

TUGAS AKHIR - EC224701

NAVIGASI DAN PEMETAAN OTOMATIS BERBASIS ROS PADA TINY DRONES

Aras Aymanusa Sakti

NRP 0721 19 4000 0031

Dosen Pembimbing

Ahmad Zaini, S.T, M.Sc.

NIP 19750419 200212 1 003

Dion Hayu Fandiantoro, S.T.,M.T.

NIP 1994 20201 1 064

Program Studi Strata 1 (S1) Teknik Komputer

Departemen Teknik Komputer

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2023



TUGAS AKHIR - EC224701

NAVIGASI DAN PEMETAAN OTOMATIS BERBASIS ROS PADA TINY DRONES

Aras Aymanusa Sakti

NRP 0721 19 4000 0031

Dosen Pembimbing

Ahmad Zaini, S.T, M.Sc.

NIP 19750419 200212 1 003

Dion Hayu Fandiantoro, S.T.,M.T.

NIP 1994 20201 1 064

Program Studi Strata 1 (S1) Teknik Komputer

Departemen Teknik Komputer

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2023

[Halaman ini sengaja dikosongkan]



FINAL PROJECT PROPOSAL - EC224701

ROS-BASED AUTONOMOUS NAVIGATION AND MAPPING ON TINY DRONES

Aras Aymanusa Sakti

NRP 0721 19 4000 0031

Supervisor

Ahmad Zaini, S.T, M.Sc.

NIP 19750419 200212 1 003

Dion Hayu Fandiantoro, S.T.,M.T.

NIP 1994 20201 1 064

Undergraduate Study Program of Computer Engineering

Department of Computer Engineering

Faculty of Intelligent Electrical and Informatics Technology

Sepuluh Nopember Institute of Technology

Surabaya

2022

[Halaman ini sengaja dikosongkan]

APPROVAL SHEET

NAVIGASI DAN PEMETAAN OTOMATIS BERBASIS ROS PADA TINY DRONES

FINAL PROJECT PROPOSAL

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Teknik pada
Program Studi S-1 Teknik Komputer
Departemen Teknik Komputer
Fakultas Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

By: **Aras Aymanusa Sakti**
NRP. 07211940000031

Approved by Final Project Proposal Examiner Team:

Ahmad Zaini, S.T, M.Sc.
NIP: 19750419 200212 1 003

(Pembimbing)

Dion Hayu Fandiantoro, S.T.,M.T.
NIP: 1994 20201 1 064

(Ko-Pembimbing)

(Penguji I)

(Penguji II)

(Penguji III)

SURABAYA
Mei, 2023

[Halaman ini sengaja dikosongkan]

PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama Mahasiswa / NRP : Aras Aymanusa Sakti / 07211940000031
Departemen : Teknik Komputer
Dosen Pembimbing / NIP : Ahmad Zaini, S.T, M.Sc. / 19750419 200212 1 003
Dion Hayu Fandiantoro, S.T.,M.T. / 1994 20201 1 064

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "Navigasi dan Pemetaan Otomatis Berbasis ROS pada Tiny Drones" adalah hasil karya sendiri, berfsifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, Mei 2023

Mengetahui
Dosen Pembimbing

Mahasiswa

Ahmad Zaini, S.T, M.Sc.
NIP. 19750419 200212 1 003

Aras Aymanusa Sakti
NRP. 07211940000031

[Halaman ini sengaja dikosongkan]

NAVIGASI DAN PEMETAAN OTOMATIS BERBASIS ROS PADA TINY DRONES

Nama Mahasiswa / NRP: Aras Aymanusa Sakti / 07211940000031

Departemen : Teknik Komputer FTEIC - ITS

**Dosen Pembimbing : 1. Ahmad Zaini, S.T, M.Sc.
2. Dion Hayu Fandiantoro, S.T.,M.T.**

Abstrak

Seiring perkembangan zaman, peralatan teknologi drone yang canggih menjadi populer di kalangan masyarakat. Drone dapat digunakan untuk melakukan navigasi dan pemetaan, terutama pada ruangan tertutup. Pemetaan yang menggunakan robot umumnya dilakukan pada robot beroda, walaupun memiliki kekurangan dimana hasil pemetaan tidak maksimal apabila tempat yang dilalui tidak datar. Sedangkan drone yang melakukan pemetaan umumnya dikendalikan secara manual, sehingga akan sulit apabila ingin memetakan area yang tidak dapat dilihat oleh pengendali. Pada penelitian ini akan dikembangkan sebuah sistem dimana navigasi dan pemetaan dapat dilakukan secara otomatis pada drone berukuran mini. Pengembangan sistem ini akan dilakukan dengan mengimplementasikan algoritma SLAM dan algoritma untuk gerak otomatis pada drone berukuran mini dengan LiDAR 4 sisi yang terpasang padanya. Hasil sementara menunjukkan proses pemetaan dapat dilakukan menggunakan Rviz, namun masih belum dapat bergerak secara autonomous.

Kata Kunci: *Teknologi Drone, Pemetaan, Navigasi Otomatis*

[Halaman ini sengaja dikosongkan]

ROS-BASED AUTONOMOUS NAVIGATION AND MAPPING ON TINY DRONES

Student Name / NRP: Aras Aymanusa Sakti / 07211940000031

Departement : Computer Engineering ELECTICS - ITS

Supervisor : 1. Ahmad Zaini, S.T, M.Sc.
2. Dion Hayu Fandiantoro, S.T.,M.T.

Abstract

As time progresses, advanced technology of drones have become popular among the public. Drones can be used for navigation and mapping, especially in enclosed spaces. Mapping using robots is usually done using wheeled robots, although they have the disadvantage of not providing optimal mapping results when the path is not flat. Meanwhile, drones that perform mapping are usually manually controlled, making it difficult to map areas that cannot be seen by the controller. In this research, a system will be developed where navigation and mapping can be automatically performed on mini drones. The development of this system will be carried out by implementing the SLAM algorithm and automatic motion algorithms on mini drones with 4-sided LiDAR mounted on them. Current result shows that the mapping process can be done and recorded using Rviz. The autonomous movement is yet to be implemented.

Keywords: *Drone Technology, Mapping, Automated Navigation*

[Halaman ini sengaja dikosongkan]

ROS-BASED AUTONOMOUS NAVIGATION AND MAPPING ON TINY DRONES

Student Name / NRP: Aras Aymanusa Sakti / 07211940000031

Departement : Computer Engineering ELECTICS - ITS

Supervisor : 1. Ahmad Zaini, S.T, M.Sc.
2. Dion Hayu Fandiantoro, S.T.,M.T.

Abstract

As time progresses, advanced technology of drones have become popular among the public. Drones can be used for navigation and mapping, especially in enclosed spaces. Mapping using robots is usually done using wheeled robots, although they have the disadvantage of not providing optimal mapping results when the path is not flat. Meanwhile, drones that perform mapping are usually manually controlled, making it difficult to map areas that cannot be seen by the controller. In this research, a system will be developed where navigation and mapping can be automatically performed on mini drones. The development of this system will be carried out by implementing the SLAM algorithm and automatic motion algorithms on mini drones with 4-sided LiDAR mounted on them. Current result shows that the mapping process can be done and recorded using Rviz. The autonomous movement is yet to be implemented.

Keywords: *Drone Technology, Mapping, Automated Navigation*

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK	ix
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	2
2 TINJAUAN PUSTAKA	3
2.1 Hasil penelitian terdahulu	3
2.2 Dasar Teori	3
3 METODOLOGI	11
3.1 Metode yang digunakan	11
4 PENGUJIAN DAN ANALISIS	19
4.1 Skenario Pengujian	19
4.2 Hasil Pengujian	21
4.3 Analisis Pengujian	24
5 DAFTAR PUSTAKA	25

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

2.1	Perbandingan fitur pada ROS 1 dan ROS 2	4
2.2	Visualisasi pemetaan pada Rviz	5
2.3	Arsitektur dari Crazyflie 2.x	7
2.4	Sistem kerja dari SLAM	8
2.5	Arsitektur dari sistem LiDAR	9
3.1	Blok diagram metodologi	11
3.2	Multi-ranger Deck yang terpasang pada Crazyflie	12
3.3	Hasil pembacaan data sensor LiDAR	12
3.4	Flow Deck yang terpasang pada Crazyflie	13
3.5	Hasil pembacaan data sensor optical flow	13
3.6	Hasil pembacaan data sensor IMU	14
3.7	Penggambaran arah pergerakan <i>roll</i> , <i>pitch</i> , dan <i>yaw</i>	14
3.8	Penggambaran arah pergerakan <i>roll</i> , <i>pitch</i> , dan <i>yaw</i>	15
3.9	Multi-ranger Deck yang terpasang pada Crazyflie	16
3.10	Flow Deck yang terpasang pada Crazyflie	16
3.11	Flowchart sistem pergerakan otomatis	17
4.1	Lingkungan buatan untuk pengujian	19
4.2	Lingkungan buatan untuk pengujian	20
4.3	Drone dalam keadaan diam pada lingkungan uji	20
4.4	Hasil pemetaan Rviz pada pengujian pertama	21
4.5	Hasil pemetaan Rviz pada pengujian kedua	22
4.6	Hasil pemetaan Rviz pada pengujian ketiga	22
4.7	Hasil pemetaan Rviz pada pengujian keempat	23
4.8	Hasil pemetaan Rviz pada pengujian kelima	23

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

2.1	Tabel spesifikasi untuk drone Crazyflie 2.x	6
4.1	Tabel ukuran dari hasil pengujian	24

[Halaman ini sengaja dikosongkan]

1 PENDAHULUAN

1.1 Latar Belakang

Seiring berkembangnya zaman, semakin banyak peralatan teknologi canggih yang dapat digunakan dalam kehidupan sehari-hari, contohnya adalah *drone*. *Drone* telah menjadi teknologi yang sangat populer dalam beberapa tahun terakhir, dan banyak orang terlibat dalam penelitian dan pengembangan *drone* untuk berbagai tujuan, termasuk militer, pengawasan, hiburan, dan lainnya. Pasar *drone* telah berkembang pesat selama satu dekade ini, dan diperkirakan akan terus berkembang di masa mendatang. Kemajuan teknologi dan penurunan harga mendorong minat dalam penggunaan *drone* untuk berbagai keperluan. Dalam bidang riset dan teknologi, penelitian terhadap *drone* dapat membantu dalam mengembangkan teknologi yang lebih baik, lebih efisien, dan lebih aman. Hal ini dapat membantu dalam meningkatkan kinerja *drone* dan meningkatkan keandalan sistem, serta mengurangi biaya operasional. Penelitian juga dapat membantu dalam mengembangkan teknologi baru yang dapat digunakan dalam aplikasi *drone* yang lebih luas, seperti penggunaan *drone* untuk pengiriman barang atau pencarian dan penyelamatan. Selain itu, *drone* juga dapat digunakan sebagai pengganti kendaraan beroda, karena *drone* bergerak di udara sehingga mempermudah akses dan meningkatkan efisiensi waktu (Pamungkas et al., 2017). Penelitian terhadap *drone* juga dapat membantu dalam mengembangkan standar dan regulasi yang diperlukan untuk mengatur penggunaan *drone*. Ini penting untuk memastikan bahwa *drone* digunakan dengan aman dan tidak merugikan orang lain atau lingkungan.

