



Table of Contents

[1 Introduction and Initial Analysis 2](#_heading=h.gjdgxs)

[1.1 Project Context 2](#_heading=h.30j0zll)

[1.2 Initial Thought Process 2](#_heading=h.1fob9te)

[2 Requirement Analysis and Specification 2](#_heading=h.3znysh7)

[2.1 User Requirements 2](#_heading=h.2et92p0)

[2.2 System Requirements 2](#_heading=h.tyjcwt)

[2.3 Tools and Technologies 2](#_heading=h.3dy6vkm)

[3 Conceptual Design 2](#_heading=h.1t3h5sf)

[3.1 System Architecture 2](#_heading=h.4d34og8)

[3.2 Interface Design 2](#_heading=h.2s8eyo1)

[3.3 Control Algorithm Design 3](#_heading=h.17dp8vu)

[4 Detailed Design and Development 3](#_heading=h.3rdcrjn)

[4.1 Component Design 3](#_heading=h.26in1rg)

[4.2 Coding and Implementation 3](#_heading=h.lnxbz9)

[4.3 Integration 3](#_heading=h.35nkun2)

[4.4 Unique Features 3](#_heading=h.1ksv4uv)

[5 Testing, Evaluation, and Optimization 3](#_heading=h.44sinio)

[5.1 Testing Strategy 3](#_heading=h.2jxsxqh)

[5.2 Performance Evaluation 3](#_heading=h.z337ya)

[5.3 Optimization 3](#_heading=h.3j2qqm3)

[6 Collaboration and Project Management 3](#_heading=h.1y810tw)

[6.1 Teamwork Dynamics 3](#_heading=h.4i7ojhp)

[6.2 Project Management 3](#_heading=h.2xcytpi)

[7 Conclusion and Reflection 3](#_heading=h.1ci93xb)

[7.1 Project Summary 3](#_heading=h.3whwml4)

[7.2 Future Work 3](#_heading=h.2bn6wsx)

[7.3 Personal and Group Reflections 3](#_heading=h.qsh70q)

[8 Appendices 4](#_heading=h.3as4poj)

[8.1 Bill of Materials 4](#_heading=h.1pxezwc)

[8.2 Electrical Wiring and System Layout 4](#_heading=h.49x2ik5)

[8.3 Code Repository 4](#_heading=h.2p2csry)

[8.4 Additional Documentation 4](#_heading=h.147n2zr)

[9 References 4](#_heading=h.3o7alnk)

# 1 Introduction and Initial Analysis

## 1.1 Project Context

## UAV (Unmanned Aerial Vehicle) merupakan salah satu teknologi yang sedang mengalami perkembangan yang pesat dan memiliki potensi yang sangat besar, baik untuk keperluan militer maupun sipil. Ada banyak tipe UAV yang dikenal. Contohnya adalah fixed wing, axial wing, coaxial wing, dan quadcopter. Quadcopter adalah jenis pesawat terbang UAV pengembangan dari helikopter yang memiliki empat motor. Salah satu kelebihan dari quadcopter dibandingkan dengan helikopter adalah memiliki empat motor sehingga quadkopter memiliki daya angkat yang lebih besar dibandingkan helicopter.

## Pemanfaatan quadcopter semakin berkembang bukan hanya sebuah permainan dan seni menerbangkan pesawat tanpa awak, namun pemanfaatan quadcopter sudah merambah diberbagai bidang diantaranya bidang militer, fotografi, 2 sinematografi, geografi, dan berbagai bidang lainnya. Salah fungsi dari quadcopter pada bidang fotografi atau sinematografi adalah untuk memantau keadaan atau live streaming daerah dibawah area terbang. Oleh karena itu perlu ditambahkan kamera pada quadcopter.

1.1.1 Tujuan

Tujuan proyek ini adalah dapat menjalankan quadcopter agar dapat menyelesaikan misi seperti yang diinginkan serta dapat menjadikan gerakan yang beragam kanan kiri atas bawah pada quadcopter yang sudah ada, untuk meningkatkan efektivitas penggunaannya dalam berbagai aplikasi. Keberhasilan proyek ini akan memajukan pengembangan quadcopter dan meningkatkan manfaatnya dalam berbagai bidang.

1.2 Initial Thought Process

Proses pengembangan quadcopter kami dimulai dengan menganalisa tujuan awal untuk mencari keberhasilan quadcopter dalam bergerak dan keseimbangannya. Meliputi pembaruan pada algoritma gerakan, perbaikan dalam navigasi, manuverabilitas, dan lain hal sebagainya.

Saat menganalisa tujuan awal kami tersebut, kami mengidentifikasi keberhasilan quadcopter kami untuk mengoptimalkan gerak kanan kiri dengan menggunakan algoritma pengendali dan hardware yang kami gunakan.

Selanjutnya, kami memilih untuk mengendalikan quadcopter tersebut, dengan berfokus pada gerakan quadcopter kami yang lebih tertata dan terstruktur sesuai yang diinginkan. Dalam bagian proses ini, kami terus mendokumentasikan perkembangan dan evaluasi pada robot quadcopter kami.

# 2 Requirement Analysis and Specification

## 2.1 User Requirements

GUI menyediakan antarmuka yang lebih intuitif dan mudah digunakan, memungkinkan pengguna untuk berinteraksi dengan perangkat lunak dan sistem tanpa harus menguasai perintah teks rumit. Ini membuat teknologi lebih dapat diakses oleh pengguna yang tidak memiliki latar belakang teknis yang kuat.

1. Antarmuka Pengguna yang Ramah Pengguna:

* Penjelasan: Antarmuka harus dirancang agar mudah dimengerti dan digunakan oleh pengguna dengan berbagai tingkat keahlian teknis. Hal ini mencakup desain yang intuitif dan kontrol yang mudah dipahami.

