

TUGAS AKHIR - VE190834

PENERAPAN HAVERSINE FORMULA DALAM NAVIGASI WAYPOINT PADA PACKAGE DELIVERY QUADCOPTER

MAULANA ZUHRI

NRP 2040221159

Dosen Pembimbing

Ir. Joko Susila, MT.

NIP 196606061991021001

Program Studi Teknologi Rekayasa Otomasi

Departemen Teknik Elektro Otomasi

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Surabaya

2023

“ Halaman Ini Sengaja Dikosongkan “



TUGAS AKHIR - VE190834

PENERAPAN HAVERSINE FORMULA DALAM NAVIGASI WAYPOINT PADA PACKAGE DELIVERY QUADCOPTER

MAULANA ZUHRI

NRP 2040221159

Dosen Pembimbing

Ir. Joko Susila, MT

NIP 196606061991021001

Program Studi Teknologi Rekayasa Otomasi

Departemen Teknik Elektro Otomasi

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Surabaya

2023

“ Halaman Ini Sengaja Dikosongkan “



FINAL PROJECT - VE190834

APPLICATION OF THE HAVERSINE FORMULA IN WAYPOINT NAVIGATION FOR PACKAGE DELIVERY QUADCOPTER

MAULANA ZUHRI

NRP 2040221159

Advisor

Ir. Joko Susila, MT.

NIP 196606061991021001

Study Program Automation Engineering

Electrical Automation Engineering Department

Vocational Faculty

Institut Teknologi Sepuluh Nopember

Surabaya

2023

“ Halaman Ini Sengaja Dikosongkan “

LEMBAR PENGESAHAN

PENERAPAN *Haversine Formula* DALAM NAVIGASI *Waypoint* PADA *Package Delivery Quadcopter*

PROYEK AKHIR

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Terapan Teknik pada
Program Studi D-4 Teknologi Rekayasa Otomasi
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh : MAULANA ZUHRI
NRP. 2040221159

Disetujui oleh Tim Penguji Proyek Akhir:

1. Ir. Joko Susila, MT.
2. Ir. Arif Musthofa, MT.
3. Enny Indasyah, S.ST, MT., M.Sc.
4. Fauzi Imaduddin Adhim, S.ST., MT.



Pembimbing

Penguji

Penguji

Penguji

SURABAYA
Juli , 2023

“ Halaman Ini Sengaja Dikosongkan “

APPROVAL SHEET

APPLICATION OF THE HAVERSINE FORMULA IN WAYPOINT NAVIGATION
FOR PACKAGE DELIVERY QUADCOPTER

FINAL PROJECT

Submitted to fulfill one of the requirements
for obtaining a Bachelor of Applied Engineering degree at
Undergraduate Study Program of Automation Engineering Technology
Departemen of Electrical Automation Engineering
Faculty of Vocation
Institut Teknologi Sepuluh Nopember

By : MAULANA ZUHRI
NRP. 2040221159

Approved by Final Project Examiner Team :

1. Ir. Joko Susila, MT.
2. Ir. Arif Musthofa, MT.
3. Enny Indasyah, S.ST, MT., M.Sc.
4. Fauzi Imaduddin Adhim, S.ST., MT.



Advisor

Examiner

Examiner

Examiner

[Handwritten signatures and initials of the examiners and advisor]

SURABAYA
July , 2023

“ Halaman Ini Sengaja Dikosongkan “

PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini :

Nama Mahasiswa / NRP : Maulana Zuhri / 2040221159
Program Studi : Teknik Elektro Otomasi
Dosen Pembimbing / NIP : Ir. Joko Susila, M.T./
NIP. 196606061991021001

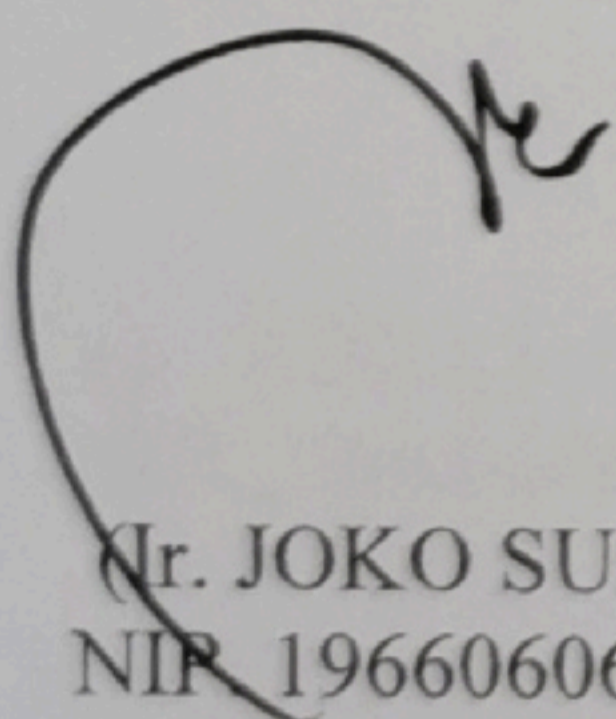
Dengan ini menyatakan bahwa Proyek Akhir dengan judul "Penerapan Haversine Formula Dalam Navigasi Waypoint Pada Package Delivery Quadcopter" adalah hasil karya sendiri dan ditulis dengan mengikuti kaidah penulisan ilmiah.

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

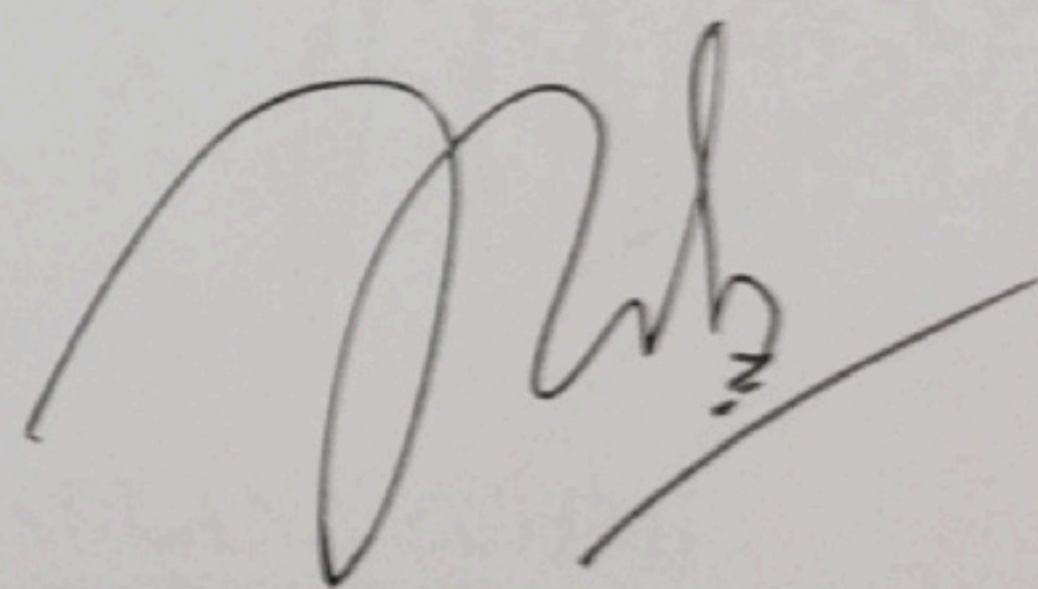
Surabaya, 06 Juli 2023

Mengetahui,
Dosen Pembimbing

Mahasiswa



(Ir. JOKO SUSILA, MT.)
NIP. 196606061991021001



(MAULANA ZUHRI)
NRP 2040221159

“ Halaman Ini Sengaja Dikosongkan “

STATEMENT OF ORIGINALITY

The undersigned below :

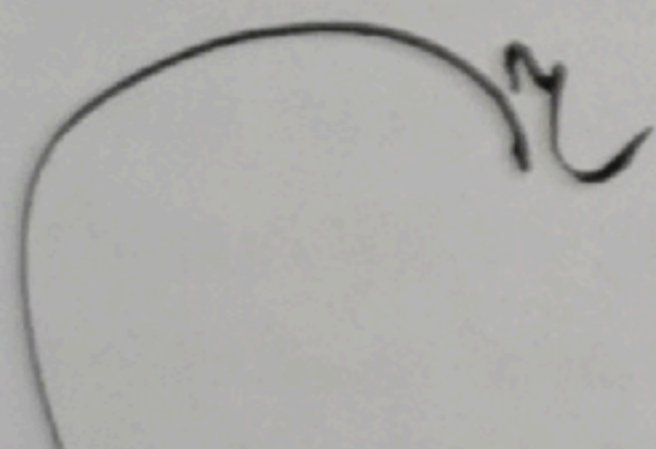
Name of student / NRP : Maulana Zuhri / 2040221159
Departement : Teknik Elektro Otomasi
Advisor / NIP : Ir. Joko Susila, M.T./
NIP. 196606061991021001

Hereby declare that the final project with the title of "Application of The Haversine Formula in Waypoint Navigation for Package Delivery Quadcopter" is the result of my own work, is original, and is written by following the rules of scientific writing. If in the future there is a discrepancy with this statement then I am willing to accept sanctions in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember

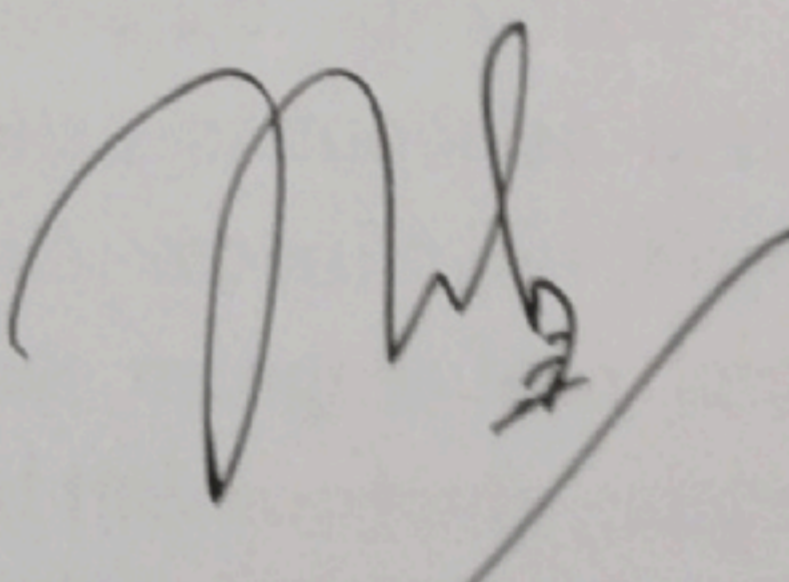
Surabaya, 06 July 2023

Verified by,
Advisor

Student



(Ir. JOKO SUSILA, MT.)
NIP. 196606061991021001



(MAULANA ZUHRI)
NRP 2040221159

“ Halaman Ini Sengaja Dikosongkan “

PENERAPAN HAVERSINE FORMULA DALAM NAVIGASI WAYPOINT PADA PACKAGE DELIVERY QUADCOPTER

Nama Mahasiswa / NRP : Maulana Zuhri / 2040221159
Departemen : Teknik Elektro Otomasi
Dosen Pembimbing : Ir. Joko Susila, M.T.

Abstrak

Seiring dengan berkembangnya teknologi dalam bidang *unmanned air vehicle*, banyak inovasi baru yang dibuat salah satunya adalah *quadcopter* yang digunakan untuk mengirim atau mengantarkan paket. *Quadcopter* adalah sebuah alat yang dapat terbang menggunakan 4 baling baling atau propeller. Karena cara terbangnya yang tidak memerlukan tempat yang besar, *quadcopter* dapat menjangkau tempat-tempat yang sempit. Dengan demikian aplikasi *quadcopter* untuk mengirimkan barang menjadi memungkinkan. Koordinat lintang dan bujur dijadikan acuan pada proses navigasi berlangsung, sebuah *quadcopter* dilengkapi dengan GPS yang digunakan sebagai sinyal umpan balik agar *quadcopter* dapat mengerti posisi terbang sekarang.

Proses berjalannya navigasi *waypoint* dilakukan dengan cara menghitung *remaining distance* antara *quadcopter* yang sedang terbang dengan koordinat pada *waypoint* yang digunakan sebagai *setpoint*. Perhitungan *remaining distance* dilakukan dengan menggunakan *haversine formula*. Rumus *haversine* merupakan permodelan matematika untuk menghitung jarak diantara dua titik pada permukaan bola, seperti bumi. Formula ini menggunakan garis lintang dan bujur dari kedua titik untuk menghasilkan jarak pada permukaan bola. Kemudian pada program diberikan radius dari titik *waypoint* yang menjadi area konfirmasi jika *waypoint* telah tercapai. Pada proyek akhir ini *haversine formula* digunakan dalam navigasi *waypoint* pada package delivery *quadcopter*.

Penerapan *haversine formula* dalam navigasi *waypoint* digunakan untuk menggantikan pengoperasian *quadcopter* secara konvensional menggunakan remote. Dengan memberikan urutan *setpoint* berupa koordinat lintang, bujur dan ketinggian (*waypoint*), *quadcopter* dapat terbang secara terprogram sesuai dengan kehendak operator. Berdasarkan pengujian perhitungan *haversine formula* mendapatkan nilai error 1.78% jika dibandingkan dengan hasil pengukuran menggunakan pita ukur tanah. *Waypoint radius* yang presisi ialah 1 meter. Maksimum *payload* yang dapat dibawa oleh *quadcopter* adalah sebesar 700 gram.

Kata Kunci : *Quadcopter, Waypoint, Payload, Haversine Formula*

“ Halaman Ini Sengaja Dikosongkan “

WAYPOINT NAVIGATION METHOD FOR PACKAGE DELIVERY QUADCOPTER

Student Name / NRP : Maulana Zuhri /2040221159
Departement : Teknik Elektro Otomasi
Advisor : Ir. Joko Susila, MT.

Abstract

The use of quadcopters for delivering packages has indeed revolutionized the field of unmanned air vehicles. With their ability to navigate small spaces and reach narrow areas, quadcopters have become a popular choice for transporting goods. During the navigation process, quadcopters rely on latitude and longitude coordinates as references. This allows them to calculate the remaining distance between their current position and the set waypoints using the Haversine formula. By using GPS as a feedback signal, quadcopters can accurately understand their flying position.

Waypoint navigation is a key aspect of quadcopter delivery. Instead of relying on remote control, operators can program a sequence of setpoints (latitude, longitude, and altitude) for the quadcopter to follow. This enables a programmed and automated flight according to the operator's preferences. This is particularly useful when operating a quadcopter from a long distance or in challenging conditions such as poor visibility or nighttime.

In this final project, the quadcopter is specifically designed to deliver goods from location A to location B. It has a maximum payload limit of 1 kg, which means it can carry packages weighing up to 1 kg during the delivery process. This project showcases the potential of quadcopters in revolutionizing the way goods are transported.

Keywords: *Quadcopter, Waypoint, Payload, Haversine Formula*

“ Halaman Ini Sengaja Dikosongkan “

KATA PENGANTAR

Puji syukur kami panjatkan kepada Allah SWT atas limpahan rahmat dan hidayah dari-Nya, sehingga saya dapat menyelesaikan Proyek Akhir ini dengan baik, begitu pula pembuatan buku Proyek Akhir ini. Proyek Akhir ini dilakukan untuk memenuhi satuan kredit semester (SKS) yang harus ditempuh sebagai persyaratan akademis di Departemen Teknik Elektro Otomasi Fakultas Vokasi Institut Sepuluh Nopember Surabaya untuk menyelesaikan program pendidikan Sarjana Terapan di Departemen Teknik Elektro Otomasi dengan judul “ METODE NAVIGASI WAYPOINT PADA UAV QUADCOPTER PENGIRIM BARANG”.

Dengan terselesainya Proyek Akhir ini saya mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Orang tua dan saudara yang senantiasa mendoakan dan memberikan dukungan dalam bentuk moril dan materi dengan tulus tiada henti.
2. Bapak Imam Arifin, ST., MT. selaku Kepala Departemen Teknik Elektro Otomasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya.
3. Ir. Joko Susila, M.T. selaku dosen pembimbing yang telah banyak memberikan masukan ilmu, waktu dan pengarahan dalam penyusunan Proyek Akhir ini.
4. Teman – teman RPL angkatan 2022 dan juga angkatan 2019 D4 Teknik Rekayasa Otomasi yang selalu memberikandoa, bantuan, semangat, dan dukungannya.
5. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Proyek Akhir ini.

Saya menyadari dan memohon maaf atas segala kekurangan pada Proyek Akhir ini. Akhirk kata, semoga Proyek Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari

Surabaya, 06 Juli 2023

Penulis

“ Halaman Ini Sengaja Dikosongkan “

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
APPROVAL SHEET	iii
PERNYATAAN ORISINALITAS	v
STATEMENT OF ORIGINALITY	vii
Abstrak	ix
Abstract	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxiii
BAB I	
PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah.....	2
1.5 Manfaat	2
BAB II	
TINJAUAN PUSTAKA.....	3
2.1 Penelitian Terdahulu	3
2.2 Gambaran Umum UAV(<i>Unmanned Aerial Vehicle</i>)	4
2.3 <i>Quadcopter</i>	6
2.4 Komponen Penyusun Quadcopter.....	6
2.4.1 <i>Frame</i>	6
2.4.2 <i>Flight Controller</i>	7
2.4.3 <i>Brushless DC Motor (BE4108-380KV)</i>	8
2.4.4 <i>Electronic Speed Control (ESC)</i>	8
2.4.5 <i>Propeller</i>	9
2.5 Navigasi Waypoint.....	9
2.6 <i>Global Positioning System (GPS)</i>	10
2.6 <i>Attitude Heading Reference System (AHRS)</i>	11
2.6 Protokol Komunikasi.....	12
2.7 <i>Harvesine Formula</i>	12
2.8 <i>Euclidean Distance</i>	13

2.9	<i>Manhattan Distance</i>	14
2.10	<i>Thrust Brushless DC Motor</i>	15
BAB III		
METODOLOGI		
3.1	Gambaran Umum Sistem	17
3.2	Desain Dan Konsep Sistem	17
3.3	Blok Diagram Proses Keseluruhan	17
3.3	Perancangan Komponen Pada Quadcopter	18
3.3.1	<i>Brushless DC Motor Dan Propeller</i>	18
3.3.2	<i>Hobbywing XRotor 40A (Electronic Speed Controller)</i>	20
3.3.3	<i>Pixhawk 6C (flight controller)</i>	21
3.3.4	<i>M8N (GPS Sensor)</i>	22
3.3.5	<i>Raspberry Pi 4B (flight computer)</i>	24
3.3.6	<i>Baterai Li-Po 6S 5000mAH</i>	25
3.3.7	<i>Tarot X Frame 650mm</i>	25
3.4	Perakitan Komponen Pada Frame	26
3.5	Perancangan Perangkat Lunak Navigasi <i>Waypoint</i>	27
3.6	<i>Wiring</i> Pada Komponen	33
3.6.1	<i>Power Wiring Diagram</i>	33
3.6.2	<i>Signal Wiring Diagram</i>	33
3.7	Perhitungan Jarak Menggunakan <i>Haversine Formula</i>	34
3.9	Perhitungan Jarak Menggunakan <i>Manhattan Distance</i>	36
3.10	Integrasi <i>Flight Computer</i> dengan <i>Flight Controller</i>	36
3.11	Kalkulasi <i>Thrust</i> Yang Diperlukan	37
3.12	Pengambilan Data Pada <i>Pixhawk 6C</i>	38
3.12.1	Pengambilan Data GPS	39
3.12.2	Pengambilan Data Sensor IMU	39
3.12.3	Pengambilan Data Ketinggian	40
3.12.4	Pengambilan Data Kecepatan	40
BAB IV		
HASIL DAN PEMBAHASAN		
4.1	Perbandingan Perhitungan <i>Haversine Formula, Euclidean Distance, dan Manhattan Distance</i>	43
4.2	Pengujian <i>Haversine Formula</i> Pada Navigasi <i>Waypoint</i>	45
4.3	Pengujian Tanpa <i>Payload</i> Dan Menggunakan <i>Payload</i>	49
4.3.1	Pengujian Tanpa Menggunakan <i>Payload</i>	49

4.3.2	Pengujian Payload 385 Gram	50
4.3.3	Pengujian Payload 708 Gram	53
BAB V		
PENUTUP		57
5.1	Kesimpulan	57
5.2	Saran	57
DAFTAR PUSTAKA		59
LAMPIRAN		61
Lampiran 1 <i>Data Sheet Motor</i>		61
Lampiran 2 <i>Manual Installation Instruction</i>		62
BIODATA MAHASISWA		63

“ Halaman Ini Sengaja Dikosongkan “

DAFTAR GAMBAR

Gambar 2. 1	Grafik Historical Waypoint Dan Jalur Terbang	3
Gambar 2. 2	Fixed Wing UAV	5
Gambar 2. 3	Single Rotor UAV	5
Gambar 2. 4	Multi Rotor Quadcopter	6
Gambar 2. 5	Frame X Pada Quadcopter.....	7
Gambar 2. 6	Flight Controller Pixhawk 6C	7
Gambar 2. 7	BLDC motor.....	8
Gambar 2. 8	Electronic Speed Control (ESC) 40A.....	8
Gambar 2. 9	Propeller	9
Gambar 2. 10	Propeller Dengan Tipe Clockwise dan Counter Clockwise	9
Gambar 2. 11	Fly-by dan Flyover pada Waypoint.....	10
Gambar 2. 12	GPS Berdasarkan Posisi Satelit.....	10
Gambar 2. 13	Sudut Roll Pitch Dan Yaw	11
Gambar 2. 14	Interface Pada Sistem AHRS	11
Gambar 2. 15	Jarak Terdekat Dari 2 Koordinat Pada Bumi	12
Gambar 2. 16	Perhitungan Titik A ke Titik D Menggunakan Haversine Formula	13
Gambar 2. 17	Dua Titik Koordinat A dan B	14
Gambar 2. 18	Titik O Sebagai Titik Bantu Imajiner	14
Gambar 2. 19	Perbedaan Euclidean Distance dengan Manhattan Distance.....	15
Gambar 2. 20	Quadcopter Thrust.....	15
Gambar 2. 21	Gerakan Pada Quadcopter	16
Gambar 3. 1	Blok Diagram Sistem Arsitektur	17
Gambar 3. 2	Blok Diagram Proses.....	18
Gambar 3. 3	(a) INRUNNER BLDC MOTOR (b) OUTRUNNER BLDC MOTOR.....	18
Gambar 3. 4	Instalasi Propeller Pada Motor	19
Gambar 3. 5	(a) 4108-380KV BLDC MOTOR, (b) 13x55" Propeller.....	19
Gambar 3. 6	Driver Motor BLDC Hobbywing XRotor 40A	20
Gambar 3. 7	Pixhawk 6C	21
Gambar 3. 8	Arsitektur Komponen Penyusun Pixhawk 6C.....	22
Gambar 3. 9	M8N GPS Sensor	22
Gambar 3. 10	Visualisasi Data Pembacaan Sensor GPS.....	23
Gambar 3. 11	Pengujian Keakuratan Pembacaan Koordinat Pada Sensor GPS	24

Gambar 3. 12 Raspberry Pi 4B.....	24
Gambar 3. 13 Baterai Li-Po 6S 5000mAH 70C.....	25
Gambar 3. 14 Tarot X Frame 650mm.....	25
Gambar 3. 15 Frame Tarot 650mm dan Instalasi Motor pada Frame	26
Gambar 3. 16 Instalasi Baling-baling dan ESC.....	26
Gambar 3. 17 Perakitan Komponen Pada Frame	27
Gambar 3. 18 Contoh Syntax Komunikasi Dari Raspberry Menuju Pixhawk.....	27
Gambar 3. 19 Fungsi Arm And Takeoff.....	28
Gambar 3. 20 Fungsi Heading Control	29
Gambar 3. 21 Syntax Posisi Sekarang Dan List Target Koordinat	29
Gambar 3. 22 Perintah Terbang Menuju Waypoint	30
Gambar 3. 23 Perhitungan Haversine Formula Pada Bahasa Pemrograman Python	30
Gambar 3. 24 Urutan Sequence Program Pada Quadcopter	31
Gambar 3. 25 Flowchart Algoritma Waypoint, (a) Flowchat Program Utama, (b) Flowchart Perhitungan Menggunakan Haversine Formula	32
Gambar 3. 26 Power Wiring Diagram.....	33
Gambar 3. 27 Signal Wiring Diagram.....	34
Gambar 3. 28 Komunikasi Antara Raspberry Pi 4 Dengan Pixhawk Melalui Komunikasi Serial ..	37
Gambar 3. 29 Pengukuran Berat Quadcopter Menggunakan Timbangan.....	38
Gambar 3. 30 Visualisasi Data Historical GPS Menggunakan Matlab.....	39
Gambar 3. 31 Visualisasi Data Imu Menjadi Atitude Menggunakan Matlab	39
Gambar 3. 32 Visualisasi Data Altitude Menggunakan Matlab.....	40
Gambar 3. 33 Visualisasi Data Kecepatan Sumbu Z Dengan Groundspeed.....	41
Gambar 4. 1 Hasil Plotting Data Dari Percobaan Pengujian Haversine Formula	44
Gambar 4. 2 Hasil Plotting Data Dari Percobaan Pengujian Euclidean Distance.....	44
Gambar 4. 3 Hasil Plotting Data Dari Percobaan Pengujian Manhattan Distance.....	45
Gambar 4. 4 Pengujian Radius 5m Pada Waypoint	46
Gambar 4. 5 Kepresisian Landing Pada Waypoint Radius 5m	46
Gambar 4. 6 Pengujian Radius 2m Pada Waypoint	47
Gambar 4. 7 Kepresisian Landing Pada Waypoint Radius 2m	47
Gambar 4. 8 Pengujian Radius 1m Pada Waypoint	48
Gambar 4. 9 Kepresisian Landing Pada Waypoint Radius 1m	48
Gambar 4. 10 Path History Pengujian Tanpa Payload	49
Gambar 4. 11 Hubungan Altitude Dan Voltase Baterai Tanpa Payload	49
Gambar 4. 12 Hubungan Altitude Dan Throttle PWM Tanpa Payload	50

Gambar 4. 13 Path History Payload 385 Gram	50
Gambar 4. 14 Payload 385 Gram	51
Gambar 4. 15 Hubungan Altitude Dan Voltase Baterai Dengan Payload 385 Gram.....	51
Gambar 4. 16 3D Trajectory	52
Gambar 4. 17 Hubungan Altitude (m) dan Throttle PWM	52
Gambar 4. 18 Path History Payload 708 Gram	53
Gambar 4. 19 Payload 708 Gram	53
Gambar 4. 20 Hubungan Altitude Dan Voltase Baterai Dengan Payload 708 Gram.....	54
Gambar 4. 21 3D Trajectory	54
Gambar 4. 22 Hubungan Altitude (m) dan Throttle PWM	55

“ Halaman Ini Sengaja Dikosongkan “

DAFTAR TABEL

Tabel 3. 1 Data Pengujian Motor Yang Telah Terpasang Propeller	20
Tabel 3. 2 Data Pembacaan Sensor GPS Berupa Koordinat (latitude, longitude).....	23
Tabel 4. 1 Perbandingan Perhitungan Haversine Formula, Euclidean Distance dan Manhattan Distance	43

“ Halaman Ini Sengaja Dikosongkan “

BAB I

PENDAHULUAN

1.1 Latar Belakang

UAV (unmanned aerial vehicle) merupakan kendaraan udara tak berawak yang dapat dikontrol dari jarak jauh baik itu menggunakan *remote control* atau terbang menggunakan sistem yang sudah terprogram (J & L, 2020). *Quadcopter* memiliki 4 motor di setiap lengannya, memiliki kemampuan lepas landas dan mendarat secara vertikal sehingga tidak membutuhkan lintasan yang panjang seperti pesawat pada saat melakukan proses lepas landas maupun mendarat. Ke-4 motor yang dimiliki memberikan kemudahan *quadcopter* dalam melakukan roll, pitch dan yaw. Ketiga hal tersebut berjalan di ketiga sumbunya yaitu x, y dan z. Hal ini memberikan beberapa keuntungan dalam hal bermanuver seperti terbang mundur, kesamping kiri atau kanan dan juga dapat merubah posisi heading dengan mudah pada sudut yaw. Ada beragam implementasi *UAV Quadcopter* guna membantu pekerjaan manusia salah satunya digunakan sebagai pengirim barang.