Drone dapat terdiri dari berbagai macam ukuran, dari yang besar seperti *drone* militer berukuran pesawat hingga yang kecil berukuran telapak tangan. Tiap ukuran memiliki kelebihan, kekurangan, dan kegunaan masing-masing. *Drone* berukuran kecil pada umumnya digunakan pada lingkungan tertutup. Dengan ukurannya yang kecil, *drone* jenis ini hanya dapat beroperasi di luar ruang udara terkontrol dan tidak bisa terbang lebih dari 150 meter di atas tanah, sehingga tidak perlu adanya izin resmi untuk operasi (Menhub, 2015). Navigasi berperan penting terhadap kemajuan umat manusia. Tanpa adanya navigasi, maka sejarah penjelajahan dunia yang telah kita ketahui tidak akan pernah terjadi. Pada zaman sekarang, navigasi dapat membantu teknologi tertentu untuk mempelajari dan menentukan arah secara akurat, misalnya pada kendaraan ataupun kompas digital. Apabila dihubungkan dengan pengembangan robot, navigasi dapat membantu menentukan posisi dan perencanaan pergerakan menuju lokasi yang diinginkan (Liani, 2017).

Pada umumnya, pemetaan ruangan oleh robot dilakukan menggunakan robot yang bergerak menggunakan roda (Quang et al., 2019). Robot pemetaan beroda memiliki keterbatasan dimana harus melalui tempat yang datar agar data peta ruangan yang didapatkan bisa akurat. Di sisi lain, *drone* yang melakukan pemetaan umumnya menggunakan *drone* berukuran besar (Dijkshoorn, 2012). Hal ini menyebabkan *drone* hanya dapat diterbangkan di ruang terbuka yang luas, cukup beresiko apabila diterbangkan di ruang tertutup yang sempit. Pemetaan di ruang tertutup harus dilakukan secara presisi agar tidak bertabrakan dengan benda-benda di ruang tersebut. Berdasarkan hal-hal tersebut, maka diperlukan sebuah penelitian untuk mengembangkan fitur navigasi dan pemetaan secara otomatis pada *drone* berukuran mini. Fitur ini dikembangkan agar *drone* berukuran mini dapat melakukan pemetaan di ruang tertutup secara presisi, seperti ruang penyimpanan barang dan area pertambangan bawah tanah. Dasar sistem navigasi dan pemetaan untuk *drone* yang akan digunakan adalah dari Crazyflie yang terintegrasi dengan Robot Operating System 2 (ROS 2).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan pada sub-bab 1.1, didapatkan rumusan masalah sebagai berikut:

1. Apakah program navigasi dan pemetaan secara otomatis berbasis ROS pada *drone* berukuran mini dapat diterapkan pada ruang tertutup?
2. Apakah ada beda antara hasil pengukuran dari *drone* yang menerapkan program navigasi dan pemetaan otomatis dengan pengukuran secara manual?

1.3 Batasan Masalah

Untuk memfokuskan permasalahan yang akan diselesaikan maka dilakukan pembatasan masalah. Berikut ini merupakan batasan masalah yang telah ditentukan:

1. Peta yang dihasilkan berupa model 2 Dimensi (2D) dari ruang dan terdiri dari 1 layer.
2. Ketinggian terbang *drone* stabil pada 30 cm di atas bidang lantai.

1.4 Tujuan

Berdasarkan rumusan masalah yang telah disebut pada subbab 1.2, didapatkan tujuan dari penelitian ini sebagai berikut:

1. Untuk mengetahui hasil penerapan program navigasi dan pemetaan secara otomatis berbasis ROS pada *drone* berukuran mini.
2. Untuk mengetahui beda hasil pengukuran manual dengan pengukuran dari *drone* yang menerapkan program navigasi dan pemetaan otomatis.

1.5 Manfaat

Adapun manfaat yang didapatkan dari dilakukannya penelitian ini sebagai berikut:

1. Membantu nav

2 TINJAUAN PUSTAKA

2.1 Hasil penelitian terdahulu

Pada jurnal Desain dan Implementasi *Path Following Drone*, Fernaldy Julius *et al.* melakukan perancangan drone yang dapat bergerak sendiri sesuai jalur yang sudah ditetapkan pada program komputer. Pengujian drone dilakukan menggunakan jalur berbentuk segitiga dengan tiga skenario pengujian, dimana pada masing-masing skenario dilakukan lima kali pengujian. Pada skenario pertama, Drone bergerak di luar ruangan dan mendarat di titik akhir segitiga. Pada skenario kedua, Drone bergerak di luar ruangan dan mendarat di setiap titik segitiga. Pada skenario ketiga, drone bergerak di dalam ruangan dan mendarat di titik akhir segitiga. Dari tiga pengujian tersebut, Fernaldy Julius *et al.* mendapatkan hasil berupa drone mengikuti jalur yang diberikan, namun tidak dapat mendeteksi perbaikan posisi drone, sehingga berpengaruh pada akurasi gerakan (Julius *et al.*, 2019).

Pada jurnal *LiDAR-Based Navigation of Tethered Drone Formations in an Unknown Environment*, Bolognini dan Fagiano membahas tentang navigasi drone yang terikat dengan menggunakan sensor *Light Detection and Ranging* (LiDAR) dalam lingkungan yang tidak dikenal. Pada jurnal ini, drone terikat dikendalikan dengan menggunakan algoritma evolusi genetik untuk mencari jalur terbaik menuju tujuan yang ditentukan. Selain itu, jurnal ini juga menjelaskan tentang pengontrolan kestabilan drone secara real-time dengan menggunakan sistem *Proportional-Integral-Derivative* (PID) dan simulasi yang dilakukan menunjukkan bahwa sistem navigasi ini mampu mencapai tujuan dengan akurasi yang cukup baik dalam lingkungan yang tidak dikenal (Bolognini & Fagiano, 2020).

Pada jurnal *Mapping and navigation with four-wheeled omnidirectional mobile robot based on robot operating system*, Quang *et al.* membahas tentang cara menggunakan Robot Operating System (ROS) untuk mengatasi masalah pemetaan dan navigasi robot beroda empat, yang dapat bergerak ke segala arah tanpa mengubah orientasinya. Menurut penelitian ini, algoritma Simultaneous Localization and Mapping (SLAM) yang menggunakan teknik odometri dan sensor LiDAR dapat digunakan untuk memetakan lingkungan sekitar robot secara real-time. Hasil penelitian ini menunjukkan bahwa ROS dapat membantu mengembangkan aplikasi robot yang rumit untuk pemetaan dan navigasi lingkungan (Quang *et al.*, 2019).

Pada tesis *Simultaneous localization and mapping with the ar. drone*, Dijkshoorn membahas penggunaan teknologi SLAM dengan menggunakan drone AR. Penulis mengusulkan pendekatan SLAM berbasis visual yang menggunakan teknik penggabungan fitur dan estimasi gerakan relatif untuk membangun peta tiga dimensi dari lingkungan sekitar *drone*. Pada tahap navigasi, algoritma pemilihan jalur yang cerdas digunakan untuk memandu drone menuju tujuan yang ditentukan. Tesis ini juga menjelaskan bagaimana peta visual dari sebuah lingkungan dapat dibuat. Metode yang digunakan berupa lokalisasi, menggunakan pendekatan baru untuk memulihkan translasi dan rotasi antara bingkai kamera dan peta. Hasil tesis menunjukkan penggunaan data visual dan teknik penggabungan fitur dalam metode SLAM memberikan kemampuan untuk memetakan lingkungan dengan detail dan memperoleh informasi posisi secara akurat (Dijkshoorn, 2012).

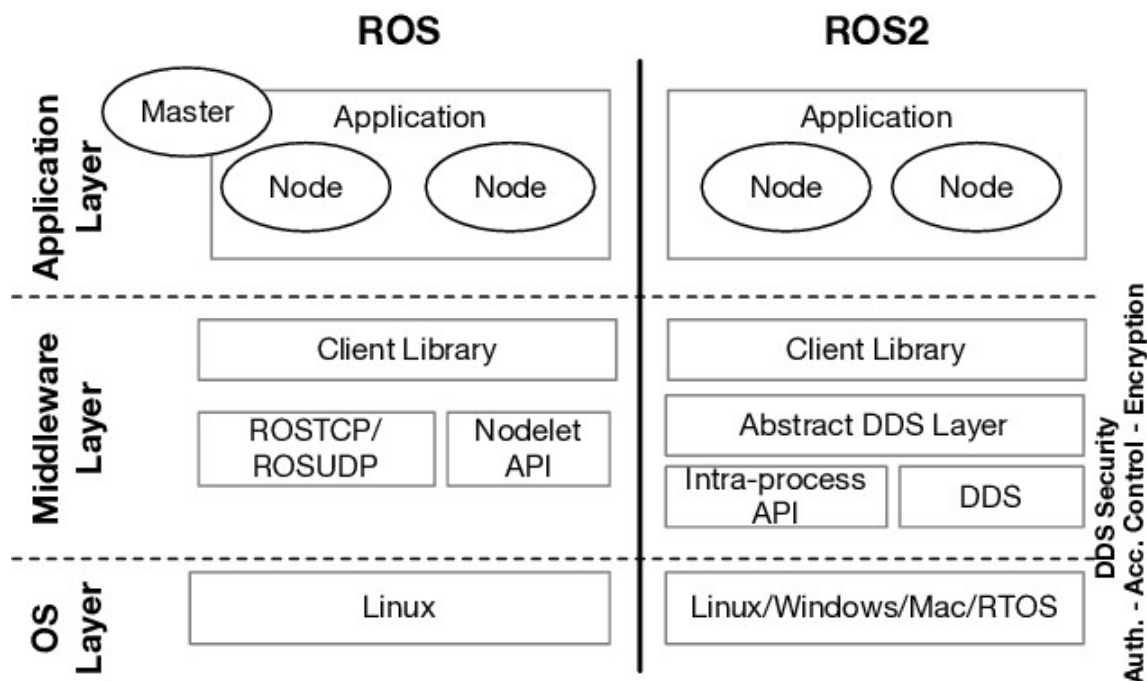
2.2 Dasar Teori

2.2.1 Robot Operating System

Robot Operating System (ROS) adalah sistem operasi yang dikembangkan khusus untuk robot. ROS merupakan platform terbuka yang dapat digunakan untuk membangun, menginte-

grasikan, dan memelihara robot dengan mudah. ROS terdiri dari sekumpulan *library* yang dapat digunakan untuk mengontrol robot, mengelola sensor, dan mengelola data yang terkumpul dari robot. ROS menyediakan struktur yang memungkinkan pengembang untuk dengan mudah membangun aplikasi yang mengontrol robot dengan menggunakan bahasa pemrograman seperti C++ atau Python. ROS juga menyediakan alat yang memungkinkan pengembang untuk dengan mudah mengotomatisasi tugas yang rumit, seperti navigasi, perencanaan jalur, dan pengenalan objek. ROS telah digunakan dalam berbagai aplikasi robotik, mulai dari robot rumah tangga sederhana hingga robot industri yang canggih (Jalil, 2022).

Saat ini ROS memiliki dua versi yang berbeda, ROS 1 dan ROS 2. ROS 2 dikembangkan karena ROS 1 memiliki beberapa kelemahan. Kelemahan yang cukup vital pada ROS 1 adalah keterbatasan performa yang membuat ROS 1 tidak bisa digunakan untuk komunikasi *real-time*. Sistem yang dapat melakukan komunikasi *real-time* akan lebih dipakai pada kehidupan sehari-hari. Oleh karena itu, para pengembang mulai beralih untuk menggunakan ROS 2 daripada ROS 1 (Park et al., 2020).



