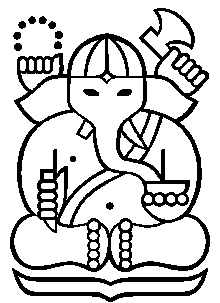
PERANCANGAN SISTEM PENELUSURAN TITIK BERBASIS KONTROL SENTRALISASI UNTUK TUGAS PEMETAAN PADA *MULTI-QUADROTOR*

LAPORAN TUGAS AKHIR

****

Oleh

Shania Argiliana NIM: 13320053

PROGRAM STUDI TEKNIK FISIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI BANDUNG  
2024

ABSTRAK

PERANCANGAN SISTEM PENELUSURAN TITIK BERBASIS KONTROL SENTRALISASI UNTUK TUGAS PEMETAAN PADA *MULTI-QUADROTOR*

Oleh

**Shania Argiliana** **NIM: 13320053**

(Program Studi Teknik Fisika)

Untuk meningkatkan efisiensi dan hasil maksimal dalam menjalankan tugas, mayoritas penelitian berfokus pada pengembangan *multi-quadrotor*. Salah satu tugas yang dapat dilakukan oleh *multi-*quadrotor adalah tugas pemetaan. Akan tetapi, pengembangan *multi-quadrotor* dapat menimbulkan potensi masalah, yaitu terjadinya tabrakan antar *quadrotor* saat menjalankan tugas. Tabrakan antar *quadrotor* dapat dihindari dengan menggunakan algoritma pemisahan kontrol sentralisasi dalam penelusuran titik refrensi. Pada algoritma pemisahan tersebut, tiap *quadrotor* memiliki jarak minimum 200 cm untuk berdekatan. Jika jarak keduanya kurang dari atau sama dengan 200 cm, kedua *quadrotor* akan bergerak saling menjauh. Selain itu, kontrol sentralisasi menyebabkan tiap *quadrotor* dianggap menjadi individu yang berbeda dan tiap *quadrotor* bergerak mencapai tujuan dengan mengikuti jalur yang telah ditentukan oleh komputer. Pada penelitian ini, *multi-quadrotor* menjalankan tugas pemetaan pada area 5m x 4m. Area tugas dibagi sesuai jumlah *quadrotor*, sehingga tiap *quadrotor* akan memiliki area masing-masing. Informasi posisi dari quadrotor diperoleh melalui pendeteksian beberapa marka fidusial persegi (MFP). Dari implementasi yang dilakukan, *quadrotor* dapat melakukan penelusuran posisi dengan berbagai arah pergerakan, yaitu maju, mundur, kanan dan kiri. Pengujian penelusuran posisi dilakukan terlebih dahulu untuk mengetahui pergerakan *quadrotor*. Dalam menjalankan tugas penelusuran posisi pergerakan arah maju, *quadrotor* A memiliki waktu tempuh 17,531 detik dan *quadrotor* B memiliki waktu 17,060 detik dengan jarak tempuh yang sama, yaitu 409,5 cm. Ketika multi-*quadrotor* menjalankan tugas pemetaan dengan menerapkan penelusuran posisi berbasis kontrol sentralisasi, *multi-quadrotor* dapat menyelesaikan tugas dalam waktu 51,137 detik dihitung dari posisi awal hingga *quadrotor* lepas landas dengan nilai RMSE masing-masing untuk *quadrotor* A sebesar 12,3 cm dan *quadrotor* B sebesar 22,5 cm.

**Kata kunci:** *quadrotor*, kontrol formasi sentralisasi*,* penelusuran titik, marka fidusial persegi

*ABSTRACT*

*DESIGN OF A CENTRALIZED CONTROL-BASED POINT TRACING SYSTEM FOR MAPPING TASKS ON MULTI-QUADROTORS*

*By*

***Shania Argiliana*** ***NIM: 13320053***

*(Engineering Physics Study Program)*

*To enhance efficiency and achieve maximum results in performing tasks, most research focuses on the development of multi-quadrotors. One of the tasks that can be carried out by multi-quadrotors is mapping. However, the development of multi-quadrotors can pose potential problems, such as collisions between quadrotors during task execution. Collisions between quadrotors can be avoided by using a centralized control separation algorithm in reference point tracking. In this algorithm, each quadrotor maintains a minimum distance of 200 cm from others. If the distance between them is less than or equal to 200 cm, the quadrotors will move away from each other. Additionally, centralized control treats each quadrotor as a separate entity, and each quadrotor moves towards its goal by following a path predetermined by the computer. In this research, multi-quadrotors perform a mapping task in an area of 5m x 4m. The task area is divided according to the number of quadrotors, so each quadrotor has its own designated area. The position information of the quadrotors is obtained through the detection of several fiducial markers (MFPs). From the implementation, the quadrotors can navigate in various directions, including forward, backward, right, and left. Position tracking tests were conducted first to determine the movement of the quadrotors. In the position tracking task moving forward, quadrotor A took 17.531 seconds and quadrotor B took 17.060 seconds to cover the same distance of 409.5 cm. When the multi-quadrotors performed the mapping task using position tracking based on centralized control, they completed the task in 51.137 seconds, measured from the initial position until the quadrotors took off, with RMSE values of 12,3 cm for quadrotor A and 22,5 cm for quadrotor B.*

***Keywords:*** *quadrotor, centralized formation control, point tracking, square fiducial marker.*

PERANCANGAN SISTEM PENELUSURAN TITIK BERBASIS KONTROL SENTRALISASI UNTUK TUGAS PEMETAAN PADA *MULTI-QUADROTOR*

HALAMAN PENGESAHAN

Oleh

**Shania Argiliana** **NIM: 13320053**

**(Program Studi Teknik Fisika)**

Institut Teknologi Bandung

Menyetujui

Tim Pembimbing

Rabu, 29 Mei 2024

|  |  |
| --- | --- |
| Pembimbing 1 | Pembimbing 2 |
| (Ir. Estiyanti Ekawati, M.T., Ph.D.)  NIP. 196908052008012020 | (Faqihza Mukhlish, S.T., M.T., Ph.D.)  NOPEG. 121110001 |
|  |  |

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan rahmat beserta karunia-Nya, serta diberi-Nya kemudahan dalam menyelesaikan penulisan Laporan Tugas Akhir yang berjudul “Perancangan Sistem Penelusuran Titik Berbasis Kontrol Sentralisasi untuk Tugas Pemetaan pada *Multi-Quadrotor*”. Penulisan Laporan Tugas Akhir ini merupakan salah satu syarat untuk menyelesaikan pendidikan tahap sarjana pada Program Studi Sarjana Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Bandung.

Laporan Tugas Akhir ini dapat terselesaikan dengan baik, tidak lepas dari bimbingan dan dukungan dari banyak pihak, sehingga dalam kesempatan ini penulis dengan segala hormat mengucapkan terima kasih kepada:

1. Ir. Estiyanti Ekawati, M.T., Ph.D., selaku dosen pembimbing pertama yang telah meluangkan waktu dan pikirannya untuk membimbing penulis dalam menyelesaikan penelitian dan penyusunan Lpaoran Tugas Akhir ini,
2. Faqihza Mukhlish, S.T., M.T., Ph.D selaku dosen pembimbing kedua yang telah meluangkan waktu dan pikirannya untuk membimbing penulis dalam menyelesaikan penelitian dan penyusunan Lpaoran Tugas Akhir ini,
3. Ir. R. Sugeng Joko Sarwono, MT., Ph.D., selaku dosen wali yang telah membimbing penulis selama menempuh pendidikan di Program Studi Sarjana Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Bandung,
4. Kedua orang tua yang selalu memberikan doa dan dukungan kepada penulis,
5. Seluruh Dosen dan Staf di Program Studi Sarjana Teknik Fisika yang telah membekali banyak ilmu pengetahuan yang bermanfaat selama penulis menjalani studi di Program Studi Sarjana Teknik Fisika,
6. Rekan-rekan di laboratorium CITA atas kebersamaan selama penelitian ini dilaksanakan,
7. Teman-teman S1 Teknik Fisika Angkatan 2020 yang memberikan dukungan dan semangat kepada penulis, dan
8. Semua pihak yang telah membantu hingga terselesaikannya Laporan Tugas Akhir ini yang tidak dapat penulis sebutkan satu per satu.

Semoga Allah SWT senantiasa memberikan limpahan rahmat dan karunia-Nya kepada semua pihak yang terlibat dalam proses penulisan Laporan Tugas Akhir ini.

Akhir kata penulis memohon maaf apabila terdapat banyak kekurangan dalam penulisan Laporan Tugas Akhir ini. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan guna penyempurnaan dan pengembangan Tugas Akhir ini ke arah yang lebih baik. Semoga segala yang tertuang dalam Laporan Tugas akhir ini memberikan manfaat bagi kita semua baik sekarang maupun dimasa yang akan datang.

Bandung, Mei 2024

Penulis

DAFTAR ISI

ABSTRAK i

[*ABSTRACK* ii](#_Toc148008190)

[HALAMAN PENGESAHAN iii](#_Toc148008191)

[KATA PENGANTAR iv](#_Toc148008192)

[DAFTAR ISI vi](#_Toc148008193)

[DAFTAR GAMBAR viii](#_Toc148008194)

[DAFTAR](#_Toc148008194) TABEL x

[DAFTAR SINGKATAN DAN LAMBANG x](#_Toc148008196)i

[BAB I PENDAHULUAN 1](#_Toc148008197)

[1.1 Latar Belakang 1](#_Toc148008198)

[1.2 Permasalahan 2](#_Toc148008199)

[1.3 Tujuan Penelitian 3](#_Toc148008200)

[1.4 Ruang Lingkup Permasalahan 3](#_Toc148008201)

[1.4.1 Batasan 3](#_Toc148008202)

[1.4.2 Asumsi 4](#_Toc148008203)

[1.5 Sistematika Penulisan 4](#_Toc148008204)

[BAB II TINJAUAN PUSTAKA 6](#_Toc148008206)

[2.1 Model Sistem *Quadrotor* 6](#_Toc148008207)

[2.1.1 Persamaan Dinamika *Quadrotor* 6](#_Toc148008202)

[2.1.2 Pendekatan Persamaan Dinamika *Quadrotor* Menggunakan FOPTD 9](#_Toc148008203)

[2.2 Pengontrol PD 10](#_Toc148008208)

[2.3 Marka Fidusial Persegi 11](#_Toc148008208)

[2.4 *Moving Average* 12](#_Toc148008208)

[2.5 Jaringan Nirkabel 12](#_Toc148008208)

[2.5.1 TCP/IP 13](#_Toc148008202)

[2.5.2 UDP 13](#_Toc148008203)

[2.6 Jenis-Jenis Konflik pada *Multi-Agent* 14](#_Toc148008208)

[2.2 Algoritma Pemisahan (*Separation*) 15](#_Toc148008208)

[BAB III METODOLOGI DAN KEMAJUAN 16](#_Toc148008213)

[3.1 Objek Penelitian](#_Toc148008207) 16

[3.2 Rancangan Penelitian 17](#_Toc148008208)

[3.3 Pengontrol 18](#_Toc148008208)

[3.4 Sistem Pemosisian 19](#_Toc148008208)

[3.4.1 Pendeteksian Marka ArUco 19](#_Toc148008203)

[3.4.2 Konversi Marka ArUco Menjadi Titik Koordinat 19](#_Toc148008203)

[3.4.3 Penentuan Posisi *Quadrotor* 20](#_Toc148008203)

[3.5 Perancangan Algoritma Pemisahan pada *Multi-Quadrotor* 25](#_Toc148008208)

[3.6 Perancangan Penelusuran Titik pada TugasPemetaan 26](#_Toc148008208)

[3.7 Komunikasi Pada *Quadrotor* 27](#_Toc148008208)

[3.8 Pengujian 28](#_Toc148008208)

[BAB IV HASIL DAN ANALISIS 31](#_Toc148008213)

[4.1 Sistem Pemosisian 31](#_Toc148008207)

[4.2 Identifikasi Karakteristik *Quadrotor* 32](#_Toc148008207)

[4.2.1 Hasil Identifikasi Karakteristik *Quadrotor* 32](#_Toc148008203)

[2.1.2 Hasil Penalaan Parameter Pengontrol PD 34](#_Toc148008203)

[4.3 Pengujian Penelusuran Posisi *Quadrotor* 36](#_Toc148008207)

[4.4 Pengujian Algoritma Pemisahan untuk Menghindari Tabrakan 40](#_Toc148008207)

[4.5 Pengujian untuk Tuga Pemetaan 42](#_Toc148008207)

[BAB V PENUTUP 46](#_Toc148008213)

[5.1 Kesimpulan 46](#_Toc148008207)

[5.2 Saran 47](#_Toc148008207)

[DAFTAR PUSTAKA 48](#_Toc148008214)

[LAMPIRAN A. DIAGRAM ALIR PENENTUAN POSISI *QUADROTOR* 50](#_Toc148008214)

[LAMPIRAN B. HASIL IDENTIFIKASI KARAKTERISTIK KECEPATAN *QUADROTOR* 51](#_Toc148008214)

[B.1 Grafik Karakteristik Kecepatan *Quadrotor* A 51](#_Toc148008207)

[B.2 Grafik Karakteristik Kecepatan Quadrotor B 53](#_Toc148008207)

[B.3 Grafik Pengontrol PD untuk *Quadarotor* A 55](#_Toc148008207)

[B.4 Grafik Pengontrol PD untuk Quadarotor A 56](#_Toc148008207)

[LAMPIRAN C. PENELUSURAN POSISI *QUADROTOR* 58](#_Toc148008214)

[C.1 Penelusuran Posisi *Quadrotor* A 58](#_Toc148008207)

[C.2 Penelusuran Posisi *Quadrotor* B 61](#_Toc148008207)

**DAFTAR GAMBAR**

[Gambar 1.1 Diagram tulang ikan rumusan permasalahan. 3](#_Toc167940065)

[Gambar 2.1 Skema gaya yang ada pada *quadrotor* dalam kerangka system *quadrotor*s [9]. 6](#_Toc167940066)

[Gambar 2.2 Ilustrasi marka ArUco [14]. 11](#_Toc167940067)

[Gambar 2.3 Proses pengolahan citra marka [5]. 11](#_Toc167940068)

[Gambar 2.4. Ilustrasi jenis-jenis konflik pada *multi-agent*: (a) konflik titik, (b) konflik tepi, (c) konflik menyusul. 15](#_Toc167940069)

[Gambar 2.5. Ilustrasi algortima pemisahan dua individu. 15](#_Toc167940070)

[Gambar 2.6. Ilustrasi jalur pada tugas pemetaan dengan pola (a) berliku-liku, (b) spiral, dan (c) S. 16](#_Toc167940071)

[Gambar 3.1. Desain karpet mat. 19](#_Toc167940072)

[Gambar 3.2. Diagram blok pengontrol PD untuk kecepatan *quadrotor.* 19](#_Toc167940073)

[Gambar 3.3. Diagram alir pendeteksian MFP ArUco. 20](#_Toc167940074)

[Gambar 3.4. Diagram alir konversi kode unik MFP menjadi koordinat pada karpet mat. 21](#_Toc167940075)

[Gambar 3.5. Hasil pendeteksian MFP pada ketinggian 80 cm. 22](#_Toc167940076)

[Gambar 3.6. Hasil pendeteksian empat marka MFP pada *quadrotor.* 22](#_Toc167940077)

[Gambar 3.7. Diagram alir pendeteksian beberapa MFP saat *quadrotor* mengalami pergerakan 23](#_Toc167940078)

[Gambar 3.8. Hasil pendeteksian dua marka MFP saat *quadrotor* mengalami pergerakan kearah kiri 24](#_Toc167940079)

[Gambar 3.9. Diagram alur perancangan algoritma pemisahan. 27](#_Toc167940080)

[Gambar 3.10. Diagram alir perancangan penelusuran titik pada tugas pemetaan. 27](#_Toc167940081)

[Gambar 3.11. Arsitektur komunikasi data. 28](#_Toc167940082)

[Gambar 3.12. Jalur pengujian individu (a) lurus terhadap sumbu x dan (b) lurus terhadap sumbu y. 29](#_Toc167940083)

[Gambar 3.13. Jalur pengujian algoritma pemisahan. 30](#_Toc167940084)