Proses pengiriman barang dengan range dalam kota secara konvensional dilakukan dengan memberikan barang ke kurir kemudian diantar oleh kurir melalui jalur darat, tentunya proses ini memiliki kekurangan yaitu terhambat oleh padatnya kendaraan sehingga durasi penjemputan dan pengiriman barang menjadi bertambah. Seiring dengan berkembangnya teknologi dibidang *aerial vehicle*, *uav quadcopter* dapat membantu menyelesaikan permasalahan tersebut dengan menggunakan *uav quadcopter* sebagai sarana pengiriman barang melalui jalur udara. Maka dibuat prototipe metode navigasi *waypoint* pada *uav quadcopter* sebagai pengirim barang.

Proses navigasi menggunakan *waypoint* membutuhkan *gps* yang terhubung dengan *flight controller*. *Gps* akan memberikan estimasi koordinat *quadcopter* ke *flight computer* melalui komunikasi serial yang terhubung dengan *flight controller*. Selanjutnya *flight computer* akan menghitung *remaining distance quadcopter* dengan koordinat yang ada di dalam urutan *waypoint* yang telah di program sebelumnya menggunakan *haversine formula*. Jika posisi relatif pada *quadcopter* dengan koordinat *waypoint* sudah tepat maka *quadcopter* akan terbang melanjutkan kearah urutan *waypoint* selanjutnya. Sistem navigasi menggunakan *waypoint* pada *uav quadrotor* sebagai pengirim barang diharapkan dapat memberi solusi pada permasalahan pengiriman barang menggunakan jalur darat.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas terdapat permasalahan yang diangkat pada proyek akhir ini seperti:

1. Bagaimana navigasi *waypoint* dilakukan pada *quadcopter*?
2. Bagaimana penerapan *haversine formula* dalam perhitungan *remaining distance* untuk navigasi *waypoint*?
3. Bagaimana pengaruh penggunaan *radius area* konfirmasi pada titik *waypoint* terhadap akurasi pencapaian *waypoint* pada navigasi *quadcopter*?

4. Bagaimana pengaruh penggunaan navigasi *waypoint* terhadap efisiensi dan kehandalan pada proses pengiriman paket berlangsung?

1.3 Tujuan

Penelitian ini bertujuan untuk menerapkan *navigasi waypoint* pada *quadcopter* yang digunakan untuk mengirim paket dengan menggunakan *waypoint* di dalamnya terdapat garis lintang dan bujur serta ketinggian yang digunakan sebagai *setpoint*. Sistem ini berjalan pada *flight controller* dan *flight controller* yang menentukan apakah barang sudah sampai, kapan *quadcopter* Kembali menuju *home point*.

1.4 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah, navigasi *waypoint* pada *quadcopter* pengirim barang memiliki batasan masalah yaitu sebagai berikut :

1. *Quadcopter* yang digunakan adalah *Quadcopter* yang stabil dan sudah dituning (tarot 650mm frame, 70C-5000mAH baterai).
2. Akselerasi tidak berubah ubah (takeoff: 10m/s², hovering: 5m/s², landing: 5m/s²).
3. *Payload* maksimal yang dapat dibawa 700 gram.
4. Range pengiriman barang maksimal 800 meter.

1.5 Manfaat

Manfaat yang diperoleh dari pengerjaan alat ini diantara lain :

1. Memberikan solusi pada jasa pengiriman barang selain menggunakan jalur darat.
2. Membantu operator tanpa harus menjalankan *quadcopter* dari jarak jauh pada saat menyelesaikan misinya.

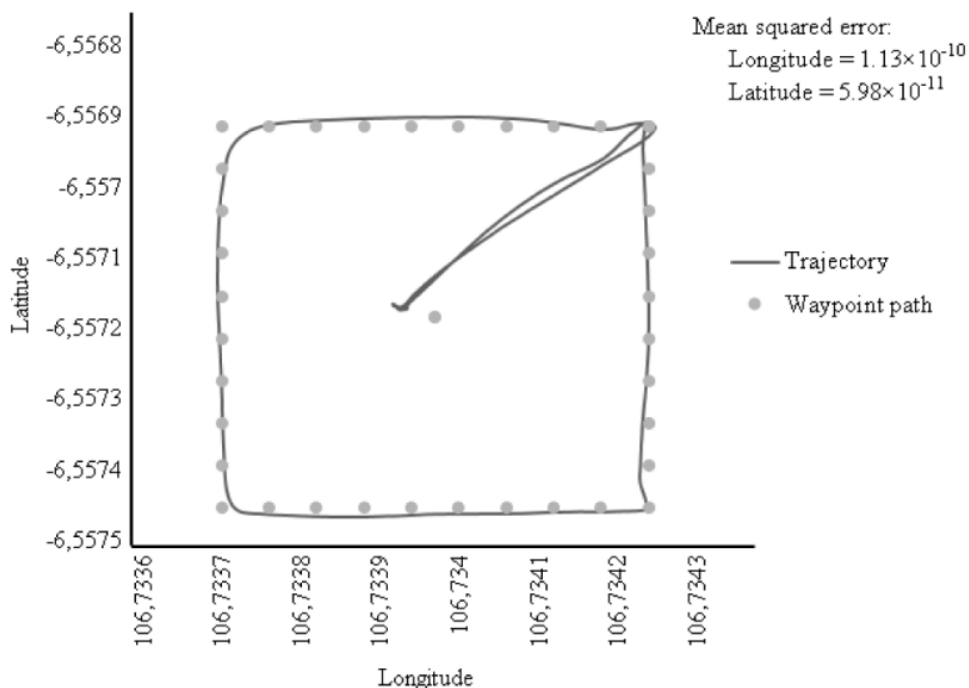
BAB II TINJAUAN PUSTAKA

Penyusunan proyek akhir memerlukan tinjauan pustaka yang digunakan sebagai acuan, di dalam tinjauan pustaka terdapat penelitian terdahulu beserta dasar-dasar teori yang berhubungan dengan proyek akhir yang digunakan sebagai landasan pemahaman pada proses pengerjaan proyek akhir. Berikut merupakan penelitian terdahulu yang digunakan sebagai dasar teori dan acuan dalam proses pengerjaan proyek akhir ini.

2.1 Penelitian Terdahulu

Pengerjaan proyek akhir ini mengambil beberapa referensi berupa penelitian yang telah dilakukan sebelumnya. Berikut adalah beberapa penelitian sebelumnya yang menunjang pengerjaan proyek akhir yang berhubungan dengan metode navigasi menggunakan *waypoint* dan desain hardware yang digunakan sebagai pembawa barang.

A. Jurnal berjudul “*Development of Autonomous UAV Quadcopters using Pixhawk Controller and Its Flight Data Acquisition*” (K. Priandana, 2020). Membahas tentang bagaimana membuat autonomus *quadcopter* yang melakukan misi dan membandingkan titik setpoint dengan titik aktual yang di lewati oleh *quadcopter*. Gambar dibawah merupakan penyajian data koordinat (lat,lon) dalam bentuk grafik yang terekam melalui *telemetry log*



Gambar 2.1 Grafik Historical Waypoint Dan Jalur Terbang
(K. Priandana, 2020)

B. Jurnal berjudul “*Pixhawk-based scalable platform for multi-purpose surface unmanned vehicle*” (J. Song, 2022). Membahas tentang bagaimana sebuah *platform* perangkat keras dapat digunakan dalam beberapa aplikasi misalnya *rover*, *Copter*, dan *boat*. Perangkat Pixhawk dapat melakukan kontrol beberapa aplikasi dikarenakan menggunakan tidak hanya

satu perangkat lunak tetapi user dapat menggunakan perangkat lunak tergantung dari aplikasi yang sedang digunakan. Selain dapat digunakan untuk beberapa aplikasi, kesatuan perangkat tersebut merupakan perangkat yang terbuka (*open source*), sehingga user dapat mengembangkan perangkat lunaknya sendiri sehingga mengikuti aplikasi yang diharapkan.

C. Jurnal berjudul "Perbandingan Metode Perhitungan Jarak Euclidean, Haversine, dan Manhattan Dalam Penentuan Posisi Karyawan" (Yusup M., 2020). Penulis melakukan penelitian mengenai perhitungan jarak antara 2 koordinat menggunakan metode euclidean, haversine dan Manhattan. Pada jurnal tersebut peneliti menghitung error yang dihasilkan dari perhitungan menggunakan euclidean, haversine dan manhattan. Kesimpulan yang didapat dari jurnal tersebut ialah metode manhattan memiliki tingkat keakurasian paling rendah yaitu sebesar 75.98% sedangkan euclidean memiliki tingkat keakurasian 98.51% yang paling tinggi haversine memiliki tingkat keakurasian 98.66%

D. Jurnal berjudul "Pengembangan Sistem Navigasi Otomatis Pada UAV(Unmanned Aerial Vehicle) dengan GPS (Global Positioning System) Waypoint (Rahmat H., 2018). Membahas pengembangan sistem navigasi otomatis pada UAV menggunakan GPS waypoint. Pada jurnal ini juga menjelaskan komponen-komponen yang digunakan dalam UAV, termasuk motor DC Brushless (BLDC), Electronic Speed Controller (ESC), motor servo, mikrokontroler Arduino Mega 2560, dan beberapa sensor seperti GY-85 (sensor gyroscope, accelerometer, dan magnetometer) dan BMP280 (sensor barometrik).

E. Jurnal berjudul "Autonomous Navigation of an Agricultural Robot Using RTK GPS and Pixhawk" (R. Moeller, 2020). Membahas sistem navigasi otonom yang dibangun untuk robot pertanian dengan menggunakan sistem GPS presisi untuk mengidentifikasi tanaman kentang yang terinfeksi virus. Sistem ini bertujuan untuk menghapus tanaman yang terinfeksi itu dari ladang dengan efektif. Perancangan robot otonom didasarkan pada mikrokontroler Pixhawk agar mudah digunakan dan dengan biaya yang terjangkau. Komponen pendukung, seperti peralatan komunikasi, motor, dan tempat home base dijelaskan secara rinci agar robot dapat mampu menghadapi cuaca dan kondisi medan. Selain itu, robot ini akan berfungsi sebagai platform uji untuk mekanisme genggam dan sensor yang akan diintegrasikan pada versi akhir yang lebih besar dan lebih canggih.

2.2 Gambaran Umum UAV(*Unmanned Aerial Vehicle*)

UAV atau Unmanned Aerial Vehicle adalah pesawat yang dapat terbang tanpa pilot di dalamnya dan dapat terbang secara otonom dengan menggunakan data sensor. UAV dapat dikendalikan secara manual melalui radio kontrol atau secara otomatis dengan mengolah data sensor sehingga dapat terbang sesuai dengan keperluan pengguna. (Laurent Nagel, 2021) Dengan kemampuan ini, UAV memiliki berbagai aplikasi yang luas, termasuk pemetaan, survei udara, pengawasan, dan banyak lagi

2.2.1. Jenis UAV

Secara umum UAV dikelompokkan dari penggerak nya

1) *Fixed wing*

UAV jenis fixed wing ini (gambar 2.2) menggunakan sayap sebagai perangkat aerodinamika yang dapat menjaga agar UAV dapat *gliding* di udara. Komponen penggerak pada fixed wing dapat berupa motor listrik maupun *internal combustion engine*.



Gambar 2. 2 *Fixed Wing UAV*
(Lauren Nagel, 2021)

2) *Rotary Wing*

Pada gambar 2.3 UAV tipe *rotary wing* menggunakan *propeller* menggantikan sayap sekaligus sebagai komponen penggerak utama dengan cara memberikan *thrust* atau gaya dorong yang mengarah ke darat agar UAV dapat terbang di udara. Dari jumlah rotor rotary wing dibagi lagi menjadi 2 yaitu *single rotor* dan *multi rotor*

- *Single Rotor*

Pada UAV tipe ini hanya menggunakan satu motor beserta satu propeller yang terpasang terlihat pada gambar 2.3 masyarakat umum menyebutnya *helicopter*.



Gambar 2. 3 *Single Rotor UAV*
(Laurent Nagel, 2021)

- *Multi Rotor*

Pada tipe UAV multi rotor (gambar2.4) memiliki lebih dari 1 motor penggerak. Tergantung penggunaannya ada yang memiliki 3 rotor, 4 rotor, 6 rotor dan 8 rotor. Semakin banyak penggerak yang terpasang maka semakin besar juga thrust yang dihasilkan. Namun beban UAV semakin bertambah sehingga diperlukan perencanaan yang matang terlebih dahulu agar kombinasi antara komponen dapat bekerja secara efisien.



Gambar 2. 4 *Multi Rotor Quadcopter*
(Laurent Nagel, 2021)

2.3 *Quadcopter*

Quadcopter adalah bagian dari kategori multirotor, yaitu UAV *rotary wing* yang mampu lepas landas dan mendarat secara vertikal. (Laurent Nagel, 2021). Untuk mengendalikan *attitude quadcopter*, kecepatan putaran propeller di keempat sisinya dapat diatur. Hal ini juga secara tidak langsung mengatur gaya angkat (*thrust*) pada setiap sisi. Dibandingkan dengan helikopter, quadcopter memiliki beberapa kelebihan. Quadcopter tidak memerlukan ekor penyeimbang seperti helikopter dan bisa menggunakan propeller *fixed pitch*, sehingga tidak perlu mengatur sudut propeller seperti helikopter.

Quadcopter yang digunakan memiliki model mekanik yang terdiri dari empat rotor yang dipasang pada sumbu silang simetris. Bentuk ini dirancang agar tipis dan kaku, sehingga menghasilkan friksi udara yang kecil. Komponen yang bergerak pada quadcopter hanyalah putaran propeller.

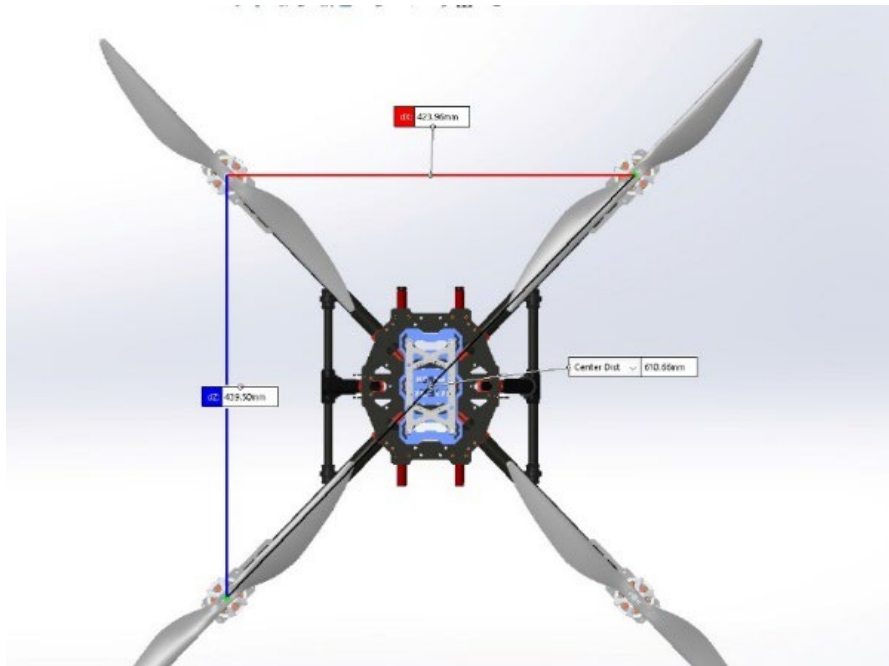
Setiap propeller pada quadcopter diputar oleh satu motor *brushless DC*, sehingga terdapat empat motor sebagai aktuator untuk menghasilkan gaya angkat. Dengan batasan menggunakan karakteristik motor dan propeller yang relatif sama, kecepatan motor yang sama akan menghasilkan kondisi melayang yang stabil di setiap propeller. Namun, pada tiap propeller yang berputar, akan menghasilkan torsi yang bergerak dalam bidang horizontal. Oleh karena itu, konfigurasi propeller terdiri dari dua propeller yang bergerak searah jarum jam dan dua propeller yang bergerak berlawanan arah jarum jam. Dalam satu sumbu, propeller berputar dengan arah yang sama.

2.4 **Komponen Penyusun Quadcopter**

Quadcopter tersusun atas beberapa komponen yaitu frame, *brushless DC* motor, driver motor, *flight controller* dan *propeller*. Fungsi dari masing masing komponen diuraikan pada penjelasan dibawah

2.4.1 *Frame*

Gambar 2.5 merupakan rangka tempat dimana semua komponen dipasang, *frame* menentukan dimensi dari *quadcopter*. Kegunaan *frame* juga beragam tergantung ukuran yang diukur dari sumbu diagonal pada *frame*. Semakin besar *frame* maka semakin besar brushless dc motor yang dapat terpasang sehingga semakin besar pula tenaga yang dihasilkan.



Gambar 2.5 *Frame X Pada Quadcopter*
(Tarotrc, 2021)

2.4.2 *Flight Controller*

Gambar 2.6 merupakan *flight controller* yang bertugas untuk mengolah dan memproses perintah terprogram yang diberikan. Pada perangkat ini dapat membaca sensor seperti GPS, IMU (Inertial Measurement Unit), barometer dan juga dapat membaca konsumsi daya melalui sensor arus yang terpasang pada power module dan juga dapat mengontrol motor. (ArduPilot Dev Team, 2022).



Gambar 2.6 *Flight Controller Pixhawk 6C*
(Ardupilot, 2022)

Pada proyek akhir ini menggunakan Pixhawk 6C. Memberikan dua pilihan sistem operasi atau *firmware*. Kedua sistem *firmware* yaitu PX4 dan Ardupilot. Keduanya bersifat *open source* dan dapat diintegrasikan dengan perangkat komputer (Raspberry). Penulis memilih Ardupilot karena banyaknya dokumentasi yang ada di internet sehingga memudahkan penulis dalam memberikan perintah dari raspberry ke *flight controller*.

2.4.3 *Brushless DC Motor (BE4108-380KV)*

Merupakan komponen penggerak utama pada quadcopter, klasifikasi motor dipisahkan menurut ukuran dan kecepatan putarnya. Untuk ukuran motor diberikan notasi dengan format XXYY. XX merupakan notasi diameter rotor dan YY merupakan notasi tinggi magnet. Semakin besar nilai pada X dan Y semakin besar pula torsi yang dihasilkan motor. Spesifikasi terlihat pada gambar 2.8



Gambar 2. 7 *BLDC motor*
(Tarotrc, 2021)

Selain ukuran rotor dan magnet terdapat parameter Constant Velocity (KV) yaitu kecepatan putar. Pada gambar 2.7 angka 380KV merupakan putaran motor permenit dalam satuan volt. Semakin tinggi kecepatan putar maka akan semakin rendah torsi. Maka semakin besar rangka yang digunakan akan semakin tinggi kecepatan putarnya.

2.4.4 *Electronic Speed Control (ESC)*

Jika terdapat *brushless dc motor* pasti membutuhkan *dc driver motor*. Pada ruang lingkup *drone*, *dc driver motor* disebut dengan ESC. ESC menerima input berupa PWM yang dihasilkan dari *flight controller*. Gambar 2.8 merupakan contoh ESC dengan maksimal sebesar 40A (Sharad Pachpute, 2020).

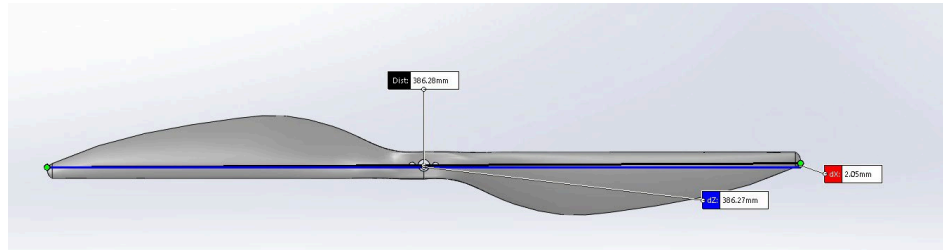


Gambar 2. 8 *Electronic Speed Control (ESC) 40A*
(Sharad Pachpute, 2020)

Pada gambar 2.8 merupakan salah satu contoh ESC dengan spesifikasi maksimum current rating sebesar 40A dengan tegangan 2-4S lipo. Hal ini berarti komponen tersebut mampu diberi tegangan maksimum hingga 4S. Huruf S merujuk pada jumlah cell pada baterai lipo dengan tegangan nominal 3.7 volt. Berarti total tegangan yang dapat diterima komponen tersebut sebesar 14.8 VDC.

2.4.5 Propeller

Atau disebut juga baling-baling (gambar 2. 9). Ada dua jenis propeller yang digunakan yaitu propeller searah jarum jam / *Clock Wise* (CW) dan Berlawanan arah jaru, jam / *Counter Clock Wise* (CCW) (gambar 2.10)



Gambar 2. 9 Propeller
(Sharad Pachpute, 2020)

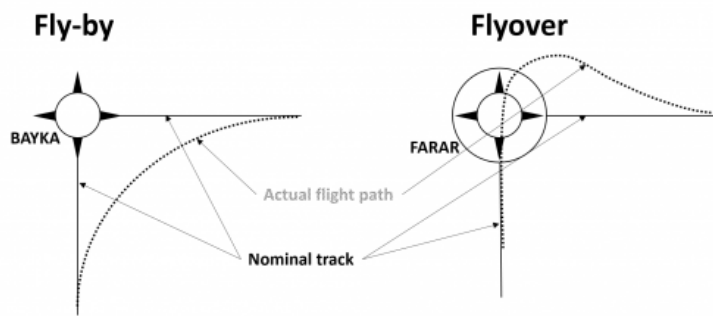
Terdapat ukuran yang beragam dari propeller, umumnya dituliskan dengan format XXYY contohnya 1045, 1150, 1355, dll. Pada nilai XX ini yaitu memberitahukan panjang propeller dan nilai YY memberitahukan nilai pitch dari proller (dalam satuan Inch) dan untuk pemilihan propeller juga wajib menyesuaikan penggunaan disesuaikan dengan penggunaan motor yang akan digunakan.



Gambar 2. 10 Propeller Dengan Tipe Clockwise dan Counter Clockwise
(Sharad Pachpute, 2020)

2.5 Navigasi Waypoint

Waypoint adalah titik referensi atau kumpulan koordinat yang digunakan untuk navigasi guna mengidentifikasi lokasi tertentu pada peta. Koordinat tersebut biasanya mencakup longitude, latitude, dan kadang-kadang ketinggian (altitude) untuk keperluan navigasi udara. Waypoint digunakan dalam berbagai jenis navigasi, baik dalam navigasi udara, laut, maupun darat di mana tidak terdapat jalur yang jelas. Bahkan dalam navigasi darat yang menggunakan robot sebagai penentu arah, waypoint tetap digunakan untuk memastikan bahwa robot tetap mengikuti rute yang ditentukan.