Gambar 2.1: Perbandingan fitur pada ROS 1 dan ROS 2

Sumber : https://www.researchgate.net/figure/Comparison-between-ROS-and-ROS2_fig_4335382592

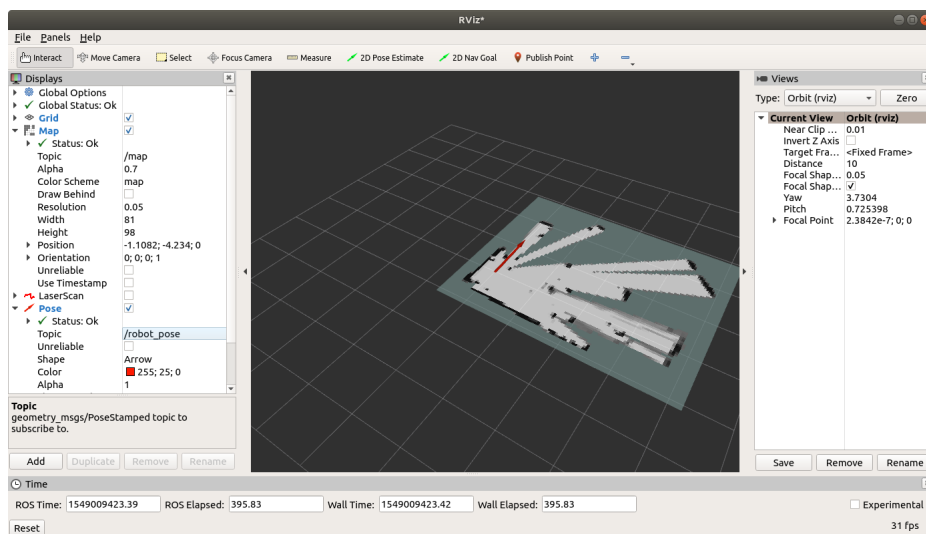
Seperti yang ditampilkan pada Gambar 2.1, ROS 2 menggunakan protokol Data Distribution Service (DDS) yang bersifat *peer-to-peer*, sedangkan ROS 1 masih menggunakan protokol TCP/UDP yang bersifat *master-slave*. Dengan penggunaan DDS sebagai *middleware*, keamanan pada sistem akan lebih terjamin (Mazzeo & Staffa, 2020).

2.2.2 Rviz

Dalam ilmu komputasi dan grafika, terdapat kebutuhan untuk merepresentasikan dan memvisualisasikan informasi secara nyata, dan berbagai macam jenis visualisasi struktur data dan algoritma telah dikembangkan untuk melakukannya. Visualisasi data selama ini cenderung kurang fleksibel dalam menampilkan struktur data dan algoritma terbaru karena hanya dapat

menerima data spesifik. Dari permasalahan tersebut, maka sebuah alat visualisasi bernama Rviz dikembangkan.

Rviz adalah salah satu alat visualisasi paling kuat yang digunakan di lingkungan ROS dari Sistem Operasi Robot. Rviz memungkinkan pengguna memvisualisasikan data sensor dan model robot dalam lingkungan 3D interaktif. Rviz dapat digunakan untuk memvisualisasikan data dari sensor seperti laser, kamera, dan pemindai kedekatan, serta untuk memvisualisasikan model robot seperti model kinematik, model grafis, dan lintasan terencana. Rviz mendukung berbagai visualisasi termasuk peta 2D dan 3D, representasi robot di ruang angkasa, tampilan kamus stereografis, tampilan data titik, tampilan gambar kamera, dan lainnya. Rviz memungkinkan pengguna untuk mempelajari dan memvalidasi perilaku robot, memahami keluaran sensor, memecahkan masalah, dan mengembangkan serta menguji algoritma navigasi dan kontrol (Kam et al., 2015).



Gambar 2.2: Visualisasi pemetaan pada Rviz

Sumber : <https://ardupilot.org/dev/docs/ros-rviz.html>

Gambar 3.10 merupakan tampilan antarmuka yang muncul ketika menggunakan Rviz. Pada bagian tengah merupakan visualisasi dari keluaran sensor yang digunakan, umumnya sensor inframerah. Pada bagian kiri dari antarmuka terdapat kotak Displays yang memberi informasi apa saja yang divisualisasikan, beserta pengaturan untuk bagaimana visualisasi tersebut disajikan. Pada bagian kanan dari antarmuka merupakan kotak Views yang berfungsi untuk mengatur orientasi dari visualisasi yang didapatkan.

2.2.3 Crazyflie 2.1

Crazyflie 2.1 adalah drone mini yang dikembangkan oleh Bitcraze, sebuah perusahaan teknologi Swedia. Crazyflie 2.1 merupakan generasi kedua dari Crazyflie, yang pertama kali dirilis pada tahun 2011. Crazyflie 2.1 sangat ringan, hanya berukuran sekitar 9.5 gram, dan memiliki ukuran yang kecil. Crazyflie 2.1 dapat dioperasikan dengan menggunakan perangkat kontrol nirkabel atau melalui komputer dengan menggunakan aplikasi yang disediakan. Crazyflie 2.1 dilengkapi dengan berbagai sensor, termasuk sensor giroskop, akselerometer, barometer, dan sensor inframerah (Giernacki et al., 2017).

Giroskop merupakan perangkat untuk mengukur dan mempertahankan orientasi. Sensor giroskop yang terpasang pada perangkat elektronik dapat mendeteksi gerakan sesuai gravitasi

yang berlaku (Rahman, 2016). Akselerometer merupakan sensor yang digunakan percepatan statis dan dinamis suatu objek. Prinsip kerja pada sensor akselerometer berupa prinsip percepatan sebuah per dengan beban yang dilepaskan bergerak dengan suatu percepatan hingga kondisi tertentu, lalu berhenti (Alfaeru & Setiawan, 2016). Barometer merupakan sensor yang mendeteksi ketinggian berdasarkan nilai tekanan udara di sekitarnya. Barometer mencari nilai ketinggian terhadap tekanan udara, sehingga pengaruh cuaca dan kecepatan angin akan memberikan efek pada nilai ketinggian barometer (Alamsyah & Rivai, 2020).

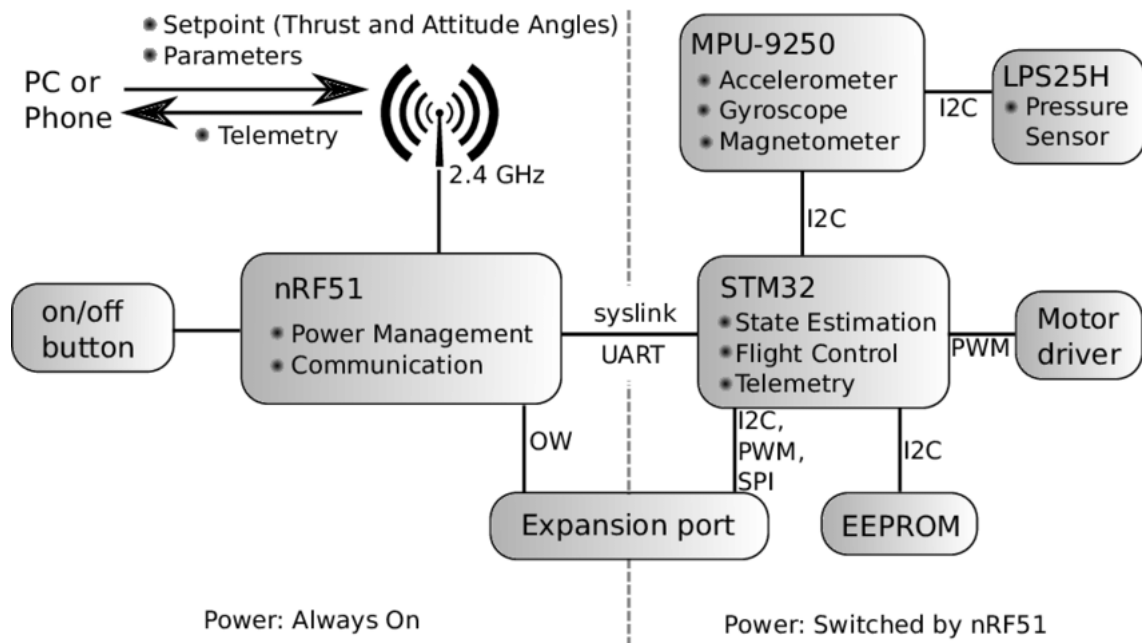
Crazyflie 2.1 cocok untuk digunakan dalam aplikasi penelitian, pengembangan, dan hiburan. Drone ini juga bisa digunakan untuk menguji perangkat lunak dan perangkat keras yang dikembangkan untuk robotik dan sistem pengendalian drone. Crazyflie 2.1 juga dapat diintegrasikan dengan perangkat lunak robotik seperti ROS untuk memudahkan pengembangan aplikasi robotik (Bitcraze, 2022).

Tabel 2.1: Tabel spesifikasi untuk drone Crazyflie 2.x

Sumber : <https://www.bitcraze.io/products/crazyflie-2-1/>

Massa	27 g
Ukuran	92 x 92 x 92 mm
Lama Daya Baterai	7 menit
Microcontroller	STM32F045
Spesifikasi IMU	3-axis akselerometer dan sensor tekanan
Connector	micro-USB
Frekuensi	2.4 GHz

Pada tabel 2.1 disebutkan spesifikasi dari Crazyflie versi 2.x, termasuk Crazyflie versi 2.1. *Drone* memiliki massa 27 gram dan ukuran 92 x 92 x 92 milimeter, sehingga dikategorikan sebagai *drone* mini. Dengan ukuran serta massa sedemikian rupa, akan sulit bagi *drone* ini untuk diterbangkan di area terbuka. Angin akan dengan mudah mengganggu terbang dari *drone*. Baterai yang digunakan dapat bertahan selama 7 menit pada penggunaan normal, sehingga *drone* tidak dapat diterbangkan terlalu lama. Crazyflie memiliki konektor USB Mikro apabila pengguna hanya ingin mengambil data tanpa perlu menerbangkannya. Penjelasan lanjut mengenai penggunaan mikrokontroler, spesifikasi *Inertial Measurement Unit* (IMU), serta frekuensi dapat dilihat pada Gambar 2.3.