[Gambar 3.14. Jalur pengujian secara berkelompok untuk tugas pemetaan. 31](#_Toc167940085)

[Gambar 4.1. Marka MFP yang terdeteksi oleh quadrotor (a) saat bergerak pada kecepatan 32](#_Toc167940086)

[Gambar 4.2. Hasil identifikasi pergerakan ke kiri dari *quadrotor* A. 33](#_Toc167940087)

[Gambar 4.3. Respons sistem input step untuk *quadrotor* A arah maju 35](#_Toc167940088)

[Gambar 4.4. Grafik resfpon sistem *quadrotor* A arah maju dengan pengontrol PD. 37](#_Toc167940089)

[Gambar 4.5. Penelusuran posisi dengan arah maju pada sumbu x (a) rekaman posisi *quadrotor* saat bergerak maju, (b) profil posisi pada sumbu x dan (c) profil posisi sumbu y. 39](#_Toc167940090)

[Gambar 4.6. Penelusuran posisi pada penelitian Giga dengan arah maju pada sumbu x (a) posisi pada sumbu x dan (b) posisi sumbu y [5]. 40](#_Toc167940091)

[Gambar 4.7. Rekaman posisi *quadrotor* pada pengujian algoritma pemisahan (a) posisi *quadrotor* pada t = 3,789 detik, (b) posisi *quadrotor* pada t = 5,550 detik, dan (c) posisi *quadrotor* pada t = 11,000 detik. 43](#_Toc167940092)

[Gambar 4.8. Penelusuran posisi *quadrotor* A dan *quadrotor* B selama 20 detik. 44](#_Toc167940093)

[Gambar 4.9. Hasil pengujian tugas pemetaan (a) posisi *quadrotor* pada koordinat *cartesius* 45](#_Toc167940094)

**DAFTAR TABEL**

[Tabel 3.1. Spesifikasi *quadrotor* Parrot AR Drone 2.0 [25]. 17](#_Toc167940095)

[Tabel 4.1. Hasil identifikasi kecepatan *quadrotor* A. 34](#_Toc167940096)

[Tabel 4.2. Hasil identifikasi kecepatan *quadrotor* B. 35](#_Toc167940097)

[Tabel 4.3. Parameter pengontrol PD untuk *quadrotor* A dan *quadrotor* B 36](#_Toc167940098)

[Tabel 4.4. Waktu tempuh pengujian penelusuran posisi tiap *quadrotor* (detik). 41](#_Toc167940099)

[Tabel 4.5. Data hasil tugas pemetaan. 46](#_Toc167940100)

DAFTAR SINGKATAN DAN LAMBANG

|  |  |  |
| --- | --- | --- |
| SINGKATAN | Nama | Pemakaian pertama kali pada halaman |
| MFP | Marka Fidusial Persegi | 2 |
| ITB | Institut Teknologi Bandung | 2 |
| CITA | *Center for Instrumentation Technology and Automation* | 4 |
| FOPTD | *First Orde Process with Time Delay* | 9 |
| SMA | *Simple Moving Average* | 12 |
| TCP/IP | *Transmission Control Protocol/Internet Protocol* | 12 |
| WiFi | *Wireless Fidelity* | 12 |
| UDP | *User Datagram Protocol* | 13 |

|  |  |  |
| --- | --- | --- |
| LAMBANG |  |  |
|  | Sudut guling | 6 |
|  | Sudut angguk | 6 |
|  | Sudut belok | 6 |
|  | Massa *quadrotor* | 6 |
|  | Matriks identitas | 6 |
|  | Vektor linear | 6 |
|  | Gaya resultan | 6 |
|  | Momen resultan | 6 |
|  | Kecepatan sudut | 6 |
|  | Matriks inersia | 7 |
|  | Momen inersia dari *quadrotor* | 7 |
|  | Gaya mekanik yang memberikan gerak rotasi pada *quadrotor* | 7 |
|  | Faktor yang menghubungkan kecepatan sudut dengan motor | 8 |
|  | Penguatan | 9 |
|  | Konstanta waktu | 9 |
|  | Waktu tunda | 9 |
|  | Penguatan proporsional | 10 |
|  | Error | 10 |
|  | Masukan | 10 |
|  | Waktu derivatif | 10 |
|  | Rangkaian aksi setiap agen | 15 |
|  | Nilai rata-rata koordinat x | 20 |
|  | Nilai rata-rata koordinat y | 20 |
|  | Radius dua titik dalam satuan pixel | 22 |
|  | Faktor konversi satuan pixel ke cm untuk koordinat x | 22 |
|  | Faktor konversi satuan pixel ke cm untuk koordinat y | 22 |
|  | Koordinat dari pendeteksian beberapa MFP | 22 |
|  | Koordinat titik pusat kamera *quadrotor* pada bingkai kamera | 22 |
|  | Pergerakan *quadrotor* arah sumbu x positif | 23 |
|  | Pergerakan *quadrotor* arah sumbu x negatif | 23 |
|  | Pergerakan *quadrotor* arah sumbu y positif | 23 |
|  | Pergerakan *quadrotor* arah sumbu y negatif | 23 |

BAB I  
PENDAHULUAN

# Pendahuluan

## Latar Belakang

Saat ini, mayoritas penelitian berfokus pada pengembangan sekelompok *quadrotor* atau *multi-quadrotor*. *Multi-quadrotor* akan berkoordinasi dan bekerjasama sebagai satu kesatuan dengan menjaga formasi yang diinginkan selama melaksanakan tugas [1]. Dengan kemampuan koordinasi dan komunikasi, *multi-quadrotor* dapat meningkatkan efisiensi dan efektivitas dalam melakukan pemetaan, pengawasan, pecarian dan penyelamatan, serta pengiriman barang [2]. Jika dibandingkan dengan *single-quadrotor*, *multi-quadrotor* memiliki beberapa kelebihan, di antaranya cakupan area tugas yang lebih luas dengan waktu yang lebih singkat, distribusi tugas yang merata untuk tiap *quadrotor*, fleksibilitas operasional karena *multi-quadrotor* dapat diatur dalam berbagai formasi dan pola sesuai tugas, dan meningkatkan presisi hasil pengamatan atau pemetaan [3]. Dalam penelusuran tugas pemetaan oleh *multi-quadrotor*, terdapat beberapa fokus kerja yang dapat dikembangkan, yaitu pengolahan citra hasil pemetaan, kontrol turbulensi antar *quadrotor*, serta perencanaan kontrol *multi-quadrotor*. Pada penelitian ini, fokus utama yang akan dilakukan adalah perencanaan kontrol untuk *multi-quadrotor* yang melibatkan koordinasi antar *quadrotor*, kontrol formasi yang digunakan, serta memastikan cakupan area dan efisiensi waktu kerja dari *multi-quadrotor.*

Pendekatan kontrol formasi yang telah dikembangkan saat ini untuk *multi-*quadrotor dalam manjalankan tugas, antara lain kontrol pemimpin-pengikut*,* perilaku*,* dan struktur virtual [4]*.* Kontrol formasi perilaku merupakan kontrol formasi yang mendistribusikan kontrolnya ke setiap *quadrotor* dan dapat melakukan *self-conguration* ketika diberi *quadrotor* tambahan, tetapi analisis matematis yang dimiliki sangat kompleks. Kontrol formasi selanjutnya ialah struktur virtual yang memberikan suatu titik virtual sebagai pemimpin, sehingga komunikasi *multi-quadrotor* dengan *ground* *segment* harus tetap ada agar sistem stabil dan tidak terjadi kesalahan. Kontrol formasi terakhir ialah pemimpin-pengikut yang merupakan kontrol formasi dengan menunjuk salah satu *quadrotor* menjadi pemimpin, tetapi jika pemimpin mengalami kegagalan, formasi *multi-quadrotor* akan menjadi kacau [5]. Pada penelitian sebelumnya, kekurangan dari kontrol formasi pemimpin-pengikut ditangani menggunakan metode siklik pemimpin-pengikut menggunakan marka fidusial persegi (MFP) [5]*.*

Apabila diidentifikasi, pengembangan *multi-qudrotor* dapat menimbulkan potensi masalah, yaitu terjadinya tabrakan antar *quadrotor*. Untuk menghindari tabrakan atau konflik yang dialami oleh *multi-quadrotor*, algoritma *separation* atau pemisahan dapat diterapkan dengan kontrol sentralisasi pada *multi-quadrotor*. Dalam algoritma tersebut, *quadrotor* akan memiliki gaya tolak antar *quadrotor* saat jarak kedua *quadrotor* kurang dari jarak minimum. Algoritma separation atau pemisahan merupakan lagoritma yang bertujuan untuk menghindari tabrakan antar *quadrotor*. Pada algoritma tersebut, setiap *quadrotor* akan menjaga jarak tertentu dari *quadrotor* lain. Dengan demikian, ketika *quadrotor* berada pada jarak terlalu dekat, *quadrotor* akan bergerak menjauh dengan *quadrotor* lainnya [6].

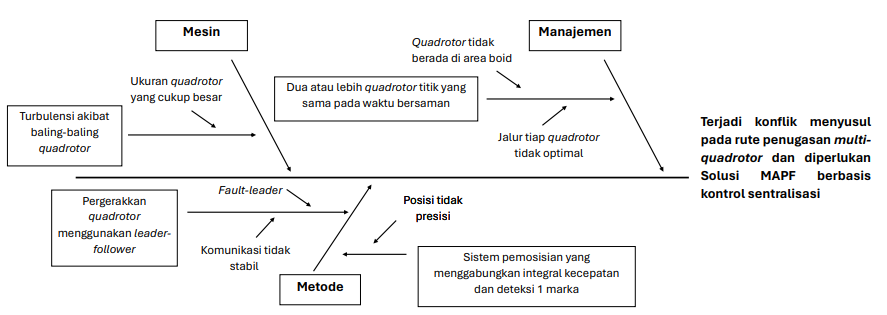
Kontrol sentralisasi pada penelitian ini digunakan untukmengirimkan keputusan yang telah diambil oleh komputer dan keputusan tersebut akan dikirimkan kembali ke *quadrotor.* Komputer akan mengirimkan jalur akan dilalui oleh *quadrotor* dalam tugas pemetaan. Kelebihan dari pendekatan ini adalah setiap *quadrotor* dianggap sebagai individu yang berbeda, sehingga apabila terjadi kegagalan tidak saling mempengaruhi. Selain itu, permasalahan terkait tabrakan yang terjadi pada konfigurasi *multi-quadrotor* saat melakukan tugas pemetaan dapat diatasi.Harapannya, hasil studi ini dapat memberikan kontribusi untuk penelitian kontrol konfigurasi multi *quadrotor* di Institut Teknologi Bandung (ITB) serta pihak-pihak terkait yang telah membantu keberhasilan studi ini dilaksanakan.

## Permasalahan

Berdasarkan latar belakang yang telah disampaikan, permasalahan yang ingin dijawab pada penelitian tugas akhir ini adalah sebagai berikut.

1. Bagaimana cara mendapatkan sistem penelusuran posisi *quadrotor* untuk pergerakan sumbu-x dan sumbu-y yang lebih baik dari penelitian sebelumnya?
2. Bagaimana cara mengontrol *multi-quadrotor* agar terhindar dari tabrakan?
3. Bagaimana cara mendapatkan performa dari *multi-quadrotor* dalam menjalankan tugas pemetaan?

Pencarian akar masalah dilakukan dengan menggunakan analisis sebab-akibat (*fishbone diagram*), seperti pada gambar berikut.



Gambar 1.1 Diagram tulang ikan rumusan permasalahan.

## Tujuan Penelitian

Berdasarkan latar belakang yang telah disampaikan serta untuk menjawab permasalahan yang ditemukan, tujuan penelitian tugas akhir ini adalah sebagai berikut.

1. Menentukan sistem penelusuran posisi *quadrotor* yang lebih baik dari penelitian sebelumnya tanpa menggunakan integrasi kecepatan.
2. Merancang sistem kontrol pada *multi-quadrotor* agar terhindar dari tabrakan.
3. Menghitung performa dari *multi-quadrotor* dalam menjalankan tugas pemetaan dengan berbasis kontrol sentralisasi.

## Ruang Lingkup Permasalahan

Berdasarkan permasalahan dan tujuan penelitian, terdapat ruang lingkup permasalahan ditentukan oleh batasan dan asumsi yang digunakan pada penelitian. Berikut merupakan ruang lingkup permasalahan yang ditentukan untuk penelitian ini.

### Batasan

Adapun batasan-batasan yang digunakan pada penelitian tugas akhir ini adalah sebagai berikut.

1. *Quadrotor* yang digunakan merupakan AR Drone 2.0 yang telah disediakan di Laboratorium CITA dengan jumlah 2 *quadrotor*.
2. *Quadrotor* bekerja pada sebuah karpet MAT yang tersusun dari 500 MFP, setiap MFP memiliki nomor identitas yang unik.
3. *Quadrotor* dapat mengetahui posisi dengan membaca titik koordinat posisi dari area yang dilewati menggunakan pendeteksian beberapa MFP.
4. Pengujian dilakukan di salah satu ruang CITA dengan ukuran 5,9 meter x 4,7 meter.

### Asumsi

Adapun asumsi yang digunakan pada penelitian tugas akhir ini adalah sebagai berikut.

1. Kecepatan dari masing-masing *quadrotor* dianggap konstan.
2. Ketinggian terbang dari masing-masing *quadrotor* dianggap sama dan konstan.
3. Sudut hadap pada setiap *quadrotor* dianggap sama dengan cara menempatkan *quadrotor* pada arah yang sama sebelum lepas landas.

## Sistematika Penulisan

Proposal penelitian tugas akhir ini terdiri dari tiga bab dengan sistematika penulisan sebagai berikut.

1. BAB I Pendahuluan

BAB I berisi tentang penjelasan singkat mengenai penelitian yang akan dilakukan, mulai dari latar belakang, perumusan masalah, tujuan penelitian, ruang lingkup dan batasan penelitian, asumsi, metode penelitian, dan sistematika penelitian.

1. Bab II Tinjauan Pustaka

Bab II berisi tinajuan pustaka yang menunjang atau digunakan dalam pembuatan tugas akhir ini.

1. Bab III Metodologi

Bab III berisi tentang objek penelitian, tahapan penelitian, pengontrol yang digunakan untuk *quadrotor*, sistem pemosisian yang digunakan pada *quadrotor*, perancangan algoritma separation atau algoritma pemisahan untuk *multi-*quadrotor agar terhindar dari tabrakan, serta pengujian penelusuran titik berbasis kontrol sentralisasi pada *multi-quadrotor* untuk tugas pemetaan.

1. Bab IV Pembahasan

Bab IV berisi tentang hasil dan analisis yang diperoleh dari penelitian, di antaranya menentukan kecepatan maksimum *quadrotor* untuk mendeteksi MFP, identifikasi karakteristik kecepatan *quadrotor* untuk menentukan parameter pengontrol PD, pengujian penelusuran posisi *quadrotor* dengan pergerakan ke arah maju, mundur, kanan, dan kiri, serta pengujian penelususran posisi berbasis kontrol sentralisasi untuk tugas pemetaan pada *quadrotor*.

1. BAB V Kesimpulan dan Saran

Bab ini berisi tentang simpulan terkait hasil yang diperoleh dari penelitian sesuai dengan batasan dan asumsi masalah. Selain itu, bab ini juga membahas mengenai saran terhadap masalah-masalah yang belum diselesaikan pada penelitian sebagai pengembangan dan perbaikan terkait penelitian ini.