1. Konsistensi:

* Penjelasan: Menyajikan elemen desain yang konsisten dan tata letak yang seragam di seluruh antarmuka membantu menciptakan pengalaman pengguna yang terpadu dan tidak membingungkan.

1. Disesuaikan:

* Penjelasan: Pengguna perlu memiliki kemampuan untuk menyesuaikan antarmuka sesuai dengan preferensi pribadi mereka. Ini dapat mencakup opsi seperti pengaturan tema, warna, atau tata letak.

1. Aksesibilitas:

* Penjelasan: Desain antarmuka harus dapat diakses oleh semua pengguna, termasuk mereka yang memiliki disabilitas. Ini dapat melibatkan integrasi fitur aksesibilitas seperti pintasan keyboard, pembaca layar, dan ukuran font yang dapat diubah.

1. Responsif:

* Penjelasan: Antarmuka harus merespons dengan cepat terhadap setiap tindakan pengguna. Responsivitas yang baik menciptakan pengalaman pengguna yang mulus.

1. Penanganan Kesalahan:

* Penjelasan: Sistem harus memberikan pesan kesalahan yang jelas dan petunjuk untuk membantu pengguna memahami dan mengatasi masalah yang mungkin muncul.

1. Umpan Balik:

* Penjelasan: Pengguna membutuhkan umpan balik visual atau auditif setiap kali mereka melakukan tindakan, seperti mengklik tombol atau memasukkan data, untuk memastikan bahwa aksi mereka telah tercatat.

1. Navigasi:

* Penjelasan: Antarmuka harus menyediakan opsi navigasi yang intuitif, memungkinkan pengguna untuk dengan mudah beralih antara berbagai bagian atau fitur sistem.

1. Kompatibilitas:

* Penjelasan: Antarmuka harus kompatibel dengan berbagai perangkat, ukuran layar, sistem operasi, dan browser agar dapat diakses oleh sebanyak mungkin pengguna.

1. Keamanan:

* Penjelasan: Sistem harus mengimplementasikan langkah-langkah keamanan yang kuat untuk melindungi data pengguna dan memastikan privasi informasi.

1. Kinerja:

* Penjelasan: Penting untuk mengoptimalkan kinerja sistem, terutama dalam aplikasi yang membutuhkan sumber daya tinggi, untuk mencegah keterlambatan atau lag.

1. Efisiensi Tugas:

* Penjelasan: Desain antarmuka harus mendukung efisiensi pengguna dalam menyelesaikan tugas-tugas mereka dengan minimal jumlah langkah.

1. Bantuan dan Dokumentasi:

* Penjelasan: Menyediakan akses mudah ke bantuan, dokumentasi, atau tooltip membantu pengguna memahami cara menggunakan sistem dan menyelesaikan masalah.

1. Skalabilitas:

* Penjelasan: Antarmuka harus dirancang dengan mempertimbangkan kemampuan untuk menampung perluasan atau pembaruan tanpa mengganggu pengalaman pengguna saat ini.

1. Dukungan Multi-Bahasa:

* Penjelasan: Jika relevan, sistem harus mendukung beberapa bahasa untuk menjangkau pengguna dengan latar belakang bahasa yang beragam.

1. Pelatihan dan Onboarding:

* Penjelasan: Proses pelatihan dan onboarding harus dirancang dengan baik untuk memastikan bahwa pengguna baru dapat dengan cepat memahami dan menggunakan sistem.

1. Daya Tarik Visual:

* Penjelasan: Antarmuka yang estetis dapat meningkatkan keterlibatan pengguna. Keseimbangan antara estetika dan fungsionalitas sangat penting.

1. Kolaborasi dan Berbagi:

* Penjelasan: Jika sistem melibatkan kolaborasi, fitur kolaborasi dan berbagi informasi harus diintegrasikan dengan baik.

## 2.2 System Requirements

**Rinci Persyaratan Teknis :**

1. Daya angkat dan kekuatan motor

* Quadcopter harus mampu menghasilkan daya angkat yang cukup untuk mengangkat dirinya sendiri dan beban yang diangkut (misalnya, kamera atau sensor).
* Motor yang digunakan harus memiliki daya tahan yang cukup dan dapat menggerakkan rotor dengan efisien.

1. Baterai dan kapasitas energi

* Baterai harus memiliki kapasitas yang cukup untuk memberikan daya selama penerbangan yang diinginkan.
* Penggunaan baterai yang efisien adalah pertimbangan penting untuk jangka waktu penerbangan yang lebih lama.

1. Stabilitas dan sensorika

* Quadcopter harus dilengkapi dengan sensor inertial (akselerometer dan giroskop) untuk memonitor perubahan posisi dan orientasi.
* Sensor tambahan seperti kompas dan GPS dapat digunakan untuk navigasi dan penentuan lokasi.

1. Sistem kendali

* Sistem kendali otomatis yang dapat mengatur pergerakan quadcopter adalah persyaratan utama.
* Controller PID (Proportional-Integral-Derivative) atau algoritma kendali lainnya mungkin diperlukan untuk menjaga kestabilan dan respons yang tepat.

1. Komunikasi dan telemetri

* Quadcopter perlu memiliki sistem komunikasi yang dapat mengirimkan data penerbangan dan menerima perintah dari pengguna atau operator.
* Komunikasi nirkabel seperti Wi-Fi atau telemetri radio umum digunakan

1. Perangkat lunak kendali

* Perangkat lunak kendali yang efisien dan handal diperlukan untuk mengatur gerakan quadcopter.
* Ini melibatkan pemrograman kendali penerbangan, navigasi, dan logika pengendalian.

**Persyaratan Fungsional :**

1. Pemantauan lingkungan

* Jika digunakan untuk pemantauan, quadcopter perlu dilengkapi dengan sensor yang sesuai (kamera, sensor termal, sensor cuaca, dll.).
* Kemampuan pemrosesan gambar dan data sensor adalah persyaratan fungsional.