Gambar 2.3: Arsitektur dari Crazyflie 2.x

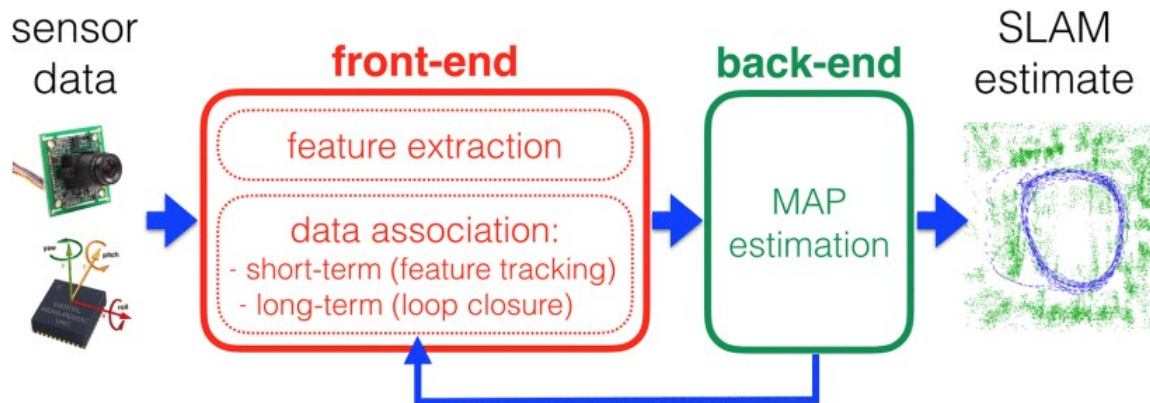
Sumber : https://www.researchgate.net/figure/Components-and-architecture-of-the-Crazyflie-20-quadcopter-Based-on-images-by-Bitcraze_fig2_317751508

Sistem komunikasi pada Crazyflie digunakan untuk mengirim *setpoint* berupa seberapa maju dan ketinggian drone, meningkatkan parameter internal, dan data telemetri seperti pembacaan sensor. Sensor IMU dan sensor tekanan terhubung dengan mikrokontroler STM32F045, berfungsi untuk mendapatkan data estimasi ketinggian drone terbang serta kontrol terbang dari drone itu sendiri. Mikrokontroler pada Crazyflie menjalankan FreeRTOS sebagai kernel sistem operasinya, sehingga memungkinkan estimasi keadaan dan kendali drone dijalankan pada frekuensi 250 Hz. Mikrokontroler nRF51 digunakan untuk komunikasi nirkabel dan manajemen daya. Kedua mikrokontroler pada Crazyflie bertukar data menggunakan protokol syslink dengan Universal Asynchronous Receiver-Transmitter (UART) sebagai antarmuka fisiknya. Crazyflie juga memiliki *port* tambahan untuk ekstensi berupa "Decks" yang bisa digunakan untuk kapabilitas tambahan (Hönig & Ayanian, 2017).

2.2.4 Simultaneous Localization And Mapping

Simultaneous Localization And Mapping (SLAM) adalah proses pembuatan peta lingkungan secara bersamaan dengan menentukan posisi robot atau drone di dalam peta tersebut. Proses ini sangat penting bagi robot atau drone yang bergerak di lingkungan yang tidak dikenal, karena memungkinkan robot atau drone untuk menentukan posisinya sendiri dan mengetahui letak objek di sekitarnya (Robertson et al., 2009). SLAM biasanya menggunakan sensor seperti laser scanner, kamera, atau ultrasonic untuk mengumpulkan data lingkungan dan membangun peta yang akurat.

SLAM juga menggunakan algoritma pemrosesan data yang kompleks untuk menentukan posisi robot atau drone secara real-time sesuai dengan data yang diperoleh dari sensor (Magnabosco & Breckon, 2013). SLAM sangat berguna bagi robot yang harus bekerja di lingkungan yang tidak dikenal, seperti robot penyelamat, robot pemadam kebakaran, atau drone penyelidik. SLAM juga dapat digunakan dalam aplikasi hiburan seperti perlombaan balap *drone*.



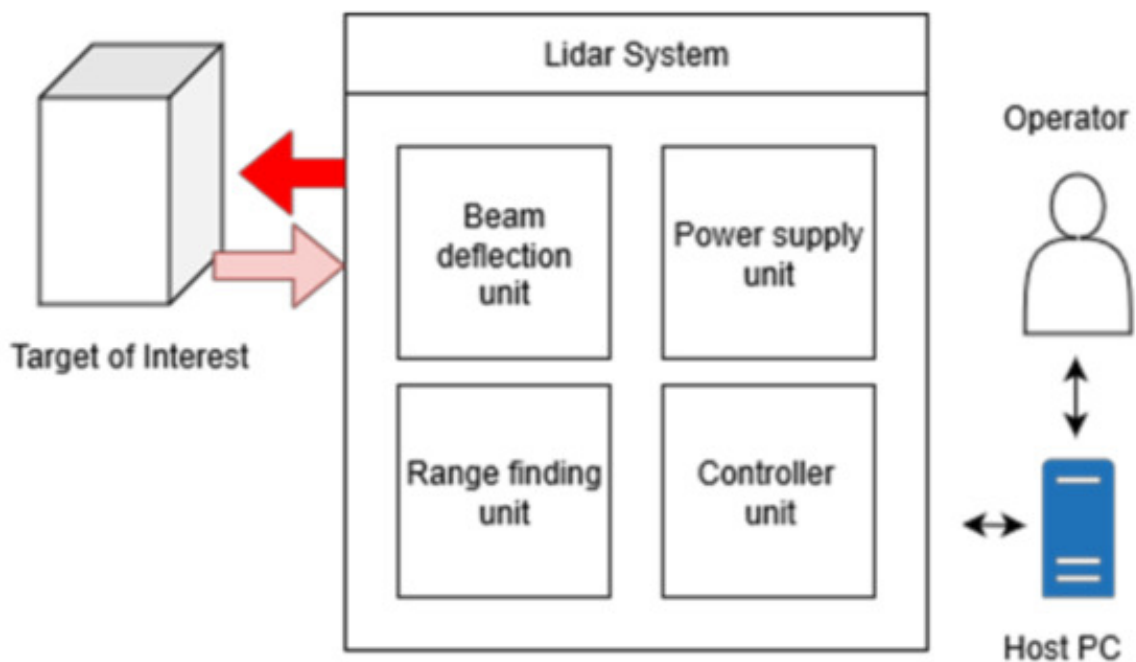
Gambar 2.4: Sistem kerja dari SLAM

Sumber : <https://ieeexplore.ieee.org/abstract/document/7747236>

Gambar 2.4 menunjukkan sistem pada SLAM memiliki dua komponen utama berupa *front end* dan *back end*. Bagian *front end* dari sistem SLAM bertugas untuk melakukan abstraksi data dari sensor menjadi model yang menerima proses estimasi. Bagian *back end* dari sistem SLAM bertugas untuk melakukan inferensi dari data yang terabstraksi oleh *front end* dari sistem SLAM (Cadena et al., 2016).

2.2.5 Light Detection and Ranging

Light Detection and Ranging (LiDAR) merupakan salah satu tipe deteksi yang mengukur properti dari cahaya untuk mendapatkan informasi. Selama beberapa dekade terakhir, LiDAR telah berkontribusi pada ilmu pengetahuan mengenai atmosfer Bumi. LiDAR dapat digunakan untuk menyelidiki parameter atmosfer yang sangat beragam (Wandinger, 2005). Saat ini, teknologi LiDAR telah mendapatkan popularitas yang besar dalam berbagai aplikasi seperti navigasi, robotika, penginderaan jauh, dan *Advanced Driving Assistance Systems* (ADAS). Popularitas ini terutama disebabkan oleh peningkatan kinerja LiDAR dalam hal deteksi jarak, akurasi, konsumsi daya, serta fitur fisik seperti dimensi dan berat.



Gambar 2.5: Arsitektur dari sistem LiDAR

Sumber : <https://www.mdpi.com/2079-9292/9/5/741>

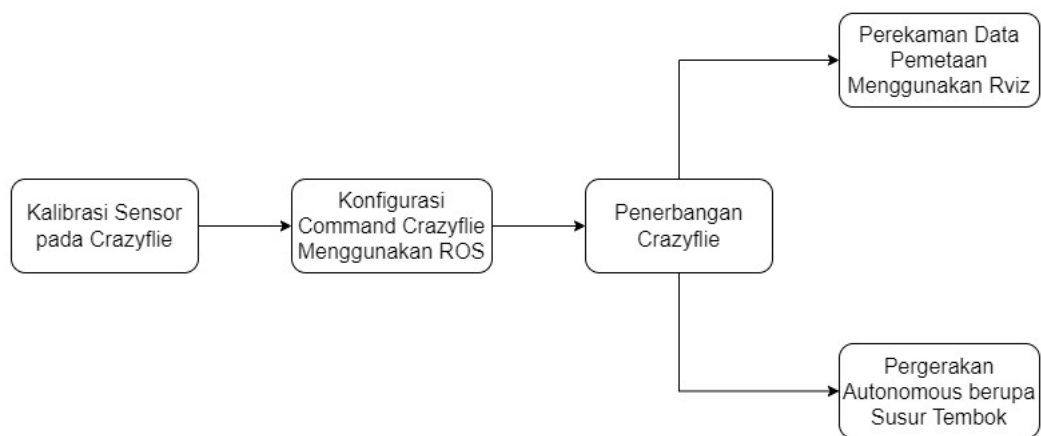
Dari penggambaran arsitektur LiDAR pada Gambar 2.5, sistem LiDAR yang berfungsi secara penuh memiliki 4 sub-sistem utama berupa deflektor sorotan, manajemen daya, laser pencari jarak, serta sistem pengendali. Keempat sub-sistem ini bersifat imperatif yang apabila ada kesalahan pada salah satu sub-sistem dapat mengakibatkan kegagalan sistem fungsionalitas dari LiDAR. Khusus sub-sistem deflektor sorotan, tanpa sub-sistem tersebut LiDAR tetap dapat berfungsi sebagai LiDAR 1 Dimensi (1D). Unit laser pencari jarak merupakan inti dari sistem LiDAR, karena memiliki komponen-komponen yang diperlukan untuk menciptakan, mengirimkan, serta menerima pulsa pendek dari laser. Beberapa komponen yang umum ditemukan pada unit tersebut diantaranya adalah laser dioda, foto dioda, *Transimpedance Amplifier* (TIA), dan *Time-to-Digital Converter* (TDC). Unit pengendali mencakup pemrosesan dasar sinyal, kontrol pembuatan sinyal, serta komunikasi dengan komputer penerima. Unit manajemen daya bertanggungjawab atas produksi kebutuhan daya untuk sistem LiDAR (Raj et al., 2020).

[Halaman ini sengaja dikosongkan]

3 METODOLOGI

Penelitian ini dilaksanakan sesuai dengan desain sistem agar mencapai target yang diharapkan. Dalam teknis pengerjaan penelitian ini akan diuraikan menjadi diagram alur.

3.1 Metode yang digunakan



Gambar 3.1: Blok diagram metodologi

Gambar 3.1 menggambarkan blok diagram dari metodologi yang dilakukan. Berikut ini tahapan-tahapan yang digambarkan untuk menjelaskan alur dari penelitian yang dilakukan:

1. Kalibrasi Sensor pada Crazyflie
2. Konfigurasi Perintah untuk Crazyflie
3. Penerbangan Crazyflie

3.1.1 Kalibrasi Sensor pada Crazyflie

Pada Crazyflie 2.1 akan dilakukan kalibrasi sensor-sensor yang digunakan. Sensor yang terpasang pada Crazyflie 2.1 berupa Multi-ranger Deck (sensor LiDAR 5 sisi), Flow Deck (sensor *optical flow*), dan sensor IMU.