BAB II  
TEORI

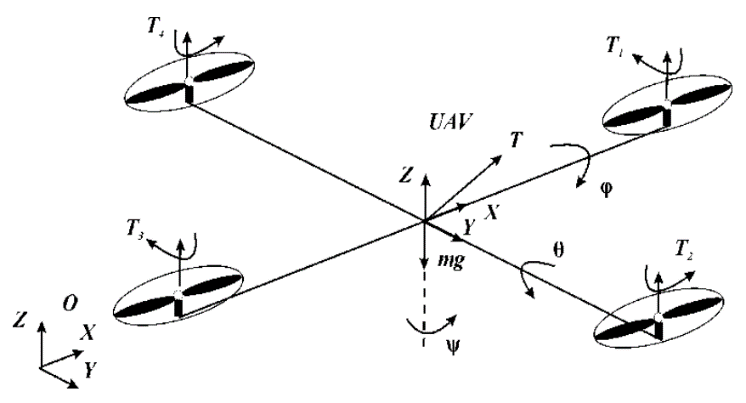
Pada bab ini akan dibahas terkait tinjauan pustaka yang akan digunakan pada penelitian, seperti persamaan dinamika *quadrotor* yang merepresentasikan model sistem *quadrotor*, persamaan fungsi transfer *first order plus time delay* (FOPTD) sebagai bentuk sederhana model sistem *quadrotor*, komunikasi yang digunakan pada quadrotor, pengontrol PD, pendeteksian marka fidusial persegi (MFP), dan algoritma pemisahan untuk menghindari tabrakan.

# Tinjauan Pustaka

## Model Sistem *Quadrotor*

### Persamaan Dinamika *Quadrotor*

*Quadrotor* merupakan kendaraan terbang nirawak yang dikembangakan dari konsep robot terbang. *Quadrotor* memiliki empat motor dengan dua pasang motor yang bergerak berlawanan arah. Selain itu, *quadrotor* dapat dikontrol secara manual dengan pengontrol jarak jauh atau menggunakan kontrol otomatis. Untuk mengontrol *quadrotor* diperlukan dinamika quadrotor [7]. Dinamika *quadrotor* dapat digambarkan pada koordinat tetap yang terhubung dengan bumi (kerangka O) dan koordinat bergerak yang bertepatan dengan pusat gravitasi *quadrotor* (kerangka UAV). Pada Gambar 2.1, variabel merupakan variabel yang berkaitan dengan sudut guling, angguk,dan belok yang digambarkan pada koordinat bumi (kerangka O) [8].



Gambar 2.1 Skema gaya yang ada pada *quadrotor* dalam kerangka system *quadrotor*s [9].

Persamaan dinimis dari quadrotor dalam ruang 3D dapat dituliskan sebagai berikut.

|  |  |
| --- | --- |
|  | 2.1 |

Dengan merupakan vektor linear, merupakan kecepatan sudut, merupakan massa *quadrotor*, merupakan matriks inersia dari objek yang dikontrol, merupakan matriks identitas, merupakan gaya resultan, merupakan momen resultan yang bekerja pada pusat massa *quadrotor*. Dengan mengabaikan gaya eksternal, gaya yang dihasilkan untuk menentukan pergerakan *quadrotor* di bidang vertikal dapat dituliskan sebagai berikut ini.

|  |  |
| --- | --- |
|  | 2.2 |

Dalam sistem koordinat Euler, orientasi *quadrotor* relatif terhadap sistem koordinat inersial XYZ sehingga dapat direpresentasikan oleh matriks rotasi R.

|  |  |
| --- | --- |
|  | 2.3 |

Dengan matriks rotasi R dituliskan seperti di bawah ini.

|  |  |
| --- | --- |
|  | 2.4 |

Pada matriks di atas, dan . Apabila Persamaan (2.3) dan Persamaan (2.4) dilakukan kombinasi, maka akan didapatkan hasil persamaan seperi di bawah ini.

|  |  |
| --- | --- |
|  | 2.5 |
|  | 2.6 |
|  | 2.7 |

Kemudian, misalkan adalah momen inersia dari *quadrotor* di sumbu X, Y, dan Z yang melewati pusat massa, merupakan kecepatan sudut yang berkaitan dengan sumbu X, Y, dan Z, serta gaya eksternal dari *quadrotor* dapat diabaikan, maka persamaan diferensial kecapatan sudut dapat direpresentasikan dalam bentuk berikut ini.

|  |  |
| --- | --- |
|  | 2.8 |
|  | 2.9 |
|  | 2.10 |

Persamaan (2.8) s.d Persamaan (2.10) di atas merupakan persamaan dinamika Euler. Ketika aksi dari motor dipertimbangkan, maka Persamaan (2.8) s.d. Persamaan (2.10) di atas dapat dituliskan sebagai berikut.

|  |  |
| --- | --- |
|  | 2.11 |
|  | 2.12 |
|  | 2.13 |

Pada persamaan di atas, merupakan jarak antar pusat massa *quadrotor* dan motor , sedangkan merupakan inersia motor. merupakan gaya mekanik yang memberikan gerakan rotasi pada *quadrotor* yang dapat ditulis persamaannya sebagai berikut.

|  |  |
| --- | --- |
|  | 2.14 |
|  | 2.15 |
|  | 2.16 |
|  | 2.17 |

pada keempat persamaan di atas merupakan faktor yang menghubungkan momen dengan kecepatan sudut dari motor ke-. Dengan mempertimbangkna identitas motor, lokais simetri dari motor, dan struktur yang kaku, momen dan pada ketinggian tetap di sumbu Z dapat dinyatakan sebagai berikut.

|  |  |
| --- | --- |
|  | 2.18 |

Dengan demikian, dinamika rotasi dari *quadrotor* dari Persamaan (2.11) s.d. Persamaan (2.13) dapat ditulis ulang dalam bentuk berikut ini.

|  |  |
| --- | --- |
|  | 2.19 |
|  | 2.20 |
|  | 2.21 |

Jika diberikan keadaan awal dan momen diketahui nilainya, koordinat dapat ditemukan dengan melakukan dua kali penurunan pada Persamaan (2.21). Berikut ini merupakan persamaan dari .

|  |  |
| --- | --- |
|  | 2.22 |

Hal yang sama dilakukan untuk posisi dan . Sehingga persamaan dan dapat dinyatakan sebagai berikut.

|  |  |
| --- | --- |
|  | 2.23 |
| |  |  | | --- | --- | |  | 2.22 | | 2.24 |

Persamaan (2.22) s.d. Persamaan (2.24) merupakan persamaan dinamika *quadrotor* secara nonlinear, sehingga persamaan ini dapat diselesaikan dengan metode numerik Runge-Kutta.

### Pendekatan Persamaan Dinamika *Quadrotor* Menggunakan FOPTD

Karena persamaan dinamika quadrotor yang kompleks, persamaan dinamika *quadrotor* tersebut dapat didekati menggunakan *First Orde Process with Time Delay* (FOPTD) agar memudahkan proses analisis dan pengembangan model kontrol [9]. FOPTD merupakan model matematis sederhana yang terdiri dari komponen dasar orde pertama dan waktu tunda. Dalam domain waktu, pendekatan FOPTD dapat dirumuskan secara umum sebagai berikut.

|  |  |
| --- | --- |
|  | 2.25 |

Dengan demikian, persamaan dinamika dari *quadrotor*, yaitu Persamaan (2.22) s.d. Persamaan (2.24) dapat dinyatakan sebagai berikut dalam domain waktu menggunakan FOPTD.

|  |  |
| --- | --- |
|  | 2.26 |
|  | 2.27 |
|  | 2.28 |

Dengan K merupakan penguatan, merupakan konstanta waktu, dan td merupakan waktu tunda. Kemudian, persamaan pada domain waktu tersebut ditransformasi ke dalam domain laplace. Secara sederhana, Persamaan (2.25) tersebut dapat ditransformasi ke dalam domain laplace seperti di bawah ini.

|  |  |
| --- | --- |
|  | 2.29 |

Persamaan (2.29) tersebut menghubungkan respons sistem dalam bentuk fungsi transfer.

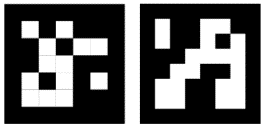
## Pengontrol PD

Pengontrol PID merupakan jenis pengontrol klasik yang digunakan dalam sistem kontrol otomatis untuk mengatur suatu sistem. Salah stau jenis pengontrol klasik tersebut adalah pengontrol PD [10 – 11]. Pengontrol proposrsional derivatif (PD) merupakan pengontrol yang terdiri dari dua parameter, yaitu penguatan proporsional dan waktu derivatif . Pengontrol derivatif digunakan untuk mempercepat respons awak suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunakanya. Dengan demikian, pengontrol derivatif digunakan dengan pengontrol lainnya. Pengontrol PD dapat dituliskan pada persamaan (2.30).

|  |  |
| --- | --- |
|  | 2.30 |

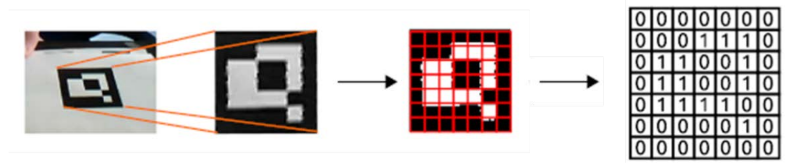
## Marka Fidusial Persegi

Marka fidusial persegi (MFP) merupakan marka berbentuk persegi yang memiliki pola tertentu di setiap markanya dan mengandung kode unik tertentu, seperti pada Gambar 2.1. Dengan demikian, marka ini digunakan untuk mengidentifikasi dan memposisikan objek secara akurat [12]. Selain itu, MFP memiliki empat titik korespondensi yang dapat berfungsi sebagai estimasi pose kamera. Terdapat beberapa pustaka yang dapat dipakai dalam pendeteksian marka, yaitu ARToolKit, Matrix, ARTag, ARToolKitPlus, QR, Maxicode, BinARyID, AprilTags dan ArUco. ArUco merupakan salah satu pustaka pendeteksi yang paling handal dan kokoh dalam mendeteksi marka, sehingga digunakan pada penelitian [13].



Gambar 2.2 Ilustrasi marka ArUco [14].

Pendeteksian MFP dilakukan dengan dua proses, yaitu deteksi marka dan indetifikasi informasi marka. Proses pendeteksian MPF dapat diilustrasikan seperti pada Gambar 2.2.



Gambar 2.3 Proses pengolahan citra marka [5].

Pada proses deteksi marka, citra yang ditangkap oleh kamera akan dianalisis untuk menemukan bentuk-bentuk persegi yang menjadi kandidat marka. Hal pertama yang dilakukan adalah proses pengambangantpada citra masukan menggunakan metode *local threshold* untuk memperhitungkan variabel iluminasi. Kemudian, kontur diekstraksi dari hasil citrasebelumnya dan dilakukan penyaringan. Citra yang dihasilkan dari proses penyaringan tersebut akan dilakukan pendekatan poligon untuk mendapatkan bentuk persegi, sehingga citra selain bentuk segi empat poligon akan dibuang. Selain itu, penyaringan ekstra juga diterapkan untuk menghapus kontur yang terlalu kecil atau terlalu besar dan menghapus kontur yang terlalu dekat satu sama lain [15].

Setelah proses deteksi marka, proses identifikasi informasi marka akan dilakukan dengan menganalisis kode intrenal pada setiap marka. Proses identifikasi diawali dnegan mengekstraksi bit-bit yang ada pada setiap marka. Untuk mendapatkan tanda dalam bentuk kanonik, transformasi perspektif diterapkan terlebih dahulu pada proses tersbeut. Citra kanonik yang dihasilkan akan dilakukan pengambangan menggunakan metode Otsu untuk memisahkan bit putih dan hitam. Selanjutnya, citra tersebut akan dibagi menjadi sel-sel berbeda sesuai dengan ukuran tanda dan ukuran tepi. Kemudian, jumlah piksel hitam atau putih di setiap sel dihitung untuk menentukan bit putih atau hitam. Pada tahap terakhir, bit-bit tersebut dianalisis sesuai dengan kamus yang tersedia pada marka ArUco [15].

## *Moving Average*

*Moving average* merupakan salah satu konsep untuk melakukan analisis data dengan cara merata-ratakan serangkaian yang terus berubah seiring waktu. *Moving average* digunakan untuk meratakan fluktuasi periodik dalam data. Salah satu jenis *moving average* sederhana adalah *Simple Moving Average* (SMA) [16]. *Simple Moving Average* (SMA) merupakan jenis *moving average* yang digunakan untuk menghitung nilai rata-rata suatu data pada periode tertentu. Persamaan untuk menghitung SMA dapat dituliskan seperti Persamaan (2.32).

|  |  |
| --- | --- |
|  | 2.32 |

dengan merupakan nilai data setiap periode dan adalah jumlah periode waktu yang digunakan untuk menghitung SMA.

## Jaringan Nirkabel

Jaringan nirkabel didefinisikan sebagai jaringan transmisi yang tidak menggunakan kabel sebagai media dalam menghubungkan sumber data dan penerima data. Jaringan nirkabel merupakan jaringan yang lebih fleksibel dan sederhana, sehingga memungkinkan banyak pengguna terhubung secara bersamaan ke dalam jaringan [17]. Jaringan nirkabel memiliki *Access Point* yang bekerja dengan menggunakan *transceiver radio* untuk menciptakan koneksi. Sinyal radio yang ditransmisikan akan diidentifikasi oleh perangkat klien. Setelah saluran komunikasi dikonfirmasi, akses ke jaringan menjadi tersedia. Access Point nirkabel mengikuti standar protokol IEEE 802.11, umumnya standar ini diterapkan dalam WiFi atau *Wireless Fidelity* [18].

### TCP/IP

TCP/IP merupakan singkatan dari *Transmision Control Protocol/Internet Protocol*. TCP/IP adalah kumpulan protokol komunikasi yang dapat digunakan untuk menghubungkan perangkat-perangkat dalam jaringan komputer. Dengan demikian, satu komputer dengan TCP/IP dapat berkomunikasi dengan komputer lain pada satu lingkup jaringan nirkabel [19]. TCP/IP terdiri dari dua protokol utama, yaitu TCP dan IP.

Protokol TCP (*Transmision Control Protocol*) bertanggung jawab untuk memastikan pengiriman data yang andal antar perangkat dalam jaringan. TCP akan membagi data menjadi paket-paket kecil, memastikan setiap paket yang datang dengan benar, dan menyusun kembali paket-paket tersebut. Selain itu, TCP berfungsi untuk memastikan bahwa data yang dikirim menjadi paket-paket kecil tersebut tidak hilang, rusak, atau urutan paket tertukar [20].

Protokol IP (*Internet Protocol*) merupkana protokol yang berfungsi untuk menangani pengiriman paket data antar perangkat dalam jaringan. IP akan mengatur alamat unik (IP *address*) dari setiap perangkat yang terhubung ke jaringan dan memsatikan bahwa paket data dikirimkan ke tujuan yang tepat berdasarkan alamat unit tersebut [20].

### UDP

UDP atau *User Dtaagram Protocol* adalah protokol transmisi data dalam jaringan yang berbeda dengan TCP/IP. UDP bersifat sederhana karena data yang dikirim berupa datagram dan tidak memiliki informasi verifikasi, sehingga memiliki kemungkinan data aan rusak saat pengiriman. Akan tetapi, pengiriman data yang dilakukan pada protokol UDP lebih cepat dibandingkan protokol TCP/IP [21].

## Jenis-Jenis Konflik pada *Multi-Agent*

Pada *multi-agent* yang berkerja secara bersama dan saling berkoordinasi, terdapat beberapa jenis konflik yang terjadi. Jenis konflik tersebut perlu didefinisikan dengan jelas agar proses perencanaan jalur dapat dilakukan dengan tepat [22]. Untuk mempermudah definisi dari setiap konflik, konflik tersebut diibaratkan terjadi pada dua agen, yaitu dan . Berikut ini merupakan konflik-konflik yang terjadi pada *multi-agent* [23].

1. Konflik Titik

Konflik titik merupakan sebuah konflik yang terjadi antar agen apabila kedua agen berada pada titik yang sama dengan waktu bersamaan, seperti Gambar 2.4 (a). Bentuk matematis dari konflik titik dapat dituliskan sebagai berikut.

|  |  |
| --- | --- |
|  | 2.35 |

1. Konflik Tepi

Konflik tepi merupakan sebuah konflik yang terjadi karena kedua agen memberikan aksi untuk melintasi rute yang sama pada waktu bersamaan dan memiliki arah yang sama, seperti Gambar 2.4 (b). Bentuk matematis dari konflik tepi dapat dituliskan sebagai berikut.

|  |  |
| --- | --- |
|  | 2.36 |

1. Konflik Menyusul

Konflik menyusul merupakan sebuah konflik yang terjadi antar dua agen apabila karena salah satu agen akan menempati posisi yang sebelumnya ditempati oleh agen yang lain, seperti Gambar 2.4 (c). Bentuk matematis dari konflik menyusul dapat dituliskan sebagai berikut.

|  |  |
| --- | --- |
|  | 2.37 |

A diagram of a positive and a positive

Description automatically generatedA diagram of a smiling face

Description automatically generatedA diagram of a diagram of a person's face

Description automatically generated with medium confidence

1. (b) (c)

Gambar 2.4. Ilustrasi jenis-jenis konflik pada *multi-agent*: (a) konflik titik, (b) konflik tepi, (c) konflik menyusul.