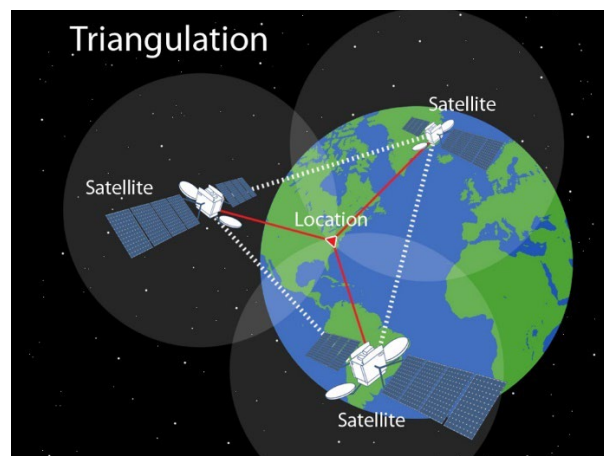


Gambar 2. 11 *Fly-by dan Flyover pada Waypoint*
(ICAO, 2018)

Pada gambar 2.11 waypoint dapat dibedakan menjadi dua jenis, yaitu waypoint *fly-by* dan waypoint *flyover*. Waypoint *fly-by* mengarah ke arah tujuan tanpa melewati langsung lokasi waypoint, sedangkan waypoint *flyover* melewati langsung lokasi waypoint tersebut . Setelah melewati satu waypoint, pilot atau pengendali harus menetapkan waypoint berikutnya yang disebut sebagai waypoint aktif (ICAO, 2018)

2.6 *Global Positioning System (GPS)*

Sistem GPS merupakan sebuah sistem navigasi dan penentuan posisi yang menggunakan satelit sebagai basisnya. Sistem ini dirancang untuk memberikan informasi posisi yang akurat secara *realtime* diseluruh dunia. Satelit-satelit GPS mengorbit di sekitar Bumi seperti yang ditunjukkan pada gambar 2.13 dan merupakan instrumen yang sangat canggih.

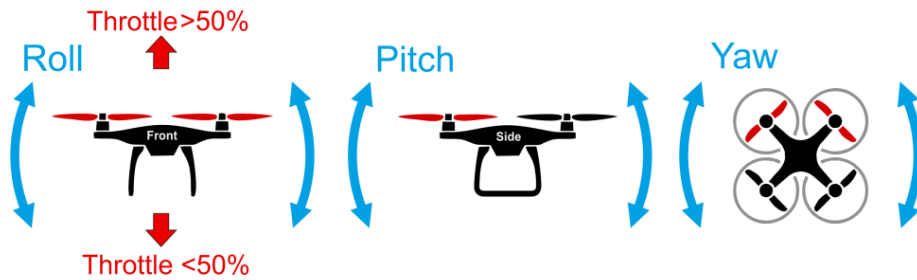


Gambar 2. 12 *GPS Berdasarkan Posisi Satelit*
(Tim Gunther, 2021)

Pada gambar 2.12 satelit yang mengorbit bumi akan mentransmisikan sinyal ke bumi setelah itu device di bumi akan menerima sinyal dari satelit. *Feedback* signal yang diterima satelit kemudian dipetakan berdasarkan arah pantulannya. Masing-masing satelit dilengkapi dengan empat jam atomik yang memiliki tingkat ketepatan hingga satu detik error dalam tiga juta tahun. Satelit-satelit GPS mengirimkan deretan kode berdasarkan sinyal waktu yang diterima, bukan dari informasi posisi atau pergerakan. (Tim Gunther, 2021)

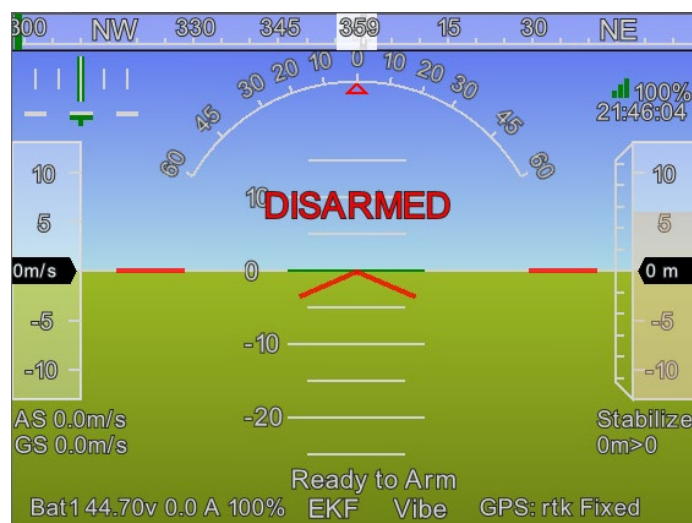
2.6 Attitude Heading Reference System (AHRS)

Didalam dunia aviasi, *attitude* dan *heading* merupakan dua variabel penting yang harus dipahami oleh pilot. Sistem ini tersusun atas *magnetometer*, *accelerometer* dan *gyroscope* yang terusun menjadi satu disebut dengan *inertial measurment unit (IMU)*. (Ardupilot Dev Team, 2020). *Attitude* merupakan sikap dari kendaraan udara menandakan apakah kendaraan ini dalam keadaan *roll* atau *pitch*, *roll* berarti kendaraan dalam keadaan miring kekanan atau kiri sedangkan *pitch* berarti kendaraan dalam keadaan menukik kedepan atau kebelakang gambaran keadaan ini ada pada gambar 2.13



Gambar 2. 13 Sudut Roll Pitch Dan Yaw
(PX4, 2020)

Kemudian *heading* merupakan arah *nose* atau hidung kendaraan udara mengarah ke utara, selatan, timur atau barat seperti pada gambar 2.13 sudut *heading* disebut dengan *yaw*. Gambar 2.14 merupakan interface dari sistem AHRS. Didalam nya terdapat kompas yang ditunjukan dengan NW atau north-west pada bagian paling atas. Dibawah kompas terdapat garis seperempat lingkaran dengan angka nol ditengah nya garis ini dibuat unruk mengetahui seberapa besar sudut *roll* pada kendaraan. Kemudian dibawah notifikasi *disarmed* terdapat garis putus putus merah yang menjadi memisah antara permukaan bumi dengan langit. Dibawah garis ini terdapat tanada seperti panah merah yang menunjukkan sudut *pitch* dengan indikator sudut berada diantaranya.



Gambar 2. 14 Interface Pada Sistem AHRS
(Ardupilot, 2020)

Dalam proyek akhir ini menggunakan firmware ardupilot sudah terdapat *AHRS* didalamnya, namun ada beberapa parameter yang harus diberikan agar sistem *AHRS* dapat berjalan.

2.6 Protokol Komunikasi

Agar *flight controller* dapat berkomunikasi dan dapat mengerti perintah yang diberikan oleh raspberry pi sebagai *flight computer* protokol komunikasi *mavlink* digunakan. Merupakan akronim dari *micro air vehicle link* yang memiliki fungsi sebagai protokol komunikasi antara *flight controller* dengan *flight computer*, maupun *ground station* melalui *serial connection*. Dirancang khusus untuk kendaraan terbang tanpa awak.

Dalam arsitektur *MAVLink*, pesawat terbang atau UAV berperan sebagai sistem kendali yang menghasilkan data telemetri seperti posisi, kecepatan, orientasi, dan sensor lainnya. Data ini dikemas dalam pesan *MAVLink* dan dikirim melalui media komunikasi ke stasiun darat atau komponen lainnya. *Ground station* atau komponen lainnya yang terhubung dengan *quadcopter* menggunakan protokol *MAVLink* dapat menerima pesan tersebut, melakukan analisis data, mengirimkan perintah kontrol, dan berinteraksi dengan sistem kendali UAV (Dan Rose, 2020).

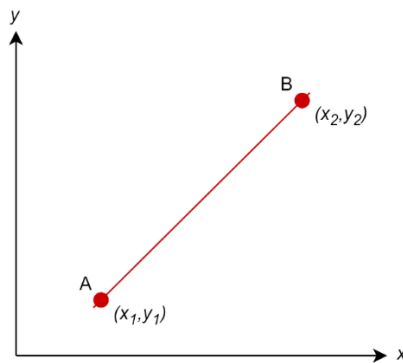
2.7 Haversine Formula

Teorema *Haversine Formula* banyak digunakan dalam bidang navigasi, kegunaannya untuk mencari jarak busur antara dua titik pada permukaan bola, permukaan bola yang dimaksud disini adalah permukaan bumi. Titik koordinat merupakan perpotongan dari garis imajiner yang membentang dari utara ke selatan (*longitude*) dan barat ke timur (*latitude*). Teori pada *haversine formula* mengasumsikan bahwa bentuk bumi bulat (*spherical earth*) dengan menghilangkan faktor bahwa bumi itu sedikit elips (*elipsoidal factor*) (Yusup M & Fahmi R. K., 2020). Untuk gambaran lebih jelasnya terlihat pada gambar 2. 15



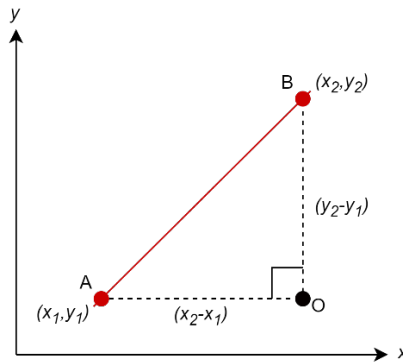
Gambar 2. 15 Jarak Terdekat Dari 2 Koordinat Pada Bumi
(Mariane, 2019)

Seperti yang terlihat pada gambar 2. 15, jarak antara 2 koordinat terdekat ditandai



Gambar 2. 17 Dua Titik Koordinat A dan B
(Sumber, Penulis)

Dari gambar 2. 16 langkah selanjutnya menggambar titik imajiner O sebagai titik bantuan pada sisi sebelah kanan sehingga akan membentuk sebuah segitiga.



Gambar 2. 18 Titik O Sebagai Titik Bantu Imajiner
(Sumber, Penulis)

Dari gambar 2. 17 jarak (A,B) dapat dihitung menggunakan rumus *pythagoras* dengan cara,

$$AB^2 = AO^2 + OB^2$$

$$AB^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2$$

Maka,

$$AB = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \dots \dots \dots (2.4)$$

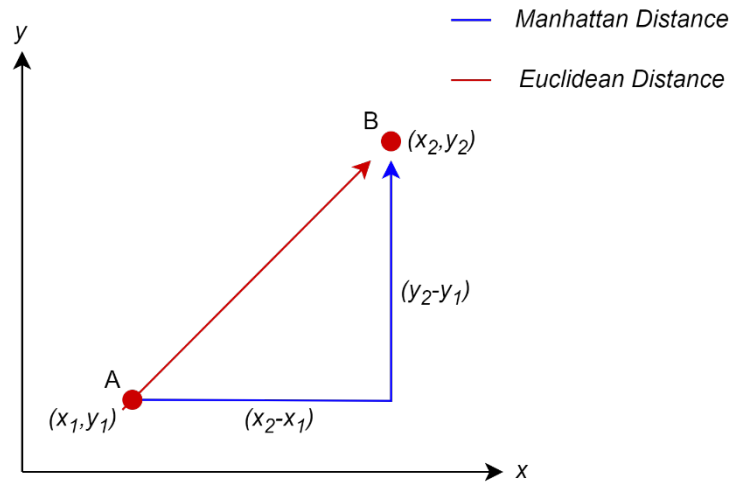
Hasil akhir AB dengan menambahkan panjang per 1^o yang ada pada bumi maka rumus *euclidean distance* menjadi,

$$d(A, B) = \sqrt{\Delta lat^2 + \Delta lon^2} \times 1.113195 \times 10^5 \dots \dots \dots (2.5)$$

2.9 Manhattan Distance

Baik jarak Manhattan maupun jarak Euclidean mewakili jarak antara dua titik pada koordinat. Jarak Manhattan mewakili jarak antara dua titik ketika bergerak sejajar dengan

masing-masing sumbu x dan y, dan jarak Euclidean mewakili jarak garis lurus antara dua titik, seperti yang diilustrasikan pada gambar 2.18 (Yusup M & Fahmi R. K., 2020)



Gambar 2. 19 Perbedaan Euclidean Distance dengan Manhattan Distance
(Sumber, Penulis)

Sehingga rumus *manhattan distace* menjadi,

$$d = \left(\sum_{i=1}^n |x_i - y_i| \right) \times 1 \text{ derajat bumi}$$

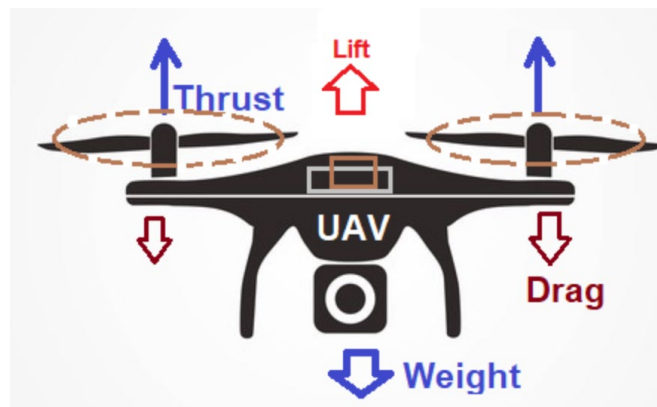
$$d = (|x_2 - x_1| + |y_2 - y_1|) \times 1.113195 \times 10^5 \dots\dots\dots(2.6)$$

Hasil akhir persamaan menjadi,

$$d = (|\Delta lat| + |\Delta lon|) \times 1.113195 \times 10^5 \dots\dots\dots(2.7)$$

2.10 Thrust Brushless DC Motor

Thrust ialah gaya angkat yang dihasilkan oleh *propeller* yang terpasang pada *brushless DC motor*. Seperti yang dijelaskan pada gambar ilustrasi 2.20 keempat propeller memberikan gaya dorong keatas karena propeller menghembuskan angin yang dihasilkan oleh putaran motor.



Gambar 2. 20 Quadcopter Thrust
(Sharad Pachpute, 2020)

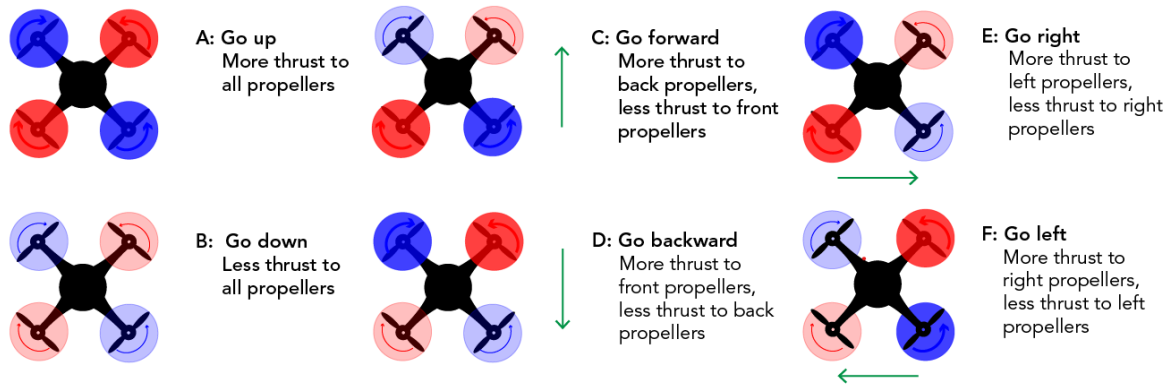
Formula perhitungan *thrust* pada *quadcopter* sebagai berikut,

$$total\ thrust = thrust\ to\ weight\ ratio \times total\ weight \dots\dots\dots(2.8)$$

Untuk mendapatkan *thrust* per motor yang nantinya dijadikan acuan dalam memilih motor maka perhitungannya menjadi,

$$thrust\ per\ motor = \frac{total\ thrust}{total\ motor} \dots\dots\dots(2.9)$$

Thrust pada *quadcopter* sendiri selain digunakan untuk acuan dalam menentukan kekuatan motor, digunakan juga dalam hal mengontrol gerakan *quadcopter*. (Sumber <https://www.calctool.org/machines-and-mechanisms/drone-motor>)



Gambar 2. 21 Gerakan Pada *Quadcopter*
(Sharad Pachpute, 2020)

Gambar 2.21 menjelaskan pengaruh *thrust* terhadap *quadcopter*. Pada bagian A jika menambahkan *thrust* maka *quadcopter* bergerak keatas, bagian B jika mengurangi *thrust* maka *quadcopter* bergerak turun, bagian C jika 2 motor dibelakang *thrust* nya lebih besar dibandingkan dengan 2 motor didepan maka *quadcopter* bergerak maju, bagian D jika dua motor didepan *thrust* nya lebih besar dari 2 motor di belakang maka *quadcopter* bergerak mundur, bagian E jika 2 motor di kiri *thrust* nya lebih besar dari 2 motor dikan maka *quadcopter* bergerak ke kanan, bagian F jika 2 motor di kakan *thrust* nya lebih besar dari 2 motor dikiri maka *quadcopter* bergerak ke kiri (Sharad Pachpute, 2020)

III METODOLOGI

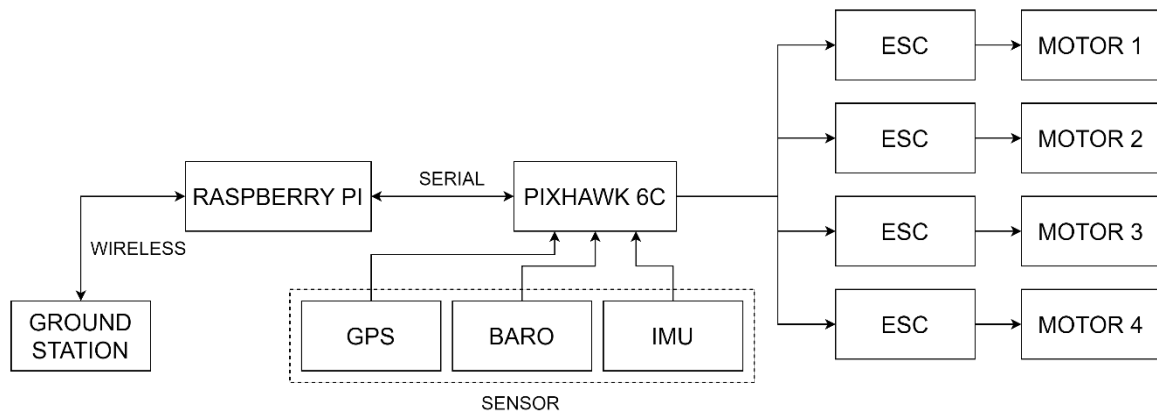
Bab ini membahas tentang perancangan dan hasil perancangan alat, serta metode yang digunakan dalam menyelesaikan permasalahan navigasi waypoint dalam proses kerjanya

3.1 Gambaran Umum Sistem

Pada pengerjaan proyek akhir ini merakit *quadcopter* difungsikan sebagai pengirim paket berdasarkan titik koordinat yang dikirim dari *ground station* ke *flight computer*. GPS pada *quadcopter* berguna untuk mengetahui lokasi dari *quadcopter* itu sendiri yang dijadikan sebuah sensor umpan balik untuk pembandingan dari posisi relatif *quadcopter* terhadap koordinat *waypoint*. Proses perhitungan ini berlangsung pada *raspberry pi* yang dijadikan sebagai *flight computer*.

3.2 Desain Dan Konsep Sistem

Pixhawk digunakan sebagai *flight controller* yang berfungsi mengatur *plant* yaitu ke-4 *brushless-dc* motor. Motor ini bertugas memberikan *thrust* agar *quadcopter* dapat terbang. Selain motor, pixhawk sebagai *controller* juga dapat memberikan data sensor seperti GPS, barometer, maupun *inertial measurment unit (IMU)* ke *raspberry pi* melalui komunikasi serial. Sebagai *flight computer*, *raspberry pi* bertugas memberikan perintah kepada *flight controller* untuk melakukan urutan *sequence* misi terbang secara otomatis.



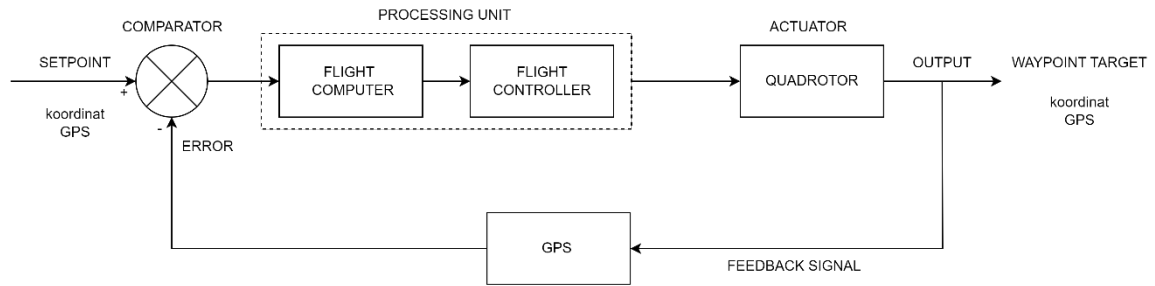
Gambar 3.1 Blok Diagram Sistem Arsitektur
(Sumber, Penulis)

Pada gambar 3.1 GPS digunakan sebagai sensor umpan balik agar dapat mengetahui posisi relatif *quadcopter* pada saat melakukan misi penerbangan. ESC yang terhubung langsung dengan pixhawk berugas sebagai driver motor dengan cara menerima sinyal PWM (*pulse width modulation*) yang dihasilkan *flight controller* kemudian menggerakkan motor sesuai dengan frekuensi yang diterima sehingga kecepatan putaran motor sesuai dengan perintah dari flight controller. Keseluruhan sistem ini bekerja dengan sumber daya baterai Li-Po 6 Cell, dengan tegangan maksimal 25.2V.

3.3 Blok Diagram Proses Keseluruhan

Blok diagram proses keseluruhan ini merupakan gambaran dasar mengenai proses dari sistem *quadcopter* pengirim barang dirancang, blok diagram dapat dilihat pada Gambar

3.2.



Gambar 3. 2 Blok Diagram Proses
(Sumber, Penulis)

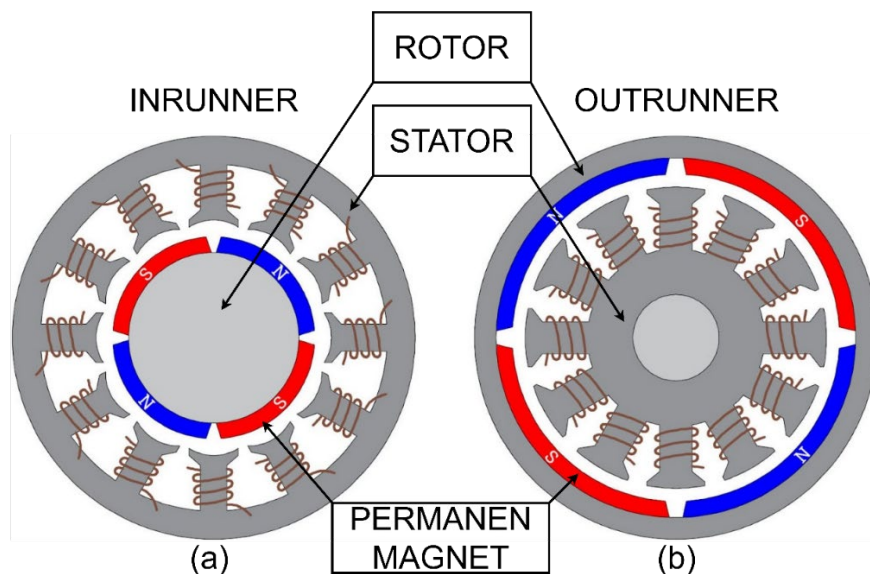
Pada blok diagram ini dapat dilihat GPS digunakan sebagai sensor umpan balik agar raspberry pi yang bertugas sebagai *flight computer* dapat mengetahui jarak relatif *quadcopter* terhadap setpoint yang berupa koordinat menggunakan perhitungan matematis dari permodelan harvisine formula. Jika hasil *comparator* masuk pada batasan threshold maka raspberry pi memerintahkan *flight controller* untuk melanjutkan *sequence* selanjutnya menuju *waypoint* selanjutnya.

3.3 Perancangan Komponen Pada Quadcopter

Pada sub bab 3.3 ini menjelaskan tiap-tiap komponen penyusun pada *quadcopter* seperti: *frame*, *brushless dc motor*, *electronic speed controller*, *propeller*, *flight controller*, *GPS sensor*, *flight computer* dan baterai.