Gambar 3.2: Multi-ranger Deck yang terpasang pada Crazyflie

Gambar 3.2 merupakan ekstensi dari Crazyflie yang bernama Multi-ranger Deck. Sensor tertanam pada lima sisi yang dibagi menjadi sensor bagian depan, sensor bagian belakang, sensor bagian kiri, sensor bagian kanan, serta sensor bagian tengah yang menghadap ke atas. Multi-ranger Deck pada Crazyflie 2.1 berguna untuk mendeteksi objek disekitar, dengan mengukur jarak dari lima arah: depan, belakang, kiri, kanan, dan atas. Sensor LiDAR yang digunakan berbasis dari sensor *Time of Flight* (ToF) VL53L1x, memiliki tingkat akurasi hingga 4 milimeter dan jarak maksimum hingga 4 meter.

```
{'range.front': 350.0, 'range.back': 105.0, 'range.left': 1935.0, 'range.right': 32766.0}
{'range.front': 353.0, 'range.back': 100.0, 'range.left': 1961.0, 'range.right': 63.0}
{'range.front': 350.0, 'range.back': 102.0, 'range.left': 1958.0, 'range.right': 32766.0}
{'range.front': 351.0, 'range.back': 100.0, 'range.left': 1949.0, 'range.right': 66.0}
{'range.front': 356.0, 'range.back': 100.0, 'range.left': 1963.0, 'range.right': 32766.0}
{'range.front': 356.0, 'range.back': 101.0, 'range.left': 32766.0, 'range.right': 66.0}
{'range.front': 363.0, 'range.back': 102.0, 'range.left': 1962.0, 'range.right': 32766.0}
{'range.front': 365.0, 'range.back': 99.0, 'range.left': 1970.0, 'range.right': 66.0}
{'range.front': 362.0, 'range.back': 102.0, 'range.left': 1991.0, 'range.right': 32766.0}
{'range.front': 360.0, 'range.back': 98.0, 'range.left': 1960.0, 'range.right': 63.0}
{'range.front': 363.0, 'range.back': 98.0, 'range.left': 1978.0, 'range.right': 32766.0}
{'range.front': 362.0, 'range.back': 104.0, 'range.left': 32766.0, 'range.right': 66.0}
{'range.front': 358.0, 'range.back': 102.0, 'range.left': 1956.0, 'range.right': 32766.0}
{'range.front': 358.0, 'range.back': 102.0, 'range.left': 1958.0, 'range.right': 67.0}
{'range.front': 362.0, 'range.back': 102.0, 'range.left': 1984.0, 'range.right': 32766.0}
{'range.front': 359.0, 'range.back': 101.0, 'range.left': 1948.0, 'range.right': 65.0}
{'range.front': 359.0, 'range.back': 100.0, 'range.left': 1977.0, 'range.right': 32766.0}
{'range.front': 363.0, 'range.back': 100.0, 'range.left': 32766.0, 'range.right': 66.0}
{'range.front': 362.0, 'range.back': 99.0, 'range.left': 1965.0, 'range.right': 32766.0}
{'range.front': 359.0, 'range.back': 102.0, 'range.left': 1960.0, 'range.right': 65.0}
{'range.front': 359.0, 'range.back': 102.0, 'range.left': 1957.0, 'range.right': 32766.0}
{'range.front': 358.0, 'range.back': 100.0, 'range.left': 1969.0, 'range.right': 65.0}
```

Gambar 3.3: Hasil pembacaan data sensor LiDAR

Setelah Multi-ranger Deck terpasang pada Crazyflie, dilakukan kalibrasi dan pengujian terhadap pembacaan sensor, seperti yang ditampilkan pada Gambar 3.3. Sebelum dijalankan, perlu menentukan parameter serta kecepatan pembacaan datanya. Data yang ditampilkan berupa data jarak yang dibaca dari masing-masing sensor pada keempat sisi dalam satuan milimeter.



Gambar 3.4: Flow Deck yang terpasang pada Crazyflie

Untuk memaksimalkan penggunaan Multi-ranger Deck, maka diperlukan ekstensi lain dari Crazyflie yang bernama Flow Deck, seperti yang ditunjukkan pada Gambar 3.4. Terdapat dua sensor yang tertanam pada *board* dari Flow Deck, sensor ToF yang sama dengan sensor pada Multi-ranger Deck dan sensor *optical flow* PMW3901. Apabila Flow Deck dipasang pada Crazyflie, maka kedua sensor tersebut akan menghadap tanah. Flow Deck berguna menambahkan kemampuan Crazyflie untuk memahami arah pergerakannya sendiri, ditambah dengan sensor VL53L1x untuk mengukur jarak Crazyflie dengan tanah.

```
{ 'stateEstimate.x': 0.4615999162197113, 'stateEstimate.y': 0.5353861451148987, 'stateEstimate.z': 0.2726270854473114 }
{ 'stateEstimate.x': 0.45830366015434265, 'stateEstimate.y': 0.5248045921325684, 'stateEstimate.z': 0.2755751609802246 }
{ 'stateEstimate.x': 0.4576396942138672, 'stateEstimate.y': 0.5236003398895264, 'stateEstimate.z': 0.2739547789096832 }
{ 'stateEstimate.x': 0.45476895570755005, 'stateEstimate.y': 0.5250220894813538, 'stateEstimate.z': 0.2718466222862244 }
{ 'stateEstimate.x': 0.4524610936641693, 'stateEstimate.y': 0.5216906666755676, 'stateEstimate.z': 0.2683148980140686 }
{ 'stateEstimate.x': 0.44625967741012573, 'stateEstimate.y': 0.5200964212417603, 'stateEstimate.z': 0.26896610856056213 }
{ 'stateEstimate.x': 0.44420748949050903, 'stateEstimate.y': 0.5140860080718994, 'stateEstimate.z': 0.26933664083480835 }
{ 'stateEstimate.x': 0.4381854236125946, 'stateEstimate.y': 0.5116848349571228, 'stateEstimate.z': 0.2674330770969391 }
{ 'stateEstimate.x': 0.4445769786834717, 'stateEstimate.y': 0.5120679140090942, 'stateEstimate.z': 0.26787376403808594 }
{ 'stateEstimate.x': 0.4412633776664734, 'stateEstimate.y': 0.5242742896080017, 'stateEstimate.z': 0.26376184821128845 }
{ 'stateEstimate.x': 0.44131821393966675, 'stateEstimate.y': 0.5279955863952637, 'stateEstimate.z': 0.26322445273399353 }
{ 'stateEstimate.x': 0.4411714971065521, 'stateEstimate.y': 0.5312199592590332, 'stateEstimate.z': 0.2639031708240509 }
{ 'stateEstimate.x': 0.44478267431259155, 'stateEstimate.y': 0.5298701524734497, 'stateEstimate.z': 0.2645721435546875 }
{ 'stateEstimate.x': 0.44132232666015625, 'stateEstimate.y': 0.5318358540534973, 'stateEstimate.z': 0.2654609978199005 }
```

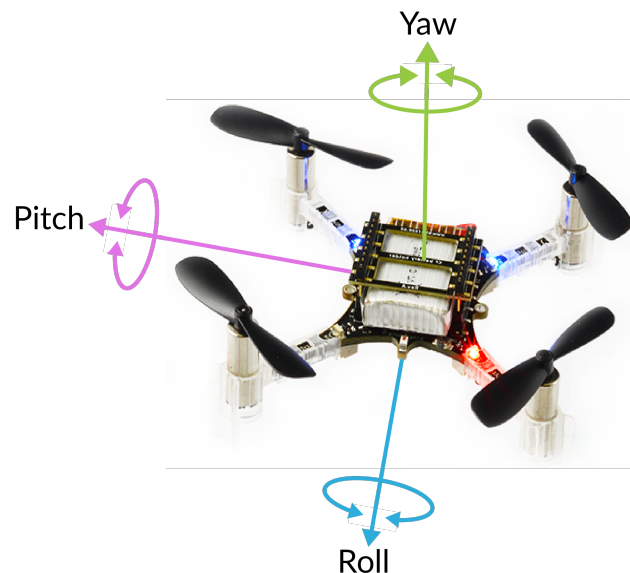
Gambar 3.5: Hasil pembacaan data sensor optical flow

Sama seperti Multi-ranger Deck, setelah Flow Deck terpasang pada Crazyflie, dilakukan kalibrasi dan pengujian terhadap pembacaan sensor sekaligus penentuan parameter dan kecepatan pembacaan data, seperti yang ditampilkan pada Gambar 3.5. Pembacaan data berupa estimasi posisi *drone* terhadap titik awal dijalankannya *drone* tersebut. 'X' merupakan koordinat horizontal bidang, 'Y' merupakan koordinat vertikal bidang, dan 'Z' merupakan koordinat ketinggian *drone* dari bidang. Ketiga parameter tersebut bergantung pada orientasi *drone* ketika pertama kali dijalankan.

```
{ 'stabilizer.roll': 0.06568686664104462, 'stabilizer.pitch': -6.206633567810059, 'stabilizer.yaw': -15.410162925720215 }
{ 'stabilizer.roll': -2.4728991985321045, 'stabilizer.pitch': -5.173481464385986, 'stabilizer.yaw': -16.535696029663086 }
{ 'stabilizer.roll': -3.3618521690368652, 'stabilizer.pitch': -3.922078847885132, 'stabilizer.yaw': -16.635404586791992 }
{ 'stabilizer.roll': -2.621164083480835, 'stabilizer.pitch': -4.737771511077881, 'stabilizer.yaw': -16.35302734375 }
{ 'stabilizer.roll': -2.9157464504241943, 'stabilizer.pitch': -3.4774255752563477, 'stabilizer.yaw': -15.827007293701172 }
{ 'stabilizer.roll': -4.4380388259887695, 'stabilizer.pitch': 0.024379048496484756, 'stabilizer.yaw': -16.828542709350586 }
{ 'stabilizer.roll': -4.881985664367676, 'stabilizer.pitch': -0.4813758432865143, 'stabilizer.yaw': -16.140605926513672 }
{ 'stabilizer.roll': -3.8034064769744873, 'stabilizer.pitch': -5.453186988830566, 'stabilizer.yaw': -15.587697982788086 }
{ 'stabilizer.roll': -1.598832368850708, 'stabilizer.pitch': -6.989500045776367, 'stabilizer.yaw': -14.72378158569336 }
{ 'stabilizer.roll': -2.367847442626953, 'stabilizer.pitch': -4.268555641174316, 'stabilizer.yaw': -13.916563987731934 }
{ 'stabilizer.roll': -2.6839656829833984, 'stabilizer.pitch': -0.5962711572647095, 'stabilizer.yaw': -13.930195808410645 }
{ 'stabilizer.roll': -2.3253397941589355, 'stabilizer.pitch': 3.1669344902038574, 'stabilizer.yaw': -14.05112075805664 }
{ 'stabilizer.roll': -2.0957090854644775, 'stabilizer.pitch': 5.254674911499023, 'stabilizer.yaw': -14.411182403564453 }
{ 'stabilizer.roll': -0.7546468838888888, 'stabilizer.pitch': 5.153007856453561, 'stabilizer.yaw': -14.834416848886403 }
```

Gambar 3.6: Hasil pembacaan data sensor IMU

Crazyflie 2.1 juga dilengkapi dengan IMU Sensor yang telah disebutkan sebelumnya pada subbab 2.2.2. Sensor ini berguna untuk mengidentifikasi orientasi dari Crazyflie ketika terbang. Hasil pembacaan data sensor IMU seperti yang ditampilkan pada Gambar 3.6. Sensor IMU telah tertanam pada *drone* Crazyflie itu sendiri, sehingga tidak perlu dilakukan kalibrasi dan hanya perlu melakukan pengujian terhadap pembacaan sensor. Hal ini bertujuan untuk melihat apakah sensor IMU yang ada pada Crazyflie dalam kondisi yang baik. Data yang ditampilkan dari sensor berupa informasi orientasi dari *drone*, berupa *roll*, *pitch*, dan *yaw*.



