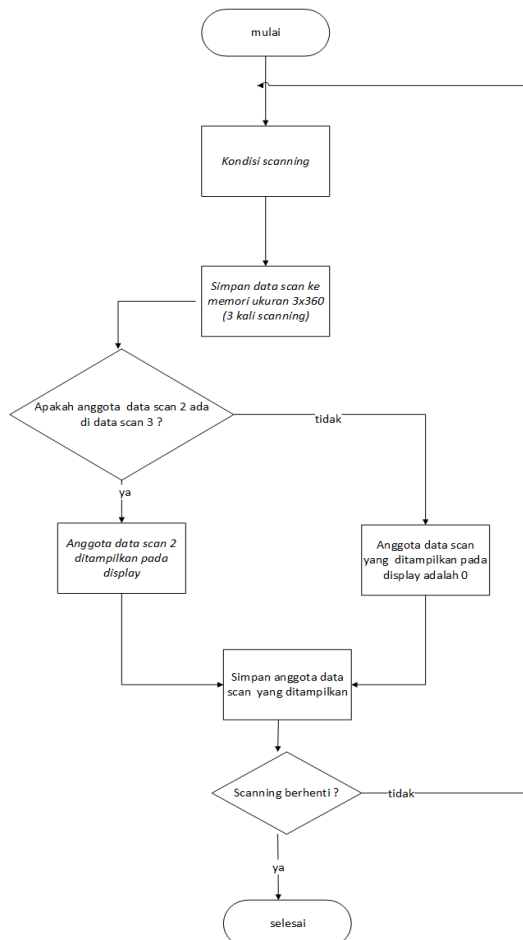
Gambar 4. Flowchart Algoritma 2



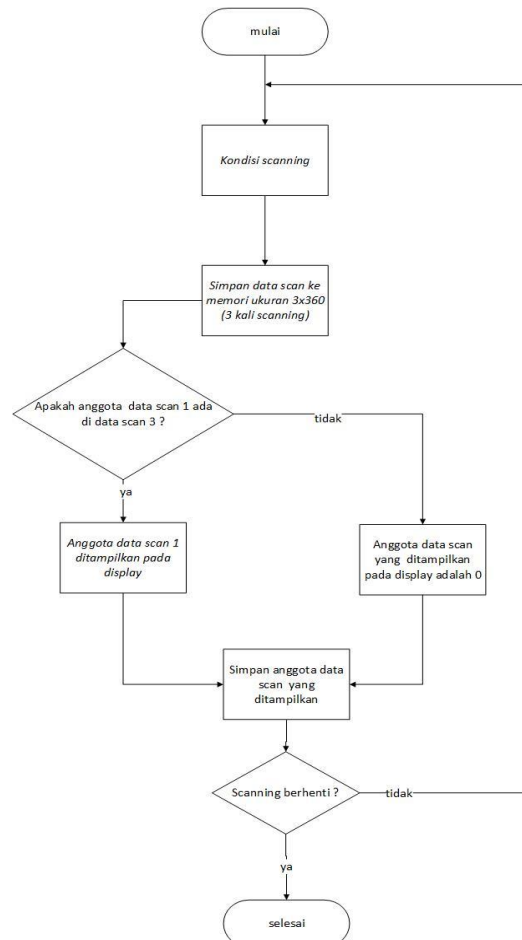
Gambar 5. Flowchart Algoritma 3



Gambar 6. Flowchart Algoritma 4



Gambar 7. Flowchart Algoritma 5



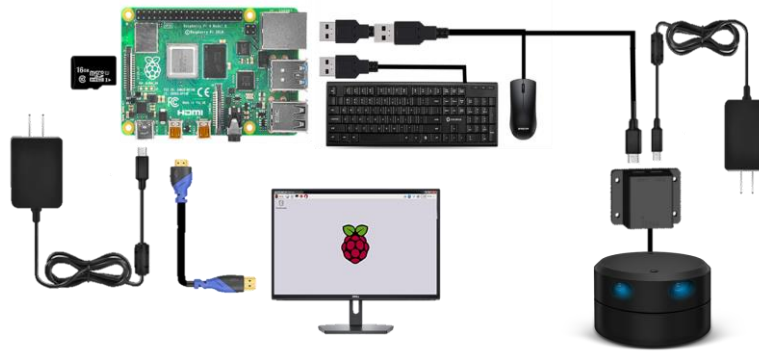
Gambar 8. Flowchart Algoritma 6

IV. IMPLEMENTASI MODEL DAN PEMBAHASAN

Bagian keempat ini dapat berupa Implementasi perancangan mekanik, perancangan algoritma *noise removal*, pembahasan Hasil dan Analisis.:

A. Perancangan mekanik

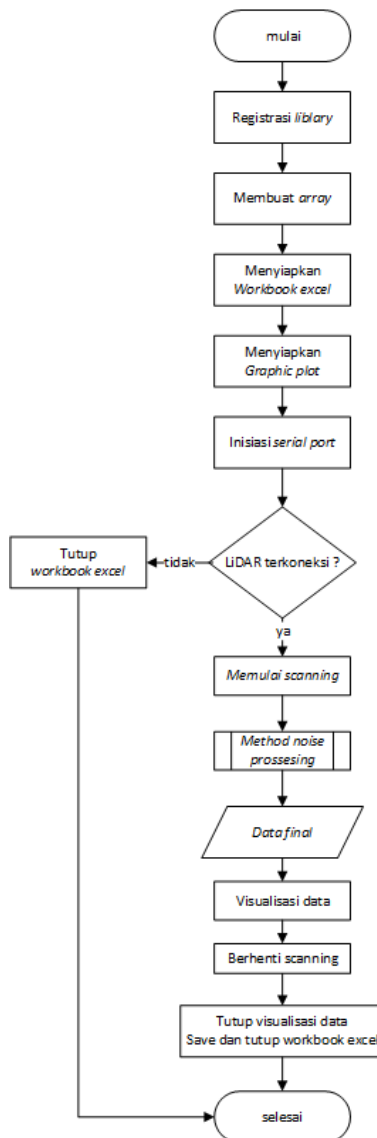
Desain sistem perancangan ini menggunakan alat yang terlihat pada Gambar 9, hubungkan perangkat pendukung seperti *keyboard*, *mouse* kedalam port USB, dan *monitor* kedalam port *micro HDMI type D*, menyalakan raspberry pi 4 dengan menghubungkan catu daya *micro USB type C* ke port catu daya. YDLIDAR G4 (LiDAR 2 D) dapat di akses dengan *adaptor board* yang memiliki port koneksi *type C* untuk komunikasi data YDLIDAR G4 dengan *development board* dalam hal ini adalah Raspberry Pi 4, dan port koneksi *type D* sebagai port catu daya.



Gambar 9. Perancangan mekanik

B. Perancangan Program Noise Remover

Untuk membuat program *noise remover point cloud data* LiDAR dengan *workspace* python pada Raspberry Pi 4, ada beberapa tahapan program yang dapat dilihat pada Gambar 10.