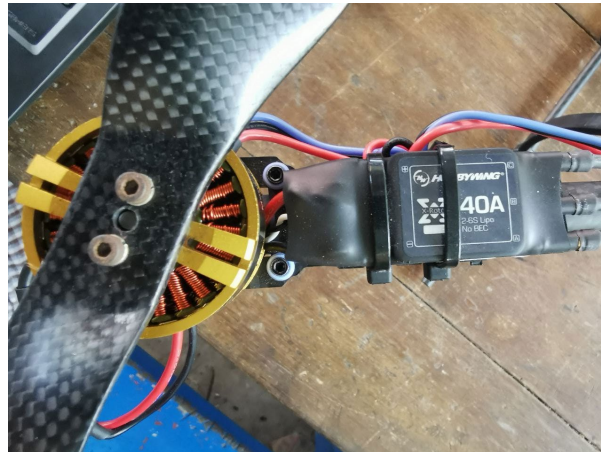
3.3.1 Brushless DC Motor Dan Propeller

Kedua komponen ini merupakan aktuator utama pada *quadcopter* seperti halnya motor dan impeler pada pompa. Ada dua jenis struktur *brushless dc motor (BLDC)* yaitu *outrunner BLDC* dan *inrunner BLDC* seperti ilustrasi pada gambar 3.3



Gambar 3. 3 (a) INRUNNER BLDC MOTOR (b) OUTRUNNER BLDC MOTOR
(Lisa Eitel, 2018)

Pada gambar 3.3 (a) merupakan *inrunner BLDC motor* dengan struktur rotor yang dipasang permanen magnet berada didalam *body motor* dengan kumparang yang mengelilinginya. Sedangkang *outrunner BLDC motor* pada gambar 3.3 (b) memiliki struktur rotor sebagai body dengan permanen magnet mengelilingi stator yang ter-*install* pada kumparan. Pada proyek akhir ini menggunakan *outrunner BLDC motor* sehingga penampang *propeller* dapat dipasang pada *body motor* seperti yang terlihat pada gambar 3.4 sehingga gaya dorong kebawah yang dihasilkan *propeller* dapat ditahan oleh motor.



Gambar 3. 4 Instalasi Propeller Pada Motor
(Sumber, Penulis)

Spesifikasi pada motor menggunakan 4108-380KV dapat terlihat pada gambar 3.5 (a) sedangkan spesifikasi *propeller* menggunakan *propeller* dengan ukuran 13x55” dengan berbahan dasar karbon dapat terlihat pada gambar 3.5 (b). Pemilihan propeller dengan bahan dasar karbon ditujukan karena karbon dengan campuran resin memiliki tingkat kekakuan yang rendah, sehingga pada putaran motor yang tinggi *propeller* cenderung tidak bengkok selain itu bahan dasar karbon ditujukan agar *propeller* tidak mudah patah jika terjadi *crash* atau *quadcopter* jatuh ke permukaan tanah.



Gambar 3. 5 (a) 4108-380KV BLDC MOTOR, (b) 13x55” Propeller
(Tarotrc, 2021)

Tabel 3. 1 Data Pengujian Motor Yang Telah Terpasang Propeller
(Sumber: Tarot Motor BLDC 4108-380KV Data Sheet)

Voltase Baterai	Propeller	Arus (A)	Thrust (g)	Daya (W)	Efisiensi (g/W)	RPM
22.2V	1355	1	260	22.8	11.4	2980
		2	460	45.6	10.0	3600
		3	650	68.4	9.5	4230
		4	770	91.2	8.4	4660
		5	860	114	7.5	4970
		6	980	136.8	7.1	5270
		7	1090	159.6	6.8	5520
		8	1180	182.4	6.4	5830
		9	1310	205.2	6.3	5970
		10	1380	228	6.0	6200
		11	1440	250.8	5.7	6330
		12	1530	273.6	5.5	6470
		12.4	1620	282.7	5.7	6560

Tabel 3.1 merupakan hasil uji coba motor 4108-380KV dengan propeller ukuran 13x55” yang telah terpasang pada motor. Dengan *thrust* merupakan gaya angkat yang dihasilkan dari *propeller* yang berputar, semakin tinggi arus pada motor maka semakin tinggi gaya angkat yang dihasilkan oleh *propeller*. Gaya angkat (*thrust*) maksimum yang dapat dihasilkan dari konfigurasi motor 4108-380KV dan *propeller* 13x55” adalah sebesar 1.6Kg pada RPM 6560. Hasil pengujian tersebut dikeluarkan secara resmi oleh produsen motor dengan rekomendasi ukuran *propeller* 13x55” sehingga, pemilihan ukuran *propeller* didasarkan pada hasil tersebut. Tabel asli dari Tabel 3.1 ada pada lembar lampiran.

3.3.2 Hobbywing XRotor 40A (Electronic Speed Controller)

Merupakan *driver motor BLDC* yang digunakan untuk menggerakkan dan mengatur putaran *BLDC motor* sesuai dengan yang dibutuhkan. Gambar 3.6 merupakan gambar dari komponen *driver motor BLDC* ini.

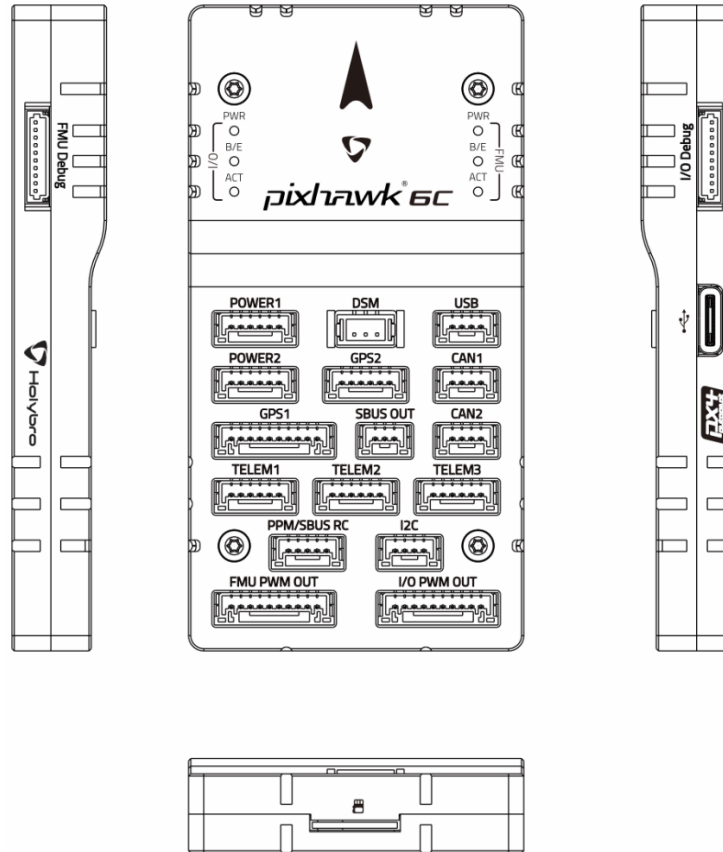


Gambar 3. 6 Driver Motor BLDC Hobbywing XRotor 40A
(Hobbywing, 2019)

Komponen ini mampu memberikan daya pada motor dengan arus konstan sebesar 40A dan arus puncak sebesar 60A, mampu menerima tegangan maksimum dari baterai LiPo-6S yang memiliki tegangan nominal sebesar 22.2 sehingga cocok digunakan pada *brushless DC motor* 4108-380KV seperti yang tertulis pada lembar manual yang terlampir.

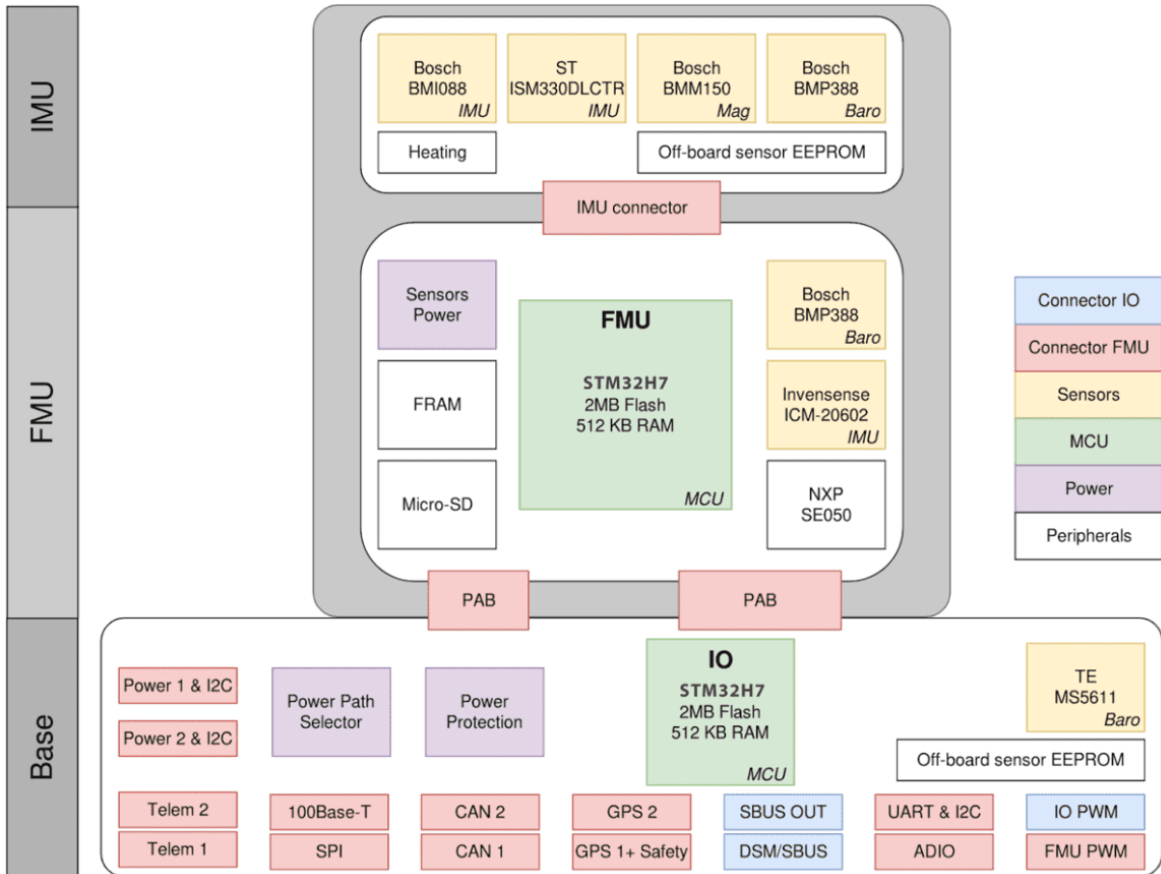
3.3.3 Pixhawk 6C (flight controller)

Merupakan sebuah merk *flight controller*. Pixhawk 6C ditujukan untuk *development quadcopter* sehingga memiliki berbagai macam *i/o connector* pada permukaan luar komponen hal ini memudahkan penggunaanya untuk merakit atau menambahkan komponen lain atau bahkan dapat dengan mudah melakukan *troubleshooting* jika ada kerusakan *hardware* pada sistem, gambaran secara lengkap ada pada gambar 3.7.



Gambar 3. 7 Pixhawk 6C
(Holybro, 2022)

Beberapa sensor telah terpasang pada komponen ini seperti: *inertial measurmernt unit*, dan *barometer sensor*. Pada bagian *inertial measurmernt unit* atau dapat disingkat IMU terdapat beberapa sensor penyusun seperti *magnetometer*, *accelerometer* dan *gyroscope* keterangan lebih jelas pada bagian IMU pada gambar 3.8 kemudian pada bagian FMU pada gambar 3.8 merupakan bagian processing unit seperti CPU pada komputer. Terdiri dari FMU STM32 H7 sebagai *processor* dengan disupport FRAM dan *power sensor* yang bekerja memberikan daya pada *processor*. Pada sisi kanan terdapat NXP SE050 yang bertugas sebagai kontroler komunikasi agar FMU STM32 H7 dapat membaca berbagai macam data dari input dan output. Yang terakhir pada bagian BASE pada gambar 3.8 merupakan bagian input dan output yang dapat disambung pada *pixhawk 6c* seperti, *Telem* digunakan untuk *telemetry*, *Power* merupakan *socket power*, *GPS* merupakan *socket* khusus untuk sensor GPS dan *socket-socket* sisanya yang difungsikan untuk *socket* input dan output. (Jessica F, 2022)



Gambar 3. 8 *Arsitektur Komponen Penyusun Pixhawk 6C*
(Jessica F, 2022)

3.3.4 M8N (GPS Sensor)

Merupakan salah satu tipe dari GPS, digunakan untuk memberikan data koordinat pada *pixhawk 6c* yang dijadikan sebagai *feedback signal*, membutuhkan tegangan input sebesar 5V dan arus minimum 150mA. Detail komponen ini ada pada gambar 3.9.

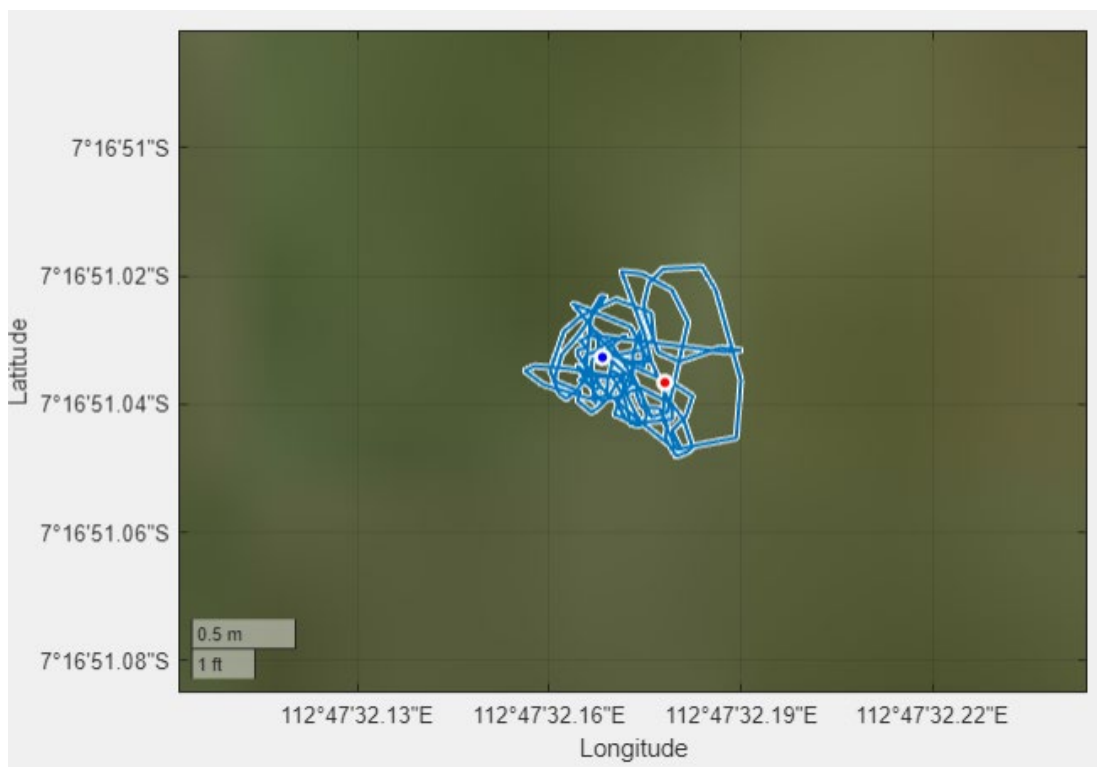


Gambar 3. 9 *M8N GPS Sensor*
(Holybro, 2022)

Tabel 3. 2 Data Pembacaan Sensor GPS Berupa Koordinat (*latitude, longitude*)
(Sumber, Penulis)

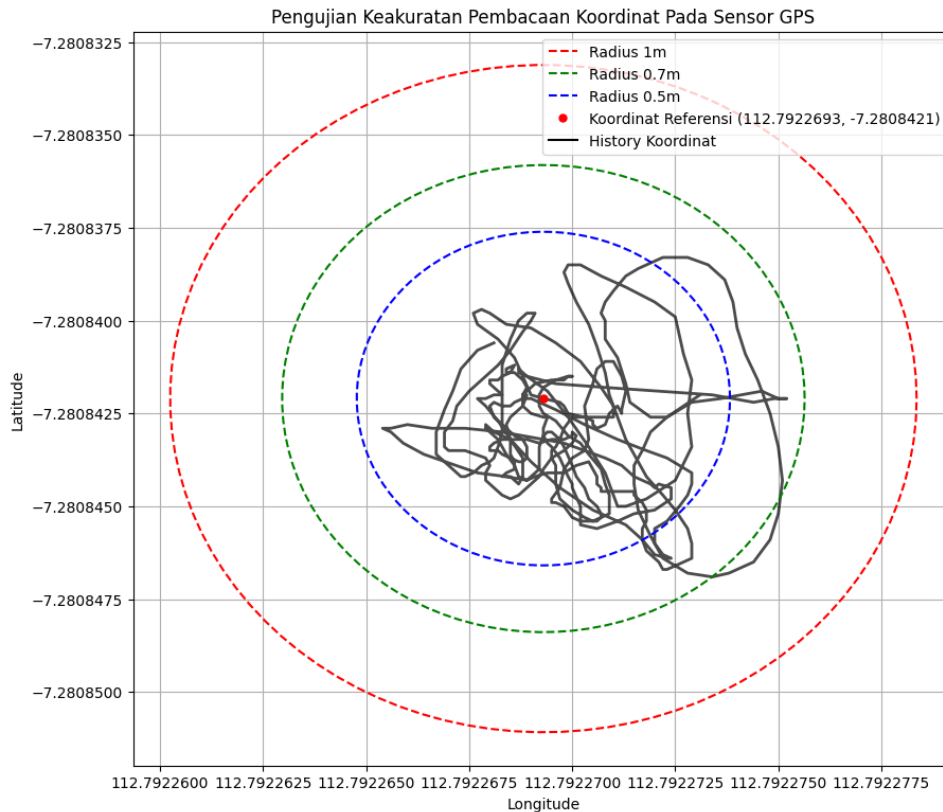
No	Koordinat Hasil Pembacaan GPS		Koordinat Referensi	
	<i>latitude</i>	<i>longitude</i>	<i>latitude</i>	<i>longitude</i>
1	-7.2808428	112.7922731	-7.2808421	112.7922693
2	-7.2808423	112.7922727		
3	-7.2808426	112.7922723		
4	-7.2808433	112.7922719		
5	-7.2808435	112.7922717		
⋮	⋮	⋮		
426	-7.2808424	112.7922669		
427	-7.2808419	112.7922671		
428	-7.2808413	112.7922673		
429	-7.2808408	112.7922677		
430	-7.2808406	112.7922681		

Data pada tabel 3.2 merupakan data yang didapatkan pada saat melakukan pengujian sensor GPS pengujian dilakukan dengan cara membandingkan koordinat yang terbaca pada drone dengan koordinat referensi.



Gambar 3. 10 Visualisasi Data Pembacaan Sensor GPS
(Sumber, Penulis)

Gambar 3.10 merupakan hasil visualisasi data yang didapatkan pada saat proses pengujian sensor dilakukan. Visualisasi data didapatkan dari *tools* program *matlab* bernama *flightLogAnalyzer* Gambar 3.11 merupakan pengolahan data yang ditampilkan secara visual guna mengetahui tingkat kepresisian dari sensor GPS M8N. Pada gambar 3.11 dapat diambil kesimpulan bahwa tingkat error dari pembacaan sensor terhadap koordinat referensi sebesar 0.7 meter dalam radius area.



Gambar 3. 11 Pengujian Keakuratan Pembacaan Koordinat Pada Sensor GPS
(Sumber, Penulis)

3.3.5 Raspberry Pi 4B (flight computer)

Pada proyek akhir ini, *raspberry pi* digunakan sebagai *flight computer* dengan tugas memberikan perintah take-off, terbang menuju target lokasi dalam satu proses *closeloop* seperti yang dijelaskan pada sub bab 3.3. Raspberry pi merupakan single board computer dengan dilengkapi 40 pin GPIO (General Purpose Input Output) dapat menjalankan proses komputasi dan dapat difungsikan sebagai kontroler dengan adanya GPIO pin. Gambar 3.12 merupakan bentuk *hardware Raspberry Pi 4B*.



Gambar 3. 12 Raspberry Pi 4B
(Raspberrypi, 2018)

3.3.6 Baterai Li-Po 6S 5000mAH

Merupakan komponen dimana daya listrik disimpan. Konfigurasi 6S yang dimaksud ialah 6 cell baterai yang dirangkai secara seri. Pada baterai Li-Po memiliki tegangan nominal sebesar 3.7 V sehingga jika ke-enam cell dirangkai secara seri menghasilkan tegangan nominal sebesar 22.2V. Pada gambar 3.13 baterai yang dipakai ialah dinogy 6S 5000mAH dengan rating arus sebesar 70C atau 70A, artinya baterai tersebut memiliki tegangan nominal 22.2V, memiliki arus discharge maksimum 70A, memiliki kapasitas sebesar 5000mAH.



Gambar 3. 13 Baterai Li-Po 6S 5000mAH 70C
(Dinogy, 2017)

3.3.7 Tarot X Frame 650mm

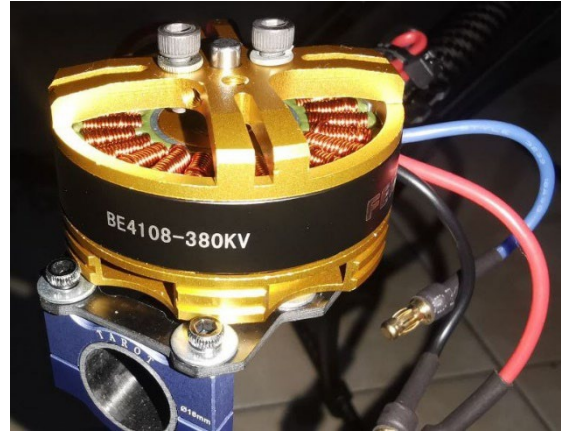
Frame merupakan sasis tempat dimana dirakitnya semua komponen yang telah disebutkan diatas. Serat karbon merupakan bahan dasar dari *tarot x frame 650mm* dibentuk, memiliki 3 layer serat karbon dengan dilapisi resin membuatnya sangat kuat sehingga tidak mudah patah ataupun bengkok dan juga memiliki keunggulan bobot yang ringan. Gambar 3.14 merupakan hasil render design dari *taror x frane 650mm*.



Gambar 3. 14 Tarot X Frame 650mm
(Tarotrc, 2021)

3.4 Perakitan Komponen Pada Frame

Pembuatan perangkat keras merupakan tahapan dalam pengerjaan proyek *quadcopter* yang stabil guna memenuhi kebutuhan batasan masalah pada bab 1. Penulis menggunakan rangka Tarrot dengan panjang lengan 650mm pada gambar 3. 14 sebelah kiri.



Gambar 3. 15 *Frame Tarrot 650mm dan Instalasi Motor pada Frame*
(Sumber, Penulis)

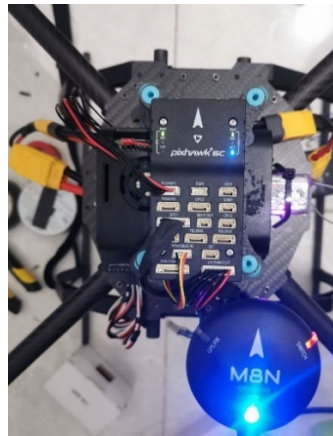
Dalam manualnya rangka tersebut sudah diklasifikasikan motor dengan ukuran 1408-380KV dan beberapa perangkat lain dalam perakitannya agar efisien dan sesuai yang di harapkan. Beberapa orang membuat rangka tersebut dengan kombinasi baling baling berukuran 13 *inch* hingga level ekstrem dengan panjang baling baling 17 *inch*. Pada proyek akhir ini menggunakan baling-baling berukuran 13 *inch* seperti yang terlihat pada gambar 3. 15.



Gambar 3. 16 *Instalasi Baling-baling dan ESC*
(Sumber, Penulis)

Ukuran baling baling tersebut dimaksudkan ketika dibutuhkan daya angkat yang lebih besar, penggantian baling baling dimaksudkan untuk memperkecil biaya riset. Penulis menggunakan rangka tersebut dikarenakan aplikasi yang digunakan yaitu mengantarkan barang membutuhkan tenaga yang lebih sehingga *frame* tersebut cocok untuk aplikasi pengangkutan beban yang diklaim pada detail spesifikasi dapat mencapai maksimal 4Kg untuk keseluruhan berat pada *quadcopter*. Untuk komponen komputasi dalam hal ini

merupakan sistem penerbangan, penulis menggunakan *Pixhawk*. *Pixhawk* yang digunakan merupakan *Pixhawk 6c* yang disertai hardware terkini dan yang tersedia di pasar lokal. *Pixhawk* digunakan karena sifatnya yang *open source*. *Pixhawk* dapat dengan mudah mengatur pergerakan *quadcopter* seperti *flight controller* pada umumnya. *Pixhawk* dipilih karena merupakan *flight controller* yang tidak terhubung pada rangka tertentu, sehingga dapat digunakan dalam remote controlling secara general. Perbedaan paling mencolok dari *pixhawk* merupakan software-nya yang dapat diatur ulang sesuai kebutuhan pengguna. Selain itu *pixhawk* mendukung berbagai API seperti *dronekit* dan *pymavlink* yang artinya *flight controller* dapat diperintah menggunakan *python*.



Gambar 3. 17 Perakitan Komponen Pada Frame
(Sumber, Penulis)

3.5 Perancangan Perangkat Lunak Navigasi *Waypoint*

Perancangan perangkat lunak dalam proyek ini meliputi visualisasi urutan program menggunakan *flowchart*, penjelasan bagaimana navigasi berjalan berdasarkan *flowchart* yang dibuat. Navigasi *waypoint* pada proyek akhir ini berisikan serangkaian perintah menggunakan *API Dronek*. *API Dronekit* merupakan sebuah perangkat lunak penghubung antara *raspberry pi* dengan *pixhawk 6c* agar kedua komponen ini dapat berkomunikasi dua arah dengan menggunakan bahasa pemrograman *python*.