Gambar 3.7: Penggambaran arah pergerakan *roll*, *pitch*, dan *yaw*

Sumber : <https://www.bitcraze.io/documentation/tutorials/getting-started-with-crazyflie-2-x/>

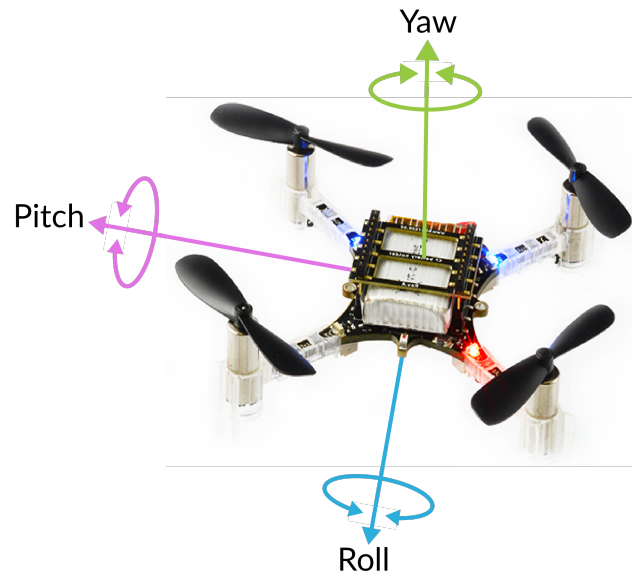
Berdasarkan Gambar 3.7, *roll* merupakan putaran dengan sumbu horizontal yang melalui bagian depan dan belakang *drone*. Hal ini menyebabkan *drone* melakukan putaran ke kiri atau kanan. *Pitch* merupakan putaran dengan sumbu horizontal yang melalui bagian kiri dan kanan *drone*. Hal ini menyebabkan *drone* melakukan putaran ke atas atau bawah. *Yaw* merupakan putaran dengan sumbu vertikal, untuk membelokkan *drone* ke kiri atau kanan. *Yaw* digunakan untuk mengubah arah kemana *drone* terbang.

3.1.2 Konfigurasi Perintah untuk Crazyflie

Konfigurasi perintah dilakukan menggunakan ROS (ROS2). Pertama-tama dilakukan penginstalan ROS2 dengan mengikuti panduan instalasi resmi. Setelah ROS2 terpasang diperlukan library khusus dari Crazyflie (cflib). Program untuk pemetaan menggunakan ROS2 pada

Crazyflie dapat menggunakan package yang tersedia pada *repository* milik Crazyflie. Dari *repository* ini dapat dikembangkan fitur pergerakan otomatis.

ROS 2 menjalankan perintah dari program dalam paket yang dibuat. Konten dari paket tersebut dapat dilihat pada Gambar 3.8.



Gambar 3.8: Penggambaran arah pergerakan *roll*, *pitch*, dan *yaw*

Sumber : <https://www.bitcraze.io/documentation/tutorials/getting-started-with-crazyflie-2-x/>

Konten pada paket ROS 2 yang dibuat terdiri dari `package.xml`, `setup.py`, `setup.cfg`, folder `nama_paket`, file `resource/nama_paket`, folder `'test'`, serta folder `'launch'`. File `package.xml` menyediakan beberapa informasi serta kebutuhan untuk paket. File `setup.py` berfungsi untuk memberi informasi apa yang harus diinstal, dimana menginstalnya, serta cara menyambungkan kebutuhan-kebutuhan lainnya ketika melakukan *compile* pada paket yang dibuat. File `setup.cfg` memberitahu dimana *script* program yang dibuat akan diinstal. Folder `nama_paket` merupakan tempat dimana pengguna membuat ROS 2 *script node* untuk dijalankan dalam paket. File `resource/nama_paket` diperlukan agar ROS 2 dapat menemukan paket yang telah dibuat. Folder `'test'` berisi 3 program Python yang berfungsi sebagai uji coba menjalankan paket, tidak akan dipakai pada paket yang memiliki *script* untuk dijalankan. Folder `'launch'` berisi file `'launch'` dalam format Python, fungsi dari file tersebut adalah agar ROS 2 dapat menjalankan beberapa *script node* sekaligus, dijalankan dengan perintah `'roslaunch'` pada terminal. Paket `'crazyflie_ros2_control'` seperti yang ditunjukkan pada 3.8 dibuat sebagai program untuk mengatur gerak otomatis pada *drone* yang dijalankan dengan ROS 2. Paket tersebut berfungsi sebagai kontrol gerak untuk melakukan navigasi dan pemetaan secara otomatis.

3.1.3 Penerbangan Crazyflie



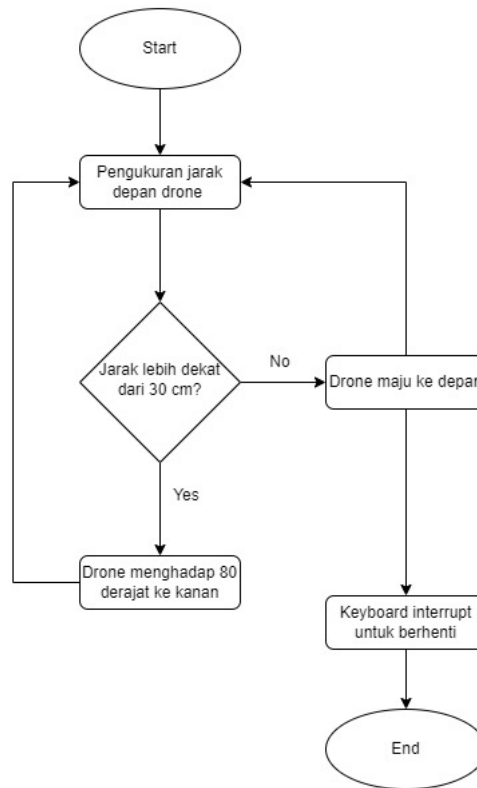
Gambar 3.9: Multi-ranger Deck yang terpasang pada Crazyflie



Gambar 3.10: Flow Deck yang terpasang pada Crazyflie

Sebelum diterbangkan, perlu dipastikan Multi-ranger Deck dan Flow Deck terpasang pada Crazyflie seperti pada Gambar 3.9 dan Gambar 3.10. Pada penerbangan Crazyflie untuk uji pemetaan secara otomatis akan dilakukan dua hal, yaitu perekaman data peta menggunakan Rviz dan pergerakan otomatis. Terdapat dua jenis pengujian yang akan dilakukan. Pengujian pertama akan menguji apakah program dapat berjalan dengan baik. Pengujian kedua berupa

lanjutan dari pengujian pertama, dimana hasil pengukuran dari navigasi dan pemetaan otomatis akan dibandingkan dengan hasil pengukuran manual.



Gambar 3.11: Flowchart sistem pergerakan otomatis

Berdasarkan Flowchart yang ditunjukkan pada Gambar 3.11, sistem pergerakan otomatis diawali dengan pengukuran jarak depan drone terhadap penghalang. Hal ini dilakukan dengan membaca sensor LiDAR bagian depan. Apabila dari sensor tersebut terbaca jarak yang lebih dekat daripada 30 cm, maka drone akan berputar ke arah kanan 80 derajat, kemudian akan dilakukan pengukuran jarak depan drone lagi. Apabila dari sensor tersebut terbaca jarak yang lebih jauh daripada 30 cm, maka drone akan maju kedepan. Selama maju ke depan drone tetap akan melakukan pengukuran jarak depan drone hingga mendapatkan jarak kurang dari 30 cm. Drone sewaktu-waktu dapat dinonaktifkan dengan menekan tombol interupsi pada terminal yang menjalankan program. Selama perintah dijalankan, drone juga akan melakukan proses pemetaan. Setelah proses pemetaan berakhir selanjutnya akan dilakukan komparasi hasil dari pergerakan manual dan pergerakan otomatis.

[Halaman ini sengaja dikosongkan]

4 PENGUJIAN DAN ANALISIS

4.1 Skenario Pengujian

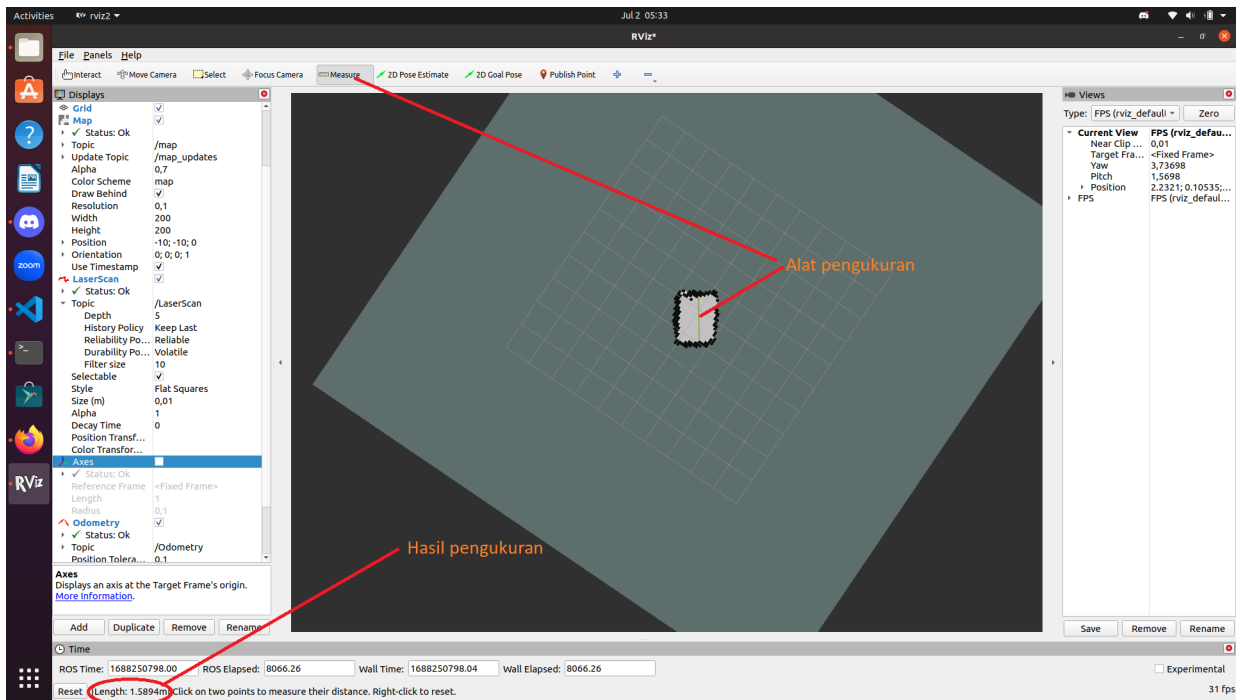
Pengujian dilakukan pada sebuah lingkungan uji buatan pada ruangan tertutup.



Gambar 4.1: Lingkungan buatan untuk pengujian

Seperti yang ditampakkan pada Gambar 4.1, lingkungan uji yang dibuat berupa area berbentuk persegi panjang berukuran $1.72 * 1.36$ meter persegi. Penutup area berupa properti padat yang tidak tembus pandang, dikarenakan sensor pada Multi-ranger Deck bertipe inframerah yang tidak bisa mendeteksi benda padat tembus pandang. Alas yang digunakan berupa matras. Matras digunakan untuk menutupi lantai, dikarenakan lantai memiliki sifat memantulkan bayangan. Sensor optical flow akan terganggu akurasi datanya apabila data yang ditangkap berupa pantulan bayangannya sendiri.

Pada lingkungan uji tersebut, drone akan diterbangkan sebanyak lima kali uji. Dari kelima uji tersebut akan dilihat akurasi dari hasil pemetaan terhadap lingkungan asli, dilakukan dengan mengukur panjang dan lebar area yang terpetakan. Untuk pengukuran dapat dilakukan pada Rviz yang dijalankan. Rviz memiliki *tools* untuk mengukur jarak dari sebuah area yang terpetakan, seperti yang ditunjukkan pada Gambar 4.2.