1. Take-off dan landing

* Quadcopter harus memiliki kemampuan untuk lepas landas dan mendarat secara otomatis atau dengan bantuan pengguna.

1. Kebijakan kemanan

* Quadcopter harus mematuhi regulasi dan kebijakan keamanan penerbangan yang berlaku.
* Kemungkinan fitur keamanan seperti geo-fencing (pembatasan area terbang) dapat diterapkan.

1. Kontrol jarak jauh

* Quadcopter perlu dapat dikendalikan dari jarak jauh melalui remote control atau aplikasi seluler.

1. Manuverabilitas

* Quadcopter harus dapat bergerak ke depan, mundur, ke samping, naik, dan turun.
* Kemampuan berputar dan mengikuti jalur tertentu juga mungkin diperlukan.

1. Regulasi

* Quadcopter harus mematuhi regulasi penerbangan yang berlaku di wilayah tempat mereka dioperasikan.

1. Keandalan dan keamanan operasi

* Quadcopter harus dirancang untuk menghindari kegagalan dan beroperasi dengan aman.
* Kemampuan darurat, seperti pendaratan darurat, harus tersedia dalam situasi yang tidak terduga.

1. Integrasi dan pemeliharaan

* Quadcopter perlu dapat diintegrasikan dengan perangkat lunak dan sistem yang ada, serta mudah dipelihara dan diperbaiki.

## 2.3 Tools and Technologies

**Software :**

1. VSCode
2. QT Designer
3. Arduino IDE

**Hardware :**

1. Arduiono Mega
2. MPU6050
3. ESC Skywalker
4. PLDC 850KV
5. Baterai LIPO 35c (2200mAh)
6. Konektpr Type B

## 2.4 Target specification

*Tabel 1. Tabel caption.*

| Feature | Description | Measurement Metric | Target Value |
| --- | --- | --- | --- |
| Stabilisasi | Sistem Manajemen Quadcopter terdiri dari beberapa komponen perangkat lunak yang bekerja bersama untuk mengelola, mengontrol, dan memantau quadcopter. Setiap komponen memainkan peran penting dalam memastikan penerbangan yang aman, stabil, dan efisien. | Metrik-metrik ini akan digunakan untuk mengukur kinerja dan keamanan sistem penerbangan otonom quadcopter. | Target spesifik yang ingin dicapai dalam pengembangan dan operasional sistem penerbangan otonom quadcopter. |
| Flowchart | Representasi grafis dari alur logika atau langkah-langkah proses dalam suatu sistem. Flowchart digunakan untuk menggambarkan urutan langkah-langkah yang harus diambil untuk menyelesaikan tugas atau mencapai suatu tujuan. | Identifikasi dan pilih parameter khusus yang akan diukur untuk mencapai tujuan. Menentukan metode atau alat yang akan digunakan untuk mengukur parameter.Melakukan pengumpulan data dan periksa apakah data sudah lengkap. Analisis data yang telah dikumpulkan. Evaluasi hasil sesuai dengan kriteria yang telah ditetapkan. Simbol akhir flowchart. | Tetapkan nilai target yang ingin dicapai. |
| Design GUI | Design merujuk pada proses merancang antarmuka pengguna grafis untuk aplikasi atau sistem perangkat lunak. GUI menyediakan cara bagi pengguna untuk berinteraksi dengan perangkat lunak menggunakan elemen visual seperti tombol, ikon, jendela, dan menu, yang mempermudah penggunaan dan pemahaman fungsi-fungsi program. |  |  |
| Manual Operation | Pelaksanaan suatu tindakan atau proses secara manual oleh manusia tanpa menggunakan otomatisasi atau bantuan mesin. Dalam konteks berbagai kegiatan atau sistem, manual operation melibatkan intervensi manusia yang aktif untuk menjalankan atau mengontrol proses tertentu. |  |  |
| Penginderaan dan hindaran objek | Modul ini berfokus pada kontrol dan navigasi quadcopter selama penerbangan. Bertanggung jawab atas stabilitas, penerbangan waypoint, dan pengambilan keputusan penerbangan. | Akurasi Navigasi: Jarak rata-rata antara posisi aktual dan posisi yang diinginkan dari waypoint. | Akurasi Navigasi: Kurang dari 1 meter dari posisi yang diinginkan. |
| Pemantauan kondisi baterai dan kesehatan perangkat keras | Modul ini memantau dan mengelola daya baterai quadcopter untuk memaksimalkan waktu penerbangan dan mencegah pendaratan yang tidak aman. | Keseimbangan Baterai: Persentase sisa daya baterai dan peringatan level baterai rendah. | Keseimbangan Baterai: Notifikasi level baterai rendah dikirimkan saat sisa daya kurang dari 20%. |
| Mode darurat dan pendaratan aman | Modul ini mengelola proses pendaratan quadcopter, baik secara otomatis maupun manual, serta menangani pendaratan darurat jika diperlukan. | Keandalan Pendaratan Darurat: Keberhasilan dalam melakukan pendaratan darurat dengan aman ketika terjadi kegagalan sistem. | Keandalan Pendaratan Darurat: 100% keberhasilan dalam pendaratan darurat tanpa merusak perangkat keras atau properti. |
| Komunikasi dengan operator melalui antarmuka pengguna | Antarmuka Pengguna adalah titik akses operator untuk memantau status quadcopter dan memberikan perintah. Ini dirancang untuk kejelasan dan kemudahan penggunaan. | Antarmuka pengguna untuk memonitor dan memberikan perintah kepada quadcopter. | 1. Pemantauan posisi, kecepatan, dan status baterai. 2. Antarmuka peta untuk menentukan rute penerbangan. 3. Kontrol manual dan otonom. 4. Notifikasi dan log kejadian. |
| Dokumentasi Pemrograman |  |  |  |

# 3 Conceptual Design

## 3.1 System Architecture

## Ilustrasi arsitektur tingkat tinggi dari sebuah sistem melibatkan representasi visual komponen-komponennya, interaksi di antara mereka, serta aliran data atau kontrol di antara mereka. Ini umumnya mencakup antarmuka pengguna grafis (GUI) dan logika kontrol yang mendasarinya.