Pada gambar 3.24 merupakan *sequence* program secara general. Didalam nya terdapat urutan perintah agar *quadcopter* dapat terbang sesuai dengan apa yang diperintahkan. Pertama diawali dengan inisiasi koneksi, proses ini berlangsung dengan cara mengakses *socket serial* pada *flight controller* menggunakan *API Dronekit* yang telah tersedia seperti yang tertera pada gambar 3.18

```
1 |from dronekit import connect, VehicleMode, LocationGlobalRelative
2
3 connection_string = '/dev/ttyACM0'
4 vehicle = connect(connection_string, wait_ready=True, baud=912600)
```

Gambar 3. 18 Contoh Syntax Komunikasi Dari Raspberry Menuju Pixhawk
(Sumber, Penulis)

Langkah pertama untuk memulai koneksi dengan meng-*import python package dronekit* pada lembar kerja *python* seperti pada gambar 3.18 baris 1, selanjutnya membuat variabel string yang berisikan alamat *socket* pada *pixhawk* seperti pada *connection_string*.

Kemudian pada `vehicle = connect` pada baris ini lah koneksi berlangsung, fungsi `wait_ready = True` akan bekerja jika koneksi tidak terjadi sehingga pengguna dapat mengerti jika terjadi `error` pada koneksi.

Setelah koneksi terjadi langkah selanjutnya pada gambar 3.19 ialah `arming` dan `takeoff` dalam dunia penerbangan `arm` berarti menyakalan mesin

```
1 # Fungsi arm & takeoff|
2 def arm_and_takeoff(aTargetAltitude):
3
4     print("Basic pre-arm checks")
5     # Menunggu Proses inialisasi vehicle
6     while not vehicle.is_armable:
7         print(" Waiting for vehicle to initialize...")
8         time.sleep(1)
9
10    print("Arming motors")
11    # Merubah Mode Terbang Menjadi GUIDED Kemudian Arming Motor
12    vehicle.mode = VehicleMode("GUIDED")
13    vehicle.armed = True
14
15    # Konfirmasi Apakah Motor Telah Menyala?
16    while not vehicle.armed:
17        print(" Waiting for arming...")
18        time.sleep(1)
19
20    print("Taking off!")
21    vehicle.simple_takeoff(aTargetAltitude) # Take off menuju target ketinggian
22
23    # Fungsi Loop Takeoff pada target altitude
24    while True:
25        print(" Altitude: ", vehicle.location.global_relative_frame.alt)
26        if vehicle.location.global_relative_frame.alt = aTargetAltitude :
27            print("Reached takeoff target altitude")
28            break
29        time.sleep(1)
30
31 arm_and_takeoff(10)
```

Gambar 3. 19 Fungsi Arm And Takeoff
(Sumber, Penulis)

Pada gambar 3.19 merupakan fungsi `arm and takeoff` yang ditulis menggunakan `API Dronekit` dengan bahasa `python`. Langkah pada fungsi tersebut ialah pertama `basic pre-arm check` pada baris 4 ditujukan untuk memastikan apakah `quadcopter` siap untuk terbang ada beberapa persyaratan seperti apakah baterai kosong, apakah semua motor telah terpasang, apakah semua sensor telah menyala. Setelah proses ini selesai langkah selanjutnya ialah menyalakan motor seperti pada gambar 3.19 baris 10. Untuk menyalakan motor harus merubah mode terbang terlebih dahulu menjadi `guided` fungsi dari mode ini ialah menggantikan remote dengan program. Jika `mode guided` berjalan maka kita tidak bisa memberikan input pada `quadcopter` menggunakan `remote`. Selanjutnya pada baris 21 merupakan `API Dronekit` berupa perintah untuk melakukan `takeoff`, pengguna hanya meng-call `API` secara otomatis `quadcopter` akan `takeoff`. Baris 24 merupakan fungsi `loop takeoff` pada `target altitude`, jika `quadcopter` telah mencapai `target altitude` maka `quadcopter` menjaga `altitude` agar tidak berubah. Baris 31 pada gambar 3.19 merupakan penggunaan fungsinya pada `main program`, angka 10 yang berada pada dalam kurung merupakan target

altitude yang harus dicapai oleh *quadcopter* pada saat melakukan *takeoff*.

Urutan *sequence* selanjutnya pada gambar 3.24 ialah *AHRS heading control* seperti yang telah dijelaskan pada bab 2.6 *quadcopter* akan selalu mengarahkan heading nya kearah *waypoint* setelah *takeoff*. Parameter yang digunakan ada pada gambar 3.20.

```
1 heading = ahrs:get_position()
2
3 waypoints = [lat, lon, alt]
4
5 def HeadingControl(vehicle,lat,long,alt,heading):
6     vehicle.mav.set_position_target_global_int_send(
7         int(1e3 * (time.time() - boot_time)),
8         0,
9         0,
10        coordinate_frame=mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT_INT,
11        type_mask=(
12            mavutil.mavlink.POSITION_TARGET_TPEMASK_AFX
13            mavutil.mavlink.POSITION_TARGET_TPEMASK_AFY
14            mavutil.mavlink.POSITION_TARGET_TPEMASK_AFZ
15            mavutil.mavlink.POSITION_TARGET_TPEMASK_YAW
16        )
17        ,lat_int=lat, lon_int=lon, alt=alt,
18        afx=0, afy=0, afz=0, yaw=heading
```

Gambar 3.20 Fungsi Heading Control
(Sumber, Penulis)

Dalam gambar 3.20 baris pertama membuat variabel heading yang isinya mendapatkan posisi dalam library *AHRS* yang ada didalam ardupilot, baris 3 merupakan list *waypoint* yang berisi *latitude*, *longitude* dan *altitude*, baris 5 hingga 16 merupakan fungsi *heading control* sedangkan baris 17 dan 18 merupakan parameter yang digunakan pada fungsi *heading control*. Isi dari parameternya adalah list *waypoint* yang ada pada baris 3, sudut *attitude* yakni *afx*, *afy* dan *afz* masing masing memiliki *setpoint* 0 derajat agar *quadcopter* dapat stabil diakhir parameter ada sudut *yaw* yang diberikan *setpoint* variabel *heading*. Tujuan diberikan *setpoint* dari *AHRS* ialah agar *quadcopter* dapat secara otomatis mengarahkan *nose* nya kearah *waypoint*

Pada gambar 3.25 (a) pengontrolan *quadcopter* menggunakan navigasi *waypoint* hal yang dilakukan pertama ialah inisialisasi posisi *quadcopter* sekarang dan membaca list *waypoint* yang diberikan penjelasan syntax ada pada gambar 3.21

```
1 current_location = vehicle.location.global_relative_frame
2
3 # Daftar waypoint
4 waypoints = [
5     LocationGlobalRelative(-7.2806894, 112.7924077, 15), # Koordinat waypoint 1
6     LocationGlobalRelative(-7.2808596, 112.7924024, 15), # Koordinat waypoint 2
7     LocationGlobalRelative(-7.2808603, 112.7920443, 15), # Koordinat waypoint 3
8     LocationGlobalRelative(-7.2807140, 112.7921060, 15), # Koordinat waypoint 4
9     LocationGlobalRelative(-7.2805670, 112.7921636, 15), # Koordinat waypoint 5
10 ]
```

Gambar 3.21 Syntax Posisi Sekarang Dan List Target Koordinat
(Sumber, Penulis)

Pada baris 1 gambar 3.21 merupakan *API Dronekit* untuk menarik data pembacaan

sensor GPS secara *realtime*. Pada garis 4 hingga 10 merupakan *tuple* atau input dalam bentuk *list* pada *python* yang berisi target koordinat beserta *altitude*.

Langkah selanjutnya pada gambar 3.25 (a) ialah perintah terbang menuju *waypoint*. Detail proses perintah terbang menuju *waypoint* ada pada penjelasan gambar 3.22 berikut.

```
1 wp_rad = 1
2
3 # Fungsi untuk mengatur target waypoint
4 def set_target_waypoint(waypoint):
5     vehicle.simple_goto(waypoint)
6
7 #terbang menuju waypoint
8 def move_to_waypoint(waypoints):
9     for waypoint in waypoints:
10        set_target_waypoint(waypoint)
11
12        while True:
13            current_location = vehicle.location.global_relative_frame
14            current_distance = get_distance_metres(current_location, waypoint)
15            rounded_distance = round(current_distance, 2)
16            print("Remaining Distance: {:.2f} meter".format(rounded_distance))
17
18            if current_distance < wp_rad: #radius waypoint
19                break
20
21            time.sleep(1)
22
23 move_to_waypoint(waypoints)
```

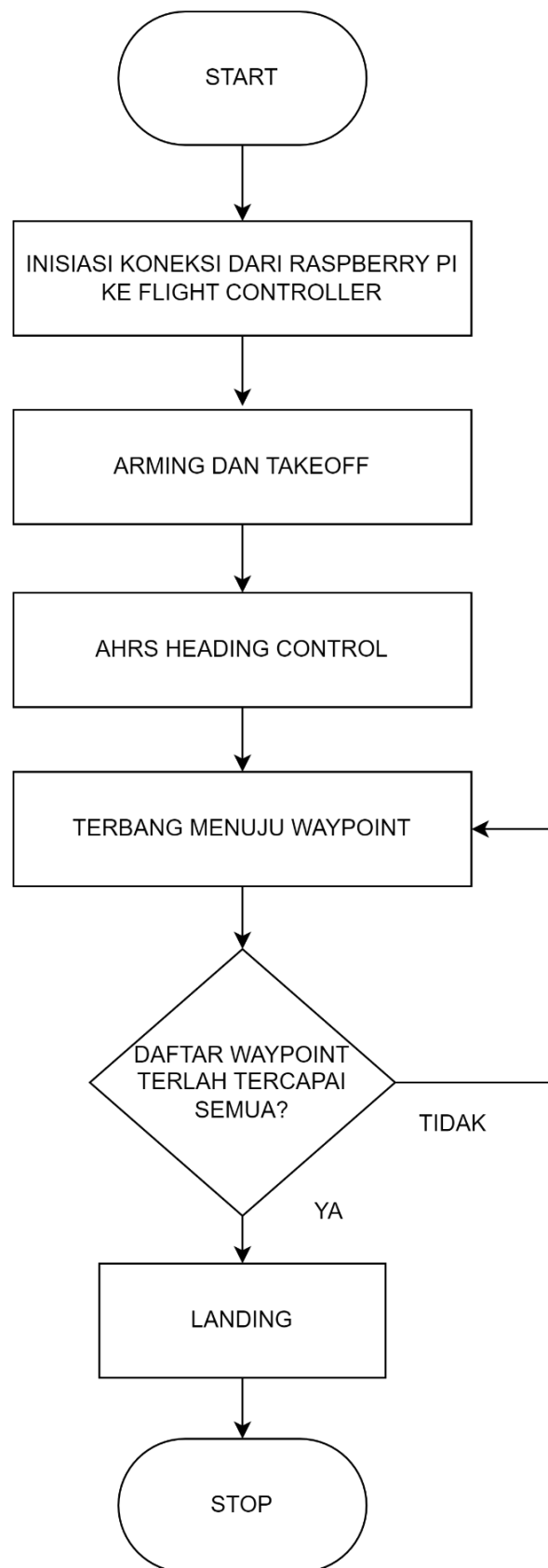
Gambar 3. 22 Perintah Terbang Menuju Waypoint
(Sumber, Penulis)

Pada baris 1 gambar 3.22 adalah variabel untuk menentukan *radius wapoint* yang ditujukan sebagai area konfirmasi jika *quadcopter* telah mencapai *target waypoint*. Pada baris 4 gambar 3.22 merupakan fungsi untuk memanggil *API Dronekit* yang memerintahkan *quadcopter* untuk menuju *waypoint*. *API* ini mengirimkan *target waypoint* (gambar 3.21 baris 4) ke *flight controller*. Selanjutnya pada baris 8 hingga 21 adalah fungsi utama *loop navigasi waypoint* yang diproses pada *raspberry pi* dengan *sequence* membaca *list target waypoint* kemudian mengirimkannya ke *pixhawk* kemudian membuat *loop* yang menghitung jarak antara *quadcopter* dengan *target waypoint*. Jika jarak *quadcopter* lebih kecil dari *radius waypoint* (gambar 3.22 baris 18) maka *loop* akan berhenti dan *quadcopter* melanjutkan menuju *waypoint* selanjutnya.

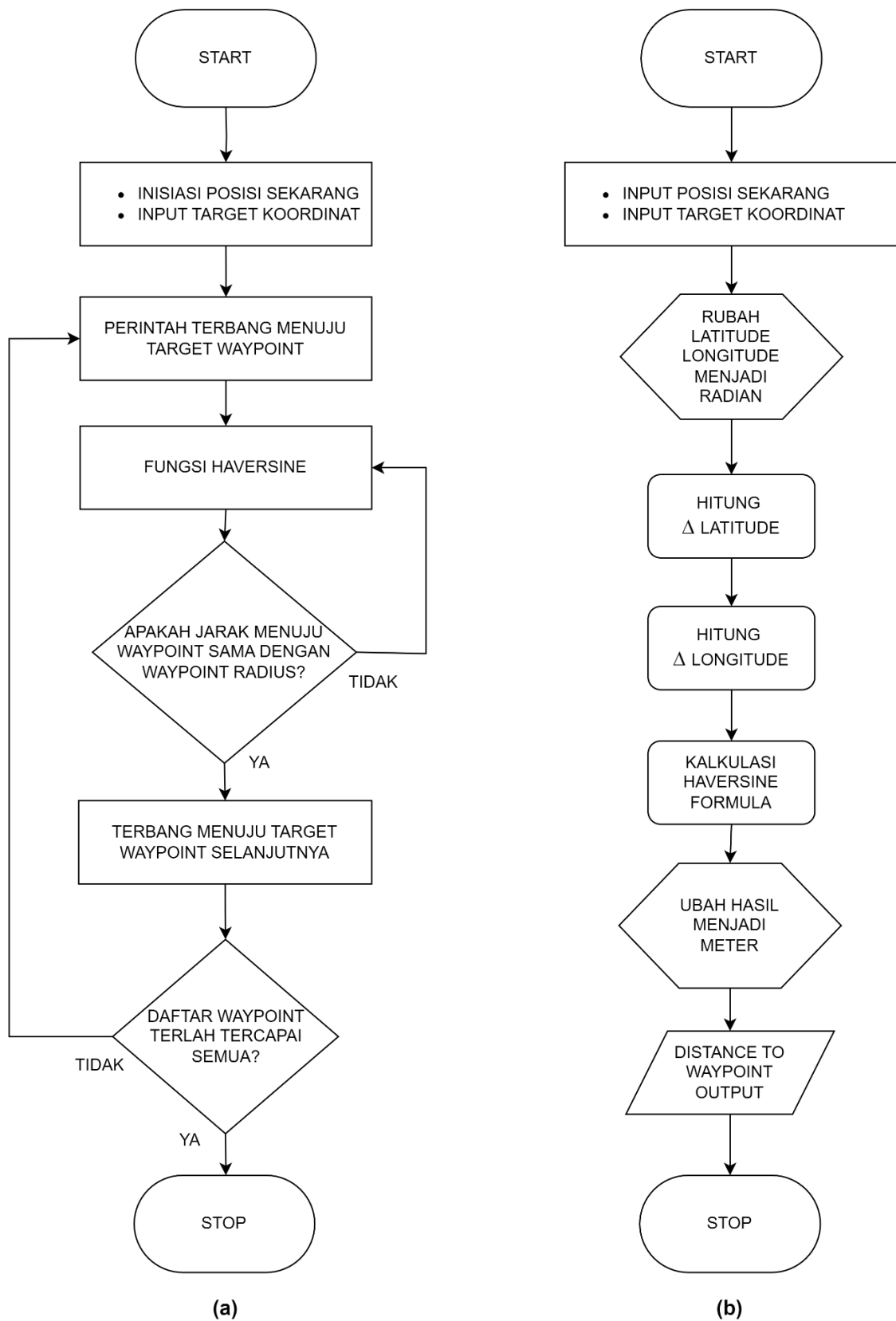
```
1 # Haversine Formula
2 def get_distance_metres(location1, location2):
3     dlat = math.radians(location2.lat - location1.lat)
4     dlong = math.radians(location2.lon - location1.lon)
5     a = math.sin(dlat / 2) * math.sin(dlat / 2) + math.cos(math.radians(location1.lat)) \
6         * math.cos(math.radians(location2.lat)) * math.sin(dlong / 2) * math.sin(dlong / 2)
7     c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
8     return 6371000 * c # Jari-jari bumi sekitar 6371000 m
```

Gambar 3. 23 Perhitungan Haversine Formula Pada Bahasa Pemrograman Python
(Sumber. Penulis)

Gambar 3.23 merupakan perhitungan *haversine formula* dalam *python* urutannya sama seperti perhitungan *haversine formula* pada bab 3.6 dan jelaskan kembali melalui *flowchart* pada gambar 3.25 (b)



Gambar 3. 24 *Urutan Sequence Program Pada Quadcopter*
(Sumber, Penulis)



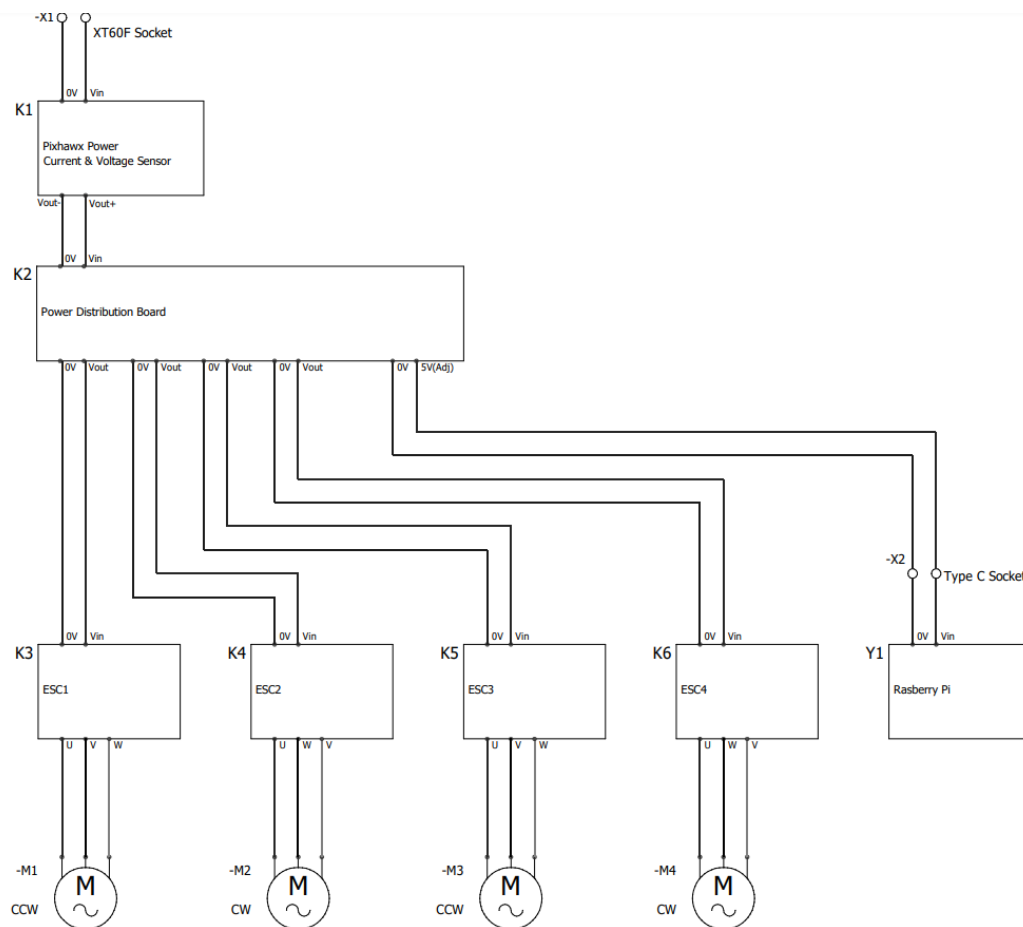
Gambar 3. 25 Flowchart Algoritma Waypoint, (a) Flowchat Program Utama, (b) Flowchart Perhitungan Menggunakan Haversine Formula
(Sumber, Penulis)

3.6 Wiring Pada Komponen

Subab ini merupakan bagaimana cara merangkai *quadcopter* dengan komponen komponen pendukungnya. Dalam subab ini wiring dipisahkan menjadi dua bagian yaitu power wiring dan signal wiring.

3.6.1 Power Wiring Diagram

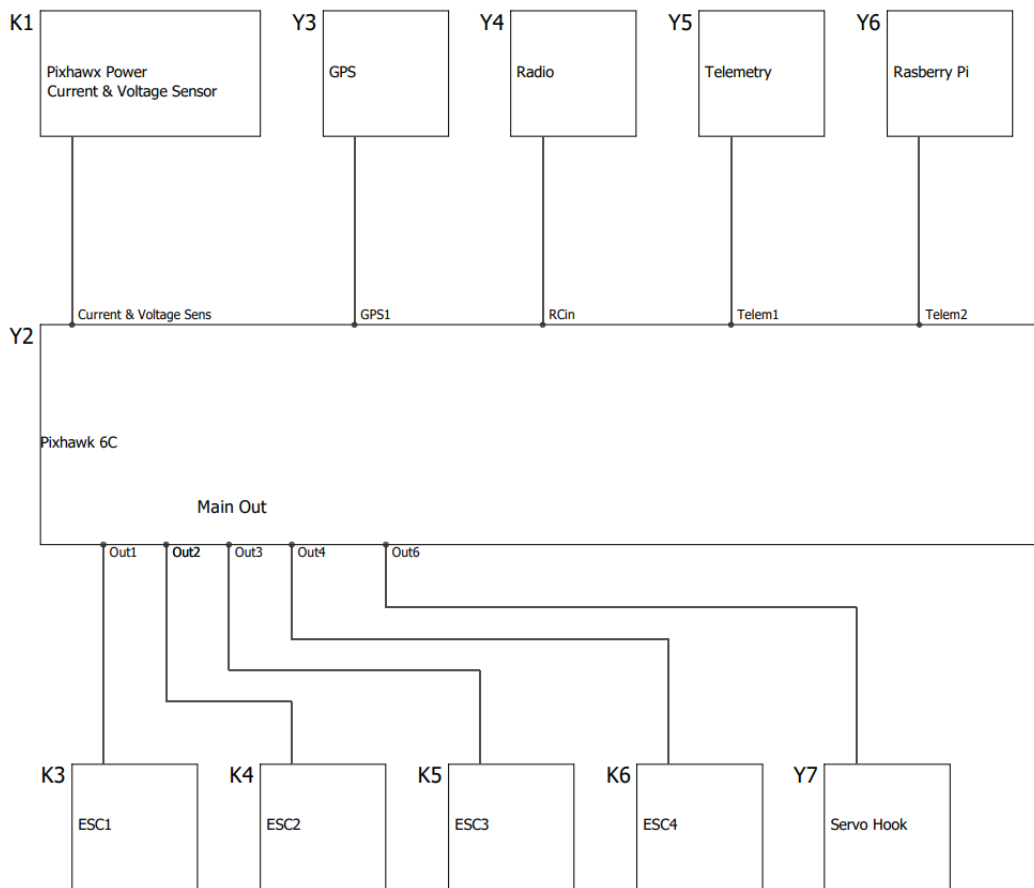
Untuk memenuhi kebutuhan daya dan menyalakan semua perangkat dibutuhkan koneksi kesemua motor. Berikut merupakan wiring power dari semua perangkat yang digunakan. Dalam Gambar 3.26 dibawah dijelaskan urutan daya yang digunakan baterai mengalir melalui sensor arus dan tegangan kemudian dibagi menjadi 5 bagian. 4 bagian digunakan untuk motor dengan dan satu bagian digunakan untuk Raspberry Pi.



Gambar 3. 26 *Power Wiring Diagram*
(Sumber, Penulis)

3.6.2 Signal Wiring Diagram

Setelah *wiring power*, dalam Gambar 3.27 terdapat *wiring* sinyal yang menunjukkan dimana komponen dipasang. Dalam diagram dibawah ini dijabarkan tempat dan soket yang dimaksud. Kemungkinan kesalahan dapat diminimalisir menggunakan soket khusus Pixhawk karena jenisnya yang dikhususkan untuk menyambungkan satu perangkat tertentu.