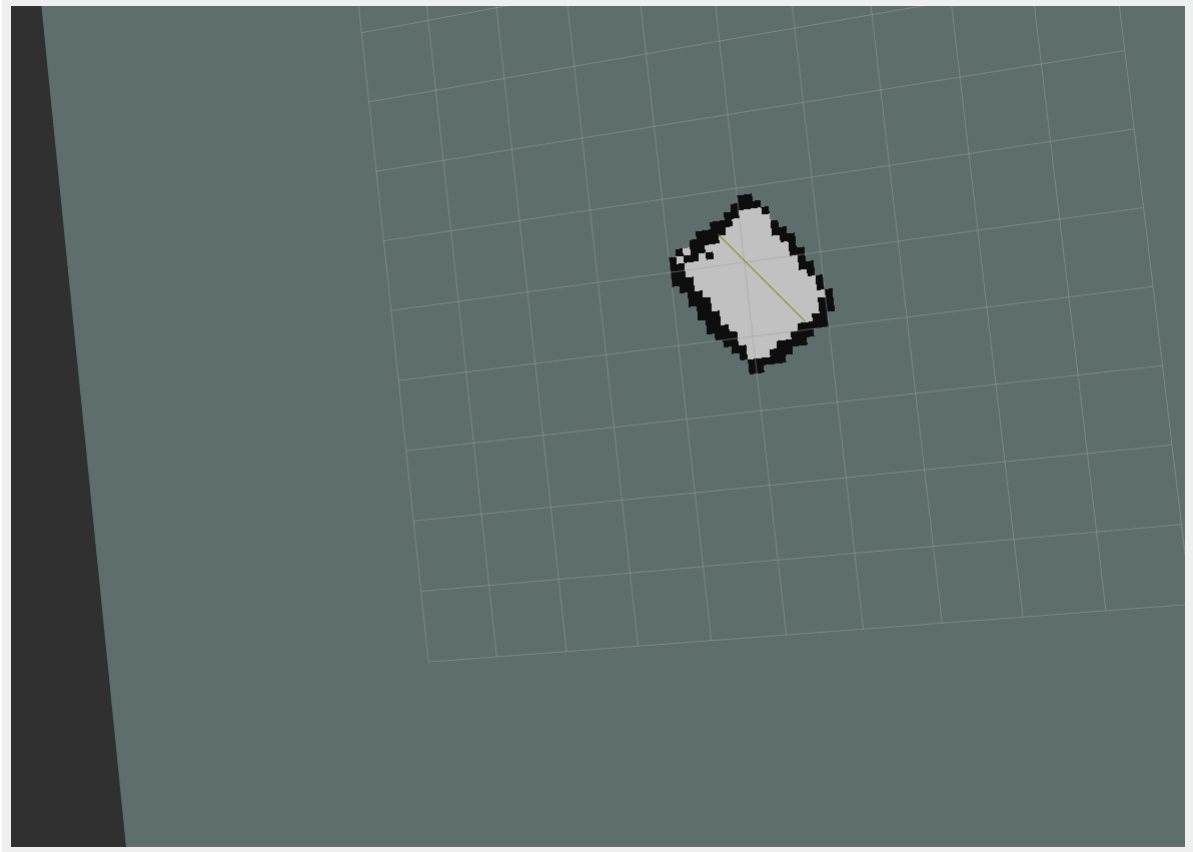
Gambar 4.2: Lingkungan buatan untuk pengujian



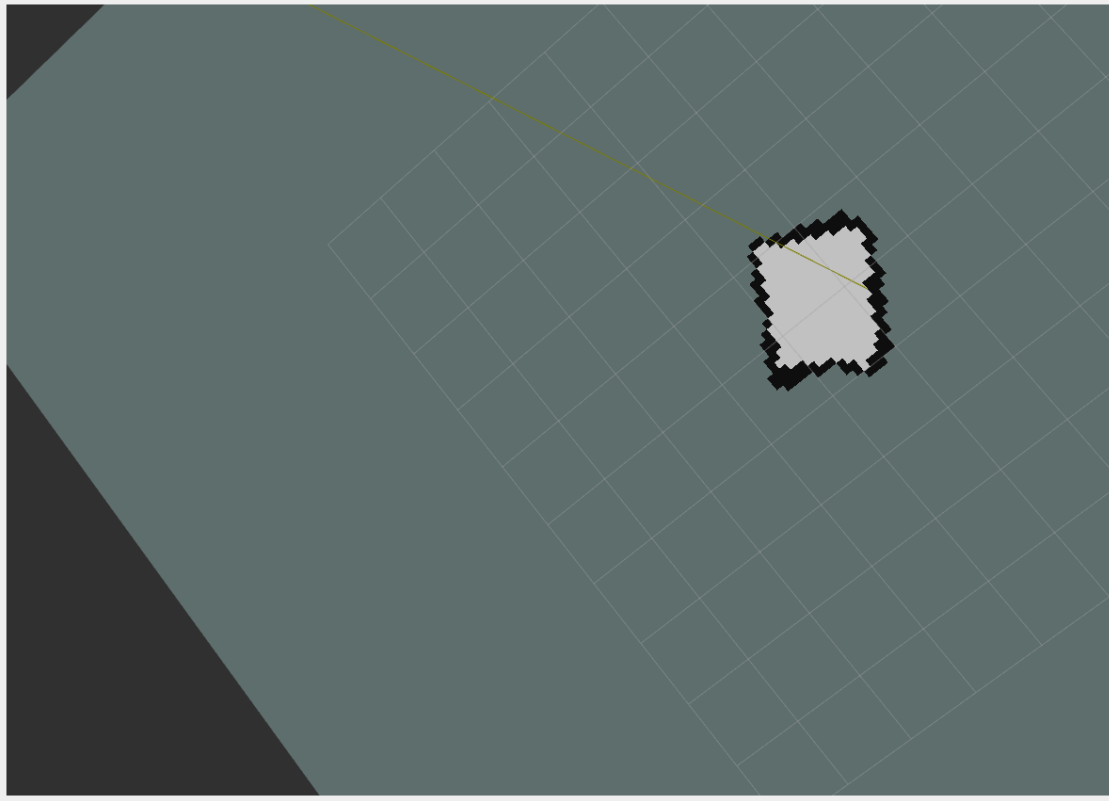
Gambar 4.3: Drone dalam keadaan diam pada lingkungan uji

4.2 Hasil Pengujian

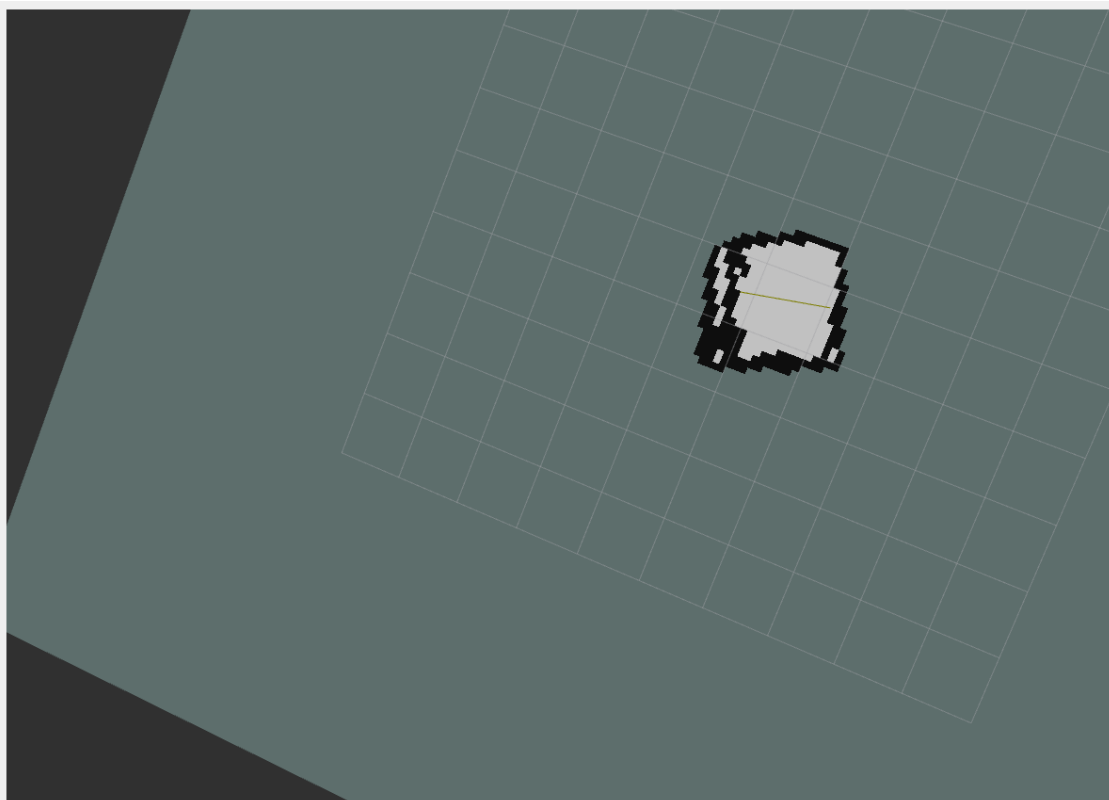
Dari penelitian yang dilakukan, Hasil data pemetaan dari pergerakan *autonomous* divisualisasikan menggunakan Rviz. Visual peta yang dihasilkan terdiri dari blok hitam dan putih. Blok hitam merupakan atau penghalang, sedangkan blok putih merupakan area bebas tanpa penghalang.



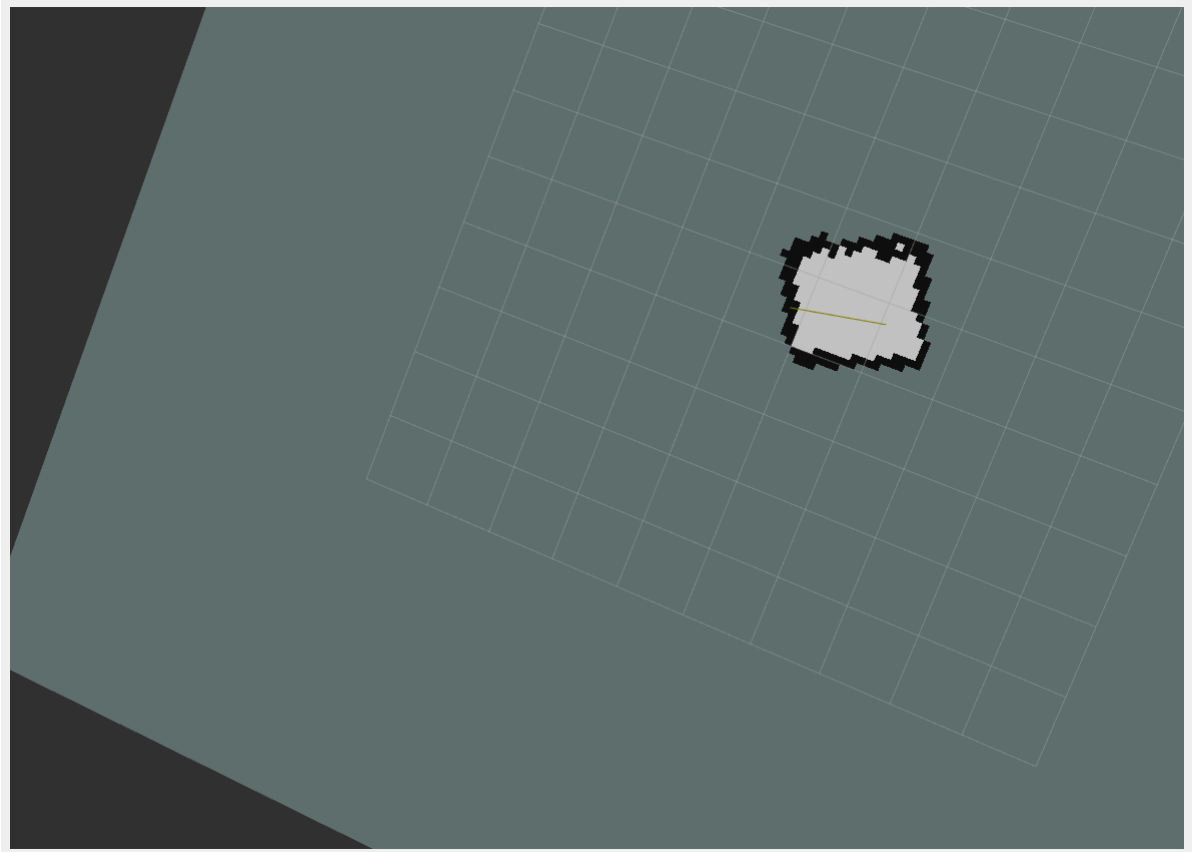
Gambar 4.4: Hasil pemetaan Rviz pada pengujian pertama