1. GUI (Antarmuka Pengguna Grafis):

GUI merupakan bagian dari sistem yang terlihat oleh pengguna, di mana pengguna berinteraksi dengan aplikasi. Komponen GUI dapat mencakup tombol, kolom input, tampilan, dan elemen lain yang dapat dilihat dan diinteraksikan oleh pengguna.

1. Logika Kontrol:

Logika kontrol adalah seperangkat aturan dan proses yang mengelola perilaku sistem. Ini mencakup proses pengambilan keputusan, algoritma, dan logika bisnis yang mengatur fungsionalitas aplikasi.

1. Komunikasi antara GUI dan Logika Kontrol:

Terdapat panah atau garis yang menunjukkan aliran informasi antara GUI dan logika kontrol. Input pengguna dari GUI (misalnya, klik tombol, pengiriman formulir) diproses oleh logika kontrol. Logika kontrol dapat mengirim pembaruan atau permintaan kembali ke GUI untuk mencerminkan perubahan atau memberikan umpan balik kepada pengguna.

1. Aliran Data:

Tunjukkan aliran data antara berbagai komponen sistem. Misalnya, jika melibatkan database, tunjukkan aliran data antara database dan logika kontrol.

1. Sistem atau Layanan Eksternal:

Jika sistem berinteraksi dengan layanan atau sistem eksternal (misalnya, API), gambarkan koneksi-koneksi ini dalam diagram arsitektur.

1. Penanganan Kesalahan:

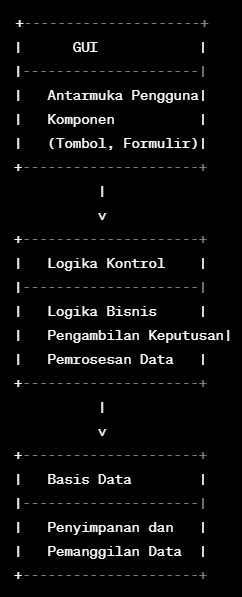
Tunjukkan bagaimana kesalahan atau pengecualian ditangani dalam sistem. Ini mungkin melibatkan pesan kesalahan yang ditampilkan di GUI atau proses penanganan kesalahan khusus dalam logika kontrol.

1. Langkah-langkah Keamanan:

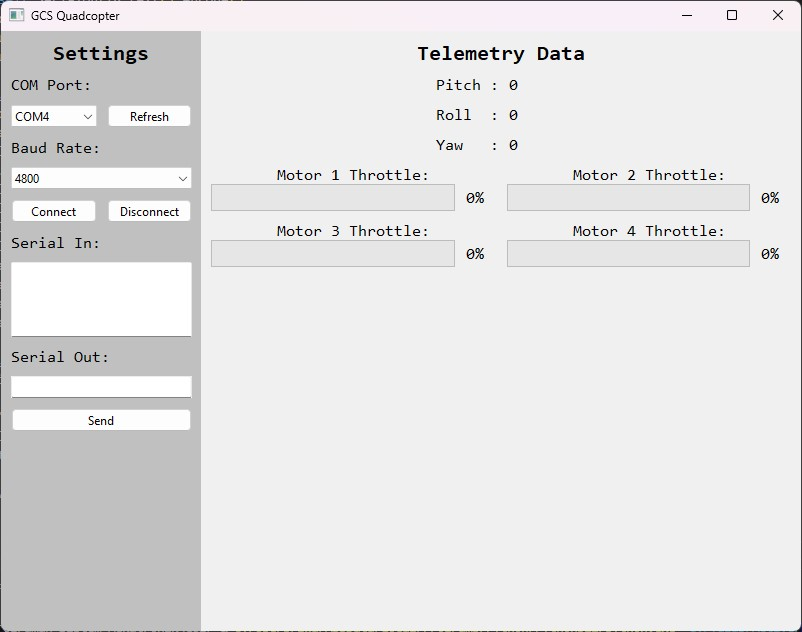
Jika ada langkah-langkah keamanan seperti otentikasi atau enkripsi, gambarkan dalam diagram arsitektur.