## Algoritma Pemisahan (*Separation*)

Algortima *separation* atau algoritma pemisahanmerupakan salah satu aturan dasar yang digunakan dalam simulasi perilaku kelompok hewan, seperti kawanan burung yang disebut “*boids*”. Algoritma pemisahan digunakan untuk menghindari tabrakan antar individu dalam satu kelompok. Dengan demikian, algoritma pemisahan dapat memastikan bahwa setiap individu menjaga jarak aman dengan individu lain saat menjalakan tugas dalam satu kelompok. Hal tersebut yang memungkinkan algoritma pemisahan dapat menghindari kepadatan yang berlebih dan tabrakan yang tidak diinginkan [6]. Perancangan algoritma pemisahan (*separation*) dilakukan dengan beberapa langkah, di antaranya identifikasi tetangga terdekat, menghitung dan mengkombinasikan vektor *avoidance*, serta penyesuaian arah dan kecepatan dari tiap individu [24]. Dengan demikian, setiap individu akan memeriksa individu lain dalam radius tertentu yang disebut sebagai zona pemisahan. Gambar 2.6 merupakan ilustrasi terkait algoritma separation.

Diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a

Description automatically generated

Gambar 2.5. Ilustrasi algortima pemisahan dua individu.

## Jenis-Jenis Jalur Pemetaan pada *Quadrotor*

Dalam menjalankan tugas pemetaan, *multi-quadrotor* dapat bergerak dengan mengikuti beberapa jalur pemetaan [3]. Berikut ini merupakan beberapa jenis jalur pemetaan yang dapat diterapkan pada *multi-quadrotor*.

* 1. Jalur Berliku-liku

Jalur Berliku-liku merupakan jalur yang berbentuk pola berliku-dan merupakan salah satu pola yang efektif untuk tugas-tugas sederhana yang membutuhkan penutupan area secara menyeluruh. Akan tetapi, jalur tersebut tidak cocok untuk area yang kompleks. Gambar 2.6 (a) merupakan ilustrasi dari jalur berliku-liku pada tugas pemetaan.

* 1. Jalus Spiral

Jalur spiral merupakan jalur pemetaan yang menyebabkan *quadrotor* terbang dengan pola spiral dari pusat menuju keluar area atau sebaliknya. Gambar 2.6 (b) merupakan ilustrasi dari jalur spiral pada tugas pemetaan.

* 1. Jalur Pola S

Jalur pola S merupakan jalur pemetaan yang menyerupai huruf “S”, sehingga *quadrotor* bergerak secara bergantian dari satu sisi ke sisi lainnya. Jalur pola S digunakan untuk pemetaan hutan atau survei agrikultur. Selain itu, jalur pola S dapat memberikan haisl pemetaan yang efisien tanpa memerlukan banyak data. Gambar 2.6 (c) merupakan ilustrasi dari jalur spiral pada tugas pemetaan.

A line drawing of a graph

Description automatically generated A black and white spiral

Description automatically generated

1. (b)

A black line on a white background

Description automatically generated

(c)

Gambar 2.6. Ilustrasi jalur pada tugas pemetaan dengan pola (a) berliku-liku, (b) spiral, dan (c) S.

BAB III   
METODOLOGI

Pada bab ini akan dijelaskan terkait metode penelitian, perancangan sistem, spesifikasi dari objek penelitian, pemilihan pengontrol, pendeteksian beberapa MFP untuk posisi *quadrotor*, dan langkah-langkah pengujian yang dilakukan pada penelitian.

# Penutup

## Objek Penelitian

Objek penelitian yang digunakan pada penelitian adalah Parrot AR Drone 2.0 dengan spesifikasi yang ditunjukkan pada Tabel 3.1.

Tabel 3.1. Spesifikasi *quadrotor* Parrot AR Drone 2.0 [25].

| **No.** | **Kategori** | **Spesifikasi** |
| --- | --- | --- |
| 1. | Kamera | Hoizontal: HD 720p 30 fps (sensor CMOS dengan lensa lebar 90 Derajat)  Vertikal: QVGA 360p 30fps |
| 2. | Tipe Perangkat Input | USB 2.0 *high speed for extentions* |
| 3. | Penyimpanan | 1 Gbit DDR2 RAM 200MHz |
| 4. | Tipe Baterai | *Lithium Polymer Battery* |
| 5. | Kapasitas Baterai | (3 sel, 11,1 V, 1000 mAh)  4 motor *brushless*, (35000 rpm, *power*: 15W)  *Discharging capacity*: 10 C  *Battery charging time*: 90 *minutes*  *Flying time*: *about* 12 *minutes* |
| 6. | Dimensi | 517 mm x 451 mm |
| 7. | Massa | Total massa 380 gram dengan *outdoor hull* dan 420 gram dengan *indoor hull* |
| 8. | Lainnya | 1 GHz 32 bit ARM Cortex A8 *processor* dengan 800MHz video DSP  Linux 2.6.32  3 *axis gyroscope* 2000/*second precison*  3 *axis accelerometer* ± 50 mg *precision*  3 *axis magnetometer* 6 *precision*  Kecepatan : 5 m/s |

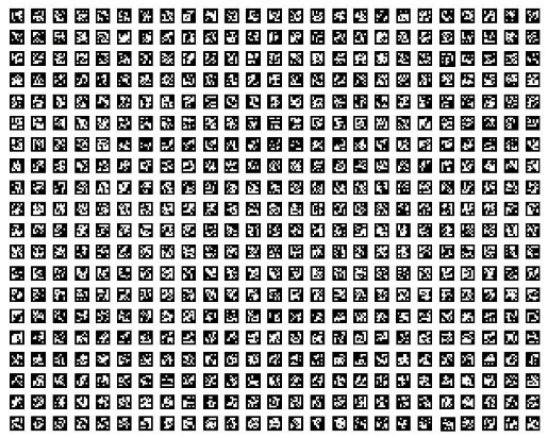
Seluruh informasi pada *quadrotor* dapat diakses menggunakan protokol *User Data Protocol* (UDP) [25]. Informasi pada *quadrotor* diakses melalui beberapa *port*, yaitu *port* 5554 untuk mengakses informasi navigasi, *port* 5555 untuk mengakses video, *port* 5556 untuk mengubah konfigurasi dan pengendalian *quadrotor*. Selain itu, prosedur yang digunakan pada Parrot AR Drone 2.0 adalah AT\*COMMAND.

Prosedur AT\*COMMAND merupakan prosedur yang menggunakan bahasa pemrograman C/C++. Akan tetapi, pada penelitian ini, prosedur AT\*COMMAND dilakukan dengan menggunakan bahasa pemrograman python. Hal tersebut dilakukan karena bahasa pemrograman python memiliki fitur pustaka yang dapat diintegrasikan untuk mendeteksi MFP. Untuk mengakses prosedur AT\*COMMAND, maka digunakan pustaka PyArdrone pada penelitian ini.

## Rancangan Penelitian

Metode penelitian yang akan digunakan secara umum mengadaptasi metode pendeteksian beberapa MFP untuk mendapatkan informasi *quadrotor*. Selain itu, penelitian ini dilakukan dengan menggunakan kontrol formasi sentralisasi yang menganggap masing-masing quadrotor merupakan individu berbeda. Dengan demikian, penelitian ini berbeda dengan penelitian sebelumnya yang menggunakan algoritma Pemimpin-Pengikut sebagai kontrol formasi *multi-quadrotor*, serta penentuan posisi yang mengintegrasikan penggunaan MFP dan pengintegralan kecepatan *quadrotor*.

Pada implementasinya, *quadrotor* dioperasikan di atas permukaan yang dilengkapi dengan karpet mat yang tersusun dari 500 marka ArUco berukuran 6x6. Susunan Marka ArUco ini berfungsi sebagai koordinat untuk pergerakkan *quadrotor* yang terlihat pada Gambar 3.1. MFP yang digunakan pada penelitian ini memungkinkan *quadrotor* untuk menentukan posisi secara presisi dan bergerak dengan tepat sesuai koordinat yang telah ditetapkan. Pada penelitian ini, karpet mat yang digunakan 500 marka ArUco berukuran 6x6 dengan ukuran karpet mat 5m x 4m.



Gambar 3.1. Desain karpet mat.

*Multi-quadrotor* akan diberikan tugas untuk melakukan pemetaan pada karpet mat. Akan tetapi, pelaksanaan tugas pemetaan untuk *multi-quadrotor* dapat berpotensi terjai tabrakan atau konflik antar *quadrotor*. Dengan demikian, penelusuran titik untuk tugas berbasis kontrol sentralisasi dan algoritma pemisahan akan diterapkan pada *multi-quadrotor*. Kontrol sentralisasi yang dirancang akan mengirimkan perintah kepada tiap *quadrotor* untuk mencapai titik tujuannya. Perintah tersebut merupakan penentuan jalur untuk setiap *quadrotor* agar tidak saling bertabrakan. Sebelum diimplementasikan pada *multi-quadrotor*, penelusuran titik pada tugas pemetaan akan disimulasikan terlebih dahulu menggunakan simulasi Pygame python. Pengujian akan dilakukan secara berkelompok untuk melihat nilai RSME yang dihasilkan dari skenario yang telah dibuat dan memastikan bahwa tidak ada *quadrotor* yang saling bertabrakan saat melaksanakan tugas.

## Pengontrol

Pengontrol yang digunakan pada penelitian ini adalah pengontrol PD yang telah terbukti pada penelitian sebelumnya memiliki karakteristik yang baik untuk *quadrotor* dan proses penalaan yang lebih mudah sehingga tidak memberikan *overshoot* maupun *undershoot* [5]. Gambar 3.2 merupakan diagram blok pengontrol PD untuk mengatur kecepatan *quadrotor*.

A black rectangle with black text

Description automatically generated

Gambar 3.2. Diagram blok pengontrol PD untuk kecepatan *quadrotor.*

## Sistem Pemosisian

Berbeda dari penelitian sebelumnya yang menggunakan integrasi kecepatan *quadrotor* dan penggunaan MFP [5], sistem pemosisian yang digunakan pada penelitian ini merupakan hasil pendeteksian dari beberapa MFP. Proses pendeteksian beberapa MFP dilakukan dengan merata-ratakan jumlah marka yang terdeteksi dan penggunaan titik pusat kamera untuk merepresentasikan titik pusat *quadrotor*. Proses sistem pemosisian tersebut dibagi menjadi beberapa langkah, di antaranya pendeteksian marka MFP, mengonversi kode unik setiap marka yang terdeteksi menjadi koordinat untuk posisi *quadrotor*, serta penentuan posisi *quadrotor* saat *quadrotor* mengalami pergerakan.

### Pendeteksian Marka ArUco

Pendeteksian MFP dilakukan dengan menggunakan OpenCV. OpenCV akan mendapatkan informasi dari setiap MFP yang terdeteksi oleh kamera *quadrotor* sesuai dengan ukuran MFP dan total MFP yang digunakan. Preses pendeteksian dibantu dengan fungsi **cv.aruco.detectMarkers**. Gambar 3.3 merupakan diagram alir dari pendeteksian MFP yang digunakan pada penelitian ini. Informasi yang diperoleh dari pendeteksian tersebut, yaitu koordinat kotak pembatas marker ‘**bboxs**’ dan identifikasi marka ‘**ids**’. Setalah informasi tersebut diperoleh, informasi tersebut akan ditampilkan pada layar pygame.

A black background with white arrows

Description automatically generated

Gambar 3.3. Diagram alir pendeteksian MFP ArUco.

### Konversi Marka ArUco Menjadi Titik Koordinat

Pada penelitian ini, jenis marka yang digunakan ialah marka MFP dengan ukuran 6x6 dan setiap marka memiliki kode unik. Selain itu, setiap marka akan mengandung informasi terkait titik koordinat sebagai informasi posisi *quadrotor*. Oleh karena itu, kode unik dari tiap marka akan dikonversi menjadi koordinat (x ; y) karpet mat dalam satuan sentimeter. Gambar 3.4 merupakan diagram alir terkait proses konversi kode unik marka menjadi titik koordinat pada karpet mat.



Gambar 3.4. Diagram alir konversi kode unik MFP menjadi koordinat pada karpet mat.

Berdasarkan Gambar 3.4, jumlah baris digunakan untuk merepresentasikan jumlah MFP yang sejajar sumbu x karpet mat dan jumlah kolom digunakan untuk merepresentasikan jumalah MFP yang sejajar pada sumbu y karpet mat. Dengan demikian, ukuran karpet mat yang digunakan adalah 25 x 20 dengan jarak antar MFP sebesar 19,5 cm. Selain itu, Gambar 3.4 menunjukkan bahwa MFP pada titik kootdinat (0 ; 0) merupakan MFP dengan kode unik 1. Penentuan titik koordinat untuk setiap MFP dihasilkan dengan proses perulangan sesuai jumlah kolom dan jumlah baris yang digunakan. ketika *quadrotor* berhasil mendeteksi MFP dan MFP tersebeut dikonversi menjadi titik koordinat pada karpet MAT, informasi tersebut akan dikirimkan pada komputer sebagai koordinat posisi. Akan tetapi, pada penelitian ini, MFP yang terdeteksi oleh kamera *quadrotor* berjumlah lebih dari satu, sehingga posisi *quadrotor* ditentukan berdasarkan jumlah MFP yang terdeteksi.

### Penentuan Posisi *Quadrotor*

Apabila setiap kode unik MFP telah dikonversi menjadi titik koordinat (x ; y), maka titik koordinat tersebut akan digunakan menjadi informasi posisi *quadrotor*. Lampiran A merupakan diagram alir terkait pendeteksian MFP menjadi informasi posisi *quadrotor* dengan MFP yang terdeteksi lebih dari 1. Pada penelitian, *quadrotor* akan terbang dan mnejalankan tugas pada posisi ketinggian 80 cm dari permukaan tanah, sehingga jumlah MFP yang terdeteksi oleh *quadrotor* sebanyak 4x2 atau 8 MFP, seperti Gambar 3.5. Titik koordinat (x ; y) yang diperoleh dari beberapa pendeteksian MFP dapat dituliskan kedalam persamaan matematika di bawah ini.

|  |  |
| --- | --- |
|  | (3.1) |
|  | (3.2) |

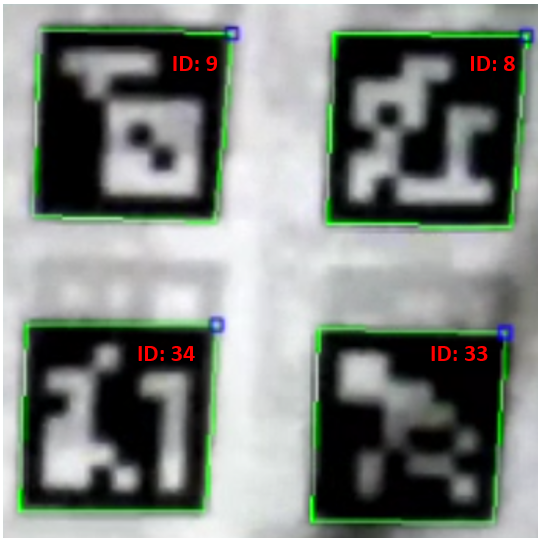
Dengan merupakan nilai rata-rata dari titik koordinat x, merupakan nilai rata-rata dari titik koordinat y, dan n merupakan jumlah marka yang terdeteksi oleh kamera *quadrotor*.

A close-up of a computer keyboard

Description automatically generated

Gambar 3.5. Hasil pendeteksian MFP pada ketinggian 80 cm.