Gambar 10. Flowchart Program Noise Remover

C. Tahapan pengujian (lihat Tabel 1)

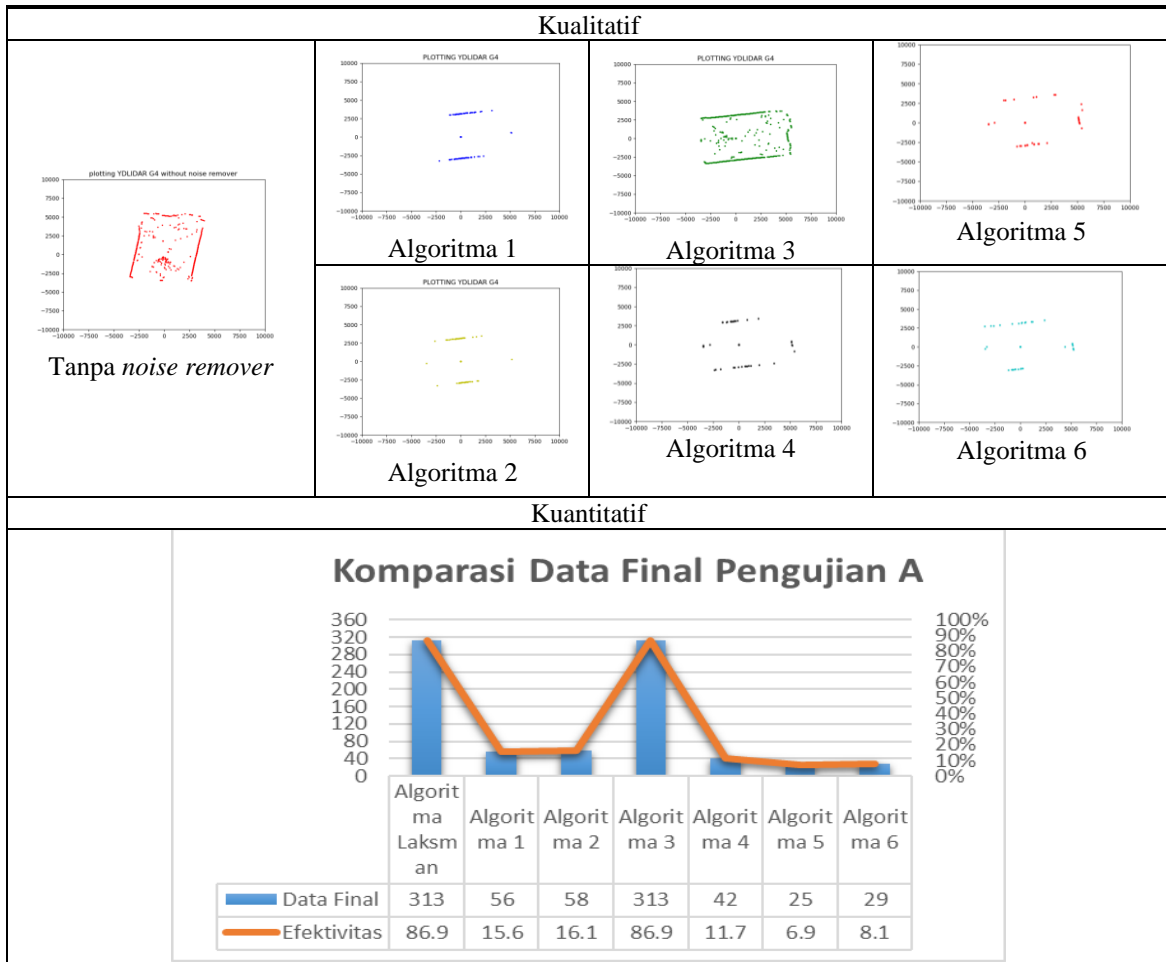
TABEL I
TAHAPAN PENGUJIAN

Subab	A	B
Tahapan Penelitian	Pengujian pertama	Pengujian kedua
Keterangan	Mengetahui perbandingan jumlah dan visualisasi data dalam kondisi ruangan tertutup.	Mengetahui perbandingan jumlah dan visualisasi data LiDAR dalam kondisi ruangan dengan jendela terbuka.
Hasil	Data kualitatif dan kuantitatif Algoritma 1,2,3,4,5,6	Data kualitatif dan kuantitatif Algoritma 1,2,3,4,5,6