1. Lapisan atau Tingkatan

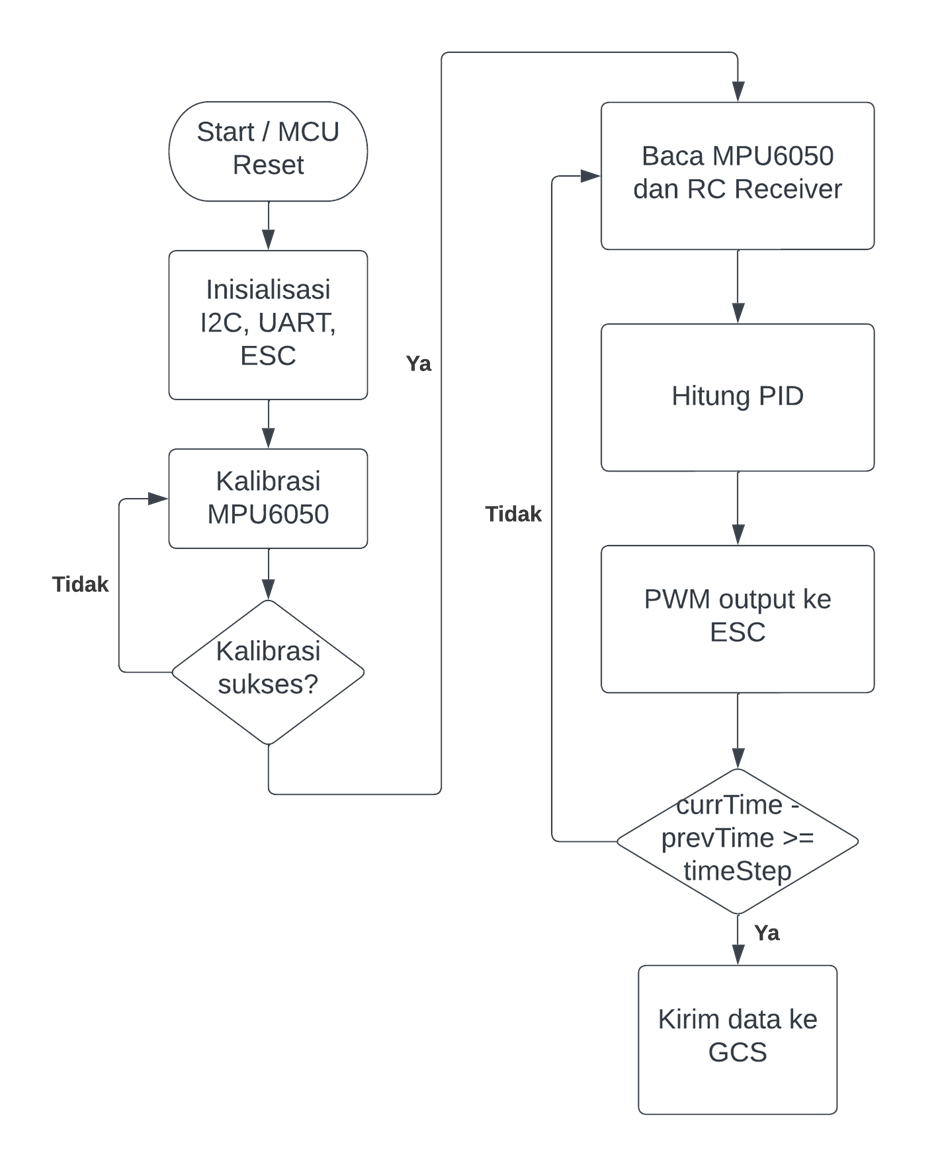
Jika sistem Anda terstruktur dengan lapisan atau tingkatan yang berbeda (misalnya, lapisan presentasi, lapisan logika bisnis, lapisan akses data), gambarkan lapisan ini dalam arsitektur.