Misalkan terdapat empat MFP yang terdeteksi, seperti pada Gambar 3.6. Saat MFP yang terdeteksi memiliki masing-masing kode unik 8, 9, 33, dan 34, maka masing-masing kode unik tersebut akan digantikan dengan koordinat (x,y), seperti (156 ; 0), (175,5 ; 0), (156 ; 19,5), (175,5 ; 19,5). Koordinat yang terdeteksi dihitung dengan menggunakan Persamaan (3.1) dan Persamaan (3.2) karena jumlah MFP yang terdeteksi lebih dari satu. Dengan demikian, posisi *quadrotor* akan menjadi (165,75 ; 9,75) dan titik hijau pada Gambar 3.6 merepresentasikan nilai rata-rata dari MFP yang terdeteksi.



ID: 33

ID: 34

ID: 8

ID: 9

Gambar 3.6. Hasil pendeteksian empat marka MFP pada *quadrotor.*

Namun, saat *quadrotor* mengalami pergerakan, terdapat marka yang tidak terdeteksi, seperti Gambar 3.8. Pada kondisi tersebut, *quadrotor* tidak dapat menggunakan nilai rata-rata dari koordinat marka yang terdeteksi. Dengan demikian, sistem pemosisian menggunakan deteksi beberapa MFP diintegrasikan dengan menggunakan interpolasi. Gambar 3.7 merupakan potongan dari diargam alir Lampiran A terkait pendeteksian beberapa MFP dengan kondisi *quadrotor* mengalami pergerakan.

A screenshot of a computer program

Description automatically generated

Gambar 3.7. Diagram alir pendeteksian beberapa MFP saat *quadrotor* mengalami pergerakan

Pada kondisi pendeteksian beberapa MFP sata *quadrotor* mengalami pergerakan, titik pusat kamera pada *quadrotor* akan digunakan sebagai titik pusat *quadrotor*. Untuk mengetahui posisi *quadrotor* saat kondisi Gambar 3.8, dilakukan interpolasi antara titik merah sebagai titik pusat *quadrotor* dengan koordinat titik tengah dari MFP yang terdeteksi.

A close-up of a grid

Description automatically generated

Gambar 3.8. Hasil pendeteksian dua marka MFP saat *quadrotor* mengalami pergerakan kearah kiri

Pada kondisi tersebut, penentuan posisi dari *quadrotor* akan ditentukan dengan menghitung posisi titik merah dengan menggunakan interpolasi penentuan titik (x ; y) pada bidang 2 dimensi. Jarak pixel antara titik hijau (titik tengah MFP yang terdeteksi) titik merah (titik pusat kamera) dihitung dengan menggunakan Metode *Eucledian Distance.* Jarak pixel kedua titik pada Gambar 3.8 dapat dituliskan sebagia berikut.

|  |  |
| --- | --- |
|  | (3.3) |

Dengan merupakakan titik koordinat dari titik hijau Gambar 3.8 dan Dengan merupakakan titik koordinat dari titik merah Gambar 3.8. Untuk mendapatkan nilai jarak dari kedua titik Persamaan (3.3) dalam satuan sentimeter, konversi satuan dari pixel ke cm perlu dilakukan. Konversi tersebut perlu dilakukan dengan mengetahui area yang tertangkap oleh kamera *quadrotor* pada ketinggian tertentu. Pada penelitin ini, tinggi *quadrotor* selama melaksanakan tugas dianggap konstan dengan ketinggian 80 cm. Sehingga, bingkai berukukan 640 x 360 pixel yang ditampilkan oleh *quadrotor* menangkap area sebesar 82 cm x 46 cm. Dengan ukuran yang telah dimiliki, maka dilakukan kalibrasi kamera sebagai penentuan faktor konversi Persamaan (3.3). Faktor konversi ( yang dipilih untuk mengonversi Persamaan (3.3) dapat dihitung sebagai berikut.

|  |  |
| --- | --- |
| dan | (3.4) |

|  |  |
| --- | --- |
| dan | (3.4) |

merupakan faktor koreksi untuk sumbu x dan merupakan faktor konversi sumbu y. Persamaan (3.4) menunjukkan faktor konversi kedua sumbu bernilai 0.128 cm/pixel. Maka dari itu, Persamaan (3.3) dapat dituliskan menjadi Persamaan (3.5).

|  |  |
| --- | --- |
|  | (3.5) |
| dan | (3.6) |
| dan | (3.7) |

Persamaan (3.6) dan Persamaan (3.7) digunakan untuk menentukan posisi dari *quadrotor* saat bergerak lurus ke arah sumbu x dan sumbu y. Apabila *quadrotor* bergerak serong kanan dan serong kiri, maka Persamaan (3.6) dan Persamaan (3.7) tidak dapat digunakan, sehingga koordinat titik merah dihitung dengan menggunakan jarak untuk masing-masing sumbu x dan sumbu y, seperti Persamaan (3.6).

|  |  |
| --- | --- |
| dan | (3.8) |

Persamaan (3.8) dikonversi dengan menggunakan faktor konversi Persamaan (3.4), sehingga akan diperoleh seperti Persamaan (3.5). Persamaan (3.6) dan Persamaan (3.7) akan digunakan ke dalam sistem pemosisi *quadrotor* dengan menggunakan beberapa pendeteksian MFP. Apabila *qaudrotor* bergerak menuju sumbu x poitif (dan sumbu x negatif ( maka titik pusat quadrotor yang tehitung dalam satuan cm dapat dituliskan, seperti Persamaan (3.9) dan Persamaan (3.10). Saat *quadrotor* bergerak menuju sumbu y poistif (dan sumbu y negatif (, maka maka titik pusat quadrotor yang tehitung dalam satuan cm dapat dituliskan, seperti Persamaan (3.11) dan Persamaan (3.12). Saat *quadrotor*  bergerak serong kanan dan serong kiri, maka titik pusat quadrotor yang tehitung dalam satuan cm dapat dituliskan, seperti Persamaan (3.13) dan (3.14).

|  |  |
| --- | --- |
| dan | (3.9) |
| dan | (3.10) |
| dan | (3.11) |
| dan | (3.12) |
| dan | (3.13) |
| dan | (3.14) |

Titik merupakan titik koordinat dari hasil rata-rata pendeteksian beberapa MFP. Sedangkan, titik merupakan titik pusat kamera yang diasumsikan sebagai titik pusat *quadrotor* dan diperoleh dengan Persamaa (3.9) sampai dengan Persamaan (3.14). Adapun dengan masing-masing memiliki arti sebagai jarak antar dua titik yang telah dikonversi ke dalam satuan sentimeter.

## Perancangan Algoritma Pemisahan pada *Multi-Quadrotor*

Pada penelitian ini, *multi-quadrotor* terdiri dari 2 *quadrotor* dan bergerak secara individu mengikuti jalur yang telah ditentukan oleh program untuk mencapai titik tujuan. Saat menyelesaikan tugas secara bersama-sama, kedua *quadrotor* diharapkan dapat menghindari tabrakan. Oleh karena itu, algoritma pemisahan pada *quadrotor* dirancang untuk menjaga jarak tiap *quadrotor*. Pada penelitian ini, radius zona pemisahan sebesar 100 cm atau jarak minimum antar pusat *quadrotor* sebesar 200 cm. Dengan demikian, algoritma pemisahan berfungsi untuk menjaga jarak antar *quadrotor* dengan menjauhi *quadrotor* satu sama lain apabila jarak antar *quadrotor* sama atau kurang dari 200 cm.

Akan tetapi, perancangan algoritma pemisahan dapat menyebabkan *quadrotor* keluar dari kelompok karena jaraknya yang terlalu besar. Oleh karena itu, penelitian ini memberikan algoritma tambahan, yaitu algoritma kohesi yang berfungsi untuk mendekatkan jarak *quadrotor* apabila jarak antar *quadrotor* sama atau lebih dari 400 cm, seperti pada penelitian di tahun 2022 [5]. Dengan demikian, jarak antar *quadrotor* harus lebih besar dari 200 cm dan lebih kecil dari 400 cm. Untuk memenuhi hal tersebut, diagram alir perancangan algoritma pemisahan untuk *multi-quadrotor* dilihat pada Gambar 3.9.

A black background with white squares and rectangles

Description automatically generated

Gambar 3.9. Diagram alur perancangan algoritma pemisahan.

## Perancangan Penelusuran Titik pada Tugas Pemetaan

Pada penelitian ini, perancangan penelusuran titik yang digunakan berbasis kontrol sentralisasi. Pada sistem yang akan dibuat, *multi-quadrotor* terdiri dari 2 *quadrotor* dan bergerak secara individu mengikuti jalur yang telah ditentukan oleh program untuk mencapai titik tujuan. Untuk merancang jalur yang akan dilalui oleh tiap *quadrotor* agar tehindar dari tabrakan, posisi dari setiap *quadrotor* perlu ditinjau. Selain itu, jarak antar *quadrotor* harus lebih besar dari 200 cm dan lebih kecil dari 400 cm sesuai algoritma yang dirancang untuk menghindari tabrakan. Gambar 3.10 merupakan diagram alir yang digunakan dalam merancang penelusuran titik pada tugas pemetaan.

A black background with white squares

Description automatically generated

Gambar 3.10. Diagram alir perancangan penelusuran titik pada tugas pemetaan.

Program yang dibuat untuk kedua *quadrotor* identik, tetapi setiap *quadrotor* memiliki trajektori masing-masing*.* Selain itu, apabila salah satu *quadrotor* telah sampai lebih dahulu dititik tertentu dari trajektori yang ditentukan dan akan melakukan pergeseran tempat ke arah sumbu y, *quadrotor* tersebut akan menunggu *quadrotor* lain hingga *quadrotor* lain bergeser. Dengan demikian, kedua quadrotor diharapkan dapat bergerak bersama ke arah sumbu x. Pada bagian ini, konflik yang diangkat untuk memecahkan masalah adalah konflik menyusul.

## Komunikasi pada *Quadrotor*

Pada penelitian ini, kontrol yang sentralisasi yang dirancang menyebabkan masing-masing *quadrotor* dianggap sebagai individu yang berbeda, sehingga tiap *quadrotor* memiliki program yang identik. Akan tetapi, hal yang membedakan dari masing-masing *quadrotor* adalah titik koordinat yang akan dicapai oleh tiap *quadrotor*, masing-masing *quadrotor* juga harus mengetahui posisi dari *quadrotor* lain untuk menghindari tabrakan pada pemetaan. Oleh karena itu, komunikasi antar *quadrotor* untuk berbagi informasi posisi dan komunikasi antara perangkat komputasi (program) dengan *quadrotor* untuk kontrol sentralisasi diperlukan pada penelitian ini. Gambar 3.11 merupakan arsitektur komunikasi yang digunakan pada penelitian.

A diagram of a computer system

Description automatically generated

Gambar 3.11. Arsitektur komunikasi data.

Pada Gambar 3.11, dapat dilihat bahwa komunikasi antara perangkat komputasi (program) dengan q*uadrotor* dalam mengirim perintah dan menerima data navigasi dan citra menggunakan protokol UDP. Sedangkan, komunikasi antara program (*ground station*) yang dijalankan berupa TC/IP.

## Pengujian

Pengujian dilakukan kapada *multi-quadrotor* yang terdiri dari dua *quadrotor*. Pengujian dilakukan saat *quadrotor* belum lepas landas hingga *quadrotor* mendarat. Lokasi pengujian dilakukan di Kampus ITB dalam ruangan yang tertutup karena pengujian dibatasi tanpa adanya gangguan eksternal seperti angin. Selain itu, pengujian dilakukan secara individu untuk masing-masing *quadrotor* dan secara berkelompok. Pengujian secara individu dilakukan untuk penelusuran posisi *quadrotor* dalam mengikuti titik referensi sebagai jalur yang harus dilewati oleh *quadrotor*. Pengujian ini dilakukan dengan tujuan melihat pergerakan *quadrotor* terkait perintah yang diberikan. Jalur pengujian yang harus dilalui oleh masing-masing *quadrotor* memiliki arah pergerakan maju, mundur, kanan, dan kiri yang diilustrasikan pada Gambar 3.12.

A grid of squares with red dots

Description automatically generated A grid with red dots and black squares

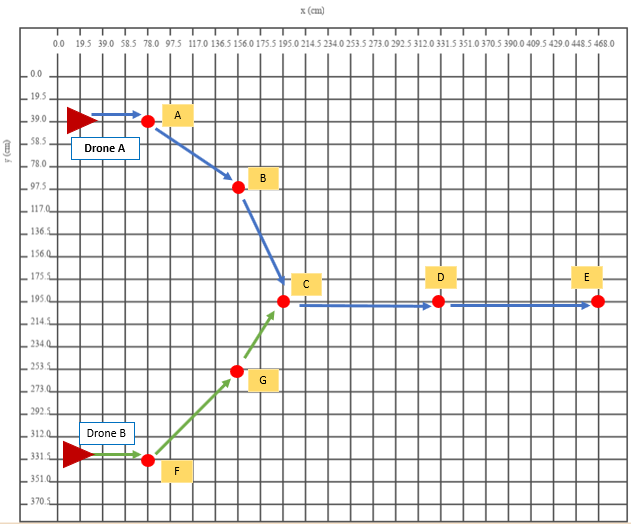
Description automatically generated

1. (b)

Gambar 3.12. Jalur pengujian individu (a) lurus terhadap sumbu x dan (b) lurus terhadap sumbu y.

Selain itu, pengujian juga dilakukan secara berkelompok. Pengujian berkelompok yang dilakukan pertama kali adalah pengujian algoritma pemisahan dengan skenario *quadrotor* A dan *quadrotor* B akan bertemu di titik tertentu. Pada kondisi tersebut, respon dari kedua *quadrotor* akan dilihat. Gambar 3.13 merupakan jalur pengujian yang akan dilakukan untuk uji coba algoritma pemisahan. Berdasarkan Gambar 3.13, *quadrotor* A akan bergerak mengikuti titik A – B – C – D – E dan *quadrotor* B akan bergerak mengikuti titik F – G – C. *Quadrotor* mulai bergerak pada waktu yang bersamaan dengan radius minimum antar *quadrotor* sebesar 200 cm. Berdasarkan skenario tersebut, ketika *quadotor* A dan *quadrotor* B bergergerak ke titik C, radius kedua *quadrotor* akan bernilai kurang dari 200 cm dan memungkinkan terjadinya tabrakan antar *quadrotor*. Oleh karena itu, pada bagian ini, algoritma pemisahan akan bekerja memisahkan kedua *quadrotor* hingga berjarak 200 cm satu sama lain.

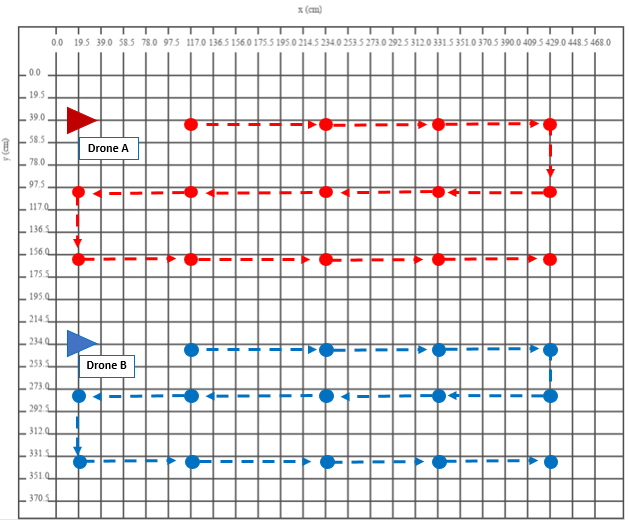
x (cm)



y (cm)

Gambar 3.13. Jalur pengujian algoritma pemisahan.

Pengujian berkelompok yang dilakukan selanjutnya tugas pemetaan dengan area 5m x 4m, seperti pada Gambar 3.14. Berdasarkan Gambar 3.14, masing-masing *quadrotor* akan memiliki area pemetaannya masing-masing berbentuk S. *Quadrotor* 1 akan bergerak mengikuti jalur berwarna merah dan *quadrotor* 2 akan bergerak mengikuti jalur berwarna biru. Jarus berbentu S pada tugas pemeteaan dipilih karena pola tersebut memungkinkan *quadrotor* untuk menutupi area yang lebih rapat dan teratur, mengurangi kemungkinan bagian yang terlewat sata melakukan pemetaan, serta menghindari tumpang tindih yang berlebih dalam menjalankan tugas pemetaan.



x (cm)

y (cm)

Gambar 3.14. Jalur pengujian secara berkelompok untuk tugas pemetaan.