D. Hasil dan Pembahasan

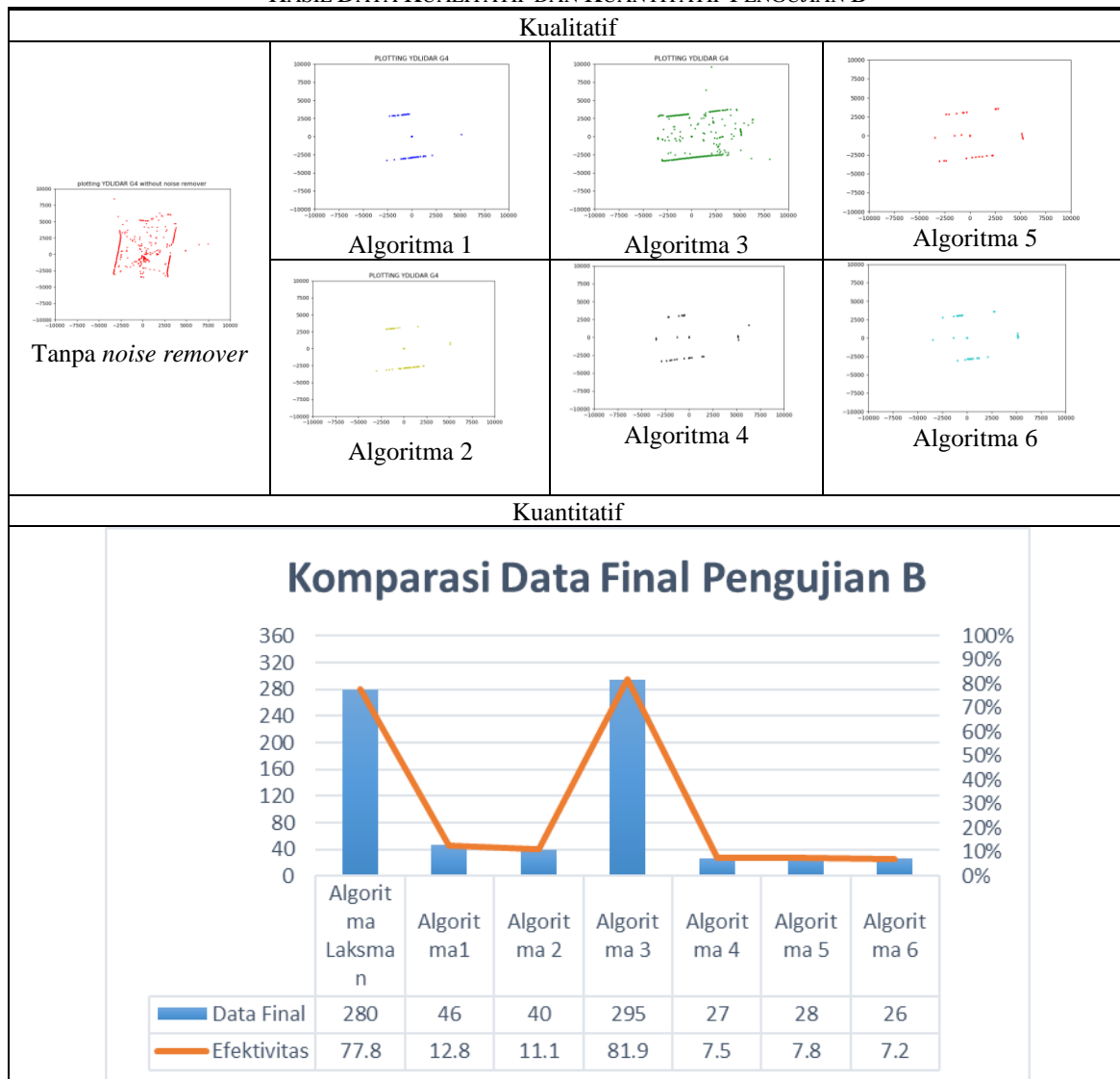
1. Data kualitatif dan kuantitatif pengujian A (lihat Tabel 2)

TABEL II
HASIL DATA KUALITATIF DAN KUANTITATIF PENGUJIAN A



2. Data kualitatif dan kuantitatif pengujian B (lihat Tabel 3)

TABEL III
HASIL DATA KUALITATIF DAN KUANTITATIF PENGUJIAN B



V. KESIMPULAN

Hasil percobaan memperlihatkan bahwa dari keenam Algoritma yang di buat mampu menampilkan visualisasi data berdasarkan sistem pemetaan 2 dimensi yang terkoreksi dari noise. Keenam Algoritma tersebut berhasil menyeleksi noise hingga 100%, meskipun kurang lebih 80% data yang dianggap benar tidak dapat disajikan. Meskipun hanya dengan 20% data benar, namun struktur objek masih dapat dikenali. Keenam algoritma ini dapat berfungsi tanpa adanya eror program dalam mengolah data pendeteksian, memvisualisasikan, merekam secara *realtime* dan menyimpan data pada Raspberry Pi 4.