Gambar 3. 27 *Signal Wiring Diagram*
(Sumber, Penulis)

3.7 Perhitungan Jarak Menggunakan *Haversine Formula*

Pada bab 2 telah dijelaskan bagaimana rumus perhitungan haversine formula. Pada sub bab 3.6 ini menjelaskan bagaimana perhitungan haversine secara bertahap dengan cara mengubah koordinat menjadi radian, kemudian menghitung jarak menggunakan haversine formula. Sebelum masuk ke perhitungan, koordinat merupakan titik potong antara 2 garis. Garis yang digunakan ialah garis lintang dan garis bujur. Pada garis lintang dan bujur, satuan yang pakai ialah derajat. Jika diketahui koordinat A (-7.2808104, 112.7922722) kemudian koordinat B (-7.2803934, 112.7918398) maka,

Menentukan variabel lat_1 yang merupakan garis *latitude*₁, lon_1 yang merupakan garis *longitude*₁, lat_2 yang merupakan garis *latitude*₂, lon_2 yang merupakan *longitude*₂.

$$lat_1 = -7.2808104$$

$$lon_1 = 112.7922722$$

$$lat_2 = -7.2803934$$

$$lon_2 = 112.7918398$$

Selanjutnya rubah nilai derajat pada lat_1 , lon_1 , lat_2 , lon_2 menjadi radian dengan perhitungan sebagai berikut,

$$rad = \frac{degree \cdot \pi}{180}$$

Substitusikan nilai variabel $lat_1, lon_1, lat_2, lon_2$ kedalam persamaan radian sehingga menjadi,

$$lat_1 = \frac{-7.2808104 \cdot \pi}{180}$$

$$= -0.1270741136 \text{ rad}$$

$$lon_1 = \frac{112.7922722 \cdot \pi}{180}$$

$$= 1.9685965206 \text{ rad}$$

$$lat_2 = \frac{-7.2803934 \cdot \pi}{180}$$

$$= -0.1270668356 \text{ rad}$$

$$lon_2 = \frac{112.7918398 \cdot \pi}{180}$$

$$= 1.9685889738 \text{ rad}$$

Setelah seluruh nilai variabel menjadi radian, substitusikan kedalam rumus (2.2),

$$a = \sin^2\left(\frac{7.2780 \times 10^{-6}}{2}\right) + (\cos(-0.1270741136) \cos(-0.1270668356) \sin^2\left(-\frac{7.5468 \times 10^{-6}}{2}\right))$$

$$a = 2.7252292164 \times 10^{11}$$

Kemudian substitusikan nilai a ke dalam persamaan *distance* (2.3), dengan r adalah radius bumi dengan nilai 6,371Km atau 6,371,000 m.

$$d(A, B) = 2r \cdot \sin^{-1}(\sqrt{2.7252292164 \times 10^{11}})$$

$$d = 66.51 \text{ m}$$

Berdasarkan hasil perhitungan jarak antara koordinat A (-7.2808104, 112.7922722) dengan koordinat B (-7.2803934, 112.7918398) menggunakan *haversine formula*, kedua titik memiliki jarak sejauh 66.51 meter.

3.8 Perhitungan Jarak Menggunakan *Euclidean Distance*

Seperti yang telah diuraikan pada bab 2, pembahasan sub bab 3.7 ini dimaksudkan untuk menguraikan bagaimana menghitung jarak antara dua titik yang memiliki koordinat menggunakan perhitungan *euclidean distance*. Jika diketahui koordinat A (-7.2808104, 112.7922722) kemudian koordinat B (-7.2803934, 112.7918398) maka,

Menentukan variabel lat_1 yang merupakan garis *latitude*₁, lon_1 yang merupakan garis *longitude*₁, lat_2 yang merupakan garis *latitude*₂, lon_2 yang merupakan *longitude*₂.

$$lat_1 = -7.2808104$$

$$lon_1 = 112.7922722$$

$$lat_2 = -7.2803934$$

$$lon_2 = 112.7918398$$

Menentukan nilai selisih antar sumbu latitude dengan sumbu longitude,

$$\Delta lat = lat_2 - lat_1$$

$$\Delta lat = -7.2803934 - (-7.2808104)$$

$$\Delta lat = 0.000417 \dots \dots \dots (3.1)$$

$$\Delta lon = lon_2 - lon_1$$

$$\Delta lon = 112.7918398 - 112.7922722$$

$$\Delta lon = -0.0004324 \dots \dots \dots (3.2)$$

Setelah mendapatkan hasil selisih kedua nya, substitusikan hasil (3.1) dan (3.2) kedalam persamaan (2.5) sehingga menjadi,

$$d(A, B) = \sqrt{0.000417^2 + -0.0004324^2} \times 1.113195 \times 10^5$$

$$d(A, B) = 66.21 \text{ m}$$

Berdasarkan hasil perhitungan jarak antara koordinat A (-7.2808104, 112.7922722) dengan koordinat B (-7.2803934, 112.7918398) menggunakan *euclidean distance*, kedua titik memiliki jarak sejauh 66.21 meter.

3.9 Perhitungan Jarak Menggunakan *Manhattan Distance*

Seperti yang telah diuraikan pada bab 2, pembahasan sub bab 3.8 ini dimaksudkan untuk menguraikan bagaimana menghitung jarak antara dua titik yang memiliki koordinat menggunakan perhitungan *manhattan distance*. Jika diketahui koordinat A (-7.2808104, 112.7922722) kemudian koordinat B (-7.2803934, 112.7918398) maka,

Hasil selisih *latitude* dan *longitude* sudah didapatkan pada perhitungan sub bab 3.7 sehingga, langsung substitusikan nilai (3.1) dan (3.2) ke dalam persamaan (2.7) sehingga menjadi,

$$d(A, B) = (|0.000417| + |-0.0004324|) \times 1.113195 \times 10^5$$

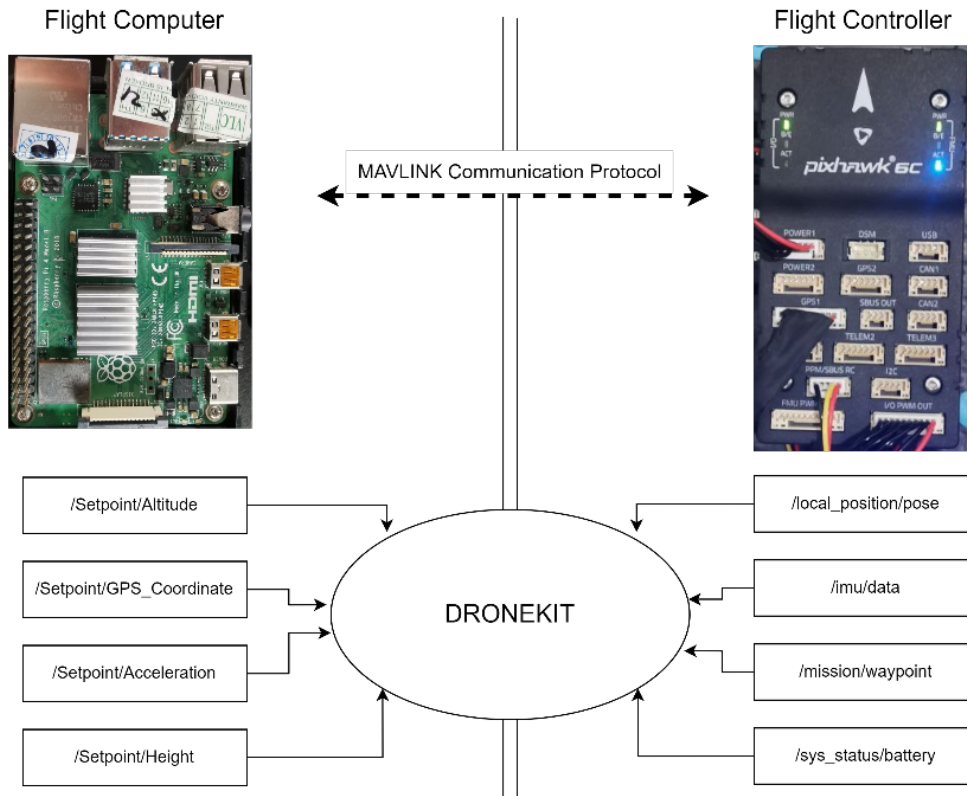
$$d(A, B) = 72.63 \text{ m}$$

Berdasarkan hasil perhitungan jarak antara koordinat A (-7.2808104, 112.7922722) dengan koordinat B (-7.2803934, 112.7918398) menggunakan *manhattan distance*, kedua titik memiliki jarak sejauh 72.63 meter. Berdasarkan hasil perhitungan haversine, euclidean dan manhattan dapat ditarik kesimpulan sementara bahwa manhattan distance memiliki perbedaan yang sangat jauh.

3.10 Integrasi *Flight Computer* dengan *Flight Controller*

Integrasi dilakukan agar terjalin komunikasi antara *flight controller* dengan *flight computer*. Komponen yang digunakan adalah *raspberry pi* sebagai *flight computer* dan *pixhawk 6C* digunakan sebagai *flight computer*. Seperti halnya pada mikrokontroler,

pixhawk digunakan untuk mengontrol *plant*, disini *plant* yang dimaksud ialah *X frame*, motor, dan driver motor. Berbeda dengan *flight controller*, *flight computer* bertugas untuk memberikan *command* kepada *flight controller* seperti, pengaturan radius pada *waypoint*, perhitungan *haversine formula*, perintah *take-off*, perintah *drop package*, perintah kembali ke *home point*, hingga perintah *landing*.



Gambar 3. 28 Komunikasi Antara Raspberry Pi 4 Dengan Pixhawk Melalui Komunikasi Serial (Sumber, Penulis)

Melalui komunikasi serial, pada Gambar 3.28 merupakan ilustrasi bagaimana *raspberry pi* dapat berkomunikasi dengan Pixhawk. Dengan menggunakan *API dronekit* maka kita dapat mengimpor nilai pembacaan sensor yang tersambung pada *flight controller*, sebaliknya urutan *command* yang terdapat pada *raspberry pi* dapat terbaca oleh *flight controller*. Kedua nya berjalan melalui protokol komunikasi bernama *Micro Air Vehicle Communication Protocol* atau dapat disebut juga sebagai *MAVLink*.

3.11 Kalkulasi Thrust Yang Diperlukan

Pada bagian ini sebelum menentukan *payload* maksimum yang dapat dibawa oleh *quadcopter* hendaknya melakukan perhitungan terlebih dahulu menggunakan rumus (2.8) dan (2.9). Beban secara keseluruhan quadcopter ialah 2.5Kg, seperti yang terlihat pada gambar 3.29, kemudian jika mengacu pada tabel 3.1 maka maksimum thrust per motor adalah 1620 gram. Sehingga perhitungannya menjadi,

$$motor\ thrust = \frac{total\ thrust}{total\ motor}$$

$$1620 = \frac{\text{thrust to weight ratio} \times \text{total weight}}{4}$$

$$1620 = \frac{2 \times \text{total weight}}{4}$$

$$\text{total berat} = \frac{1620 \times 4}{2}$$

$$\text{total berat} = 3240 \text{ gram}$$

Jika total berat yang dapat dibawa oleh *quadcopter* seberat 3240 gram dengan berat *quadcopter* sendiri 2500 gram seperti pada gambar 3.29 (a) dan 3.29 (b) maka total *payload* yang mampu dibawa *quadcopter* ialah

$$\text{payload} = \text{total weight} - \text{quadcopter weight}$$

$$\text{payload} = 3240 - 2500$$

$$\text{payload} = 740 \text{ gram}$$

Denagan demikian berdasarkan perhitungan diatas maka dapat ditarik kesimpulan, batas maksimum *payload* yang dapat dibawa oleh *quadcopter* menurut spesifikasi motor dan *propeller* yang dipakai ialah seberat 740 gram



(a)



(b)

Gambar 3. 29 Pengukuran Berat *Quadcopter* Menggunakan Timbangan
(Sumber, Penulis)

3.12 Pengambilan Data Pada *Pixhawk 6C*

Pada sub bab ini membahas bagaimana caranya mengambil data dari *pixhawk 6C*, ketika terbang *pixhawk 6C* secara otomatis merekam datanya kemudian menyimpan nya pada *sdcard* data tersebut dapat dilihat ketika *quadcopter* telah mendarat. Kemudian membukanya menggunakan *matlab* dengan apps *flightLogAnalyzer*.

3.12.1 Pengambilan Data GPS

Pengambilan data sensor GPS M8N melalui *embeded system* pada pixhawk 6c berupa *historical data*. Untuk mengaksesnya menggunakan program matlab dengan apps *flightLogAnalyzer*. Gambar 3.30 merupakan hasil visualisasi data yang terekam.

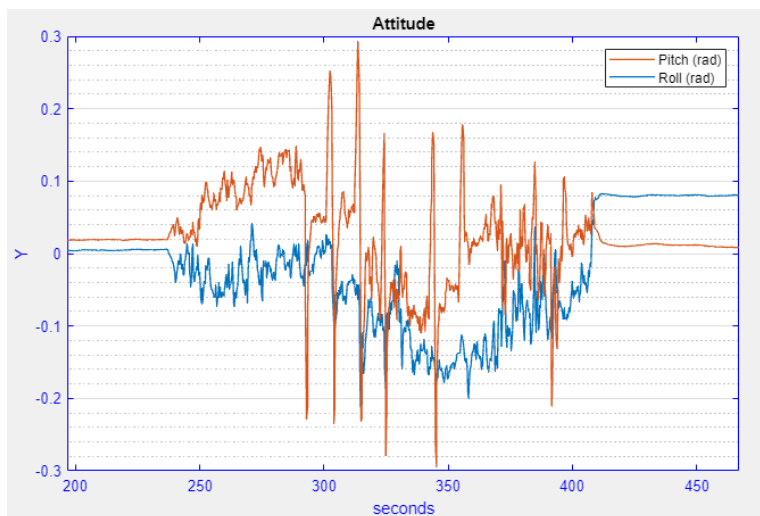


Gambar 3. 30 Visualisasi Data Historical GPS Menggunakan Matlab
(Sumber, Penulis)

Pada gambar 3.30 garis biru merupakan *historical data* dari *quadcopter*, titik merah merupakan tempat dimana *quadcopter* melakukan *takeoff* sedangkan titik biru tempat dimana *quadcopter landing*.

3.12.2 Pengambilan Data Sensor IMU

Historical data pada IMU berupa *attitude data*. Untuk mengaksesnya menggunakan program *matlab* dengan apps *flightLogAnalyzer*. Gambar 2.32 merupakan hasil visualisasi data yang terekam.

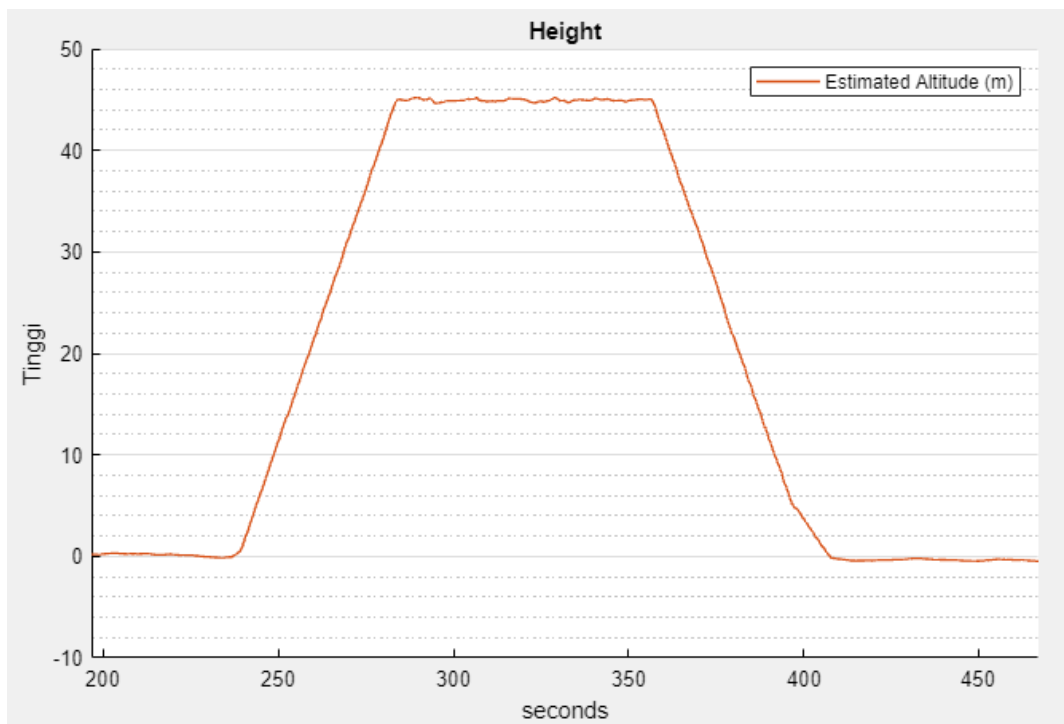


Gambar 3. 31 Visualisasi Data Imu Menjadi Attitude Menggunakan Matlab
(Sumber, Penulis)

Pada gambar 3.31 garis merah merujuk pada sudut *pitch* dalam satuan radian, sudut *pitch* mengindikasikan bahwa *quadcopter* bergerak maju terlihat bahwa sudut *pitch* mengalami kenaikan dan penurunan kedua keadaan ini menandakan bahwa *quadcopter* sedang berjalan maju dan mengerem untuk menstabilkan kecepatan. Garis biru merujuk pada sudut roll.

3.12.3 Pengambilan Data Ketinggian

Historical data ketinggian berupa *altitude* data. Untuk mengaksesnya menggunakan program *matlab* dengan apps *flightLogAnalyzer*. Gambar 2.33 merupakan hasil visualisasi data yang terekam.

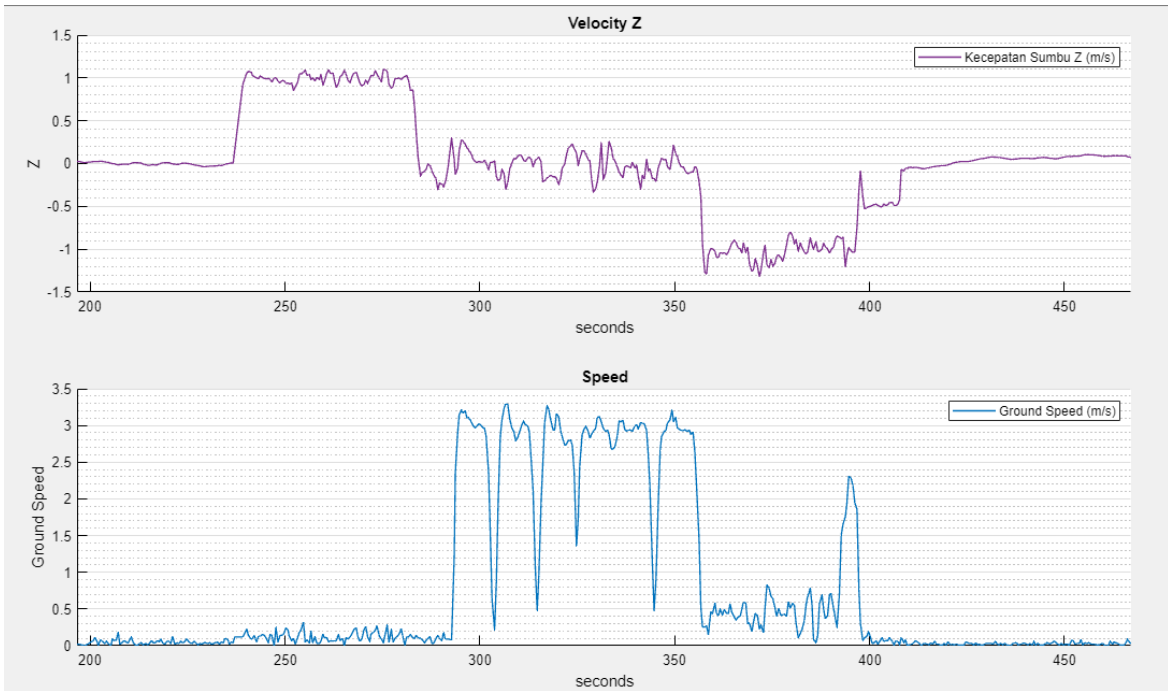


Gambar 3.32 Visualisasi Data Altitude Menggunakan Matlab
(Sumber, Penulis)

Pada gambar 3.32 merupakan visualisasi data dari *historical* ketinggian *quadcopter* garis oranye menunjukkan grafik ketinggian dengan satuan meter.

3.12.4 Pengambilan Data Kecepatan

Historical data kecepatan berupa *groundspeed* dan kecepatan pada sumbu Z. Untuk mengaksesnya menggunakan program *matlab* dengan apps *flightLogAnalyzer*. Gambar 2.33 merupakan hasil visualisasi data yang terekam. Pada gambar 3.33 saat takeoff di detik ke 250 terlihat bahwa kecepatan sumbu z naik hingga 1 m/s (grafik berwarna ungu) ketinggian *quadcopter* bertambah bisa dilihat pada gambar 3.32. Ketika *quadcopter* mencapai target takeoff nya (gambar 3.32) kecepatan sumbu z turun diiringi *groundspeed* naik (gambar 3.33 grafik warna biru). *Groundspeed* merupakan kecepatan yang terukur berdasarkan jarak dibagi satuan waktu



Gambar 3.33 Visualisasi Data Kecepatan Sumbu Z Dengan Groundspeed
(Sumber, Penulis)

“ Halaman Ini Sengaja Dikosongkan “

BAB IV HASIL DAN PEMBAHASAN

Bab ini akan membahas tentang hasil pengujian yang dilakukan terhadap sistem yang telah dirancang. Pembahasan yang ada pada bab ini merupakan pengujian dan analisa yang dilakukan untuk menguji fungsi dan keakuratan metode navigasi pada *quadcopter* menggunakan *waypoints* seperti yang dijelaskan pada tujuan awal.

4.1 Perbandingan Perhitungan *Haversine Formula*, *Euclidean Distance*, dan *Manhattan Distance*

Pengujian *haversine formula*, *euclidean distance*, dan *manhattan distance* dilakukan dengan cara membandingkan hasil perhitungan jarak antara 2 titik koordinat menggunakan *haversine formula*, *euclidean distance*, dan *manhattan distance* dengan pengukuran jarak manual menggunakan pita ukur tanah. Pada alat ukur manual menggunakan pita ukur tanah, memiliki ketelitian hingga 1 milimeter. Sedangkan koordinat yang akan di ukur didapatkan melalui aplikasi *tracking GPS* pada *smartphone* dengan tingkat akurasi sekitar 3.9 meter. Perbandingan pengukuran ini dilakukan guna mengetahui tingkat *error* yang terjadi jika pada perhitungan jarak antara 2 titik koordinat menggunakan *haversine formula*, *euclidean distance*, dan *manhattan distance*. Hasil terbaik dari pengujian dijadikan metode perhitungan jarak antara *quadcopter* dengan *waypoint*, sehingga dapat terbang secara *closeloop*.

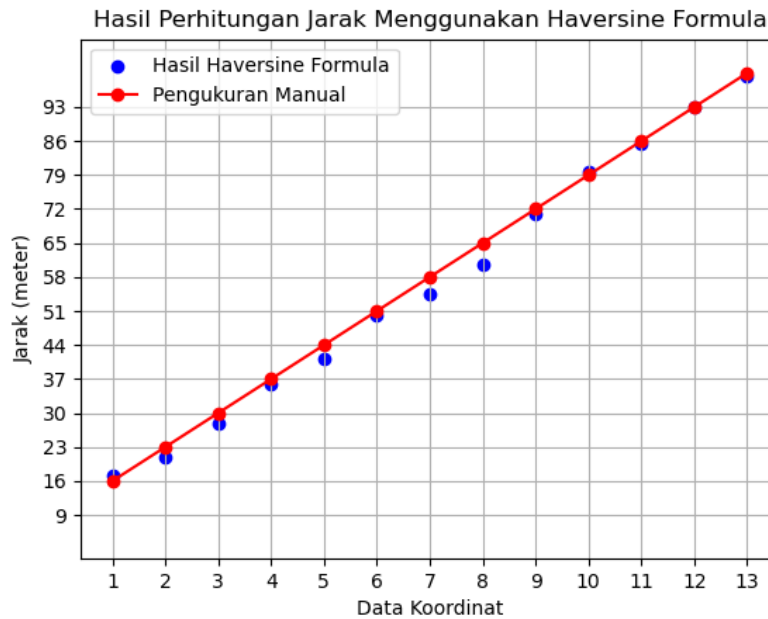
Tabel 4. 1 Perbandingan Perhitungan *Haversine Formula*, *Euclidean Distance* dan *Manhattan Distance* (Sumber, Penulis)

No.	Koordinat (lat,lon)	Pita Ukur Tanah (m)	<i>Haversine Formula</i> (m)	<i>Euclidean Distance</i> (m)	<i>Manhattan Distance</i> (m)
0	-7.3398261, 112.7900185	Titik 0			
1	-7.3399798, 112.7900212	16	17.09	17.09	17.41
2	-7.3400138, 112.7899946	23	21.03	20.82	23.55
3	-7.3400731, 112.7899742	30	27.89	27.10	32.42
4	-7.3401444, 112.7899636	37	35.90	35.21	41.54
5	-7.3401905, 112.7899508	44	41.20	40.91	48.1
6	-7.3402673, 112.7899264	51	50.09	50.11	59.36
7	-7.3403070, 112.7899210	58	54.54	54.56	64.38
8	-7.3403633, 112.7899283	65	60.55	60.37	69.84
9	-7.3404543, 112.7899021	72	71.02	70.89	82.88
10	-7.3405359, 112.7899142	79	79.75	79.47	90.62
11	-7.3405806, 112.7898742	86	85.39	85.38	100.05
12	-7.3406512, 112.7898759	93	93.05	92.88	107.72
13	-7.3407057, 112.7898595	100	99.36	99.17	115.61
<i>Mean Absolute Error</i>			1.61%	1.78%	7.65%

Perhitungan error pada tabel 4.1 menggunakan *mean absolute error* dengan cara mengukur rata-rata dari selisih mutlak antara pengukuran manual menggunakan pita ukur tanah dengan perhitungan jarak menggunakan *haversine formula*. Rumus perhitungan MAE diuraikan sebagai berikut

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \dots \dots \dots (4.1)$$

Dimana, $MAE = Mean\ Absolute\ Error$, $y_i =$ hasil *haversine*, *euclidean* dan *manhattan*, $x_i =$ pengukuran manual, $n =$ total dari banyaknya data.



