

**PERANCANGAN SISTEM KONTROL PERGERAKAN
UNTUK MENGATASI PENGARUH GANGGUAN UDARA
DALAM FORMASI *MULTIQUADROTOR***

LAPORAN KEMAJUAN TUGAS AKHIR



Oleh

Yosep Putra Setiyanto

NIM: 13320026

**PROGRAM STUDI TEKNIK FISIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI BANDUNG
2023**

ABSTRAK

PERANCANGAN SISTEM KONTROL PERGERAKAN UNTUK MENGATASI PENGARUH GANGGUAN UDARA DALAM FORMASI *MULTIQUADROTOR*

Oleh

Yosep Putra Setiyanto

NIM: 13320026

(Program Studi Teknik Fisika)

Dalam era perkembangan teknologi, penggunaan *quadrotor* telah menjadi semakin relevan dalam berbagai bidang, termasuk militer, sipil, teknologi, dan ilmu pengetahuan. Namun, penggunaan *quadrotor* tunggal untuk menyelesaikan tugas sehari-hari masih memiliki keterbatasan, seperti efisiensi yang rendah, kesulitan dalam mengeksekusi tugas kompleks, dan batasan dalam cakupan area.

Untuk mengatasi keterbatasan ini, pengembangan *swarm quadrotor* atau kelompok *quadrotor* yang bekerja bersama sebagai satu kesatuan telah menjadi fokus penelitian. *Swarm quadrotor* digunakan dalam berbagai aplikasi, termasuk pemetaan area di mana *quadrotor* perlu menjaga jarak yang dekat satu sama lain. Namun, penggunaan *swarm quadrotor* dalam jarak dekat menghadirkan tantangan terkait interaksi aerodinamika yang dapat mempengaruhi kinerja, stabilitas, dan keamanan *quadrotor*.

Penelitian ini bertujuan untuk mengembangkan sistem kontrol kestabilan yang memungkinkan *quadrotor* dalam *swarm* untuk terbang dalam jarak yang sangat dekat tanpa mengalami gangguan signifikan akibat interaksi aerodinamika. Penelitian ini melibatkan pemodelan gangguan udara dan sistem pergerakan *quadrotor*, penggunaan algoritma kontrol PID, *Backstepping*, dan *feed forward*, simulasi menggunakan Simulink pada aplikasi MATLAB dan implementasi pada tiga unit *quadrotor*.

Hasil dari penelitian sejauh ini adalah berinteraksi dengan *quadrotor* via pemrograman bahasa python dengan laptop memanfaatkan protokol komunikasi jaringan nirkabel dan memberikan perintah dari laptop. Pengukuran gangguan udara menggunakan dua skema, menunjukkan bahwa gangguan udara terbesar didapatkan dari pantulan udara dengan tanah. Data respons pergerakan *quadrotor* dari penelitian sebelumnya dapat digunakan sebagai data awal untuk mempelajari simulasi pada simulink, tetapi perlu dilakukan pengukuran dengan lintasan yang lebih panjang untuk memastikan pergerakan *quadrotor* mencapai keadaan tunak.

Untuk langkah selanjutnya proses pengerjaan tugas akhir, melanjutkan pengerjaan tahap pertama akuisisi dan analisis data dan mulai mempelajari langkah awal pengerjaan tahap kedua simulasi. Pengukuran gangguan udara akan dilakukan pada dua buah *quadrotor* yang lain. Pengukuran respons pergerakan *quadrotor* dengan lintasan yang lebih panjang.

Kata kunci: *swarm quadrotor*, kontrol, simulasi, turbulensi udara.

ABSTRACT

PERANCANGAN SISTEM KONTROL PERGERAKAN UNTUK MENGATASI PENGARUH GANGGUAN UDARA DALAM FORMASI *MULTIQUADROTOR*

By

Yosep Putra Setiyanto

NIM: 13320026

(Engineering Physics Study Program)

In the era of technological development, the use of quadrotors has become increasingly relevant in various fields, including military, civil, technology, and science. However, the use of a single quadrotor to accomplish daily tasks still has limitations, such as low efficiency, difficulty in executing complex tasks, and limitations in area coverage.

To overcome these limitations, the development of quadrotor swarms or groups of quadrotors working together as a single unit has been the focus of research. Quadrotor swarms are used in a variety of applications, including area mapping where quadrotors need to keep a close distance from each other. However, the use of quadrotor swarms in close proximity presents challenges related to aerodynamic interactions that can affect quadrotor performance, stability and safety.

This research aims to develop a stability control system that enables quadrotors in a swarm to fly in very close proximity without experiencing significant disturbances due to aerodynamic interactions. The research involves modeling the air disturbance and quadrotor movement system, the use of PID, Backstepping, and feed forward control algorithms, simulation using Simulink on MATLAB application and implementation on three quadrotor units.

The results of the research so far are interacting with the quadrotor via python language programming with a laptop utilizing the wireless network communication protocol and giving commands from the laptop. Air disturbance measurements using two schemes show that the largest air disturbance is obtained from air reflection with the ground. Quadrotor movement response data from previous research can be used as initial data to study simulations on simulink, but it is necessary to take measurements with a longer trajectory to ensure the quadrotor movement reaches a steady state.

For the next step of the final project process, continue working on the first stage of data acquisition and analysis and start studying the initial steps of working on the second stage of simulation. Air disturbance measurements will be conducted on two other quadrotors. Measurement of quadrotor movement response with a longer trajectory.

Keywords: *swarm quadrotor, control, simulation, air turbulence.*

**PERANCANGAN SISTEM KONTROL PERGERAKAN
UNTUK MENGATASI PENGARUH GANGGUAN UDARA
DALAM FORMASI *MULTIQUADROTOR***

HALAMAN PENGESAHAN

Oleh

Yosep Putra Setiyanto NIM: 13320026
(Program Studi Teknik Fisika)

Institut Teknologi Bandung

Menyetujui

Tim Pembimbing

Tanggal

Pembimbing 1

(Ir. Estiyanti Ekawati, M.T., Ph.D.)

NIP. 196908052008012020

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 Kemajuan Tugas Akhir yang berjudul “Perancangan Sistem Kontrol Pergerakan Untuk Mengatasi Pengaruh Gangguan Udara Dalam Formasi *Multiquadrotor*”. Penulisan Laporan Kemajuan 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 Kemajuan 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. Prof. Brian Yulianto, S.T., M.Eng., Ph.D., selaku Dekan Fakultas Teknologi Industri, Institut Teknologi Bandung,
2. Iwan Prasetyo, S.T., M.T., Ph.D., selaku Ketua Program Studi Sarjana Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Bandung,
3. Ir. Estiyanti Ekawati, M.T., Ph.D., selaku pembimbing yang telah meluangkan waktu dan pikirannya untuk membimbing penulis dalam menyelesaikan penelitian dan penyusunan Laporan Kemajuan Tugas Akhir ini,
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. Salsabila Gitta Arifani yang memberikan dukungan dalam proses pengambilan data dan penyusunan Laporan Kemajuan Tugas Akhir ini serta senantiasa memberikan semangat kepada penulis,
7. Rekan-rekan di laboratorium CITA atas kebersamaan selama penelitian ini dilaksanakan,

8. Teman-teman S1 Teknik Fisika Angkatan 2020 yang memberikan dukungan dan semangat kepada penulis, dan
9. Semua pihak yang telah membantu hingga terselesaikannya Laporan Kemajuan Tugas Akhir ini yang tidak dapat penulis sebutkan satu per satu.

Akhir kata penulis memohon maaf apabila terdapat banyak kekurangan dalam penulisan Laporan Kemajuan 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 Kemajuan Tugas akhir ini memberikan manfaat bagi kita semua baik sekarang maupun dimasa yang akan datang.

Bandung, Desember 2023

Penulis

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	ix
DAFTAR SINGKATAN DAN LAMBANG	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Permasalahan.....	3
1.3 Tujuan dan Sasaran	4
1.4 Batasan dan Asumsi	5
1.5 Sistematika	5
BAB II STUDI LITERATUR	7
2.1 Sistem Dinamik <i>Quadrotor</i>	7
2.2 Protokol Komunikasi <i>Quadrotor</i>	10
2.2.1 <i>Transmission Control Protocol/Internet Protocol</i>	11
2.2.2 <i>User Datagram Protocol (UDP)</i>	11
2.3 Sistem Kontrol pada <i>Quadrotor</i>	11
2.3.1 <i>PID Control</i>	12
2.3.2 Backstepping-Integral Control	13
2.3.3 Feed Forward Control	14
2.4 Simulasi Menggunakan MATLAB/Simulink	15
2.4.1 <i>Dryden Wind Turbulence Model</i>	16
BAB III METODOLOGI DAN KEMAJUAN	18
3.1 Metodologi	18
3.1.1 Akuisisi dan Analisis Data	19

3.1.2	Simulasi	20
3.1.3	Implementasi dan Validasi	21
3.2	Kemajuan	22
3.2.1	Flowchart Pelaksanaan Tugas Akhir	22
3.2.2	Set up <i>Quadrotor</i> dan Medan Terbang.....	24
3.2.3	Set Up Aplikasi, <i>Library</i> dan Pemrograman	27
3.2.4	Uji Terbang.....	30
3.2.5	Pengukuran Gangguan Udara.....	31
3.2.6	Pengukuran Respons Pergerakan <i>Quadrotor</i>	34
BAB IV RENCANA SELANJUTNYA DAN PENUTUP		38
4.1	Rencana Selanjutnya	38
4.2	Penutup.....	38
DAFTAR PUSTAKA		39