Proses pengujian dilakukan hingga mencapai nilai RMSE yang dapat diterima. Perhitungan RMSE dilakukan dengan menghitung galat dari posisi *quadrotor* secara aktual dengan titik referensi yang diberikan. Perhitungan nilai RMSE untuk posisi *quadrotor* dapat situliskan ke dalam persamaan matematika berikut.

|  |  |
| --- | --- |
|  | (3.15) |
|  | (3.16) |
|  | (3.17) |
|  | (3.18) |

( merupakan posisi *quadrotor* secara aktual sesuai dengan kondisi nyata dan ( merupakan posisi titik refrensi yang diberikan kepada *quadrotor*.

BAB IV  
HASIL DAN ANALISIS AWAL

Pada bab ini akan dijelaskan terkait hasil dan analisis yang diperoleh dari penelitian yang telah dilakukan, di antaranya identifikasi sistem pemosisian *quadrotor* dengan menentukan kecepatan maksimum *quadrotor* untuk mendeteksi MFP, identifikasi kecepatan *quadrotor* untuk memperoleh parameter pengontrol, pengujian penelusuran posisi *quadrotor* dengan pergerakan ke arah yang berbeda, pengujian algoritma pemisahan, dan pengujian penelusuran titik berbasis kontrol sentralisasi untuk tugas pemetaan pada *quadrotor*.

# HASIL PENELITIAN

## Sistem Pemosisian

Proses pengambilan citra marka menggunakan kamera bawah *quadrotor* saat *quadrotor* berada pada kondisi terbangdan memberikan perintah maju. Pada penelitian sebelumnya, pembacaan MFP dapat terdeteksi hingga nilai masukan kecepatan sebesar 0,200, sehingga nilai kecepatan diatas 0,200 tidak dapat mendeteksi MFP [5]. Pada penelitian sebelumnya, ukuran marka yang dideteksi oleh *quadrotor* berukuran 60 cm x 60 cm, sementara pada penelitian yang dilakukan berukuran 12,5 cm x 12,5 cm. Oleh karena itu, *quadrotor* akan mengambil citra marka dengan kecepatan 0,100. Gambar 4.1 (a) menunjukkan marka yang terdeteksi oleh *quadrotor* pada kecepatan 0,100. Pada Gambar 4.1 (a), resolusi citra marka MFP yang ditangkap oleh kamera *quadrotor* saat *quadrotor* bergerak dengan kecepatan 0,100 tidak sebaik marka yang terdeteksi saat *quadrotor* dalam kondisi diam, seperti Gambar 4.1 (b). Meskipun resolusi citra marka yang terdeteksi rendah, *quadrotor* tetap mendapatkan infromasi ID dari marka tersebut. Pendeteksian beberapa marka dapat memberikan informasi posisi *quadrotor* yang presisi.

A blurry picture frame with green border

Description automatically generated A black square with green border

Description automatically generated

1. (b)

Gambar 4.1. Marka MFP yang terdeteksi oleh quadrotor (a) saat bergerak pada kecepatan

0,1000 dan (b) saat *quadrotor* diam.

## Identifikasi Karakteristik *Quadrotor*

Identifikasi karakteristik dilakukan pada kecepatan *quadrotor* untuk menentukan parameter pengontrol yang akan digunakan pada penelitian ini. Identifikasi karakteristik dilakukan pada masing-masing *quadrotor* dengan kondisi ruangan yang sama dan pergerakan arah yang sama.

### Hasil Identifikasi Karakteristik *Quadrotor*

Proses identifikasi pergerakan *quadrotor* dilakukan dengan cara memberikan perintah bergerak maju, mundur, kanan, dan kiri. Pada penelitian sebelumnya, identifikasi pergerakan *quadrotor* dilakukan dengan memberikan masukan arah dan nilai variasi kecepatan dari 0,100 hingga 1,00 dengan interval 0,100 dan nilai kecepatan yang dipilih untuk proses identifikasi sebesar 0,200 [5]. Pada penelitian ini dilakukan identifikasi pergerakan *quadrotor* pada nilai kecepatan 0,100. Nilai tersebut dipilih untuk memenuhi kebutuhan tugas *quadrotor* dalam menangkap citra pemetaan, sehingga kecepatan yang dimasukkan pada *quadrotor* harus sekecil mungkin. Respons kecepatan yang didapatkan untuk proses identifikasi memiliki bentuk serupa dengan FOPTD pada penelitian sebelumnya [5], sehingga identifikasi respons dapat dilakukan dengan pendekatan FOPTD. Gambar 4.2 merupakan respons sistem dengan nilai kecepatan 0,100 dari *quadrotor* A. Data lengkap terkait identifikasi kecepatan *quadrotor* A pada Lampiran B.

Gambar 4.2. Hasil identifikasi pergerakan ke kiri dari *quadrotor* A.

Persamaan fungsi alih FOPTD dari pergerakan tersebut dapat dituliskan, seperti Persamaan (2.29) dengan parameter dan yang relatif terhadap karakteristik kecepatan yang diidentifikasi. Parameter tersebut kemudian diuji performanya dengan menggunakan nilai SSE yang dituliskan pada Persamaan (3.1).

|  |  |
| --- | --- |
|  | (3.1) |

Parameter FOPTD untuk setiap arah pegerakan *quadrotor* A ditampilkan pada Tabel 4.1. Fungsi alih kecepatan dengan arah maju pada *quadrotor* A diekspresikan pada Persamaan (3.2). Untuk pergerakkan arah lain, parameter dan dapat disubstitusikan ke Persamaan (2.29).

Tabel 4.1. Hasil identifikasi kecepatan *quadrotor* A.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arah** | **Parameter FOPTD** | | | **SSE** |
|  |  |  |
| Maju | 2,998 | 3,543 | 0,974 | 0,112 |
| Mundur | 1,681 | 0,280 | 0,075 | 0,070 |
| Ke Kanan | 1,430 | 0,957 | 0,56 | 0,148 |
| Ke Kiri | 6,655 | 0,523 | 0,7421 | 1,579 |

|  |  |
| --- | --- |
|  | (3.2) |

Persamaan (3.2) diberikan input step untuk melihat respons fungsi transfer dalam simulasi. Gambar 4.3 merupakan grafik respons input step dari fungsi transfer Persamaan (3.2) dengan menggunakan perangkat lunak MATLAB. Berdasarkan Gambar 4.3 tersebut, sistem memiliki *undershoot* yang diakibatkan oleh komponen *zero* dari fungsi transfer. Komponen zero tersebut merupakan bagian dari waktu tunda, Gambar 4.2 pun menunjukkan adanya waktu tunda pada sistem selama 0,974 detik.

A graph with a blue line

Description automatically generated

Gambar 4.3. Respons sistem input step untuk *quadrotor* A arah maju

Proses identifikasi pergerakan pada *quadrotor* B identik dengan proses identifikasi yang dilakukan pada *quadrotor* A dan dilakukan di ruangan dengan kondisi yang sama. Tabel 4.2 merupakan parameter pergerakan *quadrotor* B. Hasil grafik identifikasi untuk *quadrotor* B dapat dilihat pada Lampiran B.

Tabel 4.2. Hasil identifikasi kecepatan *quadrotor* B.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arah** | **Parameter FOPTD** | | | **SSE** |
|  |  |  |
| Maju | 2,999 | 0,976 | 1,692 | 0,524 |
| Mundur | 3,541 | 0,858 | 0,564 | 1,289 |
| Ke Kanan | 1,182 | 1 | 0,425 | 0,342 |
| Ke Kiri | 2,353 | 0,723 | 0,555 | 0,152 |

### Hasil Penalaan Parameter Pengontrol PD

Fungsi alih kecepatan dalam bentuk FOPTD, seperti Persamaan (3.2) akan digunakan untuk desain pengontrol PD. Persamana tersebut memiliki parameter waktu tunda, yaitu yang perlu diubah dengan menggunakan pendekatan Padé orde pertama, sehingga fungsi FOPTD Persamaan (3.2) dapat disimulasikan pada sistem. Berikut merupakan penurunan fungsi FOPTD dengan menggunakan pendekatan Padé.

|  |  |
| --- | --- |
|  | (3.3) |

Dengan mensubstitusi Persamaan (3.3) ke dalam Persaman (2.29), bentuk fungsi FOPTD dapat diubah menjadi Persamaan (3.4). Parameter dan yang diperoleh dari Tabel 4.1 dan 4.2 dapat dsubstitusikan ke Persamaan (3.4). Apabila dan untuk Persamaan (3.2) disubstitusikan ke Persamaan (3.4), maka diperoleh fungsi FOPTD dalam bentuk pendekatan Padé, seperti Persamaan (3.5).

|  |  |
| --- | --- |
|  | (3.4) |
|  | (3.5) |

Dengan menggunakan bantuan perangkat lunak MATLAB dan fitur aplikasi PID *Tuner*, maka parameter pengontrol PD untuk masing-masing *quadrotor* dapat diperoleh, seperti pada Tabel 4.3. Hasil pengontrolan dari *quadrotor* A untuk bergerak maju ditunjukkan oleh Gambar 4.4. Berdasarkan Gambar 4.4, *overshoot* yang dimiliki oleh sistem dapat tereduksi sehingga keluaran tersebut memberikan hasil menuju satu seusai input yang diberikan.

Tabel 4.3. Parameter pengontrol PD untuk *quadrotor* A dan *quadrotor* B

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arah** | ***Quadrotor* A** | | ***Quadrotor* B** | |
|  |  |  |  |
| Maju | 0,487 | 0,7905 | 0,501 | 0,846 |
| Mundur | 0,11 | 0,038 | 0,217 | 0,282 |
| Kanan | 0,597 | 0,479 | 0,639 | 0,213 |
| Kiri | 0,106 | 0,371 | 0,302 | 0,362 |

A graph with a red line

Description automatically generated

Gambar 4.4. Grafik resfpon sistem *quadrotor* A arah maju dengan pengontrol PD.

## Pengujian Penelusuran Posisi *Quadrotor*

Proses penelusuran posisi *quadrotor* dilakukan secara individu dengan memberikan perintah kepada *quadrotor* untuk terbang mengikuti titik referensi yang diberikan dan *quadrotor* akan mendarat jika titik referensi terakhir telah dicapai. Pengujian penelusuran posisi dilakukan dengan menggunakan sistem pemosisian deteksi beberapa MFP yang telah dirancang sebelumnya pada Bab III. Titik referensi yang ditentukan membuat pergerakan *quadrotor* mengarah maju, mundur, ke kanan, dan ke kiri. Jumlah titik referensi yang diberikan kepada *quadrotor* bergantung dengan arah pergerakan *quadrotor* karena kondisi ruangan dan ukuran karpet mat yang terbatas.

Gambar 4.5 (a) merupakan hasil pengujian pergerakan maju pada *quadrotor* A. Betdasarkan gambar 4.5 (s), pergerakan arah maju pada *quadrotor* dilakukan dengan memberikan titik referensi sebanyak lima titik, yaitu (78 ; 156), (156 ; 156), (234 ; 156), (312 ; 156), dan (390 ; 156). Pada sata proses pengujian, quadrotor tidak langsung mendapatkan nilai referensi posisi pada sumbu x dan y, hal tersebut dikarenakan *quadrotor* memerlukan waktu untuk melakukan lepas landas dan siap bergerak di udara. Oleh karena itu, titik refrensi diberikan pada detik waktu ke 10, seperti Gambar 4.5. Selain itu, pada Gambar 4.5, posisi awal *quadrotor* tidak di titik x = 0, tetapi di titik x = 19,5. Hal tersebut dilakukan agar *quadrotor* terhindar dari tabrakan dengan dinging ruangan saat lepas landas.

Pada proses pengujian ini, *quadrotor* menghadap sejajar dengan sumbu x, sehingga pergerakan maju *quadrotor* ditandai dengan bertambahnya nilai koordinat pada sumbu x, seperti Gambar 4.5 (b). Proses pergantian posisi dilakukan sama seperti pada penelitian sebelumnya [5], bahwa pergantian titik refrensi tidak dilakukan dengan adanya jeda, melainkan titik refrensi akan berubah ketika posisi *quadrotor* berada dalam radius 30,000 cm dari titik target terkini. Untuk menyelesaikan penelusuran titik yang diberikan pada pergerakan maju, waktu yang dibutuhkan *quadrotor* A menjalankan tugas tersebut adalah 17,531 detik.

Gambar 4.5 (c) merupakan posisi *quadrotor* A pada sumbu y. Pada kondisi awal dari Gambar 4.5 (c), posisi quadrotor tidak berada di titik y = 156, hal ini dikarenakan saat lepas landas quadrotor mengalami pergerakan yang tidak stabil drai gaya dorong yang dihasilkan di awal. Akan tetapi, posisi tersebut tidak jauh dari nilai y = 156, sehingga posisi quadrotor di sumbu y masih dapat diterima. Jika dilihat drai Gambar 4.3 (b), posisi *quadrotor* di sumbu y hingga akhir penelusuran titik refrensi tidak mengalami pergeseran yang cukup besar dan posisi tersebut dapat diterima.

Selesai

Mulai

(a)

(b)

(b)

Gambar 4.5. Penelusuran posisi dengan arah maju pada sumbu x (a) rekaman posisi *quadrotor* saat bergerak maju, (b) profil posisi pada sumbu x dan (c) profil posisi sumbu y.

Apabila posisi sumbu y pada *Quadrotor* A Gambar 4.5 (c) dibandingkan dengan Gambar 4.6 (c) yang merupakan posisi *quadrotor* A pada sumbu y dari hasil penelitian Giga di tahun 2022 dengan nilai posisi y = 90, dapat dilihat bahwa posisi *quadrotor* A pada sumbu y Gambar 4.5 (b) lebih terjaga pada rentang nilai yang diberikan jika dibandingkan dengan posisi *quadrotor* pada Gambar 4.6 (b). Hal tersebut dikarenakan sistem pemosisian yang digunakan tidak lagi menggunakan kombinasi sistem pemosisian dan sistem pengontrol PD, melainkan menggunakan pendeteksian beberapa MFP dan penggunaan titik acuan yang merupakan titik tengah kamera sebagai titik pusat *quadrotor*. Proses penelusuran posisi pada pergerakan arah mundur diberikan titik refrensi yang merupakan kebalikan dari titik refrensi pergerakan maju *quadrotor*, yaitu (312 ; 156), (234 ; 156), (156 ; 156), (19,5 ; 156). Grafik pergerakan arah mundur *quadrotor* A dapat dilihat pada Lampiran C.

(a)

(b)

Gambar 4.6. Penelusuran posisi pada penelitian Giga dengan arah maju pada sumbu x (a) posisi pada sumbu x dan (b) posisi sumbu y [5].

Proses penelusuran posisi pada pergerakan arah kanan dan kiri dilakukan dengan menyimpan posisi *quadrotor* tetap menghadap sumbu x. Oleh karena itu, pergerakan *quadrotor* ke arah kanan dan kiri ditandai dengan berubahnya nilai korrdinat pada sumbu y. Titik referensi yang digunakan untuk pengujian arah kanan dan kiri lebih sedikit dibandingkan pengujian arah maju. Hal tersebut dikarenakan ukuran koordinat sumbu y yang lebih kecil dibandingkan koordinat x pada karpet mat yang dirancang, seperti pada Gambar 3.9. Selain itu, jumlah titik referensi dikurangi untuk menghindari tabrakan dengan dinding dan mengurangi efek turbulensi akibat baling-baling. Pergerakan arah kanan dan kiri pada *quadrotor* A dapat dilihat pada Lampiran C. Dengan demikian, waktu tempuh *quadrotor* A untuk melakukan pergerakan ke arah kanan sebesar 15,084 detik dan waktu tempuh pergerakan ke arah kiri sebesar 14,509 detik.