Gambar 4.1 Hasil Plotting Data Dari Percobaan Pengujian Haversine Formula (Sumber, Penulis)

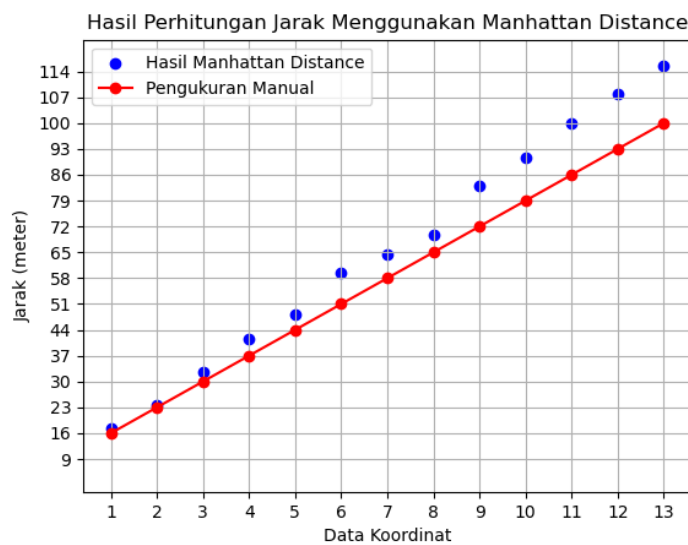
Pada gambar 4.1 pengumpulan data diambil dengan menggunakan pita ukur tanah yang memiliki panjang maksimal 100 meter. Pada tiap titik diberi jarak 7 meter agar referensi jarak yang terukur pada chart bersifat linear, sehingga dapat dengan mudah pada titik mana saja yang terdapat nilai error tinggi.



Gambar 4.2 Hasil Plotting Data Dari Percobaan Pengujian Euclidean Distance (Sumber, Penulis)

Gambar 4.2 merupakan hasil plotting dari perhitungan jarak menggunakan *euclidean distance* kemudian dibandingkan dengan pengukuran manual. Hasilnya tidak berbeda dengan gambar 4.1 yang menggunakan *haversine formula*. Dari perhitungan nilai error pun menunjukkan perbedaan yang sedikit. Haversine formula memiliki nilai error sebesar 1.67% sedang kan euclidean distance memiliki nilai error sebesar 1.78%.

Selanjutnya gambar 4.3 merupakan hasil plotting dari perhitungan *manhattan distace*. Terlihat bawasanya nilai perhitungan jarak pada manhattan distance cenderung lebih besar jika dibandingkan dengan jarak pengukuran manual menggunakan pita ukukur tanah. Hasil ini juga sama jika dibandingkan dengan perhitungan manual yang dijabarkan pada bab 3.8 jika dibandingkan dengan perhitungan manual bab 3.7 dan 3.6.



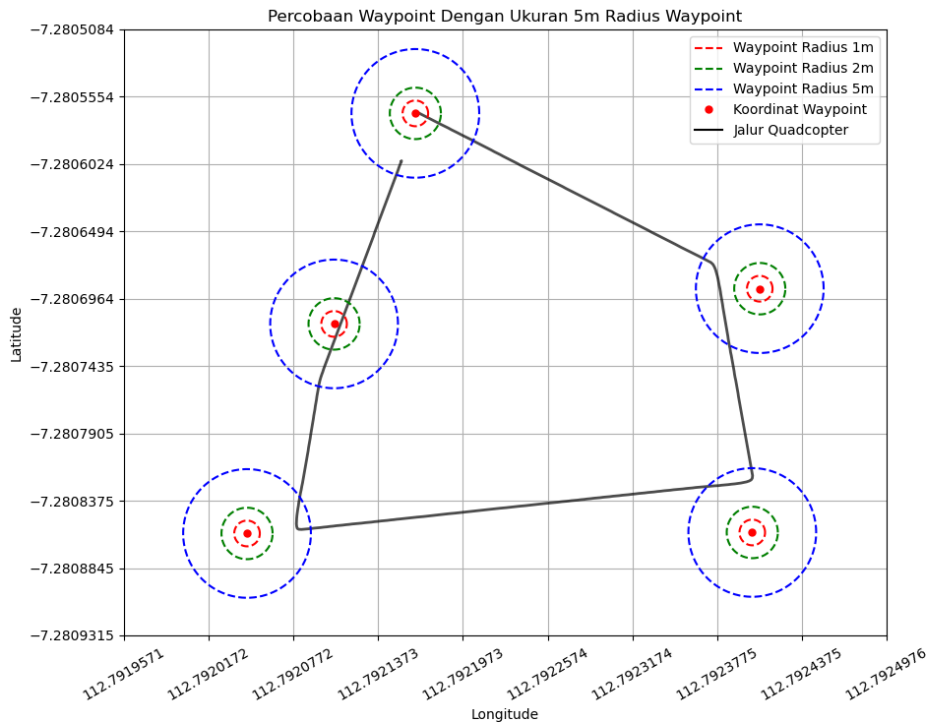
Gambar 4.3 Hasil Plotting Data Dari Percobaan Pengujian Manhattan Distance (Sumber, Penulis)

Error yang dihasilkan pada perhitungan manhattan distance pada gambar 4.3 semakin besar jika jarak bertambah. Dari tabel 4.1 nilai error yang dihasilkan jika menggunakan perhitungan manhattan distance ialah 7.65%.

4.2 Pengujian *Radius Waypoint* Pada Navigasi *Waypoint*

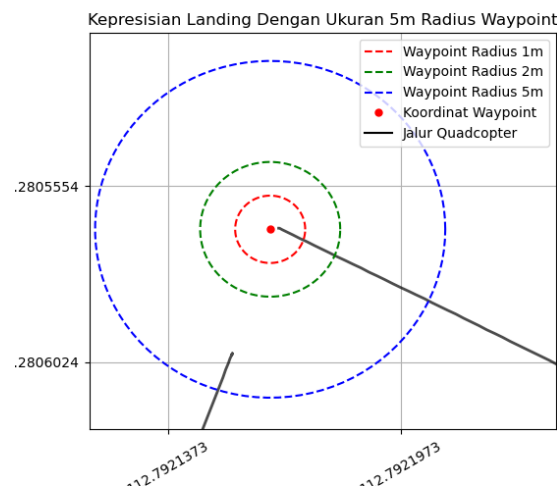
Dengan mengambil kesimpulan bahwa *haversine formula* lebih baik dari dua metode pengujian lainnya, maka pada sub bab 4.2 ini dilakukan pengujian navigasi waypoint menggunakan perhitungan *haversine formula*.

Pada tahapan pengujian ini dilakukan dengan cara merubah nilai radius pada *waypoint* nilai radius yang digunakan pada saat pengujian ialah 5 meter, 2 meter, dan 1 meter. Semakin kecil nilai radius maka semakin besar keakuratan dari penggunaan metode perhitungan *haversine formula* dalam navigasi *waypoint*. Tujuan dari merubah ukuran radius ialah untuk mencari kepresisian *quadcopter* terhitung dari *takeoff*, kemudian menjalankan misi *waypoint* dan yang terakhir kembali ke *home waypoint*. Kepresisian ini sangat berpengaruh pada *quadcopter* jika digunakan untuk mengirimkan paket, kepresisian dalam hal mengirim ke tujuan dan juga kembali ke *home waypoint*.

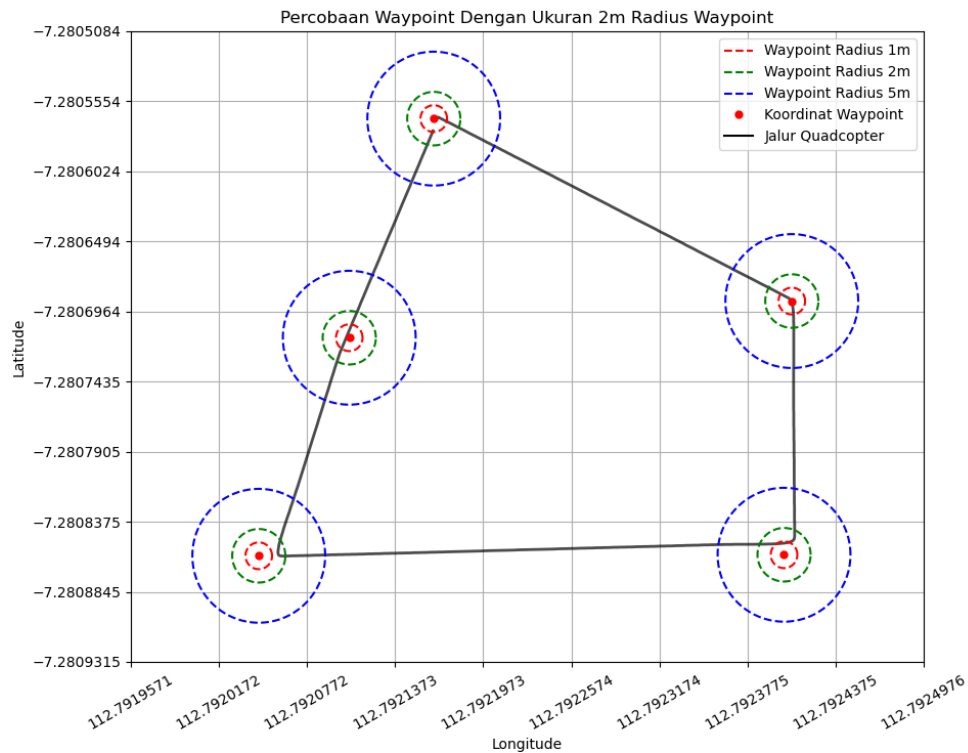


Gambar 4.4 Pengujian Radius 5m Pada Waypoint
(Sumber, Penulis)

Pada gambar 4.4 terlihat bahwa quadcopter cenderung memotong koordinat *waypoint* yang diberikan, hal ini terjadi karena radius *waypoint* yang diberikan nilai nya terlalu besar yakni 5 meter. Sisi baiknya *quadcopter* dapat mempersingkat *flight time* namun kepresisian terbang untuk menuju *target waypoint* jadi terganggu. Hal ini terlihat pada saat quadcopter memasuki *area radius waypoint* yakni garis biru yang merupakan radius 5 meter dari *waypoint*, *quadcopter* langsung mengubah arah nya kearah *waypoint* selanjutnya. Gambar 4.5 merupakan zoom in dari gambar 4. 4 menunjukkan lokasi *takeoff* dan *landing* menjadi berbeda.

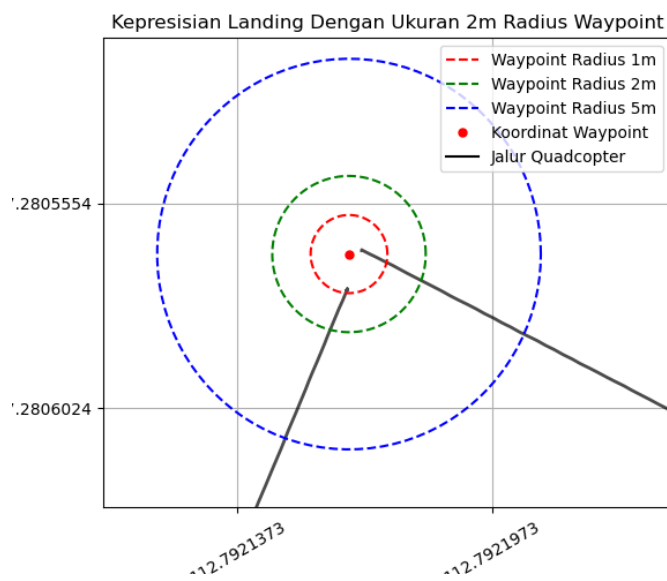


Gambar 4.5 Kepresisian Landing Pada Waypoint Radius 5m
(Sumber, Penulis)

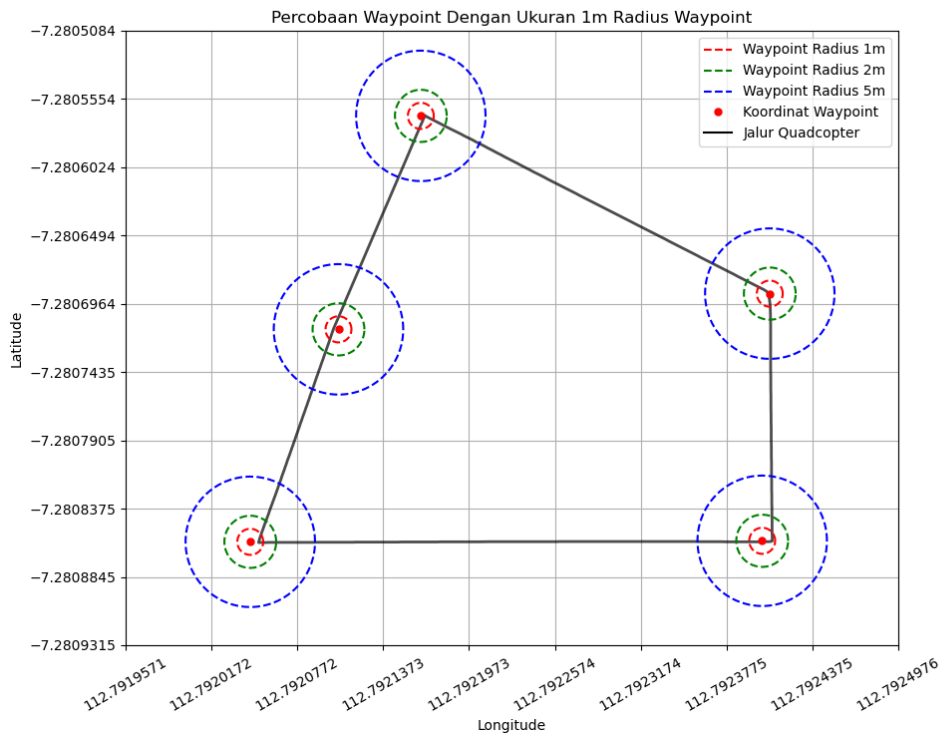


Gambar 4.6 Pengujian Radius 2m Pada Waypoint
(Sumber, Penulis)

Pada gambar 4.6 tingkat kepresisian quadcopter ketika terbang menuju target waypoint telah meningkat, terlihat di kelima titik pada gambar 4.6 quadcopter cenderung mendekati titik pusat yang merupakan titik waypoint. Pada gambar 4.7 jarak antara titik dimana quadcopter takeoff dan landing telah mengecil, hal ini mengindikasikan kepresisian telah meningkat.

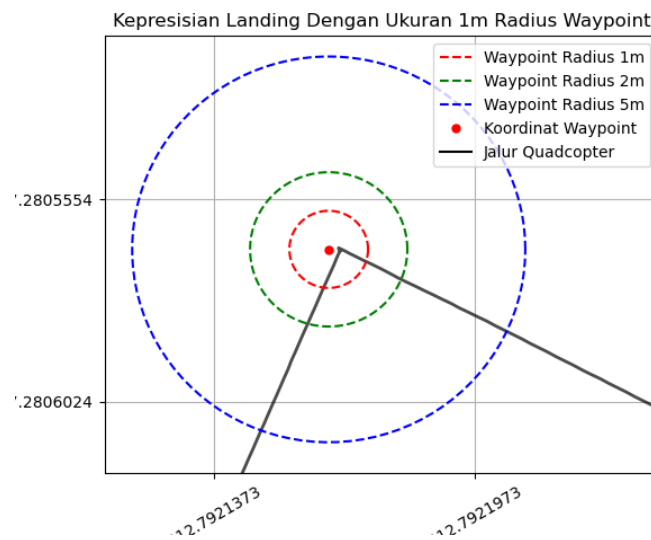


Gambar 4.7 Kepresisian Landing Pada Waypoint Radius 2m
(Sumber, Penulis)



Gambar 4. 8 *Pengujian Radius 1m Pada Waypoint*
(Sumber, Penulis)

Pada pengujian 1m *radius waypoint* seperti yang terlihat pada gambar 4.8 ketepatan *quadcopter* pada saat menuju *waypoint* semakin meningkat. Terlihat pada gambar 4.8 *quadcopter* berbelok menuju *waypoint* selanjutnya ketika telah mencapai radius area 1m. Hal ini dibuktikan juga pada gambar 4.9 ketepatan lokasi *landing* dan *takeoff* berada di titik yang relatif sama. Gambar 4.9 merupakan *zoom-in* dari gambar 4.8 dengan lokasi waypoint paling atas. Dimana *takeoff* dan *landing* terjadi.



Gambar 4. 9 *Kepresisian Landing Pada Waypoint Radius 1m*
(Sumber, Penulis)

Kesimpulan yang didapat setelah melakukan pengujian berbagai radius pada *waypoint* ialah semakin kecil *radius waypoint* yang diberikan maka semakin presisi quadcopter pada saat terbang menuju *target waypoint*, namun *flight time* menjadi sedikit lebih lama dikarenakan *quadcopter* tidak dapat memotong jalur, seperti yang terlihat pada gambar 4.4.

4.3 Pengujian Tanpa *Payload* Dan Menggunakan *Payload*

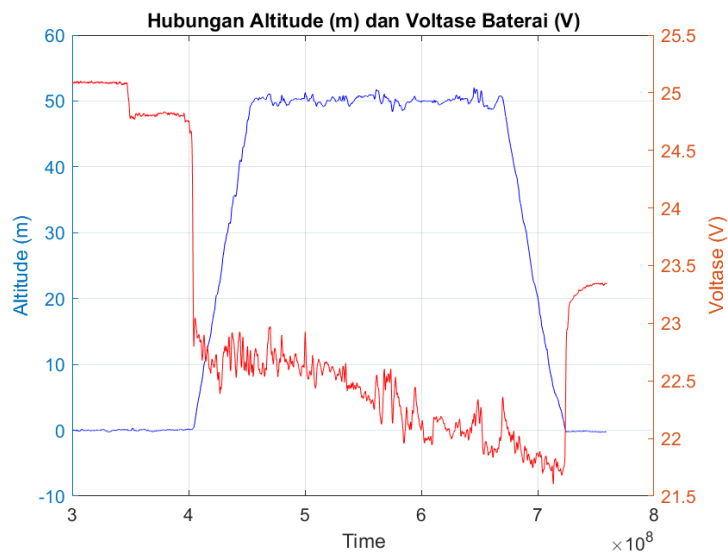
Pengujian ini dilakukan guna mengetahui bagaimana keadaan *quadcopter* jika terbang tanpa *payload* dan terbang dengan membawa *payload*. Dengan menganalisa *drop voltage* pada baterai Li-Po.

4.3.1 Pengujian Tanpa Menggunakan *Payload*



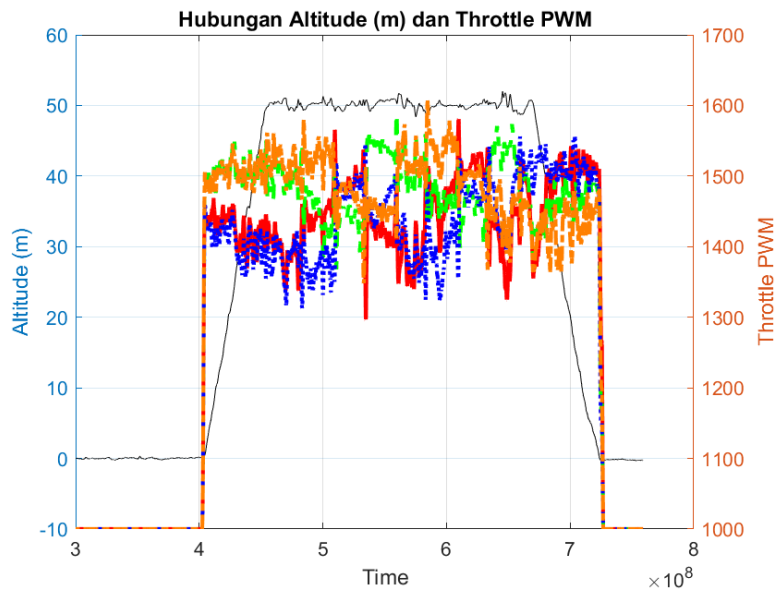
Gambar 4. 10 *Path History Pengujian Tanpa Payload*
(Sumber, Penulis)

Garis biru pada gambar 4.10 menunjukkan jalur yang telah dilalui oleh *quadcopter*. Gambar tersebut didapatkan dari *historical data GPS* yang kemudian direkam melalui *ground station* kemudian disimpan dalam bentuk *telemetry log*. Untuk mengaksesnya menggunakan program *matlab*.



Gambar 4. 11 *Hubungan Altitude Dan Voltase Baterai Tanpa Payload*
(Sumber, Penulis)

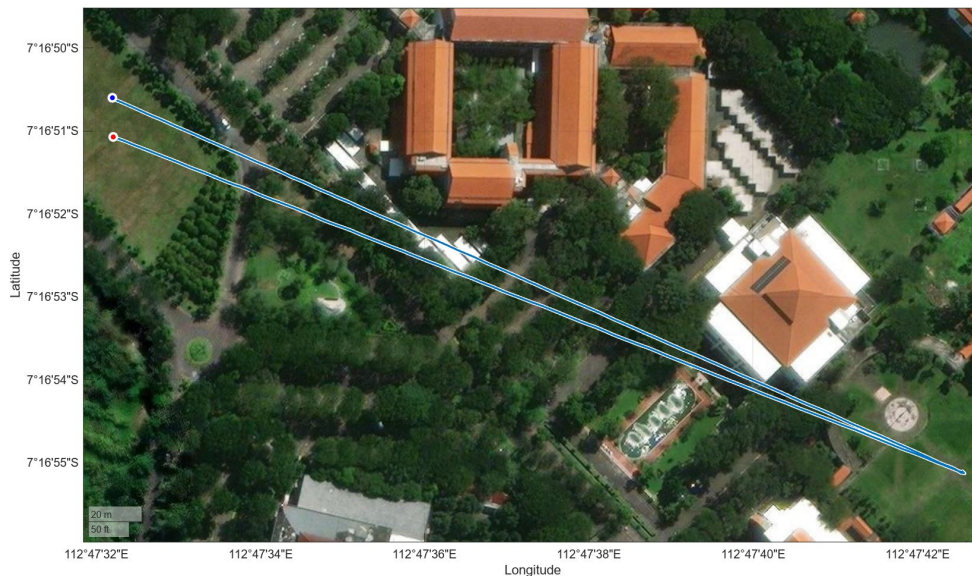
Jika melihat gambar 4.11 pada saat *quadcopter* terbang tanpa membawa *payload* tegangan baterai turun tidak kurang dari 21.5 volt, jika dibandingkan dengan gambar 4.14 tegangan baterai turun dibawah 21.5 hal ini menandakan bahwa kinerja motor tidak berat.



Gambar 4.12 Hubungan Altitude Dan Throttle PWM Tanpa Payload
(Sumber, Penulis)

Gambar 4.12 menunjukkan bahwa *throttle* yang diberikan dari *controller* ke *driver motor* hanya berkisar antara 1300 hingga 1600, dengan rentang perbedaan yang kecil menunjukkan bahwa kinerja motor tidak berat.

4.3.2 Pengujian Payload 385 Gram



Gambar 4.13 Path History Payload 385 Gram
(Sumber, Penulis)

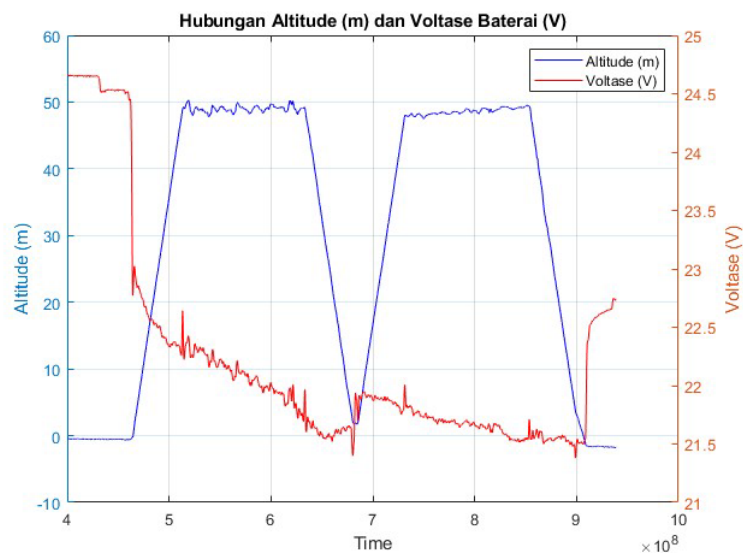
Garis biru pada gambar 4.13 menunjukkan jalur yang telah dilalui oleh *quadcopter*. Gambar tersebut didapatkan dari *historical data GPS* yang kemudian direkam melalui *ground station* kemudian disimpan dalam bentuk *telemetry log*. Untuk mengaksesnya

menggunakan program *matlab*. Terlihat bahwa jalur yang dilalui melenceng hal ini dikarenakan manual input pada *home waypoint*. Pada pengujian selanjutnya diberikan beban dengan berat 385gr terlihat pada gambar 4.14



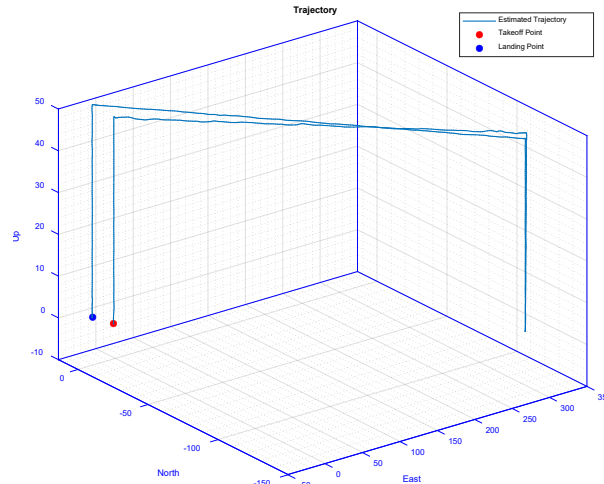
Gambar 4. 14 *Payload 385 Gram*
(Sumber, Penulis)

Gambar 4.14 merupakan hasil pengukuran berat *payload* yang digunakan pada pengujian ini dengan berat 385 gram.