DAFTAR GAMBAR

Gambar 1.2.1 Diagram tulang ikan rumusan permasalahan	4
Gambar 2.1.1 Dinamika pada <i>Quadrotor</i>	7
Gambar 2.3.1 Cara kerja sistem kontrol pada umumnya	12
Gambar 2.3.2 Skema kontrol PID	13
Gambar 2.4.1 Model turbulensi angin Dryden pada simulasi Simulink	16
Gambar 3.1.1 Diagram alir metodologi penelitian.....	18
Gambar 3.1.2 Skema akuisisi data karakteristik turbulensi udara	19
Gambar 3.1.3 Karakteristik dinamik <i>quadrotor</i>	19
Gambar 3.1.4 Skema kontrol pergerakan <i>quadrotor</i>	20
Gambar 3.1.5 Arsitektur komunikasi sistem.....	21
Gambar 3.1.6 Skema validasi performa implementasi sistem kontrol.....	22
Gambar 3.2.1 <i>Update</i> diagram alir proses pengerjaan tugas akhir	23
Gambar 3.2.2 <i>Quadrotor</i> Parrot AR.Drone 2.0	24
Gambar 3.2.3 Komponen pada <i>quadrotor</i> Parrot AR.Drone 2.0	25
Gambar 3.2.4 Antarmuka aplikasi AR.FreeFlight	25
Gambar 3.2.5 Medan terbang <i>quadrotor</i>	26
Gambar 3.2.6 Set up dan install library yang dibutuhkan.....	28
Gambar 3.2.7 Antarmuka program simulasi yang dikembangkan.....	29
Gambar 3.2.8 Program dasar untuk berinteraksi dengan <i>quadrotor</i>	29
Gambar 3.2.9 Program Dasar untuk mengontrol dan mengetahui state <i>quadrotor</i>	30
Gambar 3.2.10 Dokumentasi uji terbang menggunakan pemrograman.....	31
Gambar 3.2.11 Skema pertama pengukuran gangguan angin.....	31
Gambar 3.2.12 Skema kedua pengukuran gangguan angin	32
Gambar 3.2.13 Dinamika udara dari <i>quadrotor</i> pada arah horizontal	33
Gambar 3.2.14 Turbulensi udara yang dihasilkan <i>quadrotor</i>	34
Gambar 3.2.15 Hasil pengukuran respons kecepatan <i>quadrotor</i> 018167 berbagai arah	35

DAFTAR TABEL

Tabel 3.2.1 Hasil pengukuran kecepatan angin dengan skema pertama.....	32
Tabel 3.2.2 Hasil pengukuran kecepatan angin dengan skema kedua	32
Tabel 3.2.3 Hasil parameter FOPTD untuk arah maju.....	35
Tabel 3.2.4 Hasil parameter FOPTD untuk arah mundur	36
Tabel 3.2.5 Hasil parameter FOPTD untuk arah kanan	36
Tabel 3.2.6 Hasil parameter FOPTD untuk arah kiri	36

DAFTAR SINGKATAN

SINGKATAN	Nama	
PID	<i>Proportional Integral Derivative</i>	2
DCM	<i>Direction Cosine Matrix</i>	3
EF	<i>Earth Frame</i>	7
BF	<i>Body Frame</i>	7
OSI	<i>Open System Interconnection</i>	10
Wi-Fi	<i>Wireless Fidelity</i>	10
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>	11
UDP	<i>User Datagram Protocol</i>	11
GCS	<i>Ground Control Station</i>	21
FOPTD	<i>First Order Plus Time Delay</i>	34

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini, *quadrotor* memiliki potensi yang sangat tinggi dalam pengembangan aplikasi di bidang militer, sipil, teknologi, dan ilmu pengetahuan [1] [2]. Hal tersebut dikarenakan *quadrotor* menjadi salah satu robotik pesawat tanpa awak dengan keunggulan berupa struktur mekanik sederhana serta manuverabilitas yang dapat ditingkatkan [3]. Akan tetapi, penggunaan *single quadrotor* untuk memecahkan permasalahan sehari-hari masih memiliki keterbatasan, seperti efisiensi yang rendah dalam menjalankan tugas, sulit menjalankan tugas yang kompleks, dan tidak mampu mencakup area tugas yang luas. Maka dari itu, semakin gencar pengembangan sekelompok *quadrotor* atau *swarm quadrotor*. *Swarm quadrotor* akan berkoordinasi dan bekerja sama sebagai satu kesatuan dengan menjaga formasi yang diinginkan selama melaksanakan tugas [4].

Pengembangan *swarm quadrotor* yang semakin maju, membuatnya kini telah diaplikasikan dalam berbagai tujuan dan bidang keilmuan. Tak jarang pada aplikasi tersebut membutuhkan *swarm* dengan kerapatan atau jarak antar *quadrotor* yang berdekatan. Salah satu contohnya adalah untuk melakukan pemetaan area. Saat *quadrotor* harus menjaga jarak yang relatif aman satu sama lain, efisiensi pemetaan dapat terpengaruh karena cakupan area yang lebih luas mungkin tidak tercapai. Oleh karenanya, dibutuhkan formasi *swarm multiquadrotor* yang berdekatan satu sama lain.

Penggunaan *swarm quadrotor* yang mengharuskan jarak antar *quadrotor* berdekatan memunculkan permasalahan utama terkait dengan interaksi antar *quadrotor* dalam jarak dekat. Ketika *quadrotor* terbang dalam formasi yang rapat, interaksi aerodinamika antar *quadrotor* dapat menciptakan gangguan yang berdampak negatif pada kinerja, stabilitas, dan keamanan *quadrotor* tersebut. Pada penelitian yang dilakukan oleh Nekoo dkk (2021) telah dibuktikan bahwa rotor yang berputar pada *quadrotor* menyebabkan turbulensi udara di sekitarnya, dan akan memengaruhi kinerja dari *quadrotor* [5]. Hanya saja, penelitian tersebut

terbatas membahas turbulensi pada sumbu vertical saja belum membahas pengaruh turbulensi pada sumbu horizontal.

Penelitian yang dilakukan oleh Regula G. (2013) telah merancang algoritma kontrol yang memungkinkan *quadrotor* dalam *swarm quadrotor* untuk menjaga jarak yang aman satu sama lain. Algoritma ini memberikan solusi untuk menghindari tabrakan dan menjaga jarak yang cukup [6]. Meskipun solusi ini efektif dalam mencegah tabrakan, mereka sering kali kurang efisien dalam mengatasi gangguan aerodinamika yang muncul akibat interaksi antar *quadrotor* dalam jarak dekat [7]. Pembahasan pada penelitian tersebut cenderung mengutamakan mencegah tabrakan daripada mengatasi gangguan aerodinamika yang muncul akibat interaksi antar *quadrotor* dalam jarak dekat.

Pemahaman yang mendalam tentang pengendalian dan kompensasi gangguan ini menjadi penting untuk mengoptimalkan penggunaan *swarm quadrotor* dalam pemetaan. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem kontrol kestabilan yang memungkinkan *quadrotor* dalam *swarm quadrotor* untuk terbang berdekatan tanpa mengalami gangguan yang signifikan akibat interaksi aerodinamika antar *quadrotor*.

Pada penelitian yang dilakukan oleh Mukhlis dkk (2015) dikembangkan algoritma kontrol untuk mengompensasi pengaruh gangguan udara ketika *quadrotor* sedang terbang. Pada penelitian tersebut, gangguan udara bukan berasal dari *quadrotor* yang lain, namun berasal dari udara bebas karena uji coba dilakukan di tempat terbuka. Objek pada penelitian ini adalah sebuah *quadrotor* tunggal, bukan *swarm quadrotor*. Pada penelitian yang dilakukan oleh Kamran dkk (2014) juga dibuat algoritma kontrol *Proportional Integral Derivative* (PID) untuk mengatasi gangguan udara. Berdasarkan kedua penelitian tersebut kontrol PID terbukti telah berhasil mengurangi kesalahan rute penerbangan yang disebabkan gangguan dari udara [8] [9].

Penelitian mengenai kontrol *quadrotor* akan gangguan udara kini mulai dilakukan dengan menggunakan simulasi [10]. Salah satunya penelitian yang dilakukan oleh Dounia dkk (2020) yang memodelkan satu buah sistem pergerakan *quadrotor*

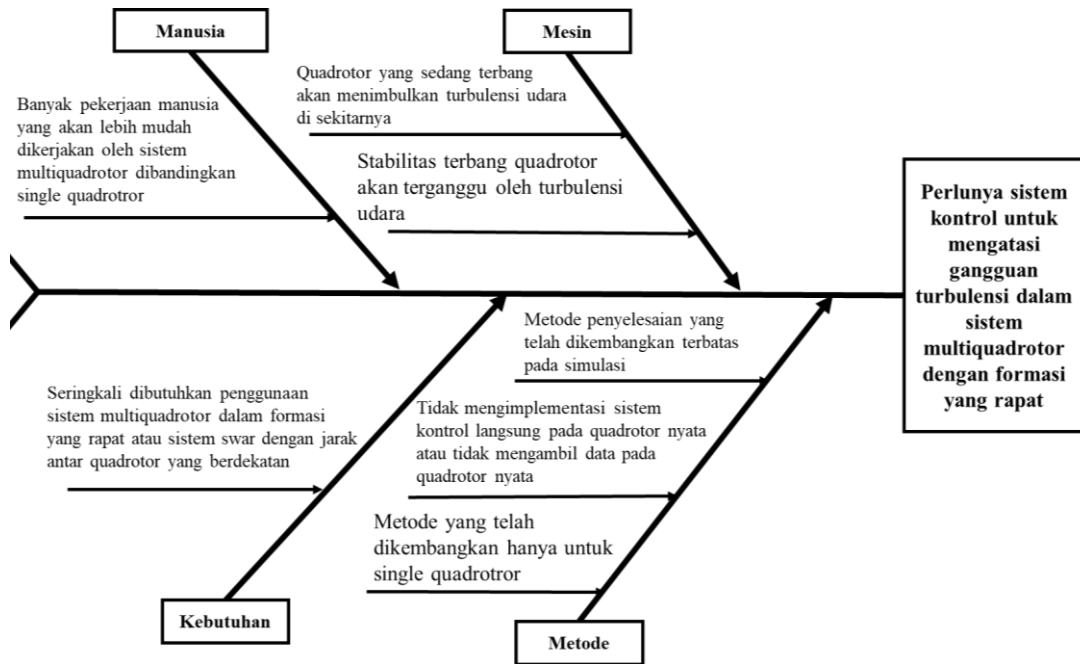
dalam alat simulasi Simulink pada aplikasi MATLAB. Gangguan udara disimulasikan dengan *Dryden Wind Turbulence Model* pada *toolbox aeroblocks*. Blok tersebut akan menerima *input* parameter seperti ketinggian, kecepatan *quadrotor*, kecepatan udara, *Direction Cosine Matrix* (DCM), dan model turbulensi [11].