Pengujian penelusuran posisi dilakukan juga pada *quadrotor* B dengan langkah-langkah yang sama, seperti pada *quadrotor* A. Perbandingan hasil pengujian untuk kedua *quadrotor* ditunjukkan pada Tabel 4.4. Berdasarkan hasil tersebut, waktu tempuh dari tiap *quadrotor* berbeda dikarenakan karakteristik *quadrotor* yang berbeda. Pergerakan *quadrotor* ke arah kanan dan kiri memiliki jarak tempuh yang lebih dekat dibadingkan dengan arah maju dan mundur, sehingga memberikan waktu tempuh yang lebih cepat juga. Akan tetapi, pada *quadrotor* B, waktu tempuh ke arah kanan lebih lambat dibandingkan ke arah kiri dikarenakan kecepatan *quadrotor* saat bergerak ke arah kanan lebih kecil dibandingkan bergerak mundur.

Tabel 4.4. Waktu tempuh pengujian penelusuran posisi tiap *quadrotor* (detik).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Arah Pergerakan *Quadrotor* (detik)** | | | |
| **Maju** | **Mundur** | **Kanan** | **Kiri** |
| ***Quadrotor* A** | 17,531 | 17,060 | 15,084 | 14,509 |
| ***Quadrotor* B** | 23,026 | 19,269 | 20,062 | 18,863 |

## Pengujian Algoritam Pemisahan untuk Menghindari Tabrakan

Proses pengujian algoritma pemisahan dilakukan dengan memberikan jalur yang bertabrakan untuk tiap *quadrotor*. Pada pengujian ini, jarak minimum antar *quadrotor* sebesar 200 cm. Jarak minimum tersebut dihitung dari titik pusat *quadrotor* A ke *quadrotor* B. Dengan demikian, ketika kedua *quadrotor* berada pada jarak kurang dari 200 cm, maka kedua *quadrotor* akan menghindar menjauhi satu sama lain. Akan tetapi, jarak maksimum antar *quadrotor* juga diatur, sehingga *quadrotor* tidak keluar dari kelompok. Jarak maksimum yang diatur sebesar 400 cm. Gambar 4.7 merupakan hasil dari pengujian algoritma pemisahan. Ketika kedua *quadrotor* berada pada jarak 200 cm, *quadrotor* akan memberikan arah pergerakan untuk menghindar. Pada proses pengujian, *quadrotor* A akan tetap bergerak menuju titik C dan *quadrotor* B akan bergerak berlawanan arah (mundur) ke titik G. Resposn pergerakkan tersebut dapat dilihat pada Gambar 4.7 , *quadrotor* B berada pada posisi (92,9 ; 275,5) saat t = 3,789 detik dan kembali ke posisi (68,5 ; 300) saat t = 5,550 detik. Respons pergerakkan yang diberikan oleh *quadrotor* B menyebabkan jarak antar *quadrotor* terjaga dengan nilai diatas 200 cm. Selain itu, *quadrotor* B akan bergerak maju menuju titik C apabila *quadrotor* A telah melewati titik C dan jarak antar *quadrotor* lebih dari 200 cm.

A graph of a graph

Description automatically generated with medium confidence

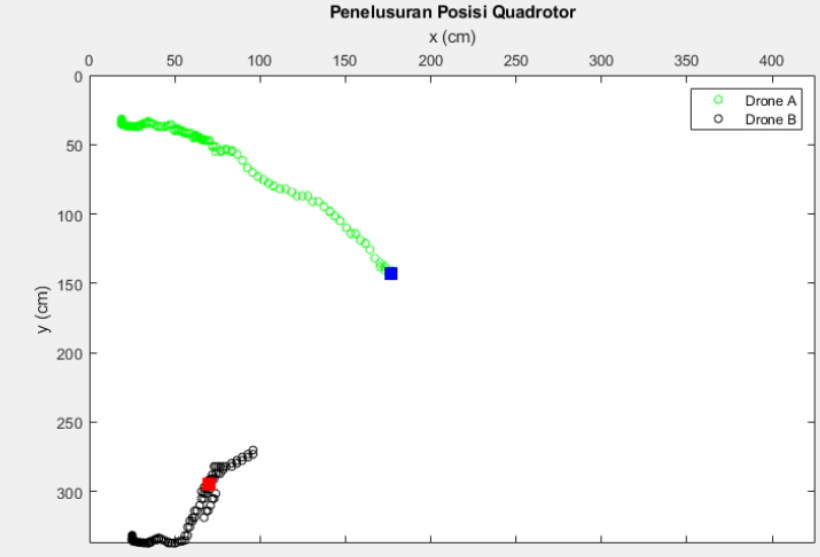
202,999 cm

300,065 cm

*Drone* B

*Drone* A

(a)



*Drone* B

220,325 cm

*Drone* A

(b)

A graph of a green line

Description automatically generated

200,955 cm

*Drone* B

*Drone* A

(c)

Gambar 4.7. Rekaman posisi *quadrotor* pada pengujian algoritma pemisahan (a) posisi *quadrotor* pada t = 3,789 detik, (b) posisi *quadrotor* pada t = 5,550 detik, dan (c) posisi *quadrotor* pada t = 11,000 detik.

## Pengujian untuk Tugas Pemetaan

Kontrol formasi yang diterapkan pada penelitian ini adalah kontrol sentralisasi. Dengan demikian, setiap *quadrotor* diasumsikan sebagai individu yang berbeda dan tidak saling mempengaruhi apabila terjadi kegagalan pada individu lainnya. Pengujian untuk *multi-quadrotor* dilakukan dengan memberikan tugas kepada kedua *quadrotor* untuk melakukan penelusuran jalur dalam tugas pemetaan, seperti Gambar 3.12.

Gambar 4.8 merupakan data yang diambil hingga detik 20,000 saat kedua *quadrotor* mengalami pergeseran pada sumbu y. Data tersebut diambil untuk melihat skenario pergerakan kedua quadrotor agar jarak keduanya lebih dari jarak minimum (2000 cm). Gambar 4.8 menunjukan bahwa *quadrotor* B sampai lebih dahulu pada titik refrensi (429 ; 234) pada detik 6,330 dan *quadrotor* A belum sampai di titik (429 ; 39). Oleh karena itu, quadrotor B akan menunggu dengan melayang di titik (429 ; 234) hingga *quadrotor* A sampai di titik (429 ; 39). Berdasarkan gambar tersebut, quadrotor A mencapai titik (429 ; 39) pada detik 10,250. Dengan demikian, skenario pergerakan *quadrotor* pada saat melakukan tugas pemetaan pada *multi-quadrotor* untuk menghindari tabrakan dan menjaga jarak antar quadrotor pada nilai 200 cm hingga 400 cm dapat digunakan pada penelitian ini. Ketika *quadrotor* A dan *quadrotor* B sudah mengalami pergerakan ke sumbu y dengan titik refrensi masing-masing (429 ; 97,5) dan (429 ; 273), maka kedua *quadrotor* akan bergerak bersama-sama pada arah sumbu x.

A graph with numbers and circles

Description automatically generated with medium confidence

Gambar 4.8. Penelusuran posisi *quadrotor* A dan *quadrotor* B selama 20 detik.

Kedua quadrotor akan melakukan tugas pemetaan dengan mengikuti titik referensi yang diberikan sebagai jalur pemetaa. Gambar 4.9 merupakan hasil penelusuran posisi *multi-quadrotor* saat melakukan tugas pemetaan. Gambar 4.9 menunjukkan posisi *quadrotor* tidak tepat berada di posisi awal, yaitu (19,5 ; 39) untuk *quadrotor* A dan (19,5 ; 234) untuk *quadrotor* B. Hal tersebut disebabkan oleh turbulensi baling baling dan gaya dorong dari *quadrotor* saat melakukan lepas landas. Namun, posisi *quadrotor* masih dalam toleransi karena tidak menyebabkan pergeseran yang besar dan jarak antar *quadrotor* tetap terjaga. Berdasarkan Gambar 4.9, kedua *quadrotor* mengikuti jalur sesuai titik refrensi yang diberikan, meskipun *quadrotor* B mengalami pergeseran dari titik referensi yang diberikan, yaitu pada posisi (281 ; 302). Pergeseran yang dialami oleh *quadrotor* B diakibatkan adanya turbulensi dari baling-baling setiap *quadrotor* serta jalur yang diberikan sebagai referensi *quadrotor* melakukan pemetaan berbentuk titik. Penggunaan jalur refrensi titik menyebabkan *quadrotor* bergerak tidak mengejar lurus mengikuti jalur refrensi yang ditentukan. Akan tetapi, pergeseran tersebut masih dalam toleransi karena pergeseran posisi *quadrotor* B di koordinat sumbu y tidak menyebabkan jarak antar *quadrotor* kurang dari jarak minimum (200 cm).

A graph with red and blue dots

Description automatically generated

(a)

(b)

Gambar 4.9. Hasil pengujian tugas pemetaan (a) posisi *quadrotor* pada koordinat *cartesius*

dan (b) jarak tiap *quadrotor* relatif terhadap waktu.

Tugas pemetaan akan selesai jika kedua *quadrotor* menemukan titik refrensi terakhir. Apabila hanya satu *quadrotor* saja yang baru sampai di titik refersensi terakhir, *quadrotor* tersebut akan berada pada kondisi melayang menunggu *quadrotor* lain mencapai titik referensi terakhir juga. Pada penelitian ini, *quadrotor* A berhasil menemukan titik referensi terakhir, yaitu titik (429,156) pada detik 49,508. Sedangkan, *quadrotor* B berhasil menemukan titik referensi terakhir, yaitu titik (429,351) pada detik 51,317. Dengan demikian, *quadrotor* akan berada pada kondisi melayang mulai dari detik 49,508 hingga detik 51,317 sehingga kedua *quadrotor* akan mendarat pada detik 51,317 sesuai dengan algoritma yang dirancang. Selain itu, Gambar 4.8 (b) menunjukkan bahwa jarak antar *quadrotor* berada pada nilai rata-rata 232,2 cm.

Untuk mengetahui performa antar *quadrotor* dalam melakukan tugas pemetaan sesuai dengan jalur yang ditentukan, maka perhitunagn perfoma tersebut menggunakan RMSE, seperti penelitian sebelumnya. Perhitungan RMSE dilakukan dengan menghitung galat antara posisi *quadrotor* secara aktual dengan titik refrensi yang diberikan. Dengan demikian, hasil yang didapat oleh *quadrotor* A adalah 12,3 cm dan *quadrotor* B adalah 22,5 cm. Selain itu, jarak tempuh tiap *quadotor* untuk melakukan tugas pemetaan jika dihitung mulai berdasarkan posisi awal hingga akhir dapat dilihat pada Tabel 4.5. Dengan demikian, pengujian untuk tugas untuk kedua *quadrotor* pada penelitian ini memberikan performa yang baik dengan luas area tugas yang terbatas, yaitu sebesar 420 cm x 350 cm.

Tabel 4.5. Data hasil tugas pemetaan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Jarak Tempuh** | **Waktu Tempuh** | **Errord** | **RMSE** |
| ***Quadrotor* A** | 1405,5 cm | 49,508 detik | 151,4 cm2 | 12,3 cm |
| ***Quadrotor* B** | 1405,5 cm | 51,317 detik | 507,4 cm2 | 22,5 cm |

BAB V  
PENUTUP

Bab ini membahas mengenai simpulan terkait hasil yang diperoleh dari penelitian sesuai dengan batasan dan asumsi masalah. Selain itu, bab ini juga membahas mengenai saran terhadap masalah-masalah yang belum diselesaikan pada penelitian sebagai pengembangan dan perbaikan terkait penelitian ini.

# HASIL PENELITIAN

## KESIMPULAN

Berdasarkan hasil dan analisis pada Bab IV, simpulan penelitian yang diperoleh adalah sebagai berikut.

* + - 1. *Quadrotor* dapat melakukan penelusuran posisi dengan menggunakan pendeteksian beberapa MFP dan interpolasi titik pada ketinggian terbang 80 cm. Dari implementasi yang dilakukan, *quadrotor* mampu memperoleh informasi posisi yang presisi meskipun resolusi marka yang terdeteksi rendah saat mengalami pergerakan. Hal tersebut diakibatkan oleh pendeteksian beberapa marka dalam satu bingkai. Selain itu, *quadrotor* mampu bergerak mengikuti titik refrensi yang diberikan dengan waktu tempuh yang berbeda-beda sesuai dengan karakteristik masing-masing. Pada penelitian ini, saat melakukan pergerakan maju, *quadrotor* A memiliki nilai galat sebesar 0,4 cm pada koordinat sumbu y dengan posisi aktual rata-rata berada pada koordinat y = 156,4 dan posisi refrensi y = 156.
      2. *Quadrotor* dapat menghindari tabrakan terhadap *quadrotor* lainnya dengan menggunakan algoritma pemisahan. Ketika jarak kedua *quadrotor* berada pada jarak minimum, yaitu kurang dari atau sama dengan 200 cm, maka algoritma pemisahan akan berjalan. Kedua *quadrotor* akan menghitung vector pergerakan dan menyebabkan *quadrotor* bergerak saling menjauhi hingga mencapai radius yang diperbolehkan. Pada penelitian ini, *quadrotor* akan kembali mengejar titik refrensi ketika jarak kedua *quadrotor* bernilai lebih dari 200 cm.
      3. Dari implementasi yang dilakukan untuk penerapan Solusi MPAF berbasis kontrol sentralisasi pada tugas pemetaan, performa *quadrotor* dihitung dengan menggunakan nilai RMSE. Pada penelitian, tugas pemetaan dilakukan dengan jarak tempuh sebesar 1405,5 cm dan waktu tempuh 51,317 detik dihitung dari posisi awal hingga posisi mendarat. Nilai RMSE yang diperoleh untuk masing-masing *quadrotor* sebesar 12,3 cm untuk *quadrotor* A dan 22,5 cm untuk *quadrotor* B. Berdasarkan hasil pengujian, *quadrotor* mampu melakukan tugas pemetaan dengan baik, meskipun *quadrotor* bekerja pada area yang terbatas, yaitu 5 m x 4 m.

## SARAN

Berdasarkan penelitian yang telah dilakukan serta simpulan yang telah diperoleh, maka terdapat beberapa hal yang disarankan sebagai arah pengembangan penelitian selanjutnya, yaitu:

1. Melakukan tambahan pengontrol agar sistem dapat digunakan dengan baik Ketika *quadrotor* mendapat gangguan eksternal akibat turbulensi dari baling – baling *quadrotor* lain, sehingga jarak antar *quadrotor* dapat diperkecil dan hasil pemetaan lebih rapat.
2. Mempertimbangkan sudut hadap *quadrotor* yang sejajar dengan arah pergerakan *quadrotor* sehingga proses pengolahan citra marka menjadi posisi lebih presisi.
3. Mempertimbangkan jarak antar *quadrotor* dalam menetukan luas area pemetaan yang akan digunakan pada penelitian, sehingga efek turbulensi antar *quadrotor* dapat dihindari.
4. Melakuakn pencarian rute untuk tugas pemetaan agar rute yang dilalui oleh *quadrotor* merupakan rute yang optimal.
5. Rute yang diberikan kepada *quadrotor* merupakan persamaan garis, sehingga pergerakkan *quadrotor* tidak akan terhenti dan *quadrotor* mengikuti lintasan yang diberika, sehingga galat posisi yang diberikan merupakan galat terdetak dengan lintasan.

DAFTAR PUSTAKA

[1] Tahir, A., Böling, J., Haghbayan, M.H., Toivonen, H.T., dan Plosila, J. (2019): Swarms of unmanned aerial vehicles – a survey, *Journal of Industrial Information Integration*, **16**, 1-7. https://doi.org/10.1016/j.jii.2019.100106.

[2] Faessler, M., Franchi, A., & Scaramuzza, D. (2018): Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories. *IEEE Robotics and Automation Letters*, 3(2), 620-626.

[3] Lin, Y., & Saripalli, S. (2017). "Path Planning Using 3D Dubins Curve for Unmanned Aerial Vehicles". 2017 *International Conference on Unmanned Aircraft Systems* (ICUAS), 322-329.

[4] Yu, J., Yan, C., dan Huang, M. (2019): Research of consistency problem for quadrotor UAV system with leader-follower, *Proceedings - 2019 Chinese Automation Congress*, CAC 2019, 616–621. https://doi.org/10.1109/CAC48633.2019.8996473

[5] Tnunay, I.H.P., Abdurrohman, M.Q., Rizqi, A.A.A., Faris, M., dan Sari, D.R.P. (2014): Perancangan sistem koordinasi dan kendali formasi uav quadrotor untuk optimalisasi mitigasi bencana, *Pekan Ilmiah Mahasiswa Nasional Program Kreativitas Mahasiswa*, 1-9.

