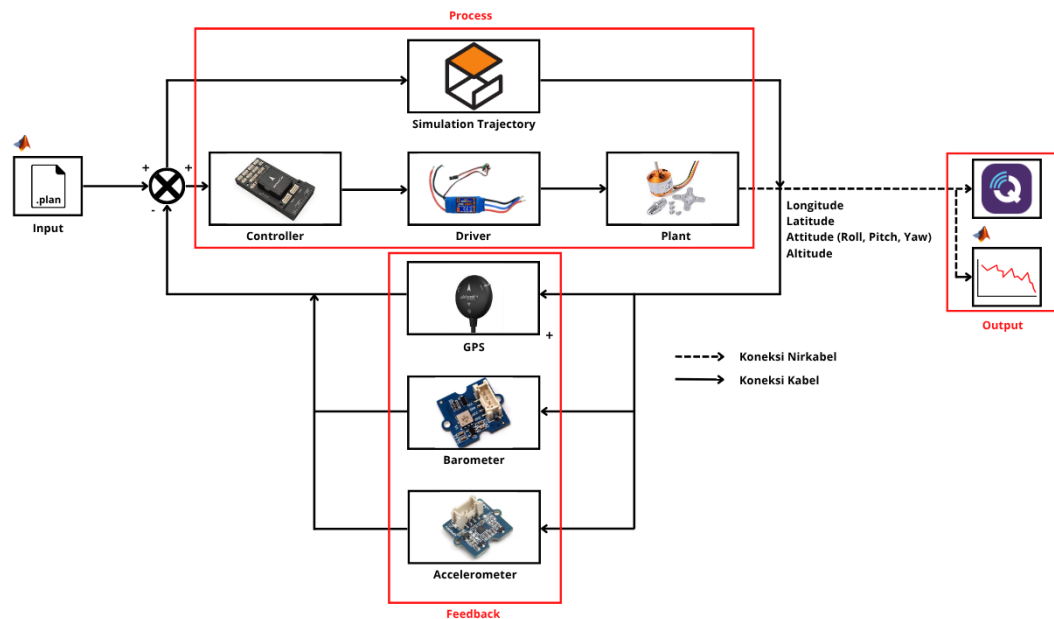


BAB III

METODOLOGI PENYELESAIAN MASALAH

Pada bab ini, akan diuraikan mengenai gambaran umum sistem, peninjauan sistem, dan diagram alur kerja yang digunakan. Selain itu, pembahasan juga mencakup berbagai metodologi tugas akhir yang diterapkan untuk mengatasi masalah yang dihadapi dalam penyelesaian Tugas Akhir ini

III.1 Gambaran Umum Sistem



Gambar III. 1 Blok Diagram Sistem

Gambar III.1 menjelaskan arsitektur sistem secara keseluruhan, dimulai dari masukan berupa berkas rencana misi (.plan) yang telah dihasilkan oleh aplikasi perancangan di MATLAB. Berkas misi ini kemudian menjadi acuan untuk dieksekusi dalam dua skenario pengujian yang berbeda: simulasi dan implementasi nyata pada wahana. Dalam skenario simulasi, berkas (.plan) diunggah ke lingkungan *Software-In-The-Loop* (SITL) menggunakan *simulator* Gazebo, yang memodelkan fisika dan lingkungan penerbangan secara virtual untuk memvalidasi logika misi tanpa risiko.

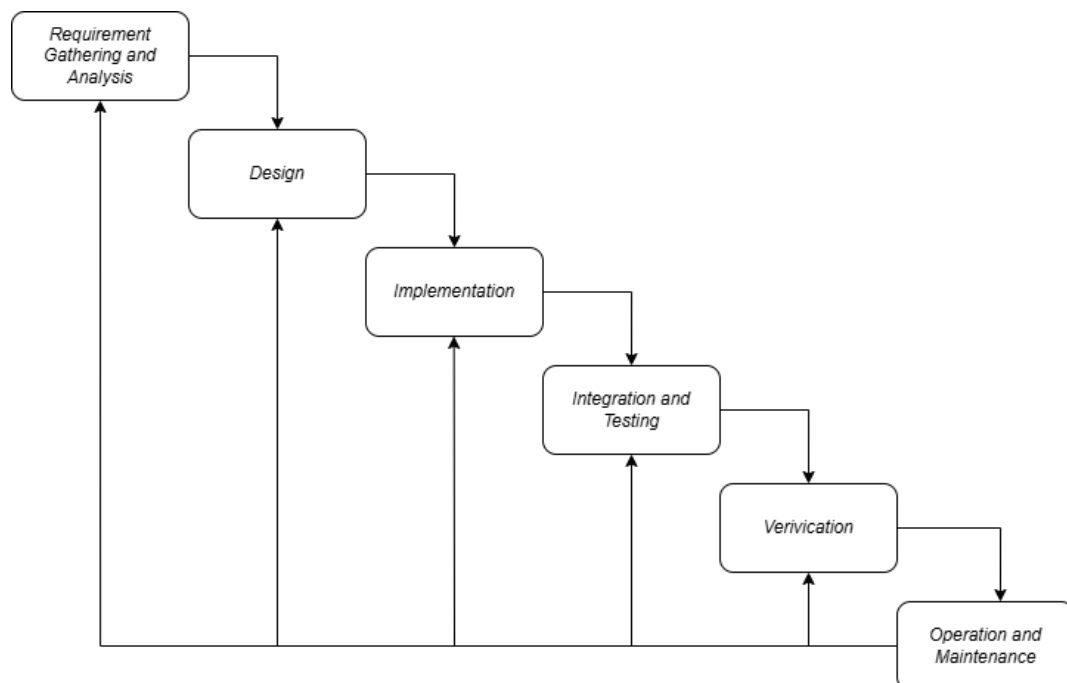
Dalam skenario implementasi nyata, berkas misi diunggah ke kendali utama wahana, yaitu *Flight Controller* (FC) Pixhawk. FC kemudian menerjemahkan setiap perintah dalam misi, seperti navigasi ke titik koordinat, menjadi sinyal

kontrol. Sinyal ini dikirimkan ke *Electronic Speed Controller* (ESC) yang bertugas mengatur kecepatan putaran Motor BLDC untuk menggerakkan wahana. Selama penerbangan, FC secara berkelanjutan menerima data umpan balik dari serangkaian sensor untuk memastikan stabilitas dan akurasi navigasi. Modul GPS menyediakan data posisi (lintang dan bujur), sementara sensor IMU (yang terintegrasi di dalam FC) memberikan data orientasi (*roll, pitch, yaw*).

Baik dari hasil simulasi maupun penerbangan nyata, data telemetri yang mencakup posisi, ketinggian, orientasi, dan kecepatan wahana akan dikirimkan kembali ke stasiun darat. Data ini kemudian ditampilkan secara *real-time* pada antarmuka QGroundControl (QGC) untuk pemantauan visual dan secara bersamaan dicatat (*logging*) oleh aplikasi kustom di MATLAB untuk analisis performa pasca-penerbangan.

III.2 Pengembangan Produk

Tugas akhir ini menggunakan metode *Waterfall* dalam proses pengembangan produknya. Metode ini memiliki beberapa tahapan yang secara garis besar dijelaskan pada Gambar III. 2



Gambar III. 2 Skema Metode *Waterfall*

Metode *Waterfall* merupakan model pengembangan aplikasi klasik yang menekankan pada tahapan yang berurutan dan sistematis. Model ini dapat

dianalogikan seperti air terjun, di mana setiap tahap dikerjakan secara berurutan mulai dari atas hingga ke bawah. Tahapan-tahapan dalam metode *Waterfall* meliputi:

1. *Requirement and Gathering Analysis*
2. *System and Software Design*
3. *Implementation*
4. *Integration and System Testing*
5. *Operation and Maintenance*

1. *Requirement and Gathering Analysis*

Tahap ini merupakan tahap pengumpulan kebutuhan secara komprehensif untuk dianalisis dan menetapkan kebutuhan yang harus dicapai oleh program. Informasi dapat diperoleh melalui wawancara, diskusi, membaca studi literatur atau survei.

III.2.1.1 Survei dan Wawancara

Untuk memahami kebutuhan dan tantangan nyata di lapangan, dilakukan sesi wawancara dengan seorang praktisi berpengalaman di industri *drone*. Wawancara ini bertujuan untuk mengumpulkan wawasan dari sudut pandang *operator* dan pengembang mengenai alur kerja, kesulitan, dan kebutuhan fungsional dalam proses perencanaan misi *drone*.



Gambar III. 3 Survei dan Wawancara

Responden dalam wawancara ini adalah Sdr. Haekal, seorang pilot *drone* bersertifikat dan *operator* di sebuah perusahaan penyedia jasa *drone* komersial dari PT. BETA. Hasil dari wawancara dirangkum dalam bentuk tabel sebagai berikut.

Tabel III. 1 Hasil Wawancara

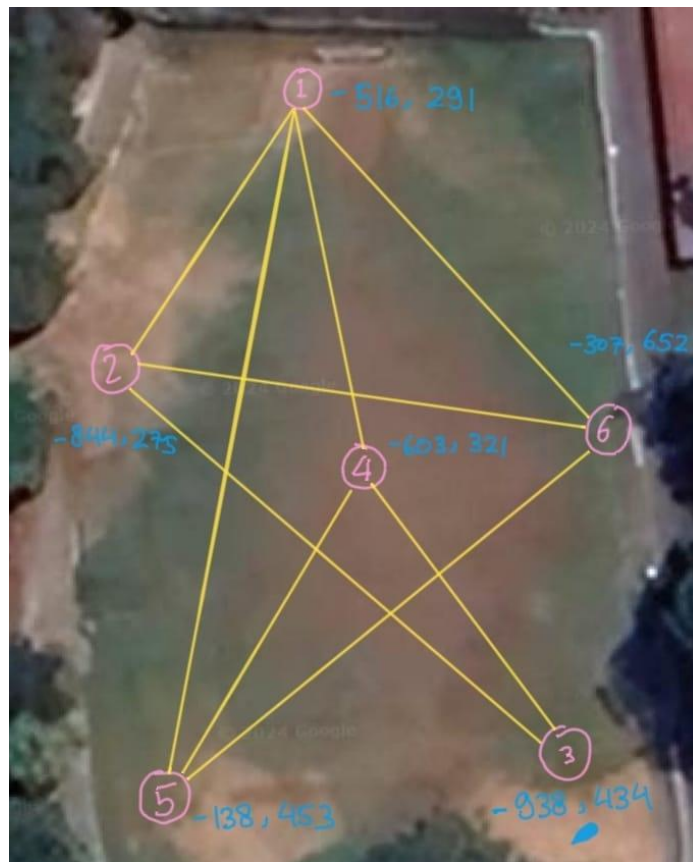
No	Pertanyaan	Jawaban
1	Apa saja hal yang diperlukan dalam penerbangan <i>drone</i> otomatis?	Selain dari unit <i>drone</i> itu sendiri, pilot perlu GCS(<i>Ground Control System</i>) atau kendali darat untuk mengatur penerbangan, serta RC (<i>Radio Control</i>) sebagai <i>safety factor</i> .
2	Bagaimana pilot menentukan rute terbaik atau terpendek antar titik?	Tidak ada sistem seperti itu, hanya berdasarkan visual dan intuisi saja.
3	Lalu bagaimana pilot membuat misi penerbangan <i>drone</i> ?	Biasanya pilot melakukannya di software QGC (QGroundControl), ada fitur yang dapat digunakan yaitu (plan). Dari fitur tersebut dapat disimpan secara luring dengan format (.plan).
4	Apakah pilot dapat melihat status <i>drone</i> ketika melakukan penerbangan?	Bisa, tapi kami kesulitan untuk menganalisisnya karena hanya berbentuk angka realtime saja, belum dapat ditampilkan dalam grafik.
5	Apakah hasil penerbangan tersebut tidak dapat di analisis sama sekali?	Bisa, tapi pilot perlu mengeluarkan SD Card yang ada pada FC, lalu meng ekstrak file (.ulg) yang ada didalamnya.

Berdasarkan hasil wawancara, didapatkan bahwa perangkat lunak standar industri seperti QGroundControl (QGC) memang telah memiliki kapabilitas dasar untuk merencanakan, mengendalikan, hingga memantau penerbangan. Namun, untuk tahap perencanaan misi yang kompleks, QGC masih memiliki kekurangan. Proses pembuatan rute multi-titik harus dilakukan secara manual dengan menghubungkan titik satu per satu, dan yang terpenting, QGC tidak menyediakan fitur optimasi untuk secara otomatis menghitung dan merekomendasikan jalur terpendek antar titik. Hal ini menyebabkan proses perencanaan menjadi tidak efisien dan hasilnya

sangat bergantung pada intuisi *operator*, yang berisiko menghasilkan rute yang suboptimal. Kesenjangan fungsionalitas inilah yang mendasari kebutuhan akan sebuah sistem perancangan misi khusus yang mampu mengatasi kekurangan tersebut.

III.2.1.2 Lintasan *Drone*

Lintasan drone berupa graf yang memiliki bobot di setiap jalurnya (*Weighted Graph*). Gambar III.4 Merupakan contoh graf berupa lintasan yang akan digunakan untuk melakukan pengujian.



Gambar III.4 Contoh Lintasan yang akan digunakan

Drone map akan terdiri dari *waypoints* berdasarkan *Longitude* dan *Latitude* dari peta satelit, Sedangkan *trajectory* yang digunakan adalah jarak garis lurus antar *waypoint*. *Drone map* akan digunakan di lingkungan Lembah POLMAN Bandung.

III.2.1.3 Algoritma Perencanaan Rute

Sistem ditargetkan dapat membuat peta dan menghitung jalur terpendek dalam berbagai skenario dan memiliki akurasi yang tinggi. Oleh karena itu perlu dilakukan riset berupa studi literatur untuk dapat menentukan algoritma *path-planning* apa

yang dapat diterapkan secara fleksibel, dan dapat mencari jalur terpendek dengan akurasi yang tinggi.

Berdasarkan studi literatur, terdapat beberapa algoritma pencarian graf (*graph search*) yang umum digunakan, di antaranya adalah Algoritma A^* , Algoritma *Bellman-Ford*, dan Algoritma *Dijkstra*. Algoritma A^* seringkali lebih cepat dalam menemukan rute karena menggunakan fungsi heuristik untuk memprioritaskan pencarian ke arah tujuan. Namun, kelemahannya adalah keoptimalan hasilnya sangat bergantung pada kualitas fungsi heuristik tersebut dan tidak selalu menjamin jalur terpendek absolut[30]. Di sisi lain, Algoritma *Bellman-Ford* memiliki keunggulan dalam menangani graf dengan bobot sisi negatif, sebuah kondisi yang tidak relevan dalam tugas akhir ini karena semua bobot merepresentasikan jarak yang selalu positif[31].

Setelah membandingkan, Algoritma *Dijkstra* dipilih sebagai metode inti dalam tugas akhir ini. Berbeda dengan A^* , *Dijkstra* tidak menggunakan estimasi, melainkan menjelajahi graf secara sistematis untuk menjamin penemuan rute terpendek yang paling optimal secara matematis pada graf dengan bobot non-negatif[32]. Mengingat tujuan utama tugas akhir ini adalah untuk merancang sebuah sistem yang andal dan dapat divalidasi keakuratannya, maka kepastian untuk mendapatkan rute yang paling optimal menjadi prioritas utama dibandingkan potensi kecepatan komputasi. Oleh karena itu, keunggulan *Dijkstra* dalam hal keandalan dan jaminan keoptimalan hasil menjadikannya pilihan yang paling sesuai untuk tugas akhir ini.

III.2.1.4 Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam perancangan sistem ini adalah sebuah unit *drone* dengan daftar komponen pada Tabel III.2.

Tabel III. 2 Daftar Komponen

No.	Nama Komponen	Tipe	Fungsi Dalam Sistem
1	Kerangka (<i>Frame</i>)	S500 <i>Quadcopter</i> <i>Frame</i>	Menyediakan struktur utama wpahana yang kokoh dan ringan.
2	<i>Flight Controller</i>	Pixhawk 4	Otak kendali utama, memproses data dan mengeksekusi misi.
3	<i>Electric Speed Control</i> (ESC)	SimonK 30A	Mengatur kecepatan putaran setiap motor BLDC.
4	Motor	BLDC 900kV	Menghasilkan gaya dorong untuk penerbangan.
5	<i>Propeller</i>	DJI Phantom 3 Genuine	Mengubah putaran motor menjadi gaya angkat aerodinamis.
6	Baterai	Li-Po Gold Racing 4000mAh 3S	Sumber daya listrik utama untuk seluruh sistem.
7	Sensor Posisi	Modul GPS (Bawaan Pixhawk)	Menyediakan data lokasi geografis untuk navigasi.
8	Komunikasi Data	Holybro SiK Telemetry V3 433MHz	Jembatan komunikasi data <i>MAVLink</i> antara <i>drone</i> dan GCS.
9	Kendali Manual	Radiolink AT10 dan R12DS <i>Transmitter</i>	Pengaman Sistem

III.2.1.5 Kebutuhan Perangkat Lunak

Perancangan dan implementasi sistem perencanaan misi ini didukung oleh serangkaian perangkat lunak yang dipilih secara spesifik berdasarkan fungsi dan keunggulannya masing-masing. Perangkat lunak utama yang digunakan dalam tugas akhir ini meliputi MATLAB, QGroundControl, Python, dan Gazebo.

1. MATLAB

MATLAB digunakan sebagai lingkungan pengembangan utama untuk aplikasi perencanaan misi. Pemilihan ini didasarkan pada tiga keunggulan utamanya yang sangat relevan dengan tugas akhir ini. Pertama, kekuatan fundamental MATLAB dalam komputasi berbasis matriks sangat ideal untuk mengimplementasikan algoritma berbasis teori graf. Kedua, fitur *App Designer* memungkinkan perancangan Antarmuka Pengguna Grafis (GUI) yang interaktif dan profesional dengan siklus pengembangan yang cepat. Ketiga, *Mapping Toolbox* yang terintegrasi menyediakan fungsi *geoplot* yang krusial untuk memvisualisasikan *waypoint* dan rute di atas peta satelit dunia nyata.

2. QGroundControl

QGroundControl (QGC) berfungsi sebagai Stasiun Kendali Darat (Ground Control Station) standar industri untuk *drone* yang menjalankan *firmware* PX4. Dalam tugas akhir ini, peran QGC sangat vital sebagai platform eksekusi dan validasi akhir. Berkas rencana misi (.plan) yang dihasilkan oleh aplikasi MATLAB secara spesifik diformat agar sepenuhnya kompatibel untuk diunggah dan dieksekusi melalui QGC. Selain itu, QGC juga digunakan untuk melakukan prosedur pemeriksaan pra-terbang (*pre-arm checklist*) yang penting untuk keselamatan, serta untuk memantau data telemetri secara langsung selama pengujian di lapangan.

3. Python

Di samping MATLAB sebagai platform utama, Python dengan *library* Pymavlink digunakan untuk tugas spesifik, yaitu pencatatan data telemetri mentah. Skrip python yang dikembangkan menggunakan editor Visual Studio Code (VSCode), berfungsi untuk mengambil data *MAVLink* dari *drone* dan menyimpannya ke dalam berkas log. Pendekatan ini dipilih karena Pymavlink memberikan akses tingkat rendah ke protokol *MAVLink*, yang sangat efisien untuk menangkap setiap pesan telemetri untuk keperluan analisis pasca-penerbangan (*post-flight analysis*).

4. Gazebo

Sebelum melakukan pengujian pada perangkat keras fisik, Gazebo *Simulator* digunakan sebagai lingkungan pengujian *Software-In-The-Loop* (SITL). Gazebo menyediakan simulasi robotika 3D yang realistis, lengkap dengan model fisika.

Penggunaan Gazebo pada tahap awal pengembangan sangat penting untuk memvalidasi fungsionalitas berkas .plan yang dihasilkan dan untuk menguji logika dasar algoritma dalam lingkungan yang aman dan terkendali, sehingga dapat mengurangi risiko kerusakan pada perangkat keras saat pengujian di lapangan.

III.2.1.6 Kebutuhan Fungsionalitas Sistem

Dalam tugas akhir ini, dibnutukan sistem perancangan misi memerlukan serangkaian fungsionalitas yang dapat memenuhi tujuan tugas akhir. Kebutuhan fungsionalitas tersebut dapat dikelompokkan sebagai berikut:

1. Manajemen Rute

Sistem yang akan dibuat harus menyediakan fungsi-fungsi berikut:

- a. Sistem harus mampu menampilkan visualisasi berupa peta berbasis satelit sebagai kanvas utama.
- b. Pengguna harus dapat menambahkan titik-titik lokasi atau *waypoint* secara interaktif dengan mengklik peta atau memasukkan koordinat.
- c. Pengguna harus bisa menghubungkan dua *waypoint* untuk menciptakan sebuah jalur penerbangan (*sisi/edge*), di mana sistem akan secara otomatis menghitung dan menampilkan jaraknya menggunakan Formula *Haversine*.
- d. Sistem harus memiliki fitur untuk menyimpan konfigurasi peta (kumpulan *waypoint* dan jalur) ke dalam sebuah berkas dan memuatnya kembali di lain waktu untuk efisiensi kerja.

2. Optimasi Rute

Sistem harus memiliki fungsionalitas yang mencakup:

- a. Pengguna dapat menentukan satu segmen misi dengan memilih titik awal dan titik tujuan dari daftar *waypoint* yang tersedia.
- b. Setelah titik awal dan tujuan dipilih, sistem harus secara otomatis menjalankan implementasi Algoritma *Dijkstra* pada graf yang sudah ada untuk menemukan rute terpendek.
- c. Hasil rute terpendek harus diberi antarmuk dengan jelas di atas peta, untuk memberikan visualisasi langsung kepada pengguna.
- d. Sistem harus memungkinkan pengguna untuk merangkai beberapa hasil pencarian rute terpendek sehingga *drone* dapat terbang ke berbagai titik.

3. Manajemen Misi

Sistem harus mampu berinteraksi dengan ekosistem *drone* standar.

- a. Sistem harus bisa mengubah hasil akhir perencanaan misi menjadi sebuah berkas berformat (.plan) yang terstruktur dan valid sesuai dengan standar yang dikenali oleh QGroundControl.
- b. Menyediakan fungsionalitas untuk memanggil skrip eksternal yang akan mengunggah berkas (.plan) yang telah dibuat ke GCS atau langsung ke wahana.

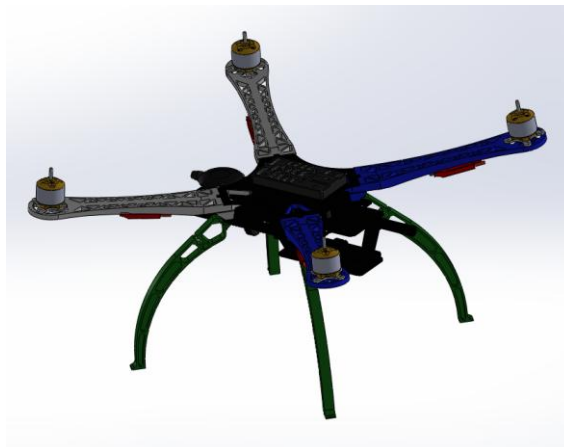
4. Pemantauan (*Monitoring*)

- a. Untuk keperluan validasi dan analisis, sistem harus menyediakan kemampuan pemantauan dasar:
- b. Koneksi Telemetri harus dapat terhubung dengan data *MAVLink* dari *drone*.
- c. Pencatatan Data (*Logging*): Sistem harus mampu menangkap dan menyimpan data telemetri penting ke dalam sebuah berkas untuk analisis pasca-penerbangan.

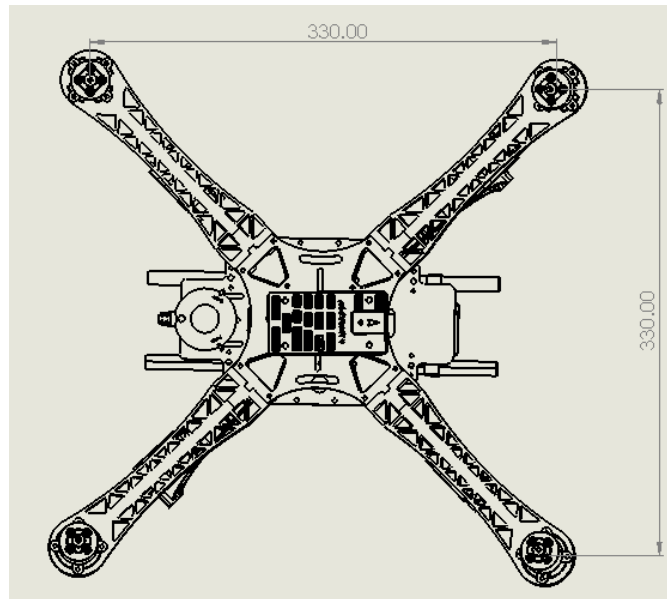
2. *System and Software Design*

III.2.2.1 Rancangan Alat

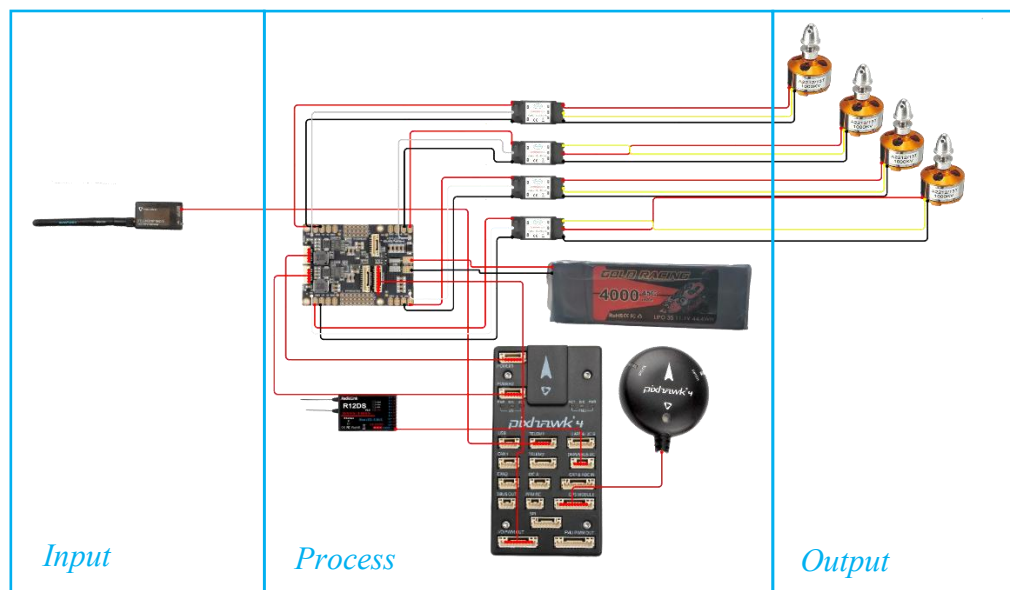
Berikut adalah rancangan mekanis dari unit *drone* yang akan digunakan dalam proyek tugas akhir ini.



Gambar III. 5 Rancangan 3D *Drone*



Gambar III. 6 Rancangan 2D Drone



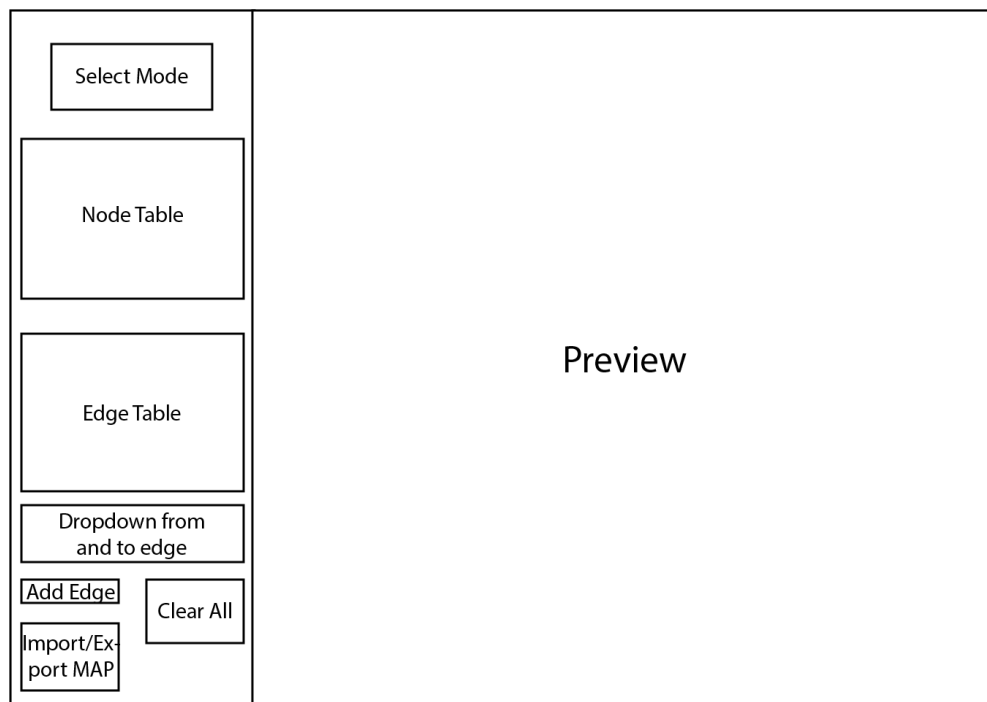
Gambar III. 7 Rancangan Elektrik

Gambar III.6 Merupakan rancangan Elektrik dari *drone*. Dari ESC ke motor *brushless* ada sebagian yang dibalik untuk memutar balikkan arah putaran motor . Baterai LiPo akan menjadi sumber daya untuk semua komponen yang terhubung, mulai dari motor, GPS, hingga Pixhawk 4. Untuk bisa menghubungkannya diperlukan *power module*. *Power module* akan terhubung ke setiap motor yang

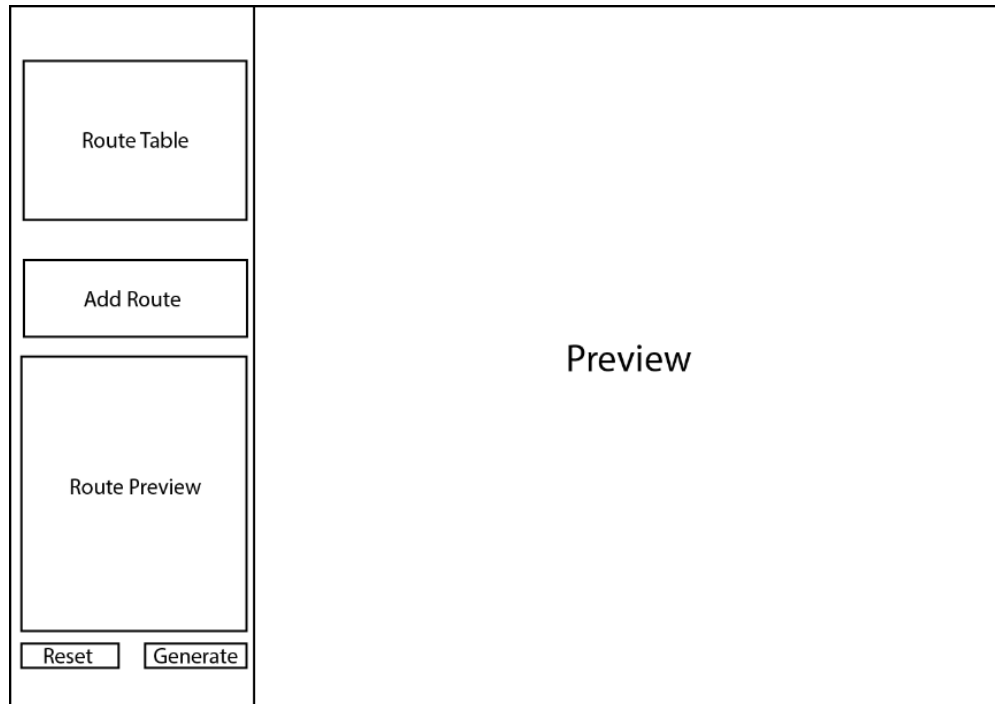
telah ditentukan sebelumnya, mulai dari motor 1 hingga motor 4. Tegangan untuk menyalakan Pixhawk, sensor dan protokol komunikasi disalurkan Melawati konektor power 1 dan power 2. Dari sana Pixhawk akan menyala dan menyalakan sensor-sensor yang ada. Lalu modul-modul yang terhubung seperti telemetry, RC, hingga GPS akan ikut menyala.

III.2.2.2 Rancangan Antarmuka (*Interface*)

Antarmuka digunakan agar *user* dapat menggunakan sistem dengan mudah. Antarmuka atau *interface* dibuat untuk semua orang. Tidak hanya untuk *operator* ataupun anak teknik. *User* yang dapat mmenggunakannya adalah siapapun yang ingin menggunakan sistem. Oleh karena itu antar muka di buat se-interaktif mungkin untuk memudahkan berbagai kalangan pengguna dan calon pengguna.

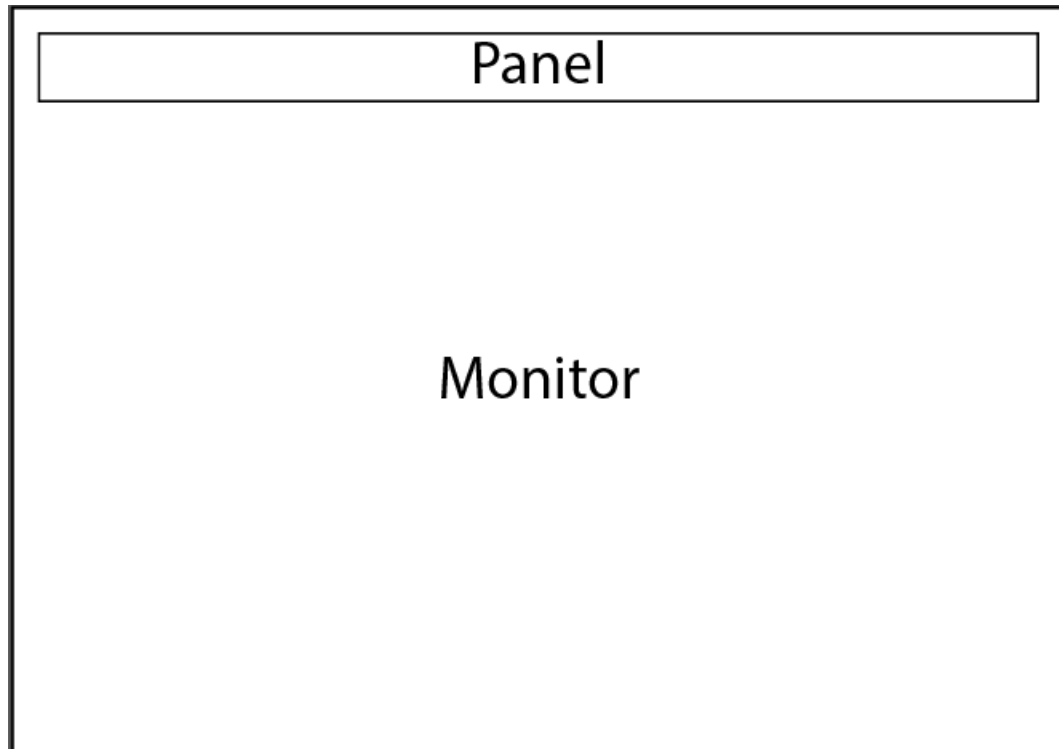


Gambar III. 8 Antarmuka *Generate Plan* halaman 1



Gambar III. 9 Antarmuka *Generate Plan* halaman 2

Generate Plan merupakan aplikasi yang digunakan untuk membuat peta terbang *drone* dan menghitung jalur terpendeknya. Pengguna dapat menggunakan peta satelit untuk memberikan Gambaran dimana pengguna memberikan titik *waypoint* nya. Halaman 1 merupakan halaman untuk membuat peta dan jalurnya. Pengguna dapat menyimpan peta secara *offline* dalam bentuk file dan dipanggil lagi nantinya untuk digunakan di lain waktu. Halaman 2 merupakan halaman tempat pengguna dapat mencari jalur tercepat dari titik awal ke titik tujuan dengan jalur terpendek yang diberikan dan menyimpannya secara *offline*.



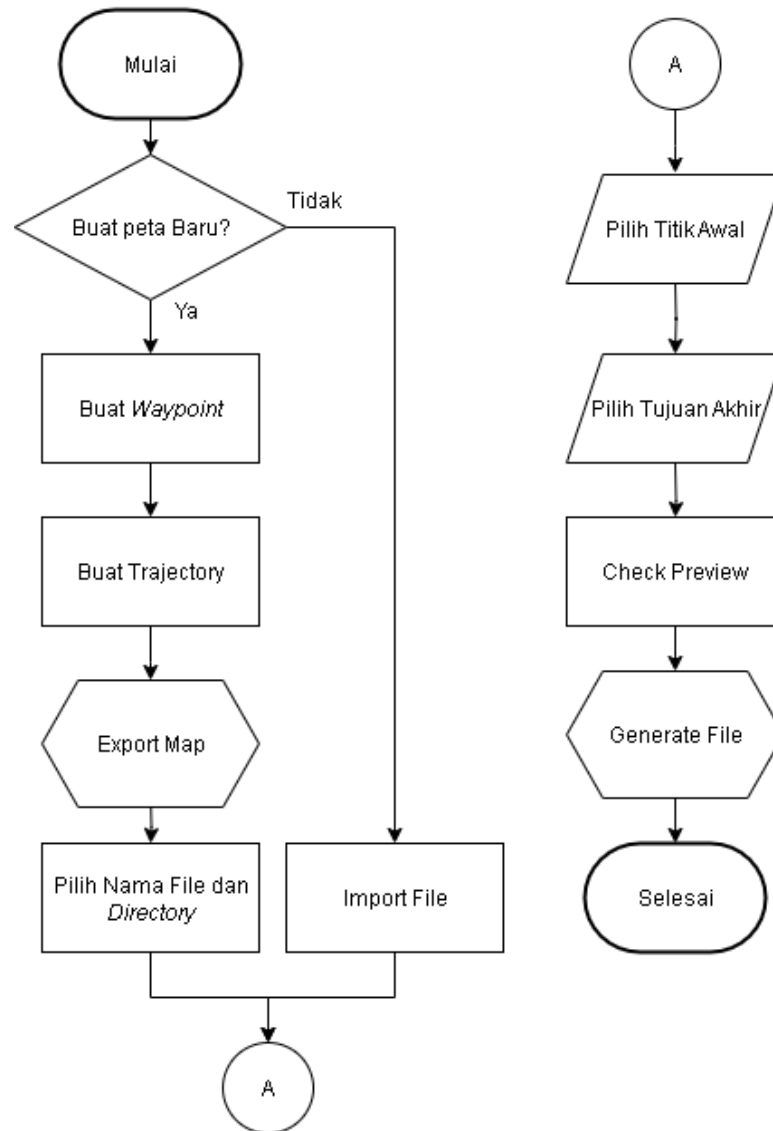
Gambar III. 10 Antarmuka *Monitor*

Monitor akan menjadi aplikasi di mana pengguna dapat melihat langsung data-data penerbangan *drone* secara *realtime* dan disimpan secara *offline* didalam file excel. Akan ada beberapa parameter yang dapat dilihat *user* pada aplikasi *monitor*. Yaitu

- Posisi *drone* (*Longitude dan Latitude*).
- Orientasi *drone* (*Roll, Pitch, Yaw*).
- Kecepatan Horizontal dan vertikal *drone*.
- Jumlah satelit yang ditangkap GPS. Dan
- Sisa Baterai.

3. Implementation

III.2.3.1 Rancangan Informatika

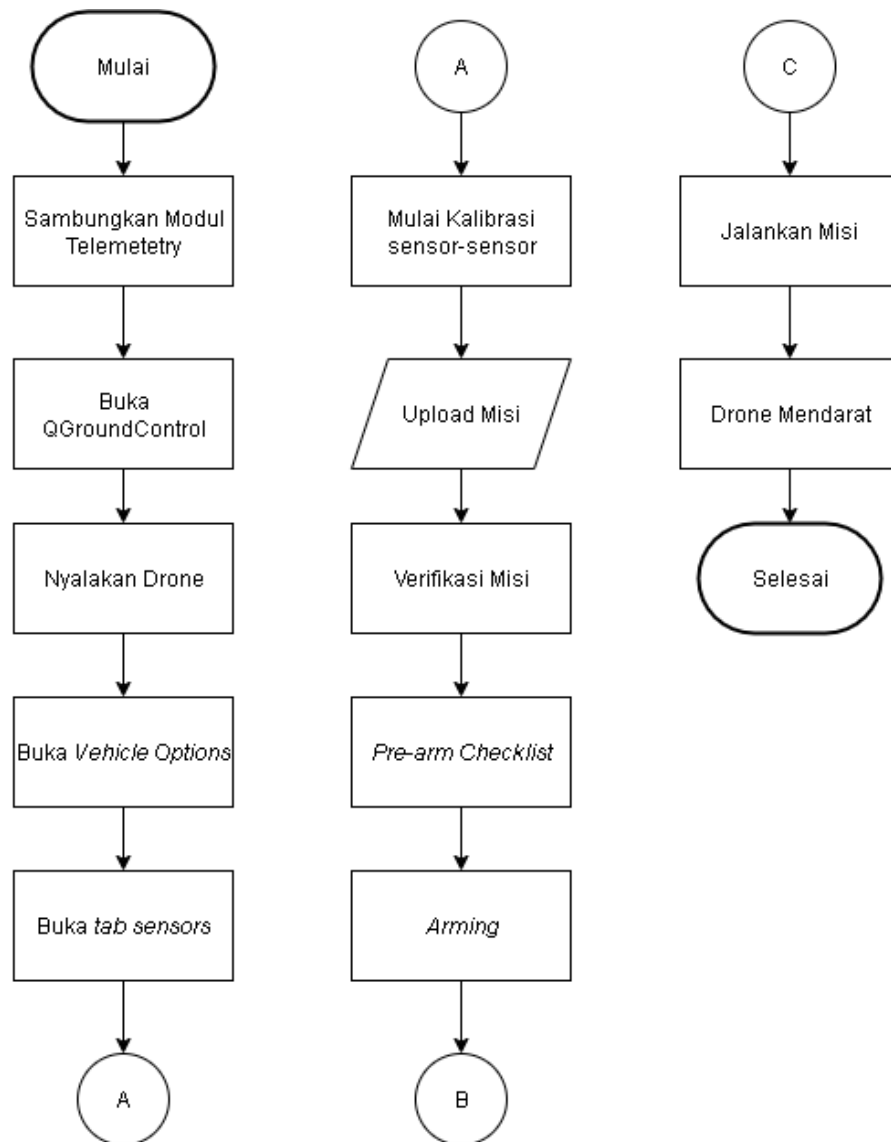


Gambar III. 11 Diagram Alur Aplikasi *Generate Map*

Alur dimulai dari pengguna memilih beberapa *waypoint* sebagai titik yang akan di kunjungi *drone*. Lalu memilih *trajectory* atau jalur yang akan dilewati sehingga peta *drone* akan terbentuk seperti *graph*. Pengguna dapat menyimpan peta dalam berkas terpisah untuk dapat digunakan di lain waktu. Setelah itu Pengguna dapat memilih dari mana dan kemana *drone* akan terbang. Setelah itu sistem akan menghitung jalur terpendek yang akan dilalui *drone* dan visualisasi akan

diperlihatkan untuk memberikan gambaran. Setelah itu Pengguna dapat membuat misi secara otomatis sesuai dengan *waypoint* dan *trajectory* yang telah ditentukan sebelumnya. Misi tersebut dapat disimpan secara luring dan dapat diterapkan ke semua jenis *drone* yang dapat menggunakan *firmware PX4 Autopilot*.

Selanjutnya adalah alur ketika seorang pengguna akan menerapkan berkas (.plan) kepada wahana secara langsung.



Gambar III. 12 Diagram Alur Penerbangan *drone*

Alur dimulai dari terhubungnya telemetry *ground module* di GCS yang ada pada laptop. Setelah itu buka aplikasi QGroundControl. Lalu setelah itu nyalakan *drone* sekaligus *air module* sehingga menghubungkan GCS dan *drone*. Setelah itu lakukan

kalibrasi sensor sebelum penerbangan hingga proses selesai. Setelah itu masuk ke *tab plan* dan *upload* misi dan verifikasi yang telah dibuat sebelumnya di aplikasi *Generate Map*. Setelah itu *drone* dapat diterbangkan dengan melakukan *arming* dan jalankan misi hingga selesai.

III.2.3.2 PYMAVLINK

```

Logger.py x +
1  from pymavlink import mavutil
2  import time
3
4  # Koneksi MAVLink
5  the_connection = mavutil.mavlink_connection('udp:127.0.0.1:14540')
6
7  print("Menunggu data MAVLink...\n")
8
9  # Buka file untuk menulis
10 log_file = open('mavlink_raw.txt', 'w')
11
12 while True:
13     msg = the_connection.recv_match(blocking=True)
14     if not msg:
15         continue
16
17     # Tulis data mentah ke file dengan timestamp
18     timestamp = time.strftime('%Y-%m-%d %H:%M:%S')
19     log_line = f"[{timestamp}] {msg}\n"
20
21     # Tulis ke file dan flush untuk real-time
22     log_file.write(log_line)
23     log_file.flush()
24
25     # Print ke console (optional)
26     print(log_line.strip())
27

```

Program ini menggunakan bahasa python dan *library* PYMAVLINK untuk berinteraksi dengan *drone* yang menggunakan protokol *MAVLINK*. `Mavutil.mavlink_connection` digunakan untuk menghubungkan Alamat UDP local pada port 14540. Alamat ini biasanya digunakan untuk berinteraksi dengan UAV. Setelah berhasil terkoneksi, program akan menampilkan pesan bahwa ia sedang menunggu *MAVLINK*. Kemudian text disimpan dalam `mavlink_raw.txt` untuk menyimpan data mentah yang diterima untuk digunakan untuk analisis lebih lanjut. Program menerima pesan dari koneksi dalam *loop* utama yang berjalan terus-menerus.. Secara keseluruhan, program ini berfungsi sebagai *logger* telemetry

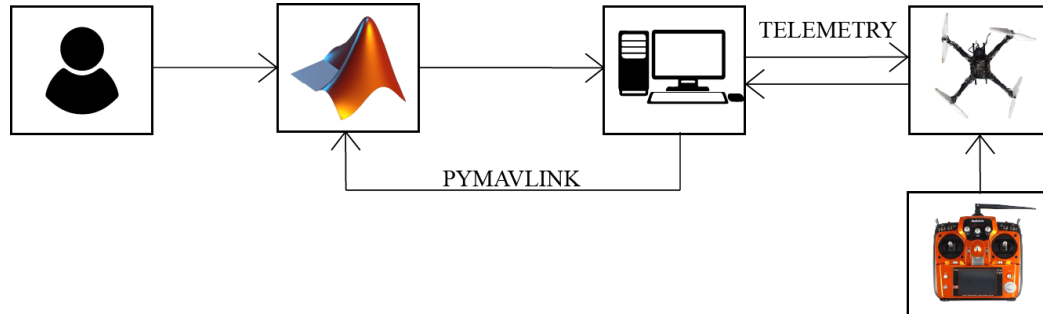
MAVLink yang memungkinkan pengawasan dan analisis data penerbangan *drone* secara langsung.

III.2.3.3 File .plan

Perangkat lunak QGroundControl menggunakan program ini sebagai berkas "Plan" untuk rencana misi *drone*. Berkas tersebut mengandung detail lengkap misi penerbangan seperti pada lampiran 3. Kecepatan jelajah (15 m/s), kecepatan mengambang (5 m/s), jenis *firmware* (12), dan mode ketinggian tertentu adalah semua bagian dari misi. *Waypoint (command 16)* dengan informasi tentang posisi, ketinggian, dan parameter lainnya dapat ditemukan di setiap item misi. Selain itu, lokasi wahana diatur dalam koordinat dan ketinggian tertentu. Sistem ini menunjukkan bagaimana MATLAB dan pymavlink terhubung ke QGroundControl untuk berkomunikasi dengan *drone*. Data telemetry juga dikirimkan kembali ke komputer untuk dipantau. Program ini bertujuan untuk otomatisasi dan pengawasan *drone*.

4. Integration and Testing

III.2.4.1 Gambaran Pengujian



Gambar III. 13 Gambaran Pengujian

Sistem akan dimulai dari pembuatan misi oleh *user* dengan menggunakan MATLAB. Pengguna dapat menentukan dimana saja *waypoint* tempat *drone* akan mendarat dan jalur yang boleh dilewati. Titik dan Jalur tersebut akan berbentuk seperti graf yang nantinya akan diterapkan algoritma. Pengguna dapat memilih darimana dan kemana tujuan *drone*, lalu sistem akan menghitung jalur tercepatnya dan membuat misi yang siap digunakan. Untuk mengunggahnya *drone* perlu terhubung dengan GCS(*Ground Control System*) baik dengan kabel ataupun nirkabel. Jika *drone* dihubungkan secara nirkabel maka Pengguna memerlukan telemetry sebagai alat komunikasi. Disamping itu user perlu melakukan *pre-arm*

terlebih dahulu, yaitu kalibrasi sensor, menentukan ketinggian, dan posisi *fix* GPS. Setelah semua aman *drone* dapat terbang dan menjalankan misi yang diinginkan. Lalu dari *drone*, log penerbangan akan dikirim Kembali ke MATLAB menggunakan protokol PYMAVLINK untuk dipantau secara *real-time*. RC (*Radio Control*) digunakan untuk mengambil alih kendali jika ada kesalahan yang dapat membahayakan.

III.2.4.2 Pengujian Sistem

Pengujian diperlukan untuk alat yang telah dirancang dan diintegrasikan menjadi satu sistem yang lengkap guna mengevaluasi sejauh mana sistem tersebut dapat memenuhi persyaratan yang ditetapkan. Pada bagian ini, dijelaskan rancangan skema pengujian yang dilakukan dalam tugas akhir ini.

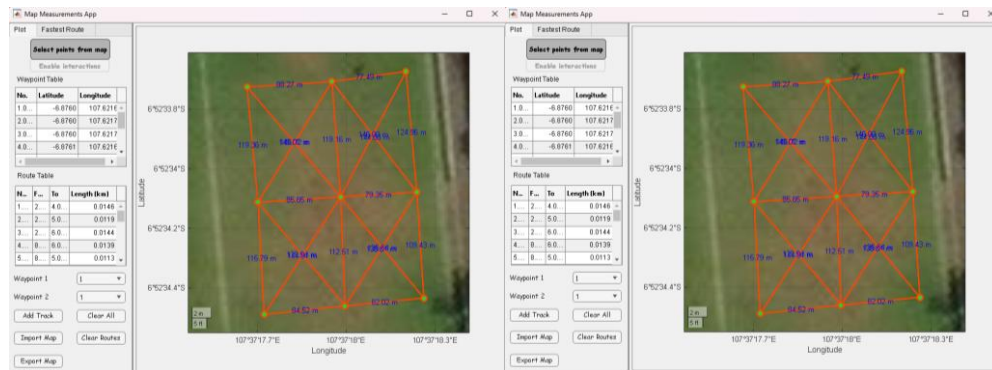
Tabel III. 3 Pengujian Sistem

No.	Pengujian	Fungsi
1	Fungsionalitas antarmuka	Pengujian fungsi dan tombol yang ada pada antar muka untuk memastikan fitur didalamnya dapat berfungsi dengan baik
2	Validasi Performa Algoritma dan Perancangan Misi	Membandingkan hasil perhitungan algoritma dengan perhitungan manual dengan ataupun tanpa jalur <i>no fly zone</i> .
3	Implementasi pada Wahana Udara Nirawak	Pengujian untuk membuktikan bahwa rencana misi dari sistem Anda dapat dieksekusi di lapangan.

III.2.4.3 Skema Pengujian Sistem

Pelaksanaan dari setiap pengujian dilakukan dengan mengikuti skema atau prosedur spesifik sebagai berikut:

1. Fungsionalitas antar muka (*Blackbox testing*)



Gambar III. 14 Antarmuka tahap awal

Pengujian ini dilakukan setelah antarmuka tahap awal selesai dibuat seperti pada Gambar III.13. Hal pertama yang akan dilakukan adalah survei kepada beberapa orang untuk mendapatkan masukan terhadap antarmuka yang telah dibuat. Survei dilakukan kepada 12 orang untuk mendapatkan hasil maksimal. Pada tabel III.4 terlihat nama-nama responden yang telah membantu mengisi survei.

Tabel III. 4 Daftar nama-nama responden

No	Nama	Program Studi
1	Mokhammad Alfian Wicaksono	TRO
2	Moch. Fikrie Dzulfikar Hamzah	TRO
3	Rais Mubarak	TRO
4	Rafi Tsaqif Athooya	TRMO
5	Dandi Cahyadi	TRO
6	Ali Baqri Fairuz Rachmat	TRMO
7	Irza Alif Miftah Al Atthar	TRMO
8	Satyo Septiyanto	TRMO
9	Kometa Ilmiah	TRO
10	Afifah Rahmi	TRMO
11	Anis Rosiani	TRO
12	Firma Dilla Sholeh	TRO

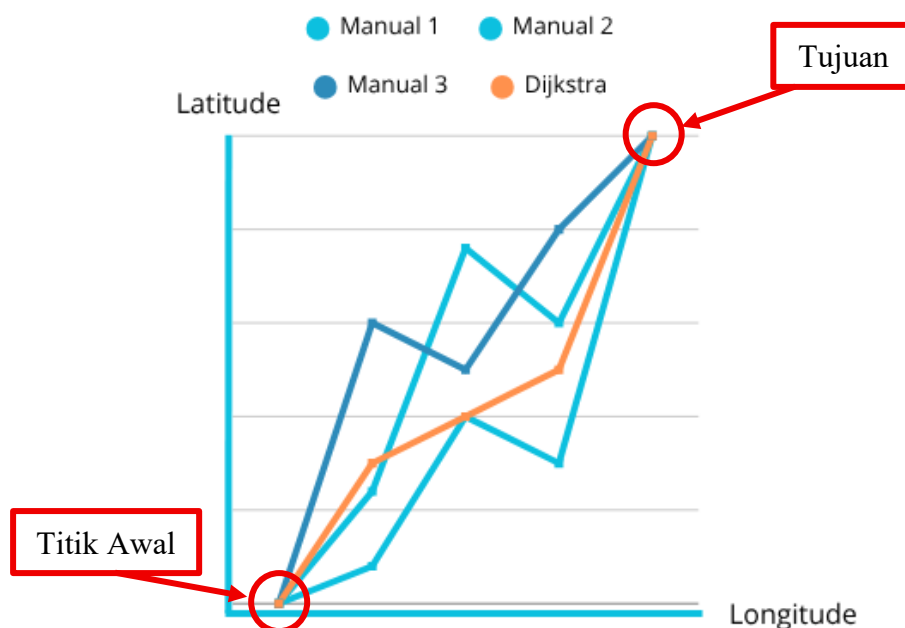
Pada Lampiran terlihat bukti nama-nama orang yang telah dilakukan survei dan memberi masukan terhadap antarmuka versi awal. Dibawah ini merupakan rangkuman masukan-masukan yang ada pada antarmuka berdasarkan survei.

- a. Inkonsistensi bahasa. jika ingin menggunakan bahasa inggris, maka ganti semua tulisannya. Begitupun dengan bahasa Indonesia.
- b. Titik awal dan titik akhir membingungkan dan tidak terlihat dengan jelas.
- c. Terdapat istilah-istilah yang sulit dimengerti oleh orang awam.

Setelah mendapatkan masukan dari beberapa responden, perbaikan pada bagian antarmuka akan dilakukan lalu pengujian dilakukan untuk memastikan semua fungsi dapat dijalankan dengan baik. Hasil dari pengujian akan disajikan dalam bentuk tabel yang menyatakan bahwa fungsi-fungsi dan tombol-tombol yang ada di antarmuka dapat berfungsi dengan baik.

2. Validasi Performa Algoritma dan Perancangan Misi

Algoritma *Dijkstra* yang digunakan berfungsi untuk mencari jalur terpendek dari titik awal menuju tujuan *drone*. Oleh karena itu perlu dilakukan validasi terhadap jalur yang dihitung algoritma *Dijkstra* merupakan jalur terpendek, dengan cara perhitungan manual ke 3 kemungkinan terdekat seperti pada Gambar III. 15.

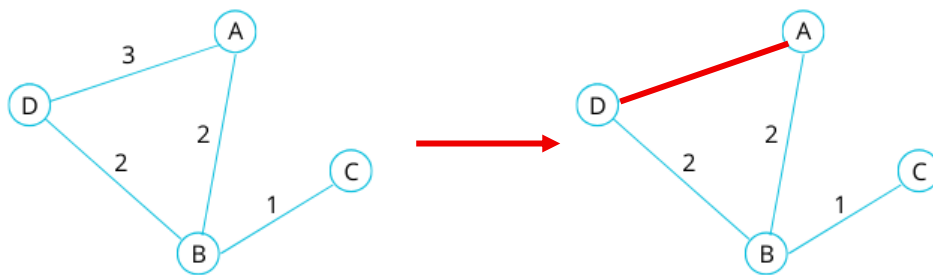


Gambar III. 15 Gambaran Pengujian

Tabel IV. 1 Contoh Tabel *waypoints*

No.	Manual 1	Manual 2	Manual 3	<i>Dijkstra</i>
1	0	0	0	0
2	12	4	30	15
3	38	20	25	20
4	30	15	40	25
5	50	50	50	50

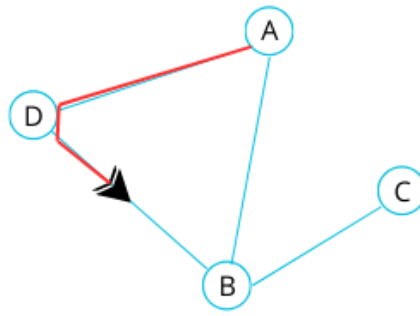
Misal, dalam suatu peta penerbangan *drone* memiliki 5 titik dengan bobot antar titik yang digambarkan dalam bentuk Matriks *adjacency* seperti pada Gambar III.15. Lalu jika salah satu jalur (Misal A ke D) tidak dapat di lewati, maka perhitungan *Dijkstra* tidak akan memperhitungkan jalur tersebut dan akan mencari alternatif jalur terdekat untuk mencapai tujuan.



Gambar III. 16 Contoh Peta

3. Implementasi dan Validasi Teknis

Pada bagian ini akan diuji bagaimana perencanaan yang dilakukan di MATLAB diterapkan ke berbagai kondisi *drone*. Respon ketika wahana diberikan perintah akan menentukan implementasi, seperti Gambar III. 17.



Gambar III. 18 Gambaran Pengujian Validasi Teknis

Beberapa perintah yang akan diuji dalam kondisi yang berbeda. Kondisi-kondisi tersebut akan disajikan dalam tabel-tabel berikut. yaitu, *Takeoff*, Menaikkan dan menurunkan ketinggian, serta menahan posisi di ketinggian tertentu. Selain itu ada juga pengujian untuk melihat implementasi berkas misi (.plan) ke wahana dan menjalankannya.

a. Lepas Landas

Pengujian ini dilakukan karena kondisi landasan mempengaruhi tingkat keberhasilan. Lepas landas di landasan yang tidak landai akan menghasilkan hasil yang berbeda dibanding jika landasannya landai. Setiap kondisi akan dilakukan 10x percobaan. Pengujian akan disajikan dalam bentuk tabel seperti pada.

b. Menaikkan dan menurunkan ketinggian

Pengujian ini dilakukan untuk melihat respon ketika diberikan perintah untuk menuju ketinggian tertentu. Pengujian ini dilakukan untuk melihat respon wahana ketika diberi perintah untuk terbang ke ketinggian tertentu. Pengujian akan dilakukan dengan 5 jenis ketinggian mulai dari 1m hingga 5m. Pengujian akan dilakukan dengan wahana lepas landas ke ketinggian tertentu lalu langsung mendarat di posisi yang sama.

c. Menahan posisi di ketinggian tertentu berdasarkan waktu

Pengujian ini dilakukan sebagai gambaran untuk menurunkan paket, karena sistem untuk menurunkan paket tidak tersedia, maka sistem akan digambarkan dengan menahan posisi di ketinggian rendah selama 30 detik,

dengan asumsi dalam 30 detik paket dapat diturunkan. Pengujian akan dilakukan dengan wahana lepas landas ke ketinggian 3m lalu menahan posisinya selama 30 detik, dan mendarat di posisi yang sama seperti pada .

d. Implementasi Misi

Pengujian akan dilakukan dengan menjalankan berkas misi (.plan) pada 2 skenario yaitu simulasi dan aktual unit wahana. Dari pengujian tersebut diambil 3 parameter. Yaitu Orientasi (*roll*, *pitch*, *Yaw*), Kecepatan (Vertikal, dan Horizontal), dan akurasi trajektori (*Longitude* dan *Latitude*).



Gambar III. 19 Gambaran Misi yang akan dijalankan

drone akan lepas landas ke ketinggian 5 m, lalu *drone* bergerak dari (-6.8762,107.6217) lalu bergerak lurus ke (-6.8762,107.6216), setelah itu bergerak lurus ke (-6.8761,107.6216), menurunkan ketinggian ke 3m, tahan posisi selama 5 detik, lalu kembali dengan jalur yang sama.

5. *Operation and Maintenance*

Bagian ini menguraikan panduan operasi dan perawatan yang digunakan untuk sistem yang dibangun, mencakup perangkat keras dan perangkat lunak pendukungnya. Tujuan dari panduan ini adalah untuk memastikan keamanan, keandalan, dan keberlangsungan fungsi sistem dalam jangka panjang. Prosedur perawatan dibagi menjadi tiga kategori, Pemeriksaan Pra-Penerbangan (sebelum terbang), Pemeriksaan Pasca-Penerbangan (setelah terbang), dan Perawatan Berkala.

III.2.5.1 Perawatan Perangkat Keras

Perawatan perangkat keras difokuskan pada pengecekan kondisi fisik unit *drone* untuk mencegah kegagalan komponen saat beroperasi.

1. Struktu dan rangka, Pemeriksaan retakan halus pada lengan atau papan utama, terutama setelah pendaratan keras. Pastikan seluruh sekrup dan baut terpasang kencang dan tidak ada yang kendur.
2. Aktuator, Pemeriksaan dilakukan pada motor dan baling-baling yang digunakan. Periksa setiap baling-baling dari adanya goresan dalam, retakan, atau perubahan bentuk. Bersihkan dari debu atau kotoran menggunakan kain lembut. Ganti baling-baling jika ditemukan kerusakan sekecil apa pun untuk menghindari getaran berlebih yang dapat mempengaruhi stabilitas terbang. Motor juga harus diperiksa secara manual apakah motor dapat berputar dengan bebas tanpa ada halangan. Dengarkan jika ada suara aneh seperti gesekan saat motor diputar. Bersihkan ventilasi motor dari debu yang menumpuk.
3. Sistem elektronik, Pastikan kabel dan konektor memiliki sambungan yang baik. Periksa semua koneksi kabel ke *flight controller*, ESC, dan motor. Pastikan tidak ada kabel yang terjepit atau konektor yang longgar. Periksa juga baterai Li-Po yang digunakan. Jangan pernah gunakan baterai yang menunjukkan tanda-tanda kembung, penyok, atau kerusakan pada kabel/konektor.

III.2.5.2 Perawatan Perangkat Lunak

Perawatan perangkat lunak bertujuan untuk menjaga kompatibilitas dan stabilitas sistem.

1. Aplikasi MATLAB, Lakukan *backup* secara berkala terhadap kode sumber aplikasi dan berkas-berkas peta (.mat) yang telah dibuat. Periksa jika ada pembaruan versi MATLAB atau *toolbox* yang digunakan yang mungkin mempengaruhi fungsionalitas.
2. *Firmware* PX4 Autopilot, Secara berkala, periksa situs resmi PX4 untuk rilis *firmware* stabil yang baru. Hindari melakukan pembaruan *firmware* sesaat sebelum misi penting tanpa melakukan uji coba terlebih dahulu.

III.2.5.3 Jadwal Perawatan Sistem

Tabel III. 5 *Maintenance* Alat

Frekuensi	Komponen	Prosedur Pemeriksaan
Pra- Penerbangan	Rangka & Struktur	Periksa visual retakan, sekrup kendur.
	Baling-Baling	Periksa dari kerusakan (retak, goresan), pastikan terpasang kencang.
	Baterai	Periksa kondisi fisik (tidak kembung/rusak), pastikan daya terisi.
Pasca-Penerbangan	Seluruh Wahana	Bersihkan dari debu dan kotoran.
	Rangka & Baling-Baling	Periksa kembali jika ada kerusakan akibat pendaratan.
	Baterai	Simpan pada tegangan penyimpanan (<i>storage voltage</i>) jika tidak akan digunakan dalam waktu dekat.
Berkala (Mingguan/Bulanan)	Sistem Elektronik	Periksa kelonggaran kabel dan konektor.
	Perangkat Lunak	Lakukan pencadangan (<i>backup</i>) kode dan data. Periksa ketersediaan pembaruan (<i>firmware</i> , QGC).