Metode tersebut dapat diaplikasikan pada simulasi dalam penelitian yang akan dilakukan. Untuk parameter-parameter *Dryden Wind Turbulence Model*, kita perlu melakukan karakterisasi turbulensi udara dari *quadrotor*. Sesuai dengan riset yang dilakukan oleh Abichandani dkk (2020) pengukuran pengaruh udara dapat dilakukan dengan menggunakan anemometer [12]. Kontrol yang digunakan dalam simulasi tersebut adalah kontrol *backstepping-integral*. Kontrol tersebut telah terbukti dapat mengatur pergerakan sebuah *quadrotor* agar dapat mengatasi gangguan turbulensi udara [10].

Penelitian ini akan fokus pada mengatasi gangguan aerodinamika yang muncul akibat interaksi antar *quadrotor* dalam jarak dekat. Penelitian ini akan merancang model matematis yang memadai untuk pergerakan *quadrotor* dan gangguan udara yang dihasilkan *quadrotor* dalam kawanan. Berdasarkan model ini akan dirancangsistem kontrol menggunakan metode PID, *Backstepping Integral*, dan *Feed Forward Control*. Sistem kontrol ini dirancang melalui perangkat ‘simulink’ pada aplikasi MATLAB dan diimplementasikan pada tiga buah *quadrotor* Parrot AR Drone 2.0 Power Edition. Diharapkan hasil studi ini dapat memberikan kontribusi untuk peningkatan kinerja kontrol konfigurasi *multiquadrotor*.

1.2 Permasalahan

Berdasarkan latar belakang yang telah disampaikan, Pencarian akar masalah dilakukan dengan menggunakan analisis sebab-akibat (*fishbone diagram*), seperti pada Gambar 1.2.1 berikut.



Gambar 1.2.1 Diagram tulang ikan rumusan permasalahan

Berdasarkan analisis pada Gambar 1.2.1 di atas, dapat disimpulkan bahwa akar permasalahan dari penelitian ini adalah diperlukannya sistem kontrol untuk mengatasi gangguan turbulensi dalam sistem *multiquadrotor* dengan formasi rapat. Untuk menyelesaikan akar permasalahan tersebut, disusun rumusan permasalahan yang akan dibahas pada tugas akhir ini adalah sebagai berikut.

1. Bagaimana karakteristik gangguan yang muncul akibat interaksi antar *quadrotor* dalam jarak dekat ?
2. Bagaimana algoritma sistem kontrol yang memungkinkan *quadrotor* untuk tetap stabil ketika terbang dalam jarak yang dekat dengan mengompensasi gangguan yang muncul ?
3. Bagaimana memastikan kinerja sistem kontrol yang dikembangkan melalui simulasi dan eksperimen bekerja dengan baik dalam situasi dunia nyata ?

1.3 Tujuan dan Sasaran

Tujuan penelitian ini adalah untuk mengembangkan sistem kontrol kestabilan yang dapat memungkinkan *drone* dalam *swarm* untuk terbang dalam jarak yang sangat

dekat satu sama lain tanpa mengalami gangguan yang signifikan. Sasaran yang ingin dicapai melalui pemenuhan tujuan tersebut adalah:

1. Menganalisis karakteristik gangguan yang muncul akibat interaksi antar *quadrotor* dalam jarak dekat.
2. Mengembangkan pemodelan sistem sesuai dengan karakteristik setiap *quadrotor* yang digunakan dalam *swarm quadrotor*.
3. Merancang algoritma kontrol yang efektif untuk mengompensasi gangguan yang muncul, sehingga *quadrotor* dalam *swarm* dapat terbang dalam jarak yang sangat dekat satu sama lain tanpa mengalami gangguan yang signifikan.
4. Menguji dan memvalidasi sistem kontrol yang dikembangkan melalui simulasi dan eksperimen untuk memastikan kinerja yang baik dalam situasi dunia nyata.

1.4 Batasan dan Asumsi

Batasan dan asumsi pada pelaksanaan tugas akhir ini adalah sebagai berikut.

1. *Quadrotor* yang digunakan dalam *swarm quadrotor* Parrot AR *Quadrotor* 2.0 Power Edition sebanyak 3 buah.
2. Kondisi lingkungan dianggap stabil dan tidak berubah-ubah selama penerbangan *quadrotor*.
3. *Quadrotor* dalam *swarm quadrotor* akan memiliki sensor yang memadai untuk deteksi dan menghindari tabrakan.
4. *Quadrotor* dalam *swarm quadrotor* akan berkomunikasi secara terus-menerus untuk berkoordinasi.

1.5 Sistematika

Laporan tugas akhir akan terdiri dari enam bab dengan sistematika sebagai berikut.

BAB I PENDAHULUAN

Bab pertama berisi latar belakang penelitian yang menjadi dasar dalam menentukan masalah. Latar belakang akan berfokus membahas permasalahan yang dialami

objek penelitian. Permasalahan yang dikaji adalah diperlukannya sistem kontrol untuk mengatasi gangguan turbulensi dalam sistem *multiquadrotor* dengan formasi rapat. Pada bab ini akan dicantumkan pula pembahasan terkait tujuan, manfaat, batasan, dan asumsi penelitian.

BAB II STUDI LITERATUR

Bab kedua berisi dasar teori dan konsep yang akan digunakan dalam penelitian sebagai acuan dasar untuk merancang sistem kontrol pergerakan udara untuk *multiquadrotor*. Adapun teori yang digunakan untuk merancang sistem kontrol adalah sistem dinamik *quadrotor*, arsitektur komunikasi *quadrotor*, sistem kontrol pada *quadrotor*, dan simulasi menggunakan MATLAB/ Simulink.

BAB III METODOLOGI DAN KEMAJUAN

Bab ketiga berisi penjelasan langkah-langkah yang akan dilakukan untuk merancang sistem kontrol pergerakan udara pada *multiquadrotor*. Langkah-langkah tersebut diilustrasikan dalam bentuk diagram alir (*flowchart*) yang dilengkapi dengan penjelasan untuk tiap langkah yang digunakan. Pada bab ini, akan dijelaskan pula hasil kemajuan penelitian.

BAB IV RENCANA SELANJUTNYA & PENUTUP

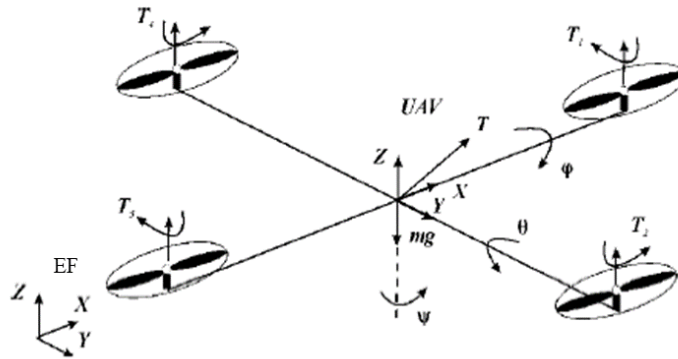
Bab empat akan berisi perencanaan proses penelitian yang akan dilakukan berdasarkan kemajuan yang telah didapat. Penelitian direncanakan agar dapat mendapatkan keluaran sesuai dengan harapan dan selesai tepat waktu. Selain itu, akan disusun pula penutup laporan kemajuan berupa harapan untuk proses penelitian yang dilakukan.

BAB II

STUDI LITERATUR

2.1 Sistem Dinamik *Quadrotor*

Quadrotor adalah salah satu jenis robot terbang dengan empat motor. Kelebihan *quadrotor* ada pada kemampuan manuvernya yang memungkinkan untuk bergerak leluasa di udara. Pergerakan *quadrotor* dapat dikendalikan dengan kontrol manual menggunakan *remote control* atau dengan sebuah sistem kontrol otomatis. Pengendalian pergerakan *quadrotor* dengan sistem kontrol otomatis memerlukan sebuah dinamika yang menggambarkan karakter dari *quadrotor* atau dinamika *quadrotor*. Dinamika *quadrotor* berdasarkan penelitian Kuchеров dkk. (2021) dapat diidentifikasi berdasarkan *earth frame* (EF) dan *body frame* (BF) seperti yang ditunjukkan Gambar 2.1.1. EF adalah sistem posisi tetap (XYZ) sedangkan BF adalah sistem posisi bergerak ($X'Y'Z'$) yang berasal dari pusat gravitasi *quadrotor*. Variabel φ , θ , ψ adalah sudut *roll*, *pitch*, dan *yaw* dalam EF [13].



Gambar 2.1.1 Dinamika pada *Quadrotor*

Persamaan untuk dinamika *quadrotor* pada ruang 3 dimensi dapat dituliskan dalam bentuk:

$$\begin{bmatrix} mI & 0 \\ 0 & J \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\Omega} \end{bmatrix} + \begin{bmatrix} \Omega \times mV \\ \Omega \times J\Omega \end{bmatrix} = \begin{bmatrix} \tilde{F} \\ \tilde{\tau} \end{bmatrix} \quad (\text{II.1})$$

$V = (V_x, V_y, V_z)^T = (\dot{x}, \dot{y}, \dot{z})^T$ adalah vektor linier, dan $\Omega = (\Omega_\phi, \Omega_\theta, \Omega_\psi)^T = (\dot{\phi}, \dot{\theta}, \dot{\psi})^T$ adalah vektor kecepatan sudut, m adalah massa *quadrotor*, $J \in \mathbb{R}^{3 \times 3}$ adalah matriks inersia dari objek yang di kontrol, I adalah matriks identitas, \mathfrak{F} adalah gaya resultan, \mathfrak{M} adalah momen resultan yang bekerja pada pusat massa dari *quadrotor*.