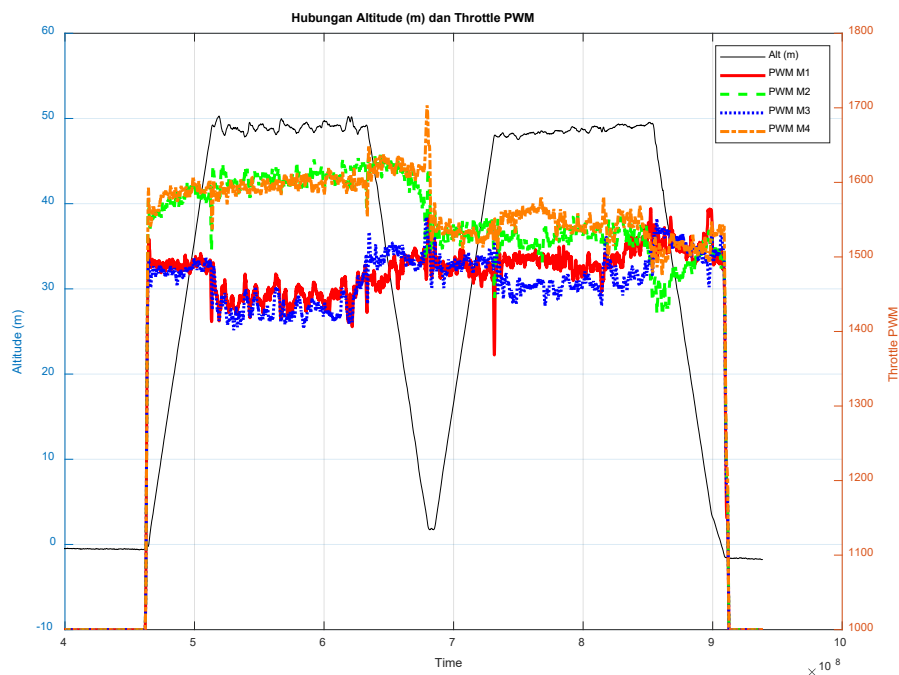
Gambar 4. 15 *Hubungan Altitude Dan Voltase Baterai Dengan Payload 385 Gram*
(Sumber, Penulis)

Pada gambar 4.15 terlihat bahwa baterai mengalami penurunan tegangan yang sangat drastis ketika *quadcopter* digunakan untuk membawa *payload*, tegangan semakin turun hingga level 21.5 volt pada saat *quadcopter* melakukan terbang rendah sebelum mengantarkan paket ke atas permukaan tanah. Ini terjadi dikarenakan *quadcopter* menahan beban sekaligus gaya gravitasi. Setelah paket dijatuhkan maka tegangan kembali naik namun karena *quadcopter* membutuhkan *power* untuk kembali terbang pada ketinggian 50m, secara perlahan tegangan kembali turun. Setelah semua proses selesai tegangan kembali ke level 23 volt.



Gambar 4. 16 3D Trajectory
(Sumber, Penulis)

Pada gambar 4.16 menjelaskan *historical trajectory* terlihat bahwa *quadcopter* bergerak mengikuti jalur dari *waypoint* yang diberikan pada gambar 4.13 pergerakan *quadcopter* cenderung lurus tanpa berbelok ditujukan guna memperkecil jarak tempuh serta, ketinggian 50 meter ditujukan guna menghindari bangunan dan beragam vegetasi dengan tinggi yang berbeda, karena *quadcopter* belum memiliki fitur *obstacle avoidance*.



Gambar 4. 17 Hubungan Altitude (m) dan Throttle PWM
(Sumber, Penulis)

Pada gambar 4.17 posisi m1 dan m3 berada di depan sedangkan posisi m2 dan m4 berada di belakang. Pada saat pengujian dilakukan posisi payload pada *quadcopter* terletak di *frame* bagian belakang. Hal ini menyebabkan bagian belakang menjadi lebih berat. Dapat terlihat pada chart bahwa kerja m2 dan m4 semakin berat dilihat dari nilai pwm yang diberikan *flight controller* ke ESC selisih rata rata nilai PWM antara M1, M3 dan M2, M4 berkisar 100 hingga 170.

4.3.3 Pengujian Payload 708 Gram



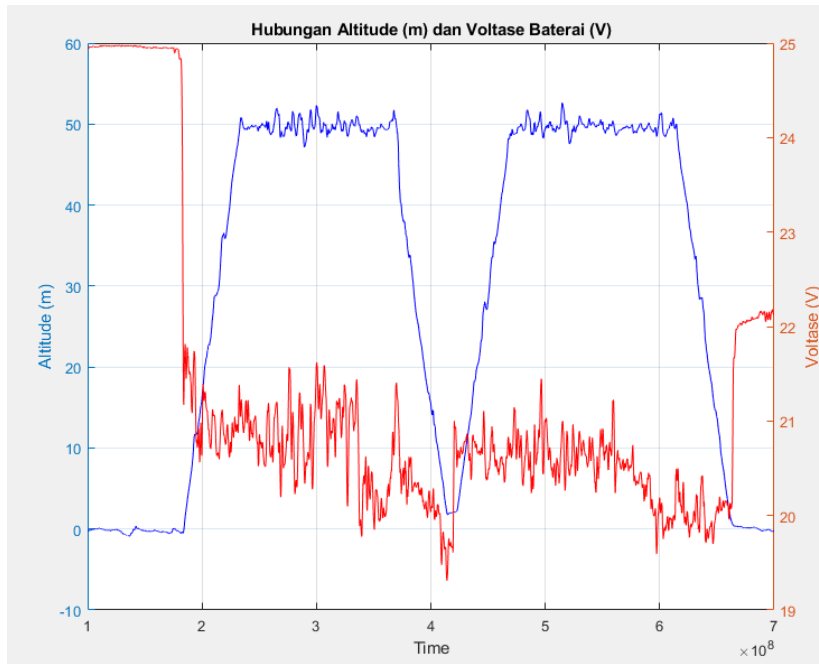
Gambar 4. 18 *Path History Payload 708 Gram*
(Sumber, Penulis)

Pada gambar 4.18 garis biru merupakan hasil historical data sensor GPS yang di plotting pada peta. Titik merah merupakan titik *takeoff* sedangkan titik biru merupakan titik *landing*.



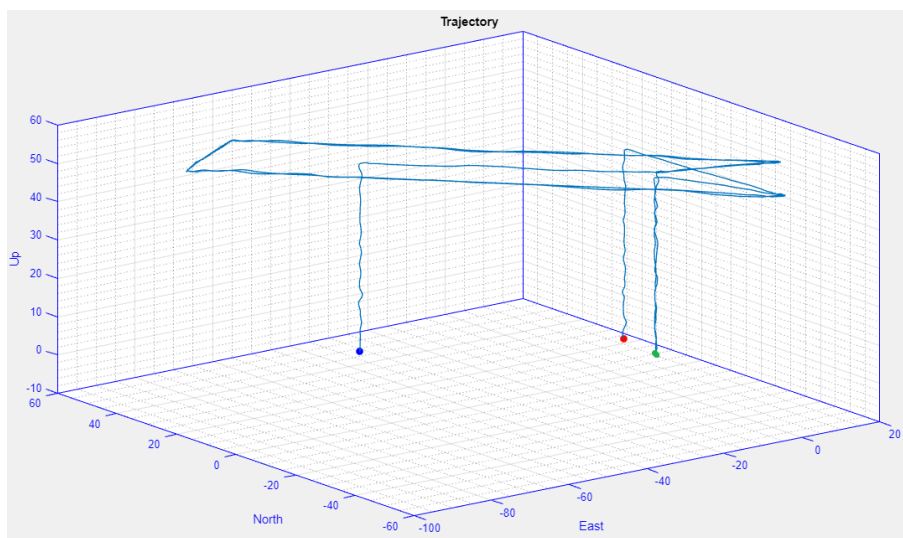
Gambar 4. 19 *Payload 708 Gram*
(Sumber, Penulis)

Dengan mengacu pada hasil kalkulasi *thrust* pada sub bab 3.11, pada percobaan ini menggunakan *payload* seberat 708 gram (gambar 4.19). Gambar 4.18 merupakan *historical data quadcopter* terbang, titik merah merupakan titik *takeoff* sedangkan titik biru merupakan titik *landing*. Percobaan tidak dilakukan long range karena ada kekhawatiran gagal terbang dikarenakan tegangan baterai yang turun. Jarak pada percobaan ini sejauh 400 meter.



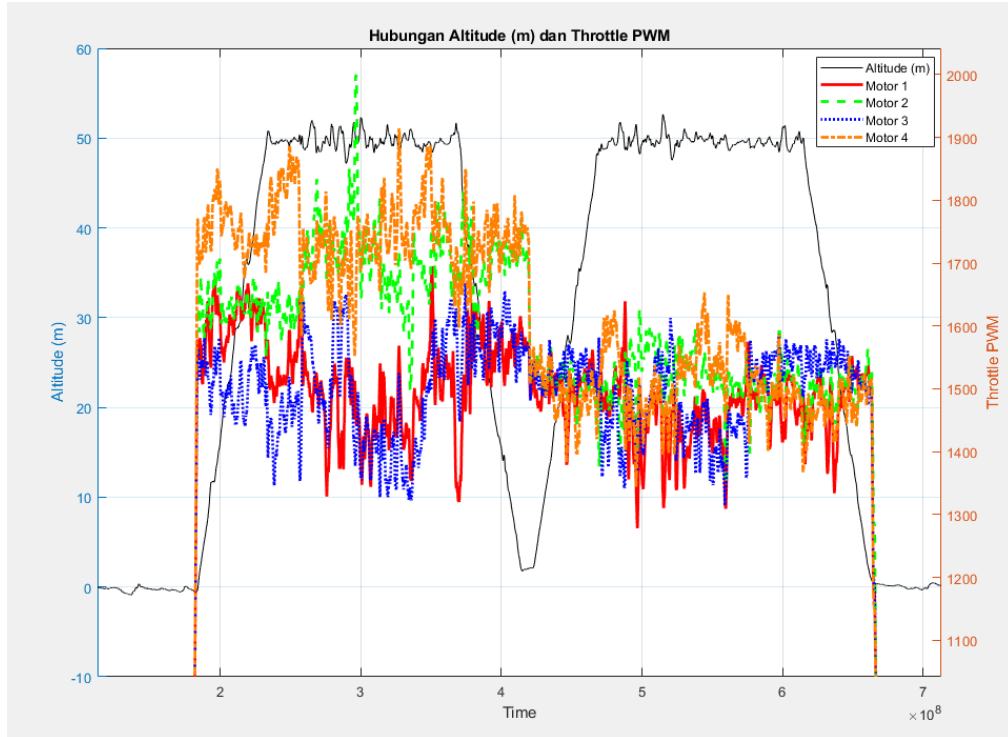
Gambar 4. 20 Hubungan Altitude Dan Voltase Baterai Dengan Payload 708 Gram
(Sumber, Penulis)

Jika membandingkan kedua *chart* pada gambar 4.20 dengan gambar 4.15 voltase baterai pada gambar 4.20 turun dengan sangat signifikan hal ini dikarenakan payload yang dibawa memiliki 708 gram hampir 2 kali lipat berat payload pada pengujian pertama. Pada keadaan turun untuk mengantarkan barang, voltase baterai drop hingga 19.5 volt hal ini sangat tidak aman karena karakteristik baterai *Li-Po* hanya mampu memberikan daya dengan minimum voltase sebesar 20.5 volt. Jika dipaksakan secara berulang maka umur dari baterai tidak bertahan lama.



Gambar 4. 21 3D Trajectory
(Sumber, Penulis)

Gambar 4.21 merupakan *historical 3D trajectory data*, titik merah merupakan lokasi *takeoff*, titik hijau merupakan lokasi *drop* barang, titik biru merupakan lokasi *landing*.



Gambar 4. 22 Hubungan Altitude (m) dan Throttle PWM
(Sumber, Penulis)

Pada gambar 4.22 *throttle* pada saat membawa *payload* terlihat lebih agresif jika dibandingkan dengan gambar 4.14, terlihat pada motor 2 di gambar 4.19 nilai pwm mencapai 2000.

“ Halaman Ini Sengaja Dikosongkan “

BAB V PENUTUP

Penutup merupakan bagian terakhir dari buku proyek akhir yang berisi tentang kesimpulan dari pengujian dan analisa yang telah dilakukan. Adapula saran terhadap proyek akhir ini untuk yang dapat digunakan untuk acuan membangun dan mengembangkan untuk penelitian selanjutnya.

5.1 Kesimpulan

Kesimpulan yang didapat dari proyek akhir dengan judul penerapan *haversine formula* dalam *navigasi waypoint* pada *package delivery quadcopter* sebagai berikut:

- Navigasi waypoint dapat dilakukan dengan mengatur radius pada titik waypoint.
- Pengujian jarak dengan cara membandingkan nilai pengukuran menggunakan pita ukur tanah dengan perhitungan jarak menggunakan *haversine formula* memiliki nilai MAE yang rendah yaitu sebesar 1.6%.
- Dengan menggunakan *Haversine Formula*, hasil perhitungan jarak antara *quadcopter* dengan *target waypoint* dijadikan acuan sebagai konfirmasi bahwa *quadcopter* telah berada di dalam *radius waypoint*.
- Nilai *radius waypoint* yang paling presisi ialah 1 meter.
- Setelah melakukan perhitungan kapasitas berat maksimum pada *quadcopter*, *payload* yang dapat dibawa memiliki berat maksimum 740 gram.
- Dari pengujian *payload* yang dilakukan dapat diambil kesimpulan bahwa semakin besar *payload* maka ketahanan baterai semakin berkurang.

5.2 Saran

Saran untuk memperbaiki sistem penerapan *haversine formula* dalam *navigasi waypoint* pada *package delivery quadcopter* sebagai berikut:

- Dibutuhkan nya fitur *obstacle avoidance* agar *quadcopter* dapat terbang secara *fully autonomous*.
- Membutuhkan beberapa preset parameter PID karena *payload* yang dibawa bervariasi.
- Menggunakan motor, *propeller*, baterai dan *frame* yang lebih besar supaya kapasitas *payload* atau menjadi lebih besar.

“ Halaman Ini Sengaja Dikosongkan “

DAFTAR PUSTAKA

- Ardupilot. (2022). Pixhawk 6C 6C Mini Flight Controller. <https://ardupilot.org/copter/docs/common-holybro-pixhawk6C.html>
- Ardupilot. (2020). Choosing a Ground Station. <https://ardupilot.org/copter/docs/common-choosing-a-ground-station.html>
- Dan Rose. (2020). Micro Air Vehicle Message Marshalling Library. <https://github.com/mavlink/mavlink/blob/master/README.md>
- Dinogy. (2017). 6S Li-Po Catalogue. <https://www.dinogylipos.com/collections/6s-lipos/6s>
- Hobbywing. (2019). XRotor 40A. <https://www.hobbywing.com/en/products/info.html?id=122>
- Holybro. (2022). M8N & M9N GPS. https://docs.px4.io/main/en/gps_compass/gps_holybro_m8n_m9n.html
- Holybro. (2022). Pixhawk 6C Ports. <https://docs.holybro.com/autopilot/pixhawk-6c/pixhawk-6c-ports>
- ICAO. (2018). Waypoint Definition. <https://skybrary.aero/articles/waypoint>
- J. Pace, & L. He. (2020). Estimating Altitude of Drones Using Batteries. 2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI). pp. 264-265, <https://doi.org/10.1109/IoTDI49375.2020.00039>.
- J. Song, H. Yuan, and B. Zhang. (2022). Pixhawk-based scalable platform for multi-purpose surface unmanned vehicle. 2022 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST). pp. 1577-1580, <https://doi.org/10.1109/IAECST57965.2022.10061960>.
- Jessica F. (2022). Latest Pixhawk Products by Holybro. <https://www.motionew.com/blog/flight-controllers/latest-pixhawk-products/>
- K. Priandana, M. Hazim, Wulandari and B. Kusumoputro. (2020). Development of Autonomous UAV Quadcopters using Pixhawk Controller and Its Flight Data Acquisition. International Conference on Computer Science and Its Application in Agriculture (ICOSICA). pp.1-6. <https://doi.org/10.1109/ICOSICA49951.2020.9243289>.
- Lauren Nagel. (2021). Types of Drones and UAVs. <https://www.tytorobotics.com/blogs/articles/types-of-drones>
- Lisa Eitel. (2018). Brushless dc motors for industrial power tools: Comparison of options including slotless motor variations. <https://www.motioncontroltips.com/motors-industrial-power-tools-comparison-of-brushless-dc-options-including-slotless-motor-variations/>
- Mariane. (2019). Lost but lovely: The haversine. <https://plus.maths.org/content/lost-lovely-haversine>
- PX4. (2020). Basic Flying. https://docs.px4.io/v1.8.2/en/flying/basic_flying.html
- R. Moeller, T. Deemyad and A. Sebastian. (2020). Autonomous Navigation of an Agricultural Robot Using RTK GPS and Pixhawk. 2020 Intermountain Engineering, Technology and Computing (IETC). pp. 1-6,

- <https://doi.org/10.1109/IETC47856.2020.9249176>.
- Raspberrypi, (2018). Raspberry pi 4. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
- Rahmat H., & Ronny M. (2018). Pengembangan Sistem Navigasi Otomatis Pada UAV (Unmanned Aerial Vehicle) dengan GPS (Global Positioning System) Waypoint. *Jurnal Teknik ITS* Vol.5, No.2, ISSN: 2337-3539
- Sharad Pachpute. (2020). Working Principle and Components of Drone. <https://cfdflowengineering.com/working-principle-and-components-of-drone/>
- Tarotrc, (2021). IRON MAN 650 Foldable Quad-copter Frame (plastic landing skid)TL65B02. <http://www.tarotrc.com/Product/Detail.aspx?Lang=en&Id=e9df4318-5564-449c-aae4-d51c5683a6e0>
- Tarotrc. (2021). Tseries1555 Carbon Fiber Propeller CW/CCW TL2831. <http://www.tarotrc.com/Product/Detail.aspx?Lang=en&Id=4fee5fc3-b655-42c8-b624-b557afcb8d2d>
- Tarotrc. (2021). IRON MAN 650 Foldable Quad-copter Frame TL65B01. <http://www.tarotrc.com/Product/Detail.aspx?Lang=en&Id=f960c8a1-cf4d-4339-adb0-8c54a29e7ea9>
- Tim Gunther. (2021). Illustration of globe with three satellites triangulating a precise location. <https://education.nationalgeographic.org/resource/triangulation-sized/>
- Yusup M., Sofia U., Fahmi Rabiul K. (2020). Perbandingan Metode Perhitungan Jarak Euclidean, Haversine, dan Manhattan Dalam Penentuan Posisi Karyawan. *Jurnal Tekno Insentif* ISSN (p): 1907-4964 ISSN <https://doi.org/10.36787/jti.v14i2.270>

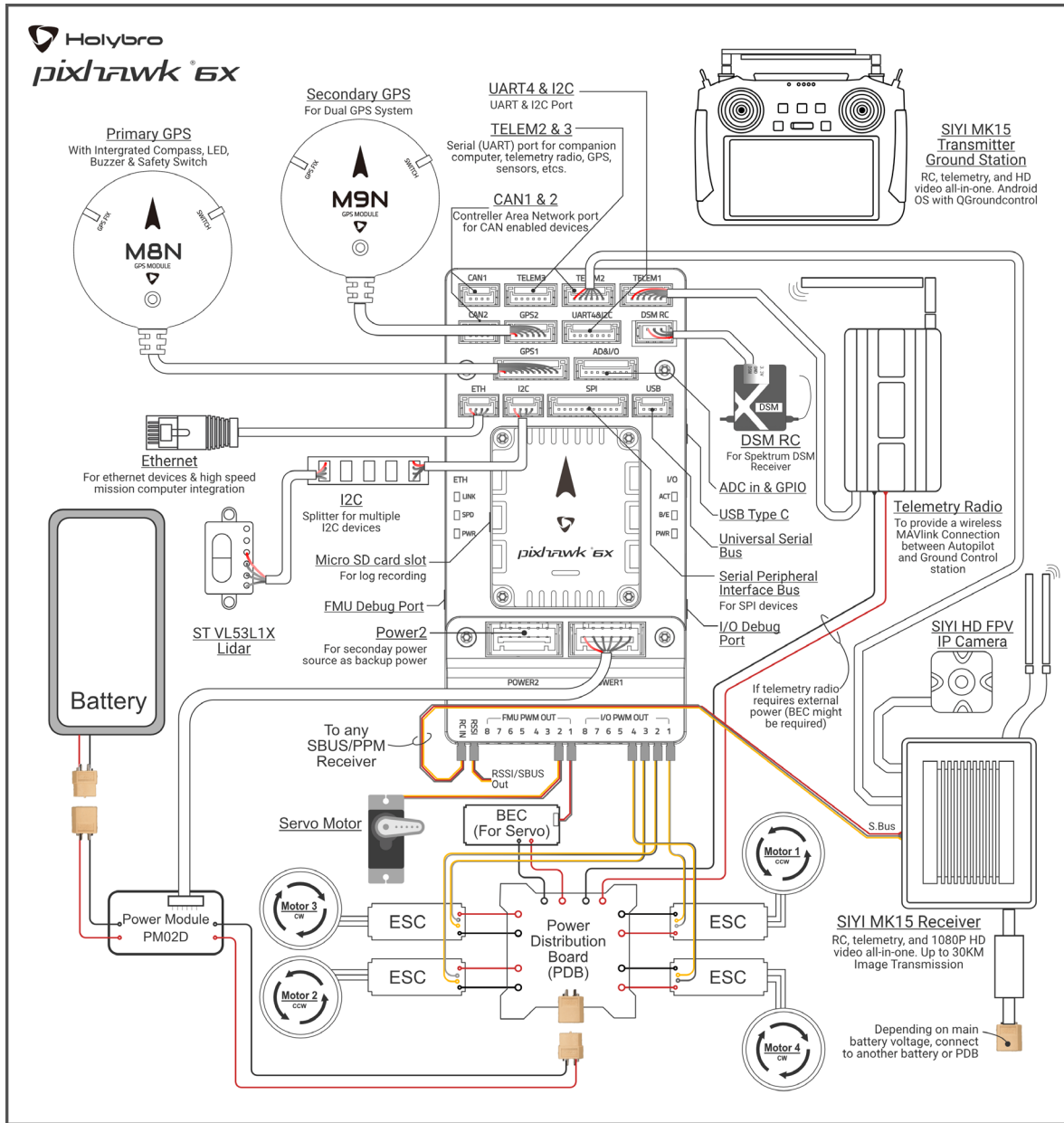
LAMPIRAN

Lampiran 1 Data Sheet Motor

Tabel Data Sheet Motor 6S 380KV 4108 Dengan Propeller Ukuran 13x5.5 inch
(Tarotrc, 2020)

TAROT 6S 380KV 4108							
电压 (V)	旋翼尺寸	电流 (A)	推力 (G)	功率 (W)	效率 (G/W)	转速 (RPM)	温度 (°C)
22.2V	1355	1	260	22.8	11.4	2980	
		2	460	45.6	10.0	3600	
		3	650	68.4	9.5	4230	
		4	770	91.2	8.4	4660	
		5	860	114	7.5	4970	
		6	980	136.8	7.1	5270	
		7	1090	159.6	6.8	5520	
		8	1180	182.4	6.4	5830	
		9	1310	205.2	6.3	5970	
		10	1380	228	6.0	6200	
		11	1440	250.8	5.7	6330	
		12	1530	273.6	5.5	6470	
		12.4	1620	282.7	5.7	6560	44

Lampiran 2 Manual Installation Instruction



Gambar Manual Installation Pada Pixhawk Controller
(Holybro Dev Team, 2022)

BIODATA MAHASISWA



Penulis yang memiliki nama lengkap Maulana Zuhri ini merupakan anak pertama dari 2 bersaudara. Penulis dilahirkan di Surabaya, 09 April 1995. Pada tahun 2010, penulis menyelesaikan Pendidikan tingkat menengah pertama di SMP Negeri 13 Surabaya. Tahun 2013 berhasil menyelesaikan pendidikan tingkat menengah atas di SMA Negeri 15 Surabaya. Pada tahun 2016 penulis telah menyelesaikan studi di Diploma III Elektro ITS kemudian pada tahun 2022 penulis melanjutkan studi di Departemen Teknik Rekayasa Otomasi Fakultas Vokasi ITS Surabaya dengan NRP 2040221159