[6] Pratama, G.V. (2021): Kontrol formasi quadrotor menggunakan metode cyclic leader follower SFM sebagai media pelacakan lokasi, Tesis Program Magister, Institut Teknologi Bandung.

[7] Reynolds, C. W. (1987). “Flocks Herds, and Schools A Distributed Behavioral Model.” *Computer Graphivs,* 21(4): 25 – 34.

[8] Mukhlish, F. (2012): Perancangan Pengontrol Proporsiional Derivatif yang Optimal dan Robus Untuk PenjejakanLintasan Pada Quadrotor, Tesis Program Magister, Institut Teknologi Bandung.

[9] Kucherov, D., Kozub, A., Sushchenko, O., dan Skrynkovskyy, R. (2021): Stabilizing the spatial position of a quadrotor by the backstepping procedure, *Indonesian Journal of Electrical Engineering and Computer Science*, **23(2)**, 1188–1199.

[10] Rafifandi, R., Asri, D. L., Ekawati, E., dan Budi, E. M. (2019): Leader–follower formation control of two quadrotor UAVs, *SN Applied Sciences*, 1(6), 1–12.

[11] Ogata, K. *Modern Control Engineering*. Prentice Hall, 2010.

[12] Y. Y. Nazaruddin. *Sistem Kontrol Otomatik: Analisis dan Desain*, I. Bandung: ITB Press, 2022.

[13] B. Li, J. Wu, X. Tan and B. Wang, "ArUco Marker Detection under Occlusion Using Convolutional Neural Network," 2020 5th *International Conference on Automation, Control and Robotics Engineering (CACRE)*, Dalian, China, 2020, pp. 706-711, doi: 10.1109/CACRE50138.2020.9230250.

[14] Mondéjar-Guerra, V., Garrido-Jurado, S., Muñoz-Salinas, R., Marín-Jiménez, M. J., dan Medina-Carnicer, R. (2018): Robust identification of fiducial markers in challenging conditions, *Expert Systems with Applications*, **93**, 336–345. <https://doi.org/https://doi.org/10.1016/j.eswa.2017.10.032>.

[15] *ArUco Marker Detection Using* Open CV, data deiperoleh melalui situs internet: https://www.delftstack.com/howto/python/opencv-aruco-markers/#google\_vignette. Diunduh pada tanggal 15 Desember 2023.

[16] *Detection od ArUco Markers*, data diperoleh melalui situs internet: <https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html>. Diunduh pada tanggal 15 Desember 2023.

[17] Hansun, Seng. (2013): A New Approach of Moving Average Method in Time Series Analysisi, *IEEE*, pp, 1 – 4, doi: 10.1109/CoNMedia.2013.6708545.

[18] Budin, S. dan Riadi I. (2019): Traffic Shaping Menggunakan Metode HTB (Hierarchical Token Bucket) pada Jaringan Nirkabel, *Buletin Ilmiah Sarjana Teknik Elektro*, **1(3),** 144-152. https://doi.org/10.12928/biste.v1i3.1100

[19] Nazir, R., laghari, A. A., Kumar, K., David, S., dan Ali, M. (2021): Survey on Wireless Network Security, *Archives of Computational Methods in Engineering*. https://doi.org/10.1007/s11831-021-09631-5

[20] Forouzan, B. A. (2005): TCP/IP Protocol Suite (3 ed.), McGraw-Hill, Inc., USA.

[21] Kessler, G. C. (2010): An Overview of TCP/IP Protocols and the Internet, *InterNIC*.

[22] AL-Dhief, F. T., Sabri, N., Latiff, N. M. A., Malik, N., Abbas, M., Albader, A., Mohammed, M. A., AL-Haddad, R. N., Salman, Y. D., dan Khanapi, M. (2018): Performance comparison between TCP and UDP protocols in different simulation scenarios, *International Journal of Engineering & Technology*, **7(4.36)**, 172–176.

[23] Ghani, A. (2023): *Pendekatan Kontrol Desentralisasi untuk Pencarian Jalur Multi Agen Pada Robot Beroda OMNI*, Skripsi Program Sarjana, Institut Teknologi Bandung, 22 – 33.

[24] Ster, R., dkk. (2019): Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks, *Twelfth Annual Symposium on Combinational Search*, **10**, 1, https://doi.org/10.1609/socs.v10i1.18510 .

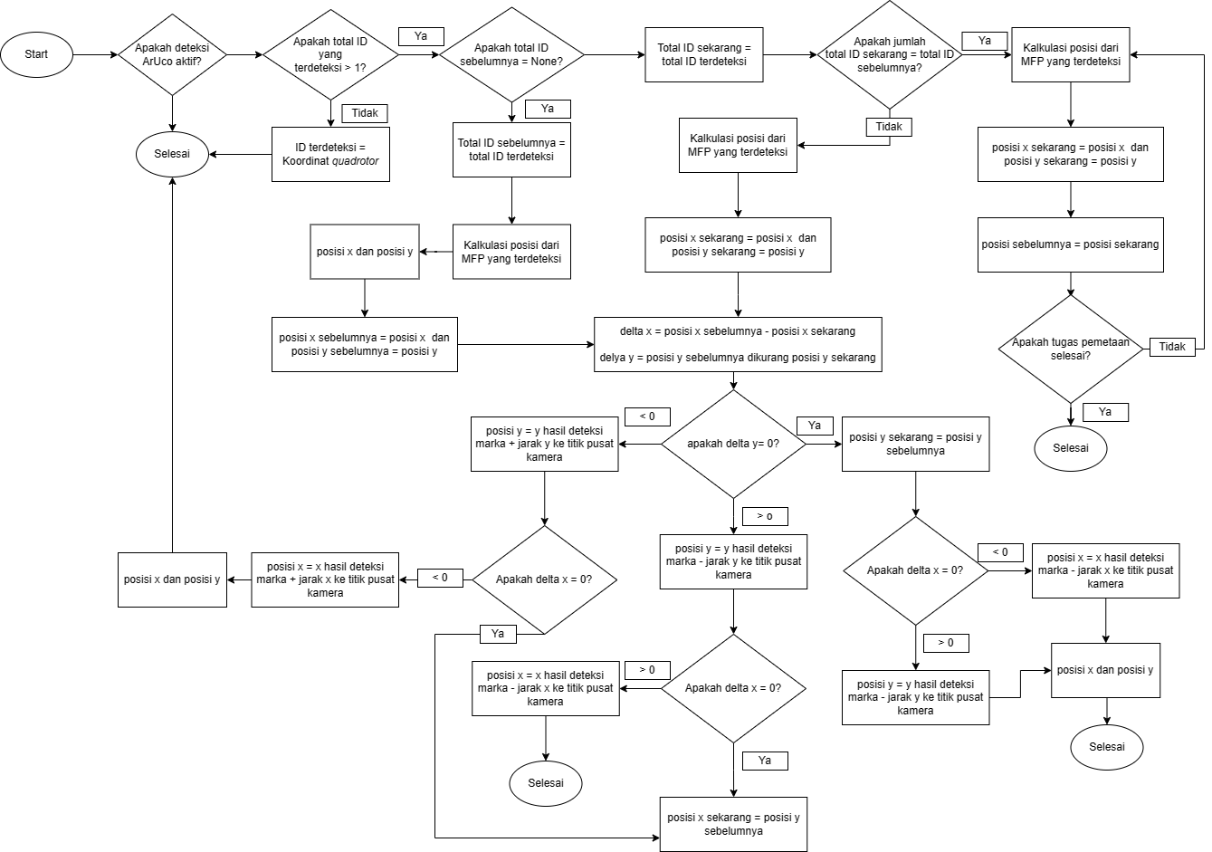
[25] Maruyama, N., Saito, D., Hashimoto, Y. et al. Dynamic organization of flocking behaviors in a large-scale boids model. *J Comput Soc Sc* **2**, 77–84 (2019). https://doi.org/10.1007/s42001-019-00037-9

[26] Piskorski, S., Brulez, N., Eline, P., dan D’Haeyer, F. (2012): Ar. Drone develepor guide, *Parrot, sdk,* **1**.

LAMPIRAN A. DIAGRAM ALIR PENENTUAN POSISI

***QUADROTOR***

Diagram alir untuk pennetuan posisi *quadrotor* ditunjukan pada gambar berikut. Berdasarkan gambar tersebut terdapat beberapa kondisi yang dirancang, di antaranya ketika *quadrotor* hanya mendeteksi satu MFP, ketika *quadrotor* baru mendeteksi MFP, serta ketika *quadrotor* mendeteksi MFP kesekian kalinya.

z

LAMPIRAN B. HASIL IDENTIFIKASI KARAKTERISTIK

***QUADROTOR***

Identifikasi karakteristik dilakukan pada *quadrotor* A dan *quadrotor* B dengan kondisi ruangan dan arah pergerakan yang sama. Proses identikiasi dilakukan untuk merancang pengontrol yang dakan digunakna pada quadrotor. Berikut merupakan hasil identifikasi karakteristik *quadrotor* A dan *quadrotor* B. Data keseluruhan dapat diakes melalui bit.ly/DataKarakteristikQuadrotor.

B.1 Grafik Karakteristik Kecepatan *Quadrotor* A

Berdasarkan data yang diperoleh dari identifikasi kecepatan *quadrotor* akan didapatkan parameter fungsi transfer orde satu atau FOPTD. Dengan menggunakan fungsi transfer tersebut, dapat ditentukan karakteristik dan pengontrol yang dapat digunakan pada penelitian ini. Berikut grafik karakteristik untuk *quadrotor* A.

|  |  |
| --- | --- |
| **Arah Pergerakan** | **Grafik Karakteristik** |
| Maju |  |
| Mundur |  |
| Kanan |  |
| Kiri |  |

B.2 Grafik Karakteristik Kecepatan *Quadrotor* B

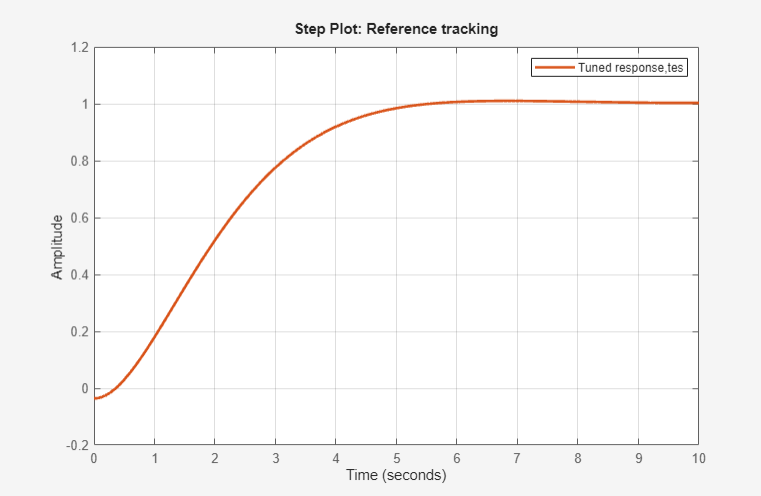
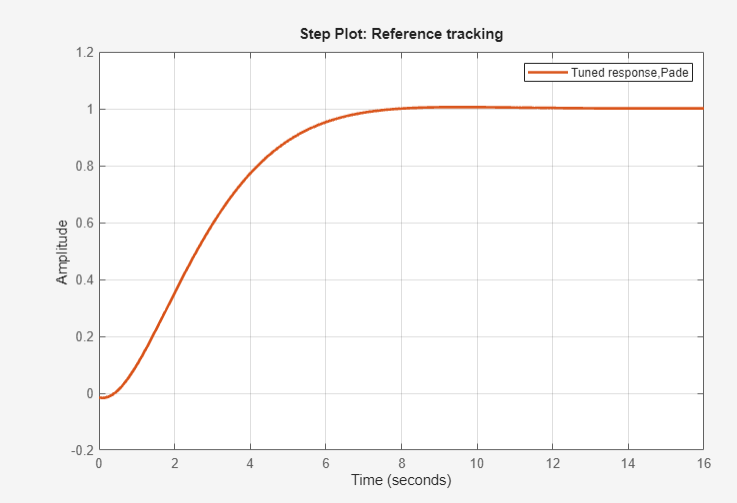
Berdasarkan data yang diperoleh dari identifikasi kecepatan *quadrotor* akan didapatkan parameter fungsi transfer orde satu atau FOPTD. Dengan menggunakan fungsi transfer tersebut, dapat ditentukan karakteristik dan pengontrol yang dapat digunakan pada penelitian ini. Berikut grafik karakteristik untuk *quadrotor* B.

|  |  |
| --- | --- |
| **Arah Pergerakan** | **Grafik Karakteristik** |
| Maju |  |
| Mundur |  |
| Kanan |  |
| Kiri |  |

B.3 Grafik Pengontrol PD untuk *Quadrotor* A

Berdasarkan data yang diperoleh dari identifikasi kecepatan *quadrotor* akan dan parameter fungsi transfer, akan dilakukan perancangan pengontrol PD untuk masing-masing *quadrotor*. Berikut merupakan diagram sistem setelah menggunakan pengontrol PD pada *quadrotor* A.

| **Arah** | **Respons Sistem Awal** | **Respons Sistem Menggunakan Pengontrol PD** |
| --- | --- | --- |
| Maju |  |  |
| Mundur |  |  |
| Kiri |  |  |
| Kanan |  |  |

B.4 Grafik Pengontrol PD untuk *Quadrotor* B

Berdasarkan data yang diperoleh dari identifikasi kecepatan *quadrotor* akan dan parameter fungsi transfer, akan dilakukan perancangan pengontrol PD untuk masing-masing *quadrotor*. Berikut merupakan diagram sistem setelah menggunakan pengontrol PD pada *quadrotor* B.

| **Arah** | **Respons Sistem Awal** | **Respons Sistem Menggunakan Pengontrol PD** |
| --- | --- | --- |
| Maju |  |  |
| Mundur |  |  |
| Kanan |  |  |
| Kiri |  |  |

LAMPIRAN C. PENELUSURAN POSISI *QUADROTOR*

Penelusuran posisi *quadrotor* dilakukan dengan memberikan titik referensi dengan pergerakan ke arah maju, kanan, dan kiri. Penelusuran posisi dilakukan untuk mengetahui pergerakan *quadrotor* sesuai jalur yang diberikan. Pengujian ini dilakukan pada tiap *quadroto* dan dilihat dari respons pergerakan *quadrotor*. Di bawah ini merupakan hasil penelusuran posisi untuk *quadrotor* A dan *quadrotor* B. Data keseluruhan dapat diakes melalui bit.ly/PenelusuranPosisiQuadrotor.

C.1 Penelusuran Posisi *Quadrotor* A

Penelusuran posisi pada *quadrotor* A dilakukan pada koordinat *cartesius* dengan menempatkan *quadrotor* sejajar dengan sumbu x. Selain itu, *quadrotor* akan bergerak maju ke arah sumbu x positif, sehingga ketika *quadrotor* maju, maka nilai koordinat x akan bertambah. Pergerakan kanan dan kiri akan dilakukan ke arah sumbu y. Berikut merupakan penelusuran posisi untuk *quadrotor* A.

1. Pergerakan Arah Maju
2. Pergerakan Arah Mundur
3. Pergerakan Arah Kanan
4. Pergerakan Arah Kiri

C.2 Penelusuran Posisi *Quadrotor* B

Penelusuran posisi pada *quadrotor* B dilakukan pada koordinat *cartesius* dengan menempatkan *quadrotor* sejajar dengan sumbu x. Selain itu, *quadrotor* akan bergerak maju ke arah sumbu x positif, sehingga ketika *quadrotor* maju, maka nilai koordinat x akan bertambah. Pergerakan kanan dan kiri akan dilakukan ke arah sumbu y. Berikut merupakan penelusuran posisi untuk *quadrotor* B.

Pergerakan Arah Maju

Pergerakan Arah Mundur

Pergerakan Arah Kanan

Pergerakan Arah Kiri