Jika gaya dari luar diabaikan, maka gaya yang dihasilkan menentukan pergerakan *quadrotor* pada sumbu z dapat ditulis:

$$\mathfrak{F}^T = (0 \quad 0 \quad \dot{z}) \quad (\text{II.2})$$

Pada sistem koordinat Euler, orientasi *quadrotor* relatif terhadap sistem koordinat inersia XYZ dapat dituliskan sebagai matriks rotasi R

$$\mathfrak{F}_V = R(\phi, \theta, \psi) \mathfrak{F} \quad (\text{II.3})$$

$$R = \begin{pmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{pmatrix} \quad (\text{II.4})$$

Matriks pada persamaan (II.4) menggunakan notasi $C_\alpha = \cos(\alpha)$ dan $S_\alpha = \sin(\alpha)$

Jika persamaan (II.3) dan (II.4) dikombinasikan maka dapat dihasilkan:

$$m\ddot{x} = \bar{u}_1(C_\psi S_\theta C_\phi + S_\psi S_\phi) \quad (\text{II.5})$$

$$m\ddot{y} = \bar{u}_1(S_\psi S_\theta C_\phi + C_\psi S_\phi) \quad (\text{II.6})$$

$$m\ddot{z} = \bar{u}_1 C_\theta C_\phi - mg \quad (\text{II.7})$$

Dimisalkan J_1, J_2, J_3 masing-masing adalah momen inersia *quadrotor* untuk sumbu X, Y, dan Z yang melalui pusat massanya. Kemudian $\dot{\phi}, \dot{\theta}, \dot{\psi}$ adalah kecepatan sudut terhadap sumbu X, Y, dan Z. Jika gaya dari luar diabaikan, persamaan diferensial dari kecepatan sudut dapat dituliskan dalam bentuk:

$$J_1 \ddot{\phi}(t) = (J_2 - J_3) \dot{\theta}(t) \dot{\psi}(t) \quad (\text{II.8})$$

$$J_2 \ddot{\theta}(t) = (J_3 - J_1) \dot{\psi}(t) \dot{\phi}(t) \quad (\text{II.9})$$

$$J_3 \ddot{\psi}(t) = (J_1 - J_2) \dot{\phi}(t) \dot{\theta}(t) \quad (\text{II.10})$$

Persamaan (II.8) – (II.10) adalah persamaan dinamika Euler. Jika menggunakan kerja motor M_1 - M_4 maka persamaannya adalah:

$$J_1 \ddot{\phi}(t) = (J_2 - J_3) \dot{\theta}(t) \dot{\psi}(t) - J_r \dot{\theta}(t) \Omega + d \bar{u}_3(t) \quad (\text{II.11})$$

$$J_2 \ddot{\theta}(t) = (J_3 - J_1) \dot{\psi}(t) \dot{\phi}(t) + J_r \dot{\phi}(t) \Omega + d \bar{u}_4(t) \quad (\text{II.12})$$

$$J_3 \ddot{\psi}(t) = (J_1 - J_2) \dot{\phi}(t) \dot{\theta}(t) + \bar{u}_2(t) \quad (\text{II.13})$$

Pada persamaan (II.11) – (II.13) d adalah jarak antara pusat massa *quadrotor* dan motor dengan \mathfrak{J}_i , J_r adalah inersia rotor, $\bar{u}_1, \bar{u}_2, \bar{u}_3, \bar{u}_4$ adalah gaya mekanik yang memberikan gerakan rotasi *drone*, ditulis pada persamaan:

$$\bar{u}_1 = k_1 \Omega_1^2 + k_2 \Omega_2^2 + k_3 \Omega_3^2 + k_4 \Omega_4^2 \quad (\text{II.14})$$

$$\bar{u}_2 = k_1 \Omega_1^2 + k_2 \Omega_2^2 - k_3 \Omega_3^2 - k_4 \Omega_4^2 \quad (\text{II.15})$$

$$\bar{u}_3 = k_4 \Omega_4^2 - k_2 \Omega_2^2 \quad (\text{II.16})$$

$$\bar{u}_4 = k_1 \Omega_1^2 - k_3 \Omega_3^2 \quad (\text{II.17})$$

k_i adalah faktor yang menghubungkan momen dengan kecepatan sudut Ω_i dari motor ke i pada persamaan (II.14) – (II.17). Mempertimbangkan identitas dari semua motor, lokasi yang simetris, dan struktur yang *rigid*, maka dapat ditentukan momen J_1 dan J_2 pada ketinggian tetap (sumbu Z) dalam persamaan (II.11) – (II.13) adalah:

$$J_1 = J_2 = J \quad (\text{II.8})$$

Dinamika rotasi dari *quadrotor* pada persamaan (II.11) – (II.13) dapat ditulis dalam bentuk:

$$\ddot{\phi}(t) = \frac{1}{J}[(J - J_3)\dot{\theta}(t)\dot{\psi}(t) - J_r\dot{\theta}(t)\Omega + d\bar{u}_3(t)] \quad (\text{II.11})$$

$$\ddot{\theta}(t) = \frac{1}{J}[-(J - J_3)\dot{\psi}(t)\dot{\phi}(t) + J_r\dot{\phi}(t)\Omega + d\bar{u}_4(t)] \quad (\text{II.12})$$

$$\ddot{\psi}(t) = \frac{1}{J}\bar{u}_2(t) \quad (\text{II.13})$$

Jika diberikan kondisi $\psi(0)$ dan momen u_3 diketahui, maka posisi $\psi(t)$ dapat diketahui dengan melakukan 2 kali diferensial:

$$\psi(t) = \psi(0) + \frac{1}{J}\bar{u}_2(t)dt^2 \quad (\text{II.13})$$

Demikian juga dengan posisi $\phi(t), \theta(t)$ [13].

2.2 Protokol Komunikasi *Quadrotor*

Swarm quadrotor merujuk pada kelompok *quadrotor* yang bekerja secara kolektif dalam suatu sistem. *Quadrotor* dalam *swarm* berkomunikasi dan berkoordinasi untuk mencapai tujuan tertentu, meniru prinsip kerja kelompok serangga atau burung dalam alam. Protokol komunikasi berupa jaringan nirkabel pada *swarm quadrotor* mengatur pertukaran informasi antara *quadrotor* dalam suatu kelompok menggunakan teknologi jaringan nirkabel.

Secara umum, jaringan nirkabel didefinisikan sebagai media transmisi yang dapat beroperasi tanpa menggunakan kabel dalam suatu sistem komunikasi. Jenis jaringan ini lebih fleksibel dan sederhana dibandingkan jenis jaringan lainnya karena banyak pengguna dapat dengan mudah terhubung ke jaringan pada saat yang bersamaan. Jaringan nirkabel adalah jenis jaringan yang menggunakan komunikasi radio nirkabel untuk menghubungkan perangkat jaringan seperti *router*, komputer, server, dan banyak lainnya. Jaringan nirkabel diterapkan pada lapisan fisik, yang merupakan lapisan 1 dari model *Open System Interconnection* (OSI). OSI adalah sebuah model referensi untuk vendor atau developer agar pembuatan perangkat lunak memiliki sifat interpolasi. Selain lapisan fisik atau *application layer*, lapisan lain yang ada pada model OSI adalah *presentation*, *session*, *transport*, *network*, dan *data-link*.

Jaringan nirkabel memiliki *Access point* yang bekerja secara efektif dengan *transceiver* radio untuk mengembangkan koneksi yang memungkinkan sinyal radio ditransmisikan dan diterima. Sinyal ini dikenali oleh perangkat *client*, dan setelah mengonfirmasi saluran komunikasi akan menyediakan akses tambahan ke jaringan. *Access point* nirkabel mengadopsi standar komunikasi nirkabel umum, yaitu protokol IEEE 802.11. Aplikasi yang paling umum dari standar ini adalah WiFi, yang dikenal sebagai *Wireless Fidelity* [14].

2.2.1 *Transmission Control Protocol/Internet Protocol*

TCP/IP adalah *Transmission Control Protocol/Internet Protocol*. TCP/IP merupakan jenis protokol yang memungkinkan komunikasi antar perangkat jaringan. Satu komputer dengan TCP/IP dapat berkomunikasi dengan komputer lain pada satu lingkup jaringan nirkabel. TCP/IP adalah protokol yang berada di lapisan *transport* pada model OSI. Pada protokol ini proses transfer data akan dilakukan dengan cara membagi data yang akan dikirimkan ke perangkat lain menjadi paket-paket kecil yang ditambahkan informasi untuk verifikasi data tersebut. Proses verifikasi pada saat penerimaan data merupakan salah satu keunggulan TCP/IP untuk memastikan bahwa data yang dikirimkan tidak mengalami kerusakan [15].

2.2.2 *User Datagram Protocol (UDP)*

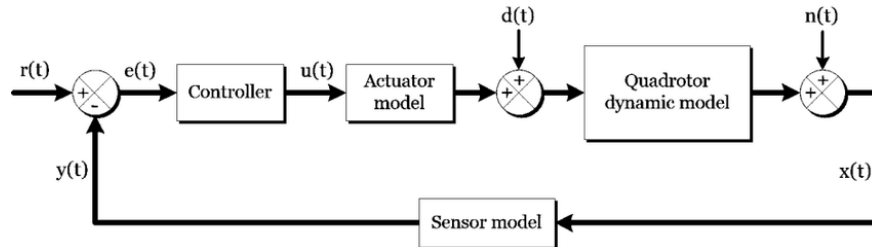
UDP adalah *User Datagram Protocol* yang merupakan salah satu protokol dengan posisi lapisan yang sama dengan TCP/IP pada model OSI yaitu pada lapisan *transport*. Tidak seperti TCP/IP data yang dikirimkan melalui protokol UDP berupa datagram dan tidak memiliki informasi verifikasi sehingga memiliki kemungkinan data akan rusak saat pengiriman. Tetapi, protokol UDP memiliki keunggulan dalam ukuran data yang kecil saat data dikirimkan. Hal tersebut membuat proses pengiriman data pada protokol UDP memiliki kecepatan yang lebih tinggi dibandingkan TCP/IP [16].

2.3 Sistem Kontrol pada *Quadrotor*

Sistem kontrol pada *quadrotor* adalah serangkaian elemen dan algoritma yang bertujuan mengatur dan menjaga kestabilan, posisi, dan gerakan *quadrotor*. Sistem

ini memanfaatkan sensor dan aktuator untuk memonitor dan mengendalikan kondisi *quadrotor* secara *real-time* [17].

Cara kerja sistem kontrol pada *quadrotor*, sesuai pada Gambar 2.3.1, dibagi menjadi tiga elemen dasar yaitu elemen *sensing* (sensor), elemen algoritma kontrol, dan elemen aktuator. *Quadrotor* dilengkapi dengan sensor seperti *gyroscopes*, *accelerometers*, dan *magnetometers* untuk mengukur orientasi dan percepatan. Algoritma kontrol digunakan untuk menerjemahkan data sensor menjadi sinyal kontrol yang sesuai. Motor dan *propeller* pada *quadrotor* berperan sebagai aktuator yang menerapkan sinyal kontrol untuk menghasilkan gerakan yang diinginkan .