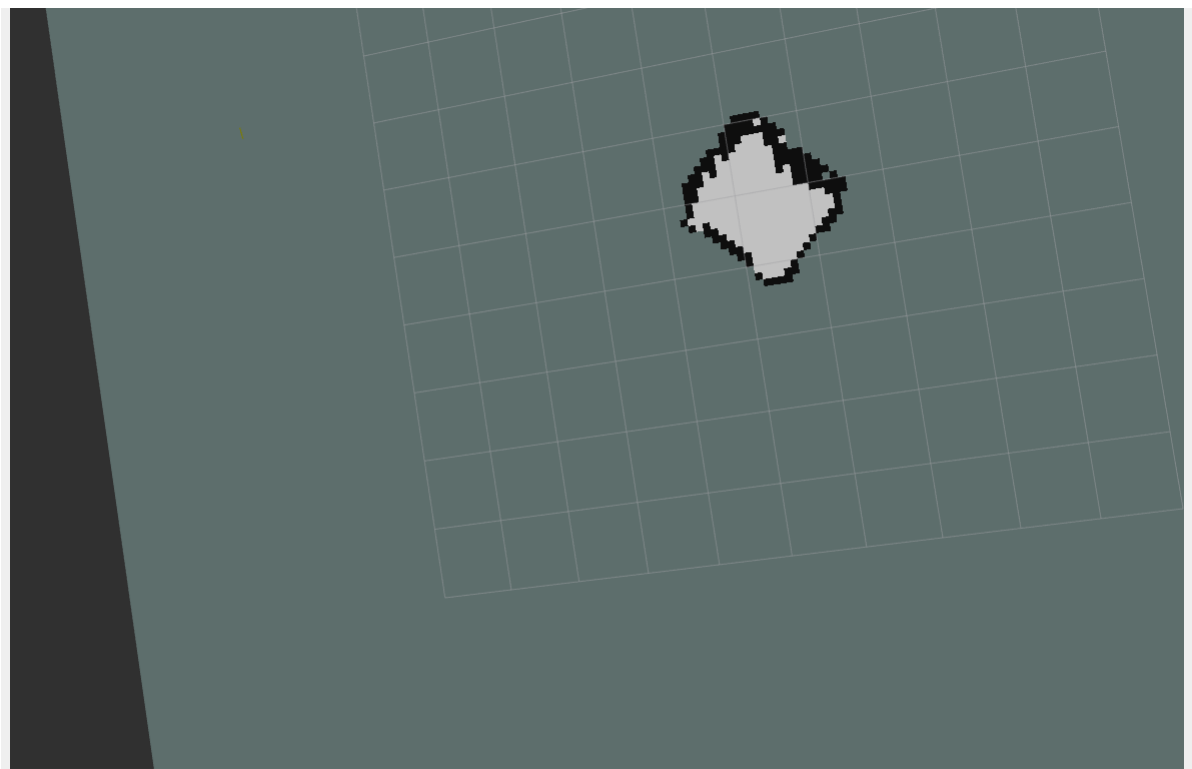
Gambar 4.5: Hasil pemetaan Rviz pada pengujian kedua



Gambar 4.6: Hasil pemetaan Rviz pada pengujian ketiga



Gambar 4.7: Hasil pemetaan Rviz pada pengujian keempat



Gambar 4.8: Hasil pemetaan Rviz pada pengujian kelima

Dari kelima pengujian, dibuat tabel hasil pengukuran area uji dari masing-masing pengujian. Tingkat kedetailan pada pengukuran area uji hingga tiga angka dibelakang koma (meter).

Tabel 4.1: Tabel ukuran dari hasil pengujian

Pengujian	Panjang	Lebar
Pertama (Gambar 4.4)	1.656 meter	1.293 meter
Kedua (Gambar 4.5)	1.669 meter	1.304 meter
Ketiga (Gambar 4.6)	1.686 meter	1.297 meter
Keempat (Gambar 4.7)	1.599 meter	1.384 meter
Kelima (Gambar 4.8)	1.715 meter	1.299 meter
Rata-rata	1.665 meter	1.315 meter

4.3 Analisis Pengujian

Pada pengujian pertama, didapatkan panjang area yang terukur pada Rviz 1.656 meter, sedangkan lebar area yang terukur pada Rviz 1.293 meter. Apabila dibandingkan dengan panjang dan lebar area yang diukur secara manual, panjang area memiliki selisih 0.064 meter, sedangkan lebar area memiliki selisih 0.067 meter.

Pada pengujian kedua, didapatkan panjang area yang terukur pada Rviz 1.669 meter, sedangkan lebar area yang terukur pada Rviz 1.304 meter. Apabila dibandingkan dengan panjang dan lebar area yang diukur secara manual, panjang area memiliki selisih 0.051 meter, sedangkan lebar area memiliki selisih 0.056 meter.

Pada pengujian pertama, didapatkan panjang area yang terukur pada Rviz 1.686 meter, sedangkan lebar area yang terukur pada Rviz 1.297 meter. Apabila dibandingkan dengan panjang dan lebar area yang diukur secara manual, panjang area memiliki selisih 0.034 meter, sedangkan lebar area memiliki selisih 0.063 meter.

Pada pengujian pertama, didapatkan panjang area yang terukur pada Rviz 1.599 meter, sedangkan lebar area yang terukur pada Rviz 1.384 meter. Apabila dibandingkan dengan panjang dan lebar area yang diukur secara manual, panjang area memiliki selisih 0,124 meter, sedangkan lebar area memiliki selisih -0.024 meter.

Pada pengujian pertama, didapatkan panjang area yang terukur pada Rviz 1.715 meter, sedangkan lebar area yang terukur pada Rviz 1.299 meter. Apabila dibandingkan dengan panjang dan lebar area yang diukur secara manual, panjang area memiliki selisih 0.005 meter, sedangkan lebar area memiliki selisih 0.061 meter.

5 DAFTAR PUSTAKA

- Alamsyah, S. A., & Rivai, M. (2020). Implementasi lidar sebagai kontrol ketinggian quadcopter. *Jurnal Teknik ITS (SINTA: 4, IF: 1.1815)*, 8(2), A109–A114.
- Alfaeru, F., & Setiawan, A. B. (2016). Implementasi sensor akselerometer dan modul gps untuk rancang bangun smartbike. *Jurnal Ilmu-Ilmu Teknik-Sistem*, 12(3).
- Bitcraze. (2022). Crazyflie 2.1. Retrieved October 27, 2022, from <https://www.bitcraze.io/products/crazyflie-2-1/>
- Bolognini, M., & Fagiano, L. (2020). Lidar-based navigation of tethered drone formations in an unknown environment. *IFAC-PapersOnLine*, 53(2), 9426–9431.
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., & Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6), 1309–1332. <https://doi.org/10.1109/TRO.2016.2624754>
- Dijkshoorn, N. (2012). Simultaneous localization and mapping with the ar. drone. *PhD diss., Masters thesis, Universiteit van Amsterdam*.
- Giernacki, W., Skwierczyński, M., Witwicki, W., Wroński, P., & Koziński, P. (2017). Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering. *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, 37–42. <https://doi.org/10.1109/MMAR.2017.8046794>
- Hönig, W., & Ayanian, N. (2017). Flying multiple uavs using ros. *Robot Operating System (ROS) The Complete Reference (Volume 2)*, 83–118. https://doi.org/10.1007/978-3-319-54927-9_3
- Jalil, A. (2022). *Panduan lengkap robot operating system (ros)*. Penerbit Andi.
- Julius, F., Putra, A. S., & Tjahyadi, H. (2019). Desain dan implementasi path following drone. *Jurnal Elektro*, 12(1), 1–10.
- Kam, H. R., Lee, S.-H., Park, T., & Kim, C.-H. (2015). Rviz: A toolkit for real domain data visualization. *Telecommunication Systems*, 60, 337–345.
- Liani, D. (2017). Sistem navigasi pada mobile robot dengan menggunakan gps (global positioning system).
- Magnabosco, M., & Breckon, T. P. (2013). Cross-spectral visual simultaneous localization and mapping (slam) with sensor handover. *Robotics and Autonomous Systems*, 61(2), 195–208. <https://doi.org/10.1016/j.robot.2012.09.023>
- Mazzeo, G., & Staffa, M. (2020). Tros: Protecting humanoids ros from privileged attackers. *International Journal of Social Robotics*, 12, 827–841.
- Menhub. (2015). Peraturan menteri perhubungan tentang pengendalian pengoperasian pesawat udara tanpa awak di ruang udara yang dilayani indonesia. Retrieved October 27, 2022, from <https://peraturan.bpk.go.id/Home/Details/103485/permenhub-no-90-tahun-2015>
- Pamungkas, D. B., Agustinah, T., & AK, R. E. (2017). Navigasi mobile robot nonholonomic menggunakan fuzzy-ant colony system. *Jurnal Teknik ITS*, 6(2), A839–A844.
- Park, J., Delgado, R., & Choi, B. W. (2020). Real-time characteristics of ros 2.0 in multiagent robot systems: An empirical study. *IEEE Access*, 8, 154637–154651. <https://doi.org/10.1109/ACCESS.2020.3018122>
- Quang, H. D., Manh, T. N., Manh, C. N., Tien, D. P., Van, M. T., Kim, D. H. T., Thanh, V. N. T., & Duan, D. H. (2019). Mapping and navigation with four-wheeled omnidirectional mobile robot based on robot operating system. *2019 International Conference on Mecha-*

tronics, Robotics and Systems Engineering (MoRSE), 54–59. <https://doi.org/10.1109/MoRSE48060.2019.8998714>

Rahman, A. (2016). Pemodelan robot beroda menggunakan umpan balik giroskop. *Jurnal Rekayasa Elektrika*, 12(2), 67–72. <https://doi.org/10.13170/aijst.v25i.4903>

Raj, T., Hanim Hashim, F., Baseri Huddin, A., Ibrahim, M. F., & Hussain, A. (2020). A survey on lidar scanning mechanisms. *Electronics*, 9(5), 741.

Robertson, P., Angermann, M., & Krach, B. (2009). Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors, 93–96. <https://doi.org/10.1145/1620545.1620560>

Wandinger, U. (2005). Introduction to lidar. In *Lidar: Range-resolved optical remote sensing of the atmosphere* (pp. 1–18). Springer.