Saran untuk pengembangan lebih lanjut dari *software* ini dapat dilakukan pengembangan noise remover dengan metode lain, implementasi *software* pada *robot operating system*, Pengujian dan pengembangan pendeteksian terhadap objek yang bergerak, lidar yang bergerak dan pengujian terhadap lingkungan dan cuaca yang berbeda.

UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yang Maha Esa serta Ucapan terima kasih atas terbitnya naskah ini pada Seminar Nasional Sains Teknologi dan Inovasi Indonesia 2021 sebagai bagian kolaborasi/kerjasama penelitian antara Badan Riset Dan Inovasi Nasional (BRIN), Universitas Nurtanio Bandung, dan Dosen/Peneliti Akademi Angkatan Udara (AAU).

REFERENSI

- [1] Data Sheet YDLIDAR G4, <https://www.ydlidar.com/Public/upload/files/2020-10-29/YDLIDAR%20G4%20Datasheet.pdf>, Diunduh pada tanggal 30 November 2020 Pukul 10.00 WIB
- [2] Raspberry Pi 4, <https://www.ros.org/about-ros/>, Diakses pada tanggal 10 Januari 2021 Pukul 16.00 WIB
- [3] Bahasa Pemrograman Python <https://www.pythonindo.com/menggunakan-idle-python/>, Diakses pada tanggal 07 Desember 2020 Pukul 13.00 WIB
- [4] Prayoga,S.,Budianto,S.,Kusuma Atmaja,AB.,system pemetaan ruangan 2D menggunakan Lidar, Vol 9 No. 1 April 2017, 73-79, April 2017.
- [5] Catapang, AN., Manuel Ramos,jr., obstacle detection using a 2D LIDAR system for an autonomous vehicle, 2016 6th internasional conference on control system, computing and engineering, 25-27 November 2016, Penang, Malaysia.
- [6] Cheng,D., Zhao,D., Wei,C., Tian,D., PCA-Based Denoising Algorithm For outdoor Lidar Point Cloud Data. Sensors 2021, <https://doi.org/10.3390/s21113703> , 26 may 2021
- [7] Lakshmanmallidi. 2019. PyLiDAR3. <https://github.com/lakshmanmallidi/PyLidar3> , diakses pada tanggal 18 April 2020.
- [8] Ravankar,AA., Kobayashi,Y., Jixin,LV., Emaru,T., Hoshino,Y., an embarrassingly parallel hopping window noise removing algorithm for lidar based robot mapping, SICE Annual conference 2014, September 9-14, 2014, Hokkaido university, Sappora, Japan.
- [9] Richard Blum, python programming for raspberry pi in 24 hours,sams teach yourself, 2014.
- [10] Zuowei,H., Yuanjiang,H., Jie,H., a method for noise removal of LIDAR point cloud, 2013 third internasional conference on intellegent system design and engineering application, Doi 10.1109/ISDDEA.2012.32.
- [11] Flavio B.P. Malavazi, Remy Guyonneau, Jean-Baptiste Fasquel, Sebastien Lagrange dan Franck Mercier LiDAR-only based navigation algorithm for an autonomous agricultural robot 2018
- [12] Angelo Nikko Catapang dan Manuel Ramos, Jr Obstacle detection using a 2D LiDAR System for an Autonomous vehicle 2016.
- [13] Ankit A. Ravankar, Yukinori Kobayashi, Jixin Lv, Takanori Emaru dan Yohei Hoshino An embarrassingly parallel hopping window noise removing algorithm for lidar based robot mapping 2014./
- [14] Satyawan, A. S., Kurniawan, D., Armi, N., dan Wijayanto, Y. N., Room Map Estimation from Two-Dimensional Lidar's Point Cloud Data, 2019 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), oktober, 2019, Tangerang, Indonesia, 2019, pp. 152-155.