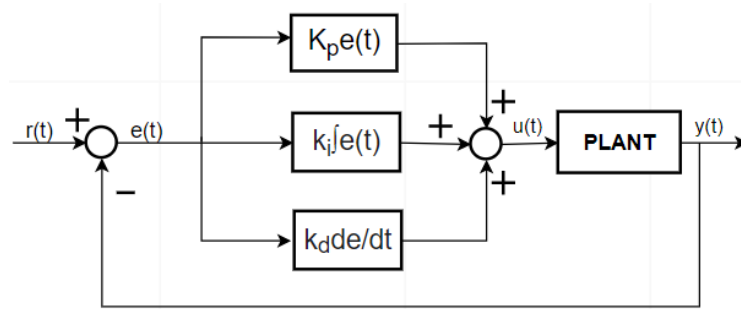
Gambar 2.3.1 Cara kerja sistem kontrol pada umumnya

Dengan adanya sistem kontrol pada *quadrotor*, terdapat banyak keuntungan atau kelebihan. *Quadrotor* mampu menjaga kestabilan *quadrotor* dalam berbagai kondisi. Memungkinkan *quadrotor* untuk melakukan manuver dengan presisi. Dapat diterapkan pada sistem *swarm* untuk menjaga formasi. Namun tidak sedikit juga kekurangan sistem kontrol pada *quadrotor*. Beberapa metode kontrol memiliki kompleksitas yang tinggi. Beberapa sistem kontrol sensitif terhadap perubahan parameter dinamika.

2.3.1 PID Control

Kontrol PID (Proporsional-Integral-Derivatif) adalah metode pengendalian yang digunakan dalam sistem otomatis untuk meminimalkan kesalahan antara *output* yang diinginkan dengan *output* aktual. Tujuan utama dari kontrol PID adalah menjaga sistem agar tetap stabil dan responsif terhadap perubahan lingkungan atau *input* yang diberikan [9].

Kelebihan dari kontrol PID terletak pada kemampuannya yang relatif mudah diimplementasikan dan dapat berfungsi baik pada sistem yang kompleks. Selain itu, PID memiliki karakter responsif yang baik terhadap perubahan *input* dan mampu bekerja stabil dalam berbagai kondisi. Meskipun demikian, terdapat kekurangan pada kontrol PID, seperti rentan terhadap *overshooting* dan membutuhkan penyetelan parameter secara manual untuk mencapai kinerja optimal .



Gambar 2.3.2 Skema kontrol PID

Skema kontrol PID dapat dijelaskan pada Gambar 2.3.2. Elemen proporsional memberikan respons sebanding dengan kesalahan saat ini, elemen integral merespons terhadap kesalahan selama waktu, sementara elemen derivatif mengurangi *overshooting* dengan mengevaluasi laju perubahan kesalahan. Persamaan umum kontrol PID dapat dirumuskan sebagai

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{d(e(t))}{dt} \quad (\text{II.14})$$

dengan $u(t)$ sebagai sinyal kontrol, $e(t)$ sebagai kesalahan saat ini, dan K_p , K_i , dan K_d sebagai konstanta kontrol proporsional, integral, dan derivatif [8] [9].

2.3.2 Backstepping-Integral Control

Kontrol Backstepping-Integral adalah metode kontrol yang dikembangkan untuk mengatasi gangguan pada sistem non-linear. Tujuannya adalah untuk mencapai pelacakan referensi yang baik dan mengompensasi gangguan dengan merancang kontrol berdasarkan prinsip gerak mundur (backstepping) dan integrasi kesalahan (integral). Kontrol ini diterapkan pada sistem yang kompleks dan non linear, seperti *quadrotor*, untuk meningkatkan kinerja dan ketahanan terhadap gangguan [18].

Cara kerja kontrol *backstepping-integral* dapat dijelaskan sebagai berikut. Pertama, sistem dinamik aslinya dipecah menjadi beberapa sub sistem virtual, dan kontrol dirancang untuk masing-masing sub sistem ini secara berurutan. Selanjutnya, elemen integral digunakan untuk menangani kesalahan statis dan merancang kontrol yang mengompensasi kesalahan dinamis pada setiap langkahnya. Ini memungkinkan sistem untuk mencapai pelacakan referensi yang akurat dan responsif terhadap gangguan.

Rumus umum kontrol *backstepping-integral* mencakup unsur integrasi kesalahan dan kontrol berbasis *backstepping* untuk masing-masing sub sistem virtual. Penggunaan integral membantu menghilangkan kesalahan statis, sedangkan prinsip *backstepping* membantu mengatasi kesalahan dinamis [19]. Berdasarkan penelitian yang dilakukan oleh Dounia dkk. (2020), Metode backstepping integral didasarkan pada Teori Stabilitas Lyapunov. Dengan mempertimbangkan model dinamik sistem, variabel kesalahan didefinisikan sebagai berikut:

$$e_{1i} = \xi_{di} - \xi_i \quad (\text{II.15})$$

$$p_i(t) = \int e_{1i}(\tau) d\tau \quad (\text{II.16})$$

Fungsi Lyapunov untuk backstepping integral direpresentasikan sebagai berikut

$$\dot{V}_{1i} = \frac{1}{2} e_{1i}^2 + \beta_i p_i^2 + \frac{1}{2} e_{2i}^2 \quad (\text{II.17})$$

$$\dot{V}_{2i} = -\alpha_{1i} e_{1i}^2 - \alpha_{2i} e_{2i}^2 \quad (\text{II.18})$$

Maka, diperoleh persamaan untuk kontrol backstepping integral

$$u_i = \frac{1}{b} [-\alpha_{1i} \beta_i p_i + (\beta_i + 1 - \alpha_{1i}^2) e_{1i} + (\alpha_{1i} + \alpha_{2i}) e_{1i} + \xi_{di} - f_i] \quad (\text{II.19})$$

Dengan $\beta_i, \alpha_{1i}, \alpha_{2i} \in \mathbb{R}$ adalah parameter-parameter yang perlu dilakukan tuning.

2.3.3 Feed Forward Control

Kontrol *feed forward* adalah suatu metode pengendalian di mana tindakan pengendalian langsung diterapkan pada sistem berdasarkan prediksi sebelumnya

mengenai gangguan atau perubahan yang akan terjadi. Tujuannya adalah untuk meminimalkan efek gangguan tanpa menunggu sistem memberikan respons terhadap gangguan tersebut. Dalam hal ini, tindakan kontrol dilakukan sebelum gangguan benar-benar memengaruhi sistem [20].

Cara kerja kontrol *feed forward* dapat dijelaskan sebagai berikut. Pertama, sistem di karakterisasi dengan baik sehingga respons sistem terhadap gangguan dapat diantisipasi. Prediksi ini kemudian digunakan untuk menghasilkan tindakan kontrol yang diterapkan pada sistem. Misalnya, pada *quadrotor*, kontrol *feed forward* dapat digunakan untuk mengompensasi gangguan turbulensi udara atau perubahan kecepatan angin yang diketahui dapat memengaruhi kinerja *quadrotor*. Rumus atau formula kontrol *feed forward* dapat dinyatakan sebagai jumlah tertimbang dari gangguan yang diantisipasi.

Kelebihan dari kontrol *feed forward* terletak pada kemampuannya untuk mengatasi gangguan tanpa menunggu sistem memberikan respons. Hal ini membuatnya sangat responsif dan efektif dalam merespons perubahan kondisi seketika. Kelebihan lainnya adalah pengurangan beban pada sistem umpan balik (*feedback*), karena sebagian besar tugas dikendalikan sebelum terjadinya gangguan. Namun, kelemahan utamanya adalah kurangnya respons terhadap ketidakpastian dan variasi yang tidak dapat diprediksi dalam sistem.

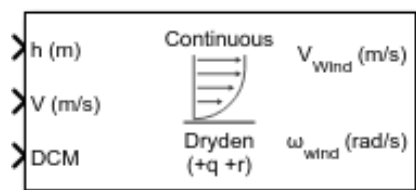
2.4 Simulasi Menggunakan MATLAB/Simulink

MATLAB/Simulink adalah perangkat lunak pemrograman yang digunakan secara luas di berbagai industri, termasuk robotika dan kendaraan tanpa awak seperti *quadrotor*. MATLAB digunakan untuk analisis dan perhitungan numerik, sedangkan Simulink adalah alat pengembangan sistem dinamika dan kontrol. Tujuan penggunaan MATLAB/Simulink dalam simulasi *quadrotor* adalah untuk merancang, menguji, dan memvalidasi algoritma kontrol pada lingkungan yang aman dan efisien sebelum diimplementasikan pada *quadrotor* fisik. MATLAB/Simulink menyediakan platform yang kuat untuk pemodelan sistem dinamis, simulasi, dan analisis performa kontrol [21].

Cara Kerja penggunaan MATLAB/Simulink dalam simulasi *quadrotor* dibagi menjadi empat tahap. Tahap yang pertama adalah pemodelan *quadrotor* yang dilakukan dengan mendefinisikan parameter dinamis dan karakteristik fisiknya menggunakan blok-blok Simulink yang sesuai. Tahap kedua melakukan simulasi gangguan udara. Gangguan udara, seperti yang dihasilkan oleh model Turbulensi Angin Dryden, dapat diintegrasikan ke dalam model Simulink untuk mengevaluasi respons *quadrotor* terhadap kondisi lingkungan yang realistis. Tahap ketiga adalah melakukan desain kontrol. Algoritma kontrol, seperti PID atau *backstepping-integral*, dapat dirancang dan diimplementasikan dalam Simulink dengan menambahkan blok kontrol ke model sistem. Tahap yang terakhir adalah melakukan analisis performa. MATLAB/Simulink memungkinkan analisis performa sistem kontrol, seperti respons transien, stabilitas, dan kinerja sistem.

2.4.1 Dryden Wind Turbulence Model

Model Turbulensi Angin Dryden adalah model matematis yang digunakan untuk menggambarkan karakteristik turbulensi udara yang memengaruhi pesawat atau kendaraan udara, seperti *quadrotor*. Model Turbulensi Angin Dryden dirancang untuk memodelkan variasi kecepatan angin dengan mempertimbangkan spektrum daya (*power spectrum*) dari fluktuasi kecepatan angin. Dengan demikian, tujuannya adalah untuk memberikan representasi yang akurat terhadap efek turbulensi udara yang dapat memengaruhi kinerja pesawat, terutama pada *quadrotor* yang sensitif terhadap perubahan kecepatan angin [11].