## 3.2 Interface Design

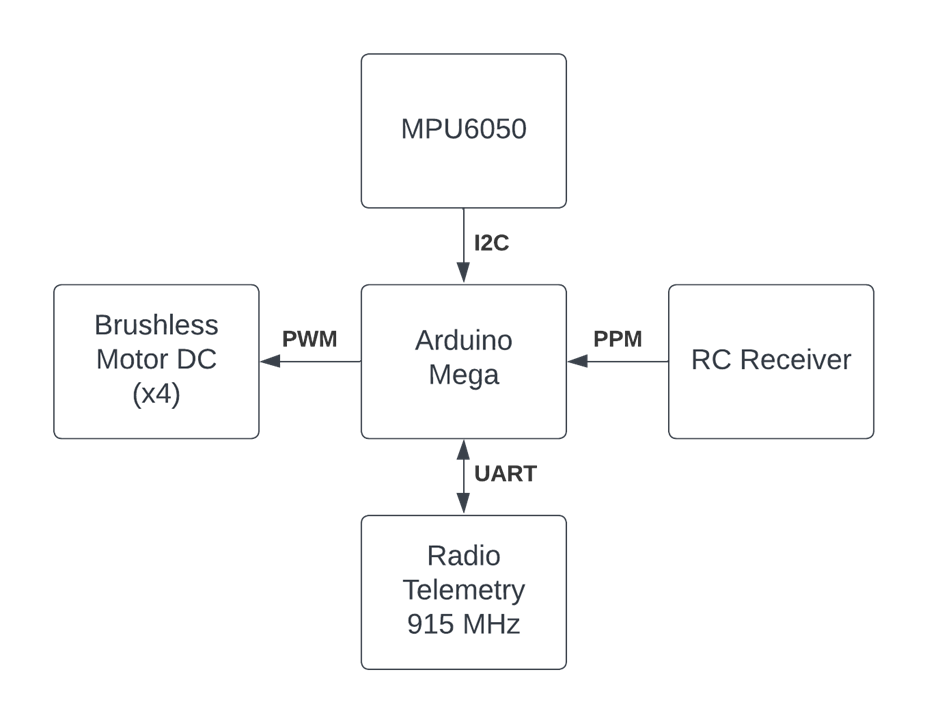


## 3.3 Control Algorithm Design



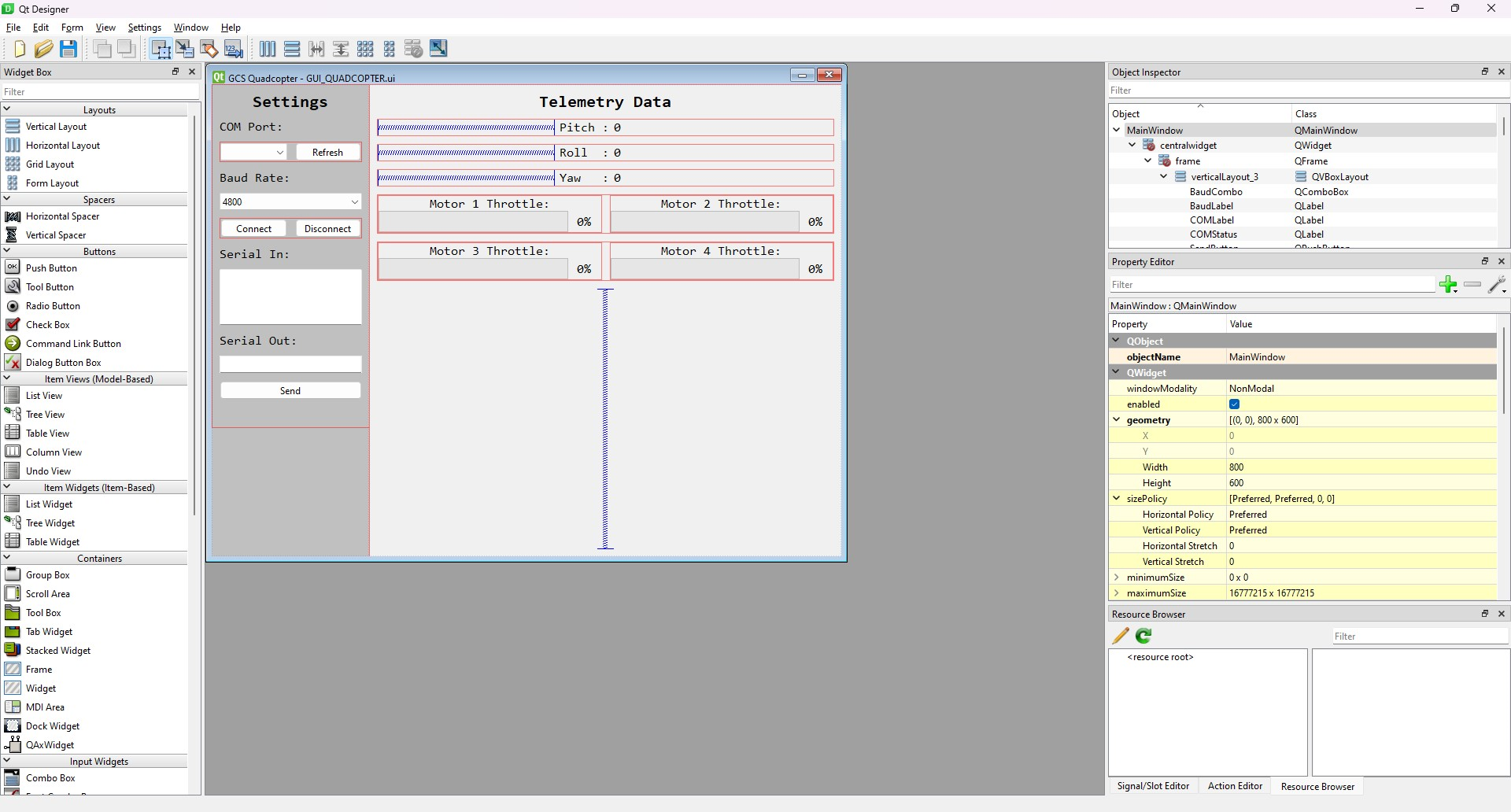
# 4 Detailed Design and Development

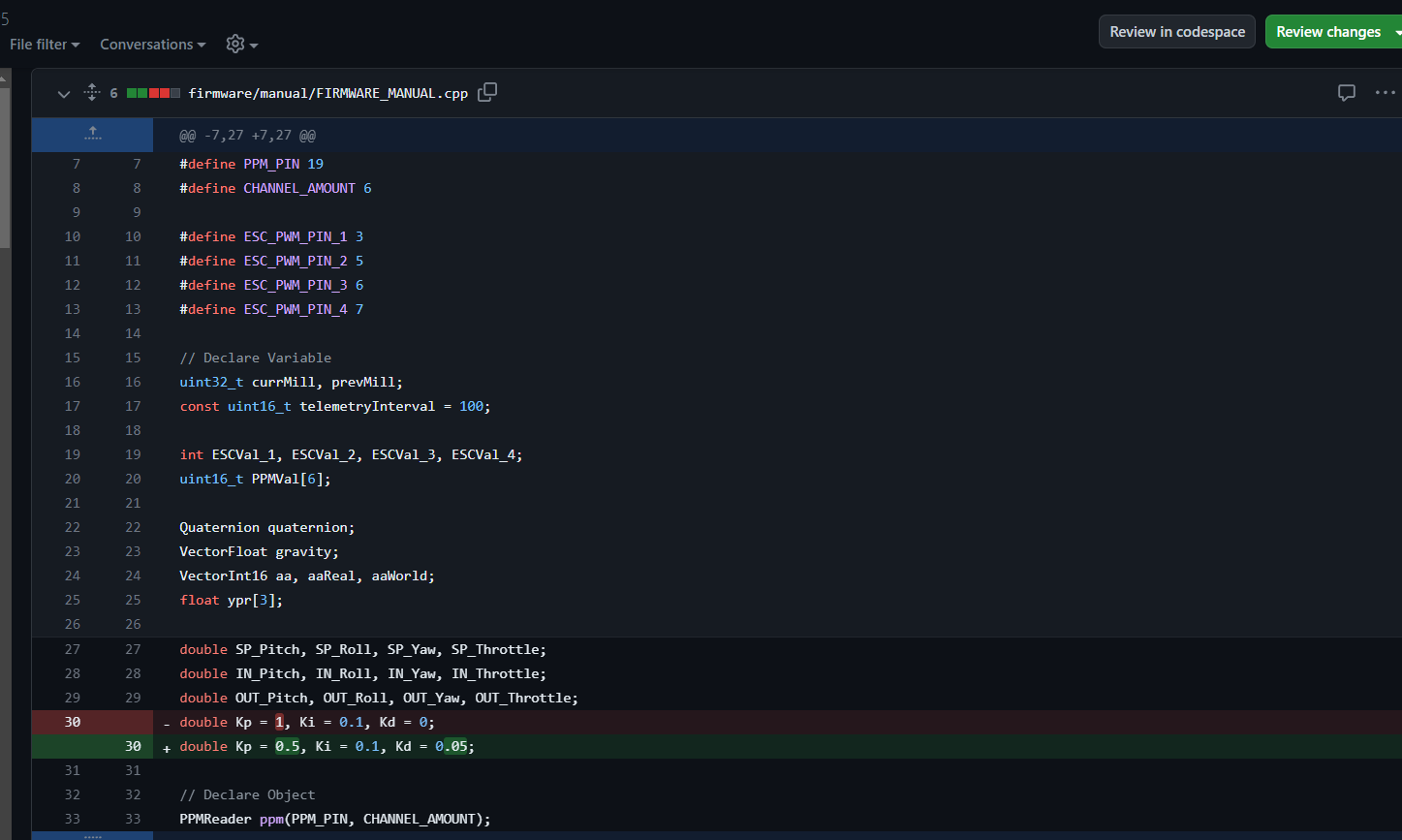
## 4.1 Component Design



## 4.2 Coding and Implementation

[Quadcopter GUI Control Beta by iwa1836 · Pull Request #5 · anh0001/Quadcopter-GUI-Control (github.com)](https://github.com/anh0001/Quadcopter-GUI-Control/pull/5)