Gambar 2.4.1 Model turbulensi angin Dryden pada simulasi Simulink

Model ini sering digunakan dalam simulasi untuk menguji dan mengembangkan algoritma kontrol yang dapat mengatasi dampak turbulensi udara. Model Turbulensi Angin Dryden pada simulasi Simulink MATLAB ditunjukkan oleh Gambar 2.4.1. Kelebihan dari model ini terletak pada kemampuannya untuk

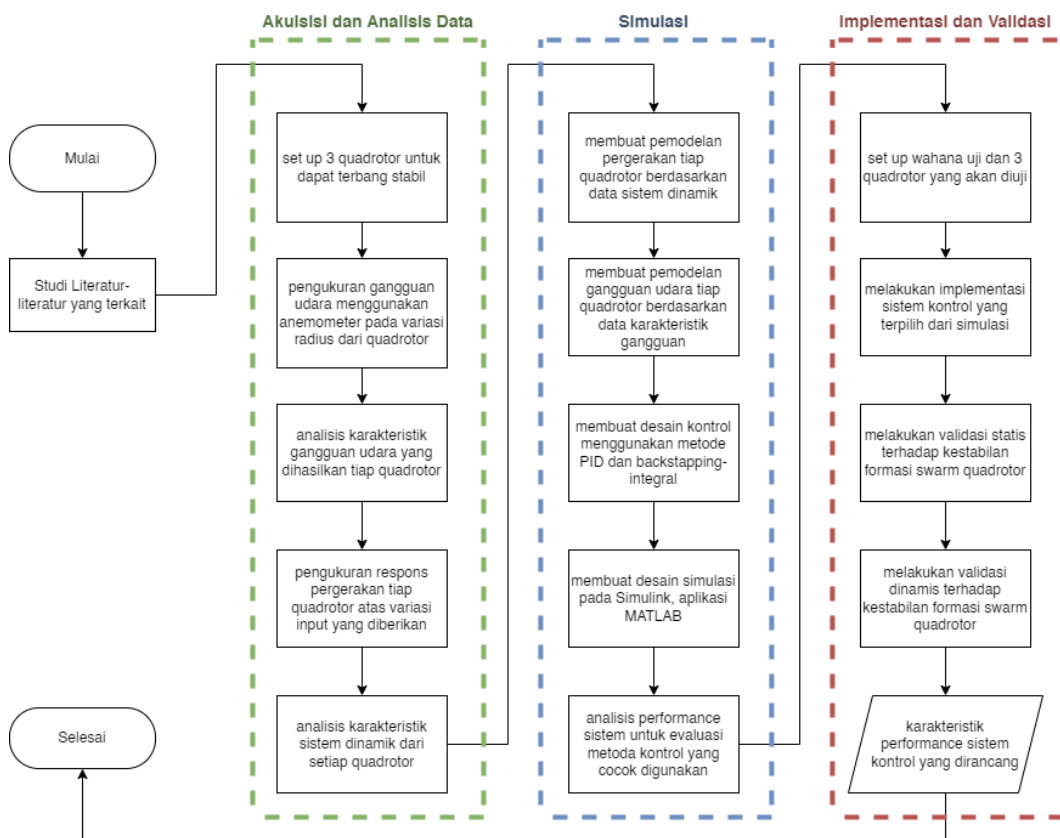
memodelkan turbulensi udara dengan mempertimbangkan variasi frekuensi dan amplitudo fluktuasi kecepatan angin. Namun, kekurangannya mungkin terletak pada asumsi-asumsi tertentu yang digunakan dalam pembentukan model, yang mungkin tidak selalu sesuai dengan kondisi atmosfer sebenarnya.

BAB III

METODOLOGI DAN KEMAJUAN

3.1 Metodologi

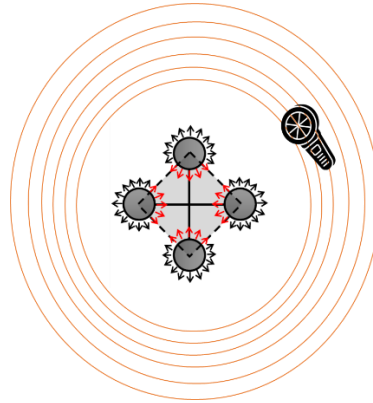
Metodologi yang akan dilakukan untuk mencapai tujuan dari proyek tugas akhir ini dijelaskan melalui Gambar 3.1.1 Diagram alir metodologi penelitian.



Gambar 3.1.1 Diagram alir metodologi penelitian

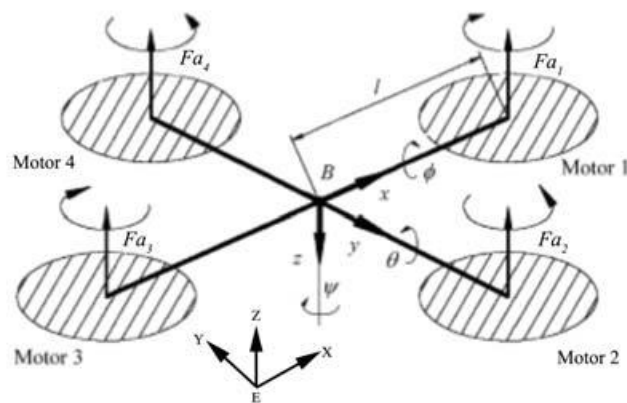
Berdasarkan Gambar 3.1.1 Diagram alir metodologi penelitian, langkah pertama yang perlu dilakukan adalah mencari literatur-literatur yang telah ada sebagai referensi terkait perancangan sistem kontrol yang optimal pada *quadrotor* untuk mengatasi pengaruh gangguan turbulensi udara. Berdasarkan Gambar 3.1.1 Diagram alir metodologi penelitian secara umum alur perancangan terdiri dari 3 tahap utama yaitu akuisisi dan analisis data, simulasi, dan implementasi.

3.1.1 Akuisisi dan Analisis Data



Gambar 3.1.2 Skema akuisisi data karakteristik turbulensi udara

Pada tahap akuisisi dan analisis data, akan dilakukan dua proses pengukuran. Proses pengukuran yang pertama adalah pengukuran gangguan turbulensi udara yang dihasilkan tiap *quadrotor* pada variasi jarak radius dari *quadrotor*. sesuai pada Gambar 3.1.2 di atas, proses pengambilan data dilakukan pada tiap *quadrotor* yang sedang terbang secara *steady* menggunakan sensor Anemometer. Data pengukuran divariasikan titik pengukuran pada beberapa radius dari *quadrotor* yang sedang terbang. Data hasil pengukuran tersebut akan dianalisis untuk mendapatkan karakteristik gangguan turbulensi udara yang muncul akibat interaksi antar *quadrotor* dalam jarak dekat.



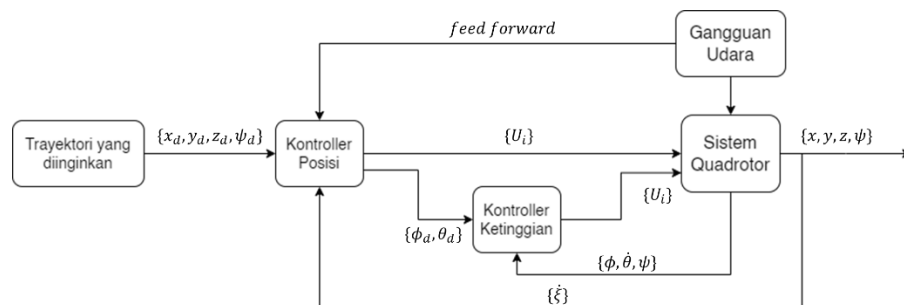
Gambar 3.1.3 Karakteristik dinamik *quadrotor*

Proses pengukuran yang kedua adalah pengukuran respons dari tiap *quadrotor* atas *input* yang diberikan. Data hasil pengukuran tersebut akan dianalisis untuk

menentukan karakteristik sistem dinamik dari setiap *quadrotor* yang akan digunakan. Berdasarkan pada Gambar 3.1.3 di atas, karakteristik dinamik terbang dari sebuah *quadrotor* meliputi roll (θ), pitch (ϕ) and yaw (ψ). Karakterisasi dilakukan menurut persamaan pada bagian Sistem Dinamik *Quadrotor* sebelumnya. Sistem dinamik tersebut nantinya akan berguna untuk mengembangkan pemodelan sistem sesuai dengan karakteristik setiap *quadrotor* yang digunakan dalam *swarm quadrotor*. Kedua proses pengukuran ini menjadi penting sebelum tahap pembuatan kontrol dalam simulasi.

3.1.2 Simulasi

Pada tahap yang kedua, yaitu tahap simulasi akan didesain pemodelan gangguan udara, pemodelan sistem, dan juga sistem kontrol yang akan coba diimplementasikan. Simulasi akan menggunakan paket perangkat lunak MATLAB yaitu Simulink. Berdasarkan data yang telah diperoleh dari tahap akuisisi dan analisis data, maka dapat dilakukan pemodelan gangguan dan pemodelan respons *quadrotor*.

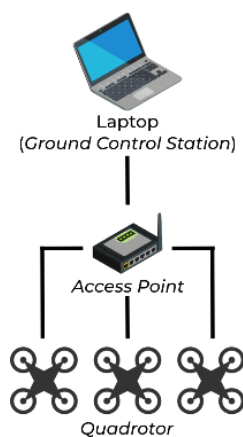


Gambar 3.1.4 Skema kontrol pergerakan *quadrotor*

Perancangan sistem kontrol dilakukan sesuai diagram pada Gambar 3.1.4 di atas. Sistem kontrol akan didesain menggunakan beberapa pilihan metode kontrol yang telah banyak dikembangkan. Pada akhir tahap simulasi, performa sistem kontrol dalam mengatasi gangguan udara akan didapatkan. Hasil performa tersebut, dapat digunakan dalam menentukan batas-batas jarak ideal antar *quadrotor* dalam skema *swarm quadrotor*.

3.1.3 Implementasi dan Validasi

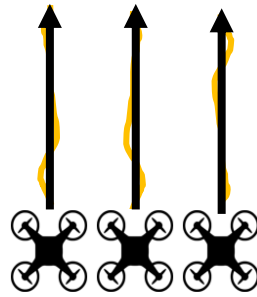
Tahap yang terakhir adalah implementasi dan validasi sistem kontrol yang dikembangkan melalui simulasi dan eksperimen untuk memastikan kinerja yang baik dalam situasi dunia nyata. Pada Gambar 3.1.5 diberikan ilustrasi arsitektur komunikasi antar *quadrotor* saat melakukan implementasi dan validasi. Arsitektur komunikasi tersebut menunjukkan jalur komunikasi dari peralatan lapangan yaitu tiga buah *quadrotor*, kemudian ke stasiun pengontrol atau *Ground Control Station* (GCS).



Gambar 3.1.5 Arsitektur komunikasi sistem

Untuk melakukan pengambilan data ke tiga *quadrotor* secara bersamaan, diperlukan jalur komunikasi yang menghubungkan mereka. Seperti pada Gambar 3.1.5 sebuah *Access Point* (AP) digunakan sebagai penghubung jalur komunikasi ini. Sehingga GCS terhubung ke AP tersebut, kemudian digunakan untuk mengubah alamat *Internet Protocol* (IP) dari masing-masing *quadrotor* dan menghubungkannya ke GCS

Validasi akan digunakan dua buah cara, yaitu statis dan dinamis. Pada validasi statis, *quadrotor* akan terbang secara tunak dengan *altitude* yang seragam. Namun, pada validasi ini akan divariasikan jarak antar *quadrotor*. Dari proses tersebut akan didapatkan rentang jarak antar *quadrotor* yang membuat formasi *swarm quadrotor* akan tetap stabil.



Gambar 3.1.6 Skema validasi performa implementasi sistem kontrol

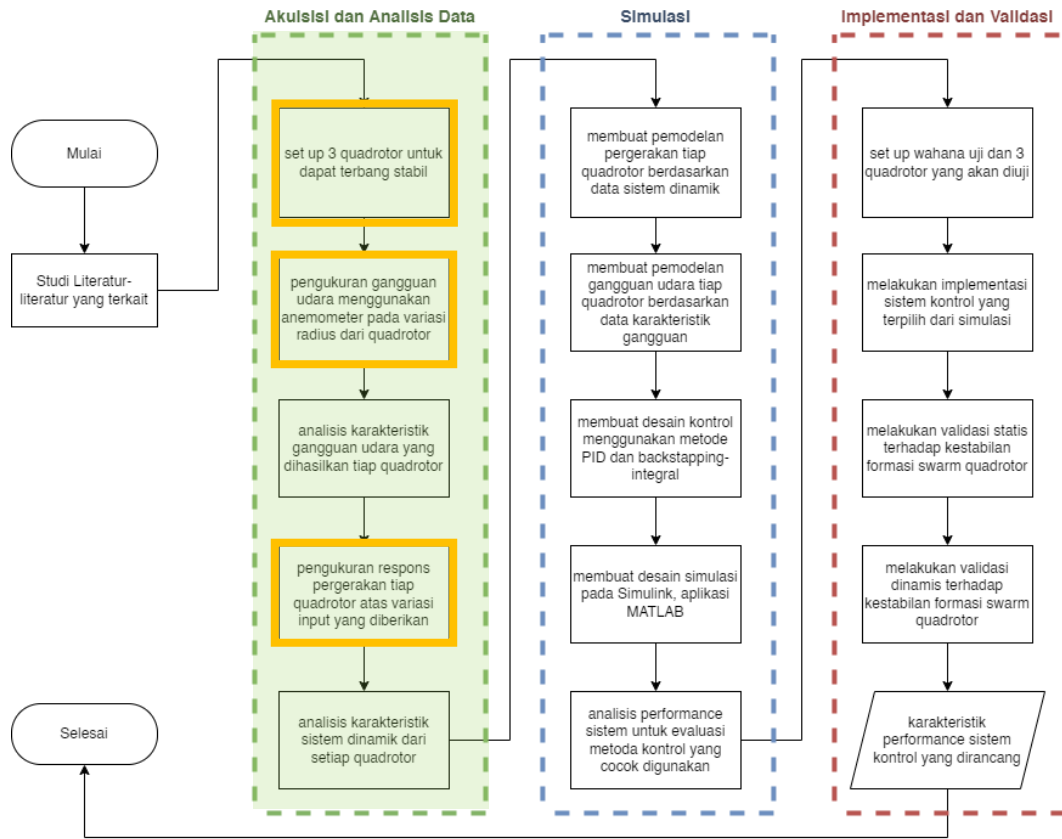
Pada validasi dinamis, dapat dilihat sesuai pada Gambar 3.1.6, *swarm quadrotor* akan dijalankan pada rute lurus saja. Namun pada validasi ini akan coba divariasikan kecepatan gerak *swarm quadrotor*. Dari proses tersebut akan didapatkan *range* kecepatan yang membuat formasi *swarm quadrotor* akan tetap stabil. Rentang jarak antar *quadrotor* dan rentang kecepatan *swarm quadrotor* akan menjadi karakteristik sistem kontrol yang dirancang.

3.2 Kemajuan

Pada bagian ini, akan dijelaskan mengenai kemajuan penelitian tugas akhir yang telah dicapai.

3.2.1 Flowchart Pelaksanaan Tugas Akhir

Pada Gambar 3.2.1 ditampilkan bagian dari diagram alir proses pengerjaan tugas akhir yang dilakukan sejauh ini.



Gambar 3.2.1 Update diagram alir proses pengerjaan tugas akhir

Berdasarkan Gambar 3.2.1, dilakukan tahap yang pertama dari tiga tahap besar proses pengerjaan tugas akhir, yaitu tahap Akuisisi dan Analisis Data. Pada tahap tersebut terdapat lima langkah yang harus dikerjakan. Pertama adalah proses *set up* tiga buah *quadrotor* untuk dapat terbang dengan stabil. Langkah tersebut telah dilakukan dan akan dijelaskan lebih detail kemajuannya pada bagian *Set up Quadrotor* dan *Medan Terbang* di bawah.

Langkah kedua adalah pengukuran gangguan udara menggunakan anemometer pada variasi radius dari *quadrotor*. Langkah tersebut telah dilakukan, tetapi untuk saat ini baru dilakukan untuk satu buah *quadrotor* dan hanya dilakukan beberapa variasi radius. Hal tersebut membuat langkah yang ketiga yaitu analisis karakteristik gangguan udara yang dihasilkan tiap *quadrotor* belum dapat dilakukan.

Langkah yang keempat adalah pengukuran respons pergerakan tiap *quadrotor* atas variasi *input* yang diberikan. Langkah tersebut masih belum dilakukan. Hal tersebut membuat langkah yang kelima yaitu analisis karakteristik sistem dinamik dari setiap *quadrotor* belum dapat dilakukan. Namun kita dapat menggunakan data

3.2.2 Set up *Quadrotor* dan Medan Terbang

Sistem yang akan dikontrol pada tugas akhir ini adalah *Quadrotor* Parrot AR.Drone 2.0. Berdasarkan hasil penelusuran, *quadrotor* ini memiliki empat mesin motor *brushless* 35.000rpm, daya 15W. *Quadrotor* ini memiliki baterai *lithium polymer battery* 3 sel, 11,1volt, 1000 mAh, dengan *discharge capacity* 10C dan waktu *charging* 90 menit. *Quadrotor* ini juga memiliki perangkat input yaitu SB 2.0 *high speed*, dan memiliki dua buah kamera. Kamera horizontal HD 720p 30fps dengan sensor CMOS, lensa lebar 90 derajat. Kamera vertikal QVGA 360p 30fps.

Pada Gambar 3.2.2 diberikan detail bentuk dari *quadrotor*. Pada gambar di sebelah kanan merupakan pengaturan *quadrotor* ketika terbang di medan *outdoor*. Pada gambar di sebelah kiri merupakan pengaturan *quadrotor* ketika terbang di medan *indoor*. Terlihat pengaturan *quadrotor* ketika terbang di medan *indoor* perlu diberikan pengaman.



Gambar 3.2.2 *Quadrotor* Parrot AR.Drone 2.0

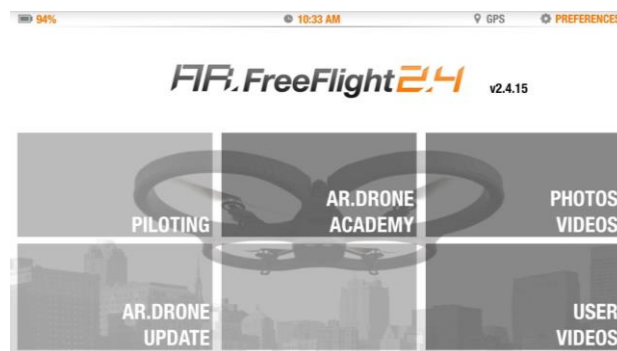
Pada langkah ini perlu dilakukan identifikasi fisik dari setiap *quadrotor* yang akan digunakan. Pihak penyedia fasilitas tugas akhir ini, yaitu Laboratorium CITA, memiliki beberapa persediaan *Quadrotor* Parrot AR.Drone 2.0. Namun, tidak seluruh *quadrotor* yang tersedia memiliki kondisi yang optimal untuk digunakan. Karena itu perlu dilakukan pemeriksaan kelengkapan komponen dan performa dari

setiap *quadrotor*. Kelengkapan komponen dan performa dari setiap *quadrotor* digambarkan pada Gambar 3.2.3 di bawah ini.



Gambar 3.2.3 Komponen pada *quadrotor* Parrot AR.Drone 2.0

Pemeriksaan dilakukan meliputi uji koneksi dengan *quadrotor*, uji sensor, uji kamera, uji mesin rotor untuk terbang dan uji ketahanan penggunaan baterai. uji koneksi dengan *quadrotor* dilakukan untuk melihat apakah *quadrotor* masih dapat dikontrol atau tidak. Uji mesin dilakukan untuk melihat apakah mesin rotor masih dapat berputar dengan baik atau tidak. Uji sensor ultrasonik dilakukan untuk melihat apakah *quadrotor* masih dapat membaca ketinggian dengan baik atau tidak. Uji kamera pada *quadrotor* dilakukan untuk mengetahui apakah *quadrotor* masih dapat menangkap citra dan mengirimkannya. Pengujian terbang dilakukan menggunakan aplikasi android komersial untuk Parrot AR.Drone 2.0 yaitu AR.FreeFlight seperti ditunjukkan pada Gambar 3.2.4 di bawah ini.



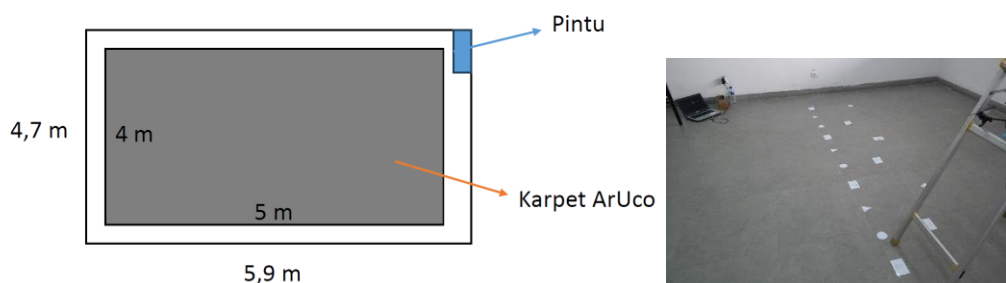
Gambar 3.2.4 Antarmuka aplikasi AR.FreeFlight

Berdasarkan hasil pemeriksaan pada delapan *set quadrotor*, didapatkan bahwa hampir semua baterai *quadrotor* sudah tidak layak digunakan lagi, dikarenakan

ketahanannya yang sangat mudah habis ketika digunakan menyuplai daya *quadrotor* untuk terbang. Didapatkan rata-rata lama terbang dengan baterai yang ada hanya sekitar 1-4 menit saja. Hampir semua *propeller* pada *quadrotor* juga mengalami deformasi atau perubahan bentuk sehingga memengaruhi performa *quadrotor* ketika terbang. Selain itu juga terdapat banyak set *quadrotor* yang mengalami kerusakan pada sensor dan juga kamera. Karena untuk tugas akhir ini hanya diperlukan tiga buah *quadrotor*, dipilih tiga buah *quadrotor* yang dianggap memiliki kelengkapan dan performa yang paling baik, yaitu *quadrotor* dengan label seri 018167, label seri 221751, dan label seri 018458.

Oleh karena itu dilakukan pembelian bagian *quadrotor* yang perlu diganti meliputi baterai LiPo 1500 mAh 11.1volt 3S, *charger* baterai LiPo B3 3S, konektor baterai LiPo, dan set *propeller*. Proses pembelian meliputi survei pasar *online* melalui *marketplace online* maupun secara *offline* dengan mengunjungi toko yang ada di sekitar Kota Bandung, kemudian mencari komponen dengan spesifikasi yang paling cocok untuk digunakan serta cara dan lama pengiriman komponen yang dibeli. Namun dalam prosesnya, terdapat kendala saat dalam proses pembelian barang. *Spare part quadrotor* sudah sangat jarang dijual sehingga memakan waktu yang lama dalam pencarian barang.

Sembari menunggu pembelian komponen *quadrotor*, dilakukan juga set up medan terbang. Uji terbang *quadrotor* dilakukan di salah satu ruangan laboratorium CITA sebagai tempat pengujian. Berdasarkan hasil pengukuran, dimensi ruangan berukuran 5.9m x 4.7m seperti yang ditunjukkan pada Gambar 3.2.5 di bawah ini.



Gambar 3.2.5 Medan terbang *quadrotor*

3.2.3 Set Up Aplikasi, *Library* dan Pemrograman

Objek utama yang digunakan pada tugas akhir ini adalah *quadrotor*. objek tersebut sebelumnya belum dipelajari dalam kegiatan perkuliahan, oleh karena itu dibutuhkan waktu lebih untuk memahami sistem dan cara kerja dari objek tersebut. Salah satu langkah yang dilakukan adalah dengan berkonsultasi dengan alumni. Dilakukan konsultasi dengan Alumni Pasca Sarjana Teknik Fisika ITB yaitu Giga Verian Pratama (23819304) yang sebelumnya pernah melakukan tesis menggunakan objek penelitian yang sama.

Pada sesi konsultasi tersebut dibahas mengenai teknis penggunaan *quadrotor*, karakteristik *quadrotor* secara kasar, hal-hal yang penting diketahui mengenai *quadrotor* dan program yang dapat digunakan. Didapatkan bahwa memang beberapa *quadrotor* yang ada memiliki kerusakan pada sebagian komponennya sehingga perlu dilakukan pemilihan *quadrotor* dengan kondisi terbaik. *Quadrotor* parrot AR.Drone 2.0 memiliki dua buah sarana komunikasi yang digunakan untuk fungsi yang berbeda. Untuk kontrol *quadrotor* menggunakan komunikasi TCP/IP, namun untuk pengiriman NavData dan MetaData digunakan protokol komunikasi UDP.

Quadrotor dapat terbang normalnya selama 10-12 menit. Performa *quadrotor* yang sebaiknya diambil untuk proses karakterisasi sistem adalah ketika baterai yang telah terisi penuh. Kemudian *quadrotor* akan diterbangkan tanpa melakukan *landing* kembali kecuali proses selesai. Hal ini disebabkan pada saat tersebut performa terbang yang diberikan *quadrotor* akan maksimal. Jika *quadrotor* dengan baterai yang baru diisi dilakukan *take off* kemudian *landing* kembali dan ketika dilakukan *take off* kembali karena baterai memang masih memiliki daya, performa terbang yang dihasilkan oleh *quadrotor* tidak sebaik saat *take off* pertama.

Untuk pemrograman, berdasarkan penelitian yang dilakukan oleh Pratama, G.V. (2021), telah mencoba beberapa *library* yang dapat digunakan untuk mengontrol *quadrotor* parrot AR.Drone 2.0 dalam bahasa Python. Didapatkan *library* yang terbaik untuk digunakan adalah PyArDrone. Pyardrone adalah sebuah *library* Python yang menyediakan antarmuka untuk mengontrol dan berinteraksi dengan

quadrotor secara program. Pyardrone dapat digunakan untuk membuat aplikasi Python yang dapat mengendalikan *quadrotor*, memperoleh data dari sensor *quadrotor*, serta melakukan berbagai tindakan lainnya seperti penerbangan otomatis, navigasi, dan pemrosesan data sensor. Kelebihan Library Pyardrone Python adalah fitur yang lengkap dan bervariasi sehingga mudah digunakan dibandingkan dengan *library* yang lain.

Sebelumnya, dilakukan *set up* aplikasi pada laptop agar dapat berinteraksi dengan *quadrotor* secara pemrograman. Digunakan aplikasi Microsoft Visual Studio Code dengan interpreter Python 3.10. Set up dilakukan dengan menyiapkan *virtual environment* serta meng-*install library* yang dibutuhkan, seperti yang ditunjukkan pada Gambar 3.2.6 di bawah ini.

```
PS D:\Kuliah\Tugas Akhir\Kak Giga\Program\Program Python\Final Program> pip install pyardrone
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)
Collecting pyardrone
  Downloading pyardrone-0.6.1.tar.gz (15 kB)
  Preparing metadata (setup.py) ... done
  Building wheels for collected packages: pyardrone
  Building wheel for pyardrone (setup.py) ... done
  Created wheel for pyardrone: filename=pyardrone-0.6.1-py3-none-any.whl size=20196 sha256=59805
  Stored in directory: c:\users\acer\appdata\local\pip\cache\wheels\c9\ad\45\957f78382f94589706
Successfully built pyardrone
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)
Installing collected packages: pyardrone
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)
Successfully installed pyardrone-0.6.1

[notice] A new release of pip available: 22.3.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS D:\Kuliah\Tugas Akhir\Kak Giga\Program\Program Python\Final Program> pip install pygame
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)
Collecting pygame
  Downloading pygame-2.5.2-cp310-cp310-win_amd64.whl (10.8 MB)
    10.8/10.8 MB 1.2 MB/s eta 0:00:00
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)
Installing collected packages: pygame
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -p (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\python310\lib\site-packages)
Successfully installed pygame-2.5.2
```

Gambar 3.2.6 Set up dan install library yang dibutuhkan

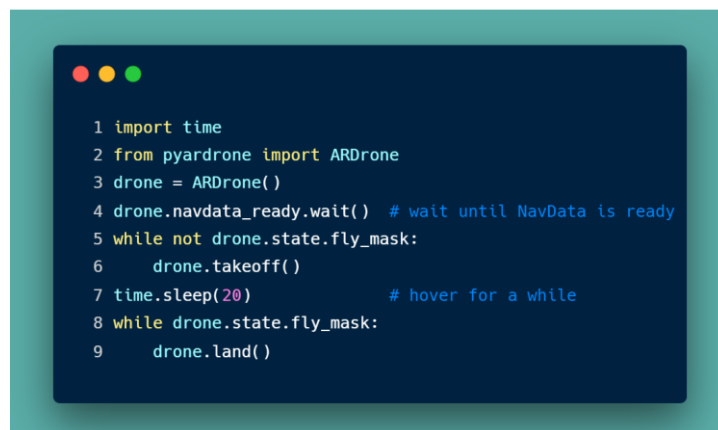
Setelah berhasil melakukan *set up* aplikasi dan *library*, berdasarkan penelitian yang dilakukan oleh Pratama, G.V. (2021), program-program yang telah dibuat dicoba untuk dijalankan. Hal ini bertujuan untuk mempelajari fungsi dan kegunaan tiap program yang telah disusun serta memahami manfaatnya untuk tugas akhir ini. Telah dibuat antarmuka untuk proses simulasi *quadrotor* yang dibuat menggunakan *library pygame*. Antarmuka tersebut dapat dikembangkan dan digunakan dalam pengerjaan tugas akhir ini kedepannya. Pada Gambar 3.2.7 merupakan detail antarmuka dari program yang telah dikembangkan sebelumnya.



Gambar 3.2.7 Antarmuka program simulasi yang dikembangkan

Dapat dilihat pada Gambar 3.2.7, antarmuka yang dihasilkan masih terlihat berantakan. Dengan demikian, akan dilakukan perbaikan tata letak dan penyempurnaan antarmuka tersebut pada penelitian tugas akhir ini.