## 4.3 Integration

Sistem komunikasi ini menggunakan radio telemetri pada frekuensi 915 MHz yang terhubung ke quadcopter melalui UART, yang selanjutnya terhubung ke laptop melalui antarmuka COM. Pada GUI (Graphical User Interface), pengguna perlu memilih COM port dan baudrate yang sesuai. Data dikirimkan dalam bentuk paket dengan representasi biner. Penting untuk memastikan bahwa konfigurasi COM port dan baudrate yang dipilih di GUI sesuai dengan pengaturan pada radio telemetri dan quadcopter. Sebagai contoh, jika COM port adalah "COM3" dan baudrate adalah 9600 bps, konfigurasi tersebut harus dipilih pada GUI untuk memastikan koneksi yang benar. Komunikasi melalui UART memungkinkan pengiriman dan penerimaan data biner antara quadcopter dan laptop. Oleh karena itu, perhatikan format dan struktur paket biner yang digunakan agar pemrosesan data dapat dilakukan dengan benar di kedua sisi komunikasi. Dengan memilih COM port dan baudrate yang sesuai, serta memastikan konsistensi format paket biner, sistem ini dapat berfungsi dengan optimal, memungkinkan pengguna untuk memantau dan mengendalikan quadcopter melalui laptop dengan menggunakan antarmuka grafis yang disediakan.

## 4.4 Unique Features

Highlight any novel features, optimizations or technologies employed.

# 5 Testing, Evaluation, and Optimization

## 5.1 Testing Strategy



Percobaan quadcopter pertama untuk mencoba stabilitas gerakan



Percobaan menggerakkan quadcopter, namun belum terlalu stabil sehingga tidak dapat membuat quadcopter terbang dengan stabil dan malah berbalik arah memutar.



Percobaan dengan menggunakan bantuan tiang penyangga, agar bisa melihat keseimbangan gerak quadcopter, dan meminimalisir terjadinya patah baling baling ato kerusakan quadcopter lainnya.



## 5.2 Performance Evaluation

**Evaluasi Kinerja Quadcopter:**

**1.** **Stabilitas Penerbangan:**

Tinjau kestabilan penerbangan quadcopter. Evaluasi sejauh mana quadcopter dapat menjaga posisi dan ketinggian yang diinginkan.

**2.** **Respons terhadap Kontrol:**

Uji respons quadcopter terhadap perintah pengendali. Pastikan bahwa quadcopter merespons dengan cepat dan akurat terhadap perubahan kontrol.

**3.** **Akurasi Navigasi:**

Nilai akurasi navigasi dan pemosisian quadcopter. Pastikan bahwa sistem navigasi seperti GPS berfungsi dengan baik.

**4.** **Durasi Baterai:**

Monitor durasi baterai quadcopter selama penerbangan. Pastikan bahwa baterai mencukupi untuk menjalankan misi atau tugas yang diinginkan.

**5.** **Keamanan:**

Tinjau protokol keamanan untuk mencegah kecelakaan atau kerusakan. Pastikan bahwa quadcopter memiliki fitur keselamatan yang memadai.

**Evaluasi Kinerja Sistem GUI:**

**1.** **User Experience (UX):**

Evaluasi pengalaman pengguna saat menggunakan GUI. Pastikan antarmuka pengguna mudah dipahami, responsif, dan ramah pengguna.

**2.** **Fungsi dan Kontrol:**

Tinjau apakah GUI menyediakan kontrol yang cukup untuk mengelola dan memonitor quadcopter. Pastikan semua fungsi yang diinginkan dapat diakses melalui antarmuka.

**3.** **Visualisasi Data:**

Periksa kemampuan GUI untuk menyajikan data dari quadcopter dengan jelas. Pastikan visualisasi data seperti peta atau grafik memberikan informasi yang berguna.

**4.** **Konektivitas:**

Uji konektivitas antara GUI dan quadcopter. Pastikan data dapat ditransfer dengan baik dan bahwa kontrol dari GUI dapat dijalankan secara efektif.

**5. Kemudahan Pemeliharaan:**

Pertimbangkan sejauh mana GUI dapat memfasilitasi pemeliharaan dan debugging. Pastikan ada mekanisme untuk melacak dan memecahkan masalah.

**6. Responsivitas GUI:**

Evaluasi responsivitas GUI terhadap perubahan kondisi quadcopter. Pastikan GUI dapat menanggapi perubahan status atau kondisi dengan cepat.

**Perbandingan dengan Target:**

**1.** **Bandingkan dengan Persyaratan Proyek:**

Bandingkan kinerja aktual quadcopter dan GUI dengan persyaratan proyek yang telah ditetapkan sebelumnya.

**2.** **Ulas Kembali Pengguna (Feedback):**

Ambil masukan dari pengguna atau pemangku kepentingan terkait dengan kinerja quadcopter dan GUI. Ulas kembali ini dapat memberikan wawasan berharga.

**3.** **Visualisasi Grafis:**

Gunakan grafik atau tabel untuk membandingkan hasil aktual dengan target yang ditetapkan. Ini dapat membantu menggambarkan perbandingan secara jelas.

Dengan melakukan evaluasi kinerja yang holistik terhadap quadcopter dan sistem GUI, Anda dapat mengidentifikasi area di mana proyek dapat ditingkatkan atau dioptimalkan untuk mencapai hasil yang diinginkan.

## 5.3 Optimization

Discuss any optimizations made to enhance system performance and user experience.

# 6 Collaboration and Project Management

## 6.1 Teamwork Dynamics

Reflect on the collaborative endeavor, roles, and contributions of team members.

## 6.2 Project Management

Document the project timeline, milestones, and management practices adopted.

# 7 Conclusion and Reflection

## 7.1 Project Summary

## Ringkasan Proyek : Quadcopter dan Sistem GUI

## **Pencapaian Utama :**

## Dapat mengintegrasikan quadcopter yang fungsional, termasuk frame, motor, baling-baling, dan pengontrol penerbangan.

## Menerapkan dan menyempurnakan algoritma kontrol penerbangan untuk memastikan penerbangan quadcopter yang stabil dan terkendali.

## **Pembelajaran :**

## Mendapatkan pengalaman berharga dalam mengintegrasikan berbagai komponen perangkat keras dan memastikan operasi harmonis dalam sistem quadcopter.

## Memperdalam pemahaman tentang dinamika penerbangan quadcopter, aerodinamika, dan dampak parameter yang berbeda terhadap performa.

## Meningkatkan keterampilan dalam pemrograman dan pengembangan perangkat lunak, terutama dalam konteks sistem kontrol waktu nyata untuk kendaraan udara tanpa awak.

## Mengembangkan keahlian dalam mengidentifikasi dan menyelesaikan masalah terkait perangkat keras, perangkat lunak, dan sistem komunikasi quadcopter.

## **Hasil:**

## Menghasilkan prototipe quadcopter yangmasi belum sepenuhnya fungsional namnun mampu terbang meski belum terlalu stabil dan manuver yang terkendali.

## **Antarmuka Pengguna Grafis (GUI):**

## Mengembangkan GUI yang ramah pengguna untuk kendali jarak jauh, memungkinkan pengguna dengan mudah mengendalikan quadcopter dengan umpan balik visual.

## Mengintegrasikan opsi konfigurasi pengaturan dalam GUI, memungkinkan pengguna untuk menyesuaikan parameter penerbangan dan pengaturan kendali.

## Melakukan pengujian dan sesi demonstrasi yang komprehensif untuk memvalidasi kinerja quadcopter, respons terhadap input kendali, dan kehandalan antarmuka GUI

## Menyusun dokumentasi rinci yang mencakup perakitan quadcopter, arsitektur perangkat lunak, dan petunjuk penggunaan GUI untuk referensi dan pengembangan potensial di masa depan.

## 7.2 Future Work

## **Pekerjaan Mendatang untuk Quadcopter dan Sistem GUI :**

## **Peningkatan Kontrol Penerbangan :**

## Eksplorasi implementasi kontrol adaptif yang lebih canggih untuk meningkatkan kinerja penerbangan quadcopter dalam berbagai kondisi cuaca dan lingkungan.

## **Integrasi Pemantauan Visual:**

## Mengintegrasikan sistem pemantauan visual yang lebih lanjut, seperti pengenalan objek dan penelusuran visual, untuk meningkatkan navigasi dan fungsi pengawasan quadcopter.

## **Pengembangan Aplikasi Industri:**

## Membuka peluang untuk penggunaan quadcopter dalam aplikasi industri, seperti pemantauan infrastruktur, survei tanah, atau inspeksi gedung.

## **Antarmuka Pengguna yang Lebih Lanjut:**

## Menambahkan elemen kontrol lanjutan dan fungsi pemrograman misi ke dalam GUI, memungkinkan pengguna untuk merencanakan dan melaksanakan misi quadcopter dengan lebih terperinci.

## **Konektivitas dan Komunikasi Lebih Lanjut:**

## Menggali opsi konektivitas jaringan yang lebih canggih untuk mendukung pertukaran data yang lebih cepat dan aman antara quadcopter dan GUI.

## **Analisis Data Sensor yang Mendalam:**

## Meneliti metode analisis data sensor yang lebih mendalam untuk memahami lingkungan sekitar quadcopter dengan lebih baik, seperti pemrosesan citra atau pemrosesan sinyal lanjutan.

## **Autonomi Tingkat Tinggi:**

## Menerapkan tingkat autonomi yang lebih tinggi, termasuk kemampuan pengambilan keputusan mandiri berdasarkan data sensor dan pemetaan otomatis.

## **Penelitian Keamanan dan Privasi:**

## Fokus pada penelitian keamanan cyber-physical untuk melindungi quadcopter dari serangan siber dan memastikan kepatuhan privasi pengguna.

## **Pengembangan Model Simulasi:**

## Membangun model simulasi yang canggih untuk menguji dan memvalidasi perangkat lunak dan algoritma quadcopter sebelum penerbangan aktual.

## **Penerapan Energi Terbarukan:**

## Mengintegrasikan teknologi energi terbarukan, seperti baterai berkinerja tinggi atau pengisian energi matahari, untuk meningkatkan efisiensi dan otonomi quadcopter.

## Dengan menggali aspek-aspek ini, proyek quadcopter dan sistem GUI dapat diperluas untuk mencakup aplikasi yang lebih luas, meningkatkan kinerja, dan menyelami area penelitian yang lebih mendalam.

## 7.3 Personal and Group Reflections

Reflect on the experience, challenges, and acquired knowledge.

# 8 Appendices

## 8.1 Bill of Materials

Detail the parts, costs, and sources.

## 8.2 Electrical Wiring and System Layout

Provide diagrams, schematics, and layout information.

## 8.3 Code Repository

Include links to the code repository, version control, and change logs.

## 8.4 Additional Documentation

Include any other relevant documentation, photos, or supporting materials.

# 9 References

Cite all references, tools, libraries, and external resources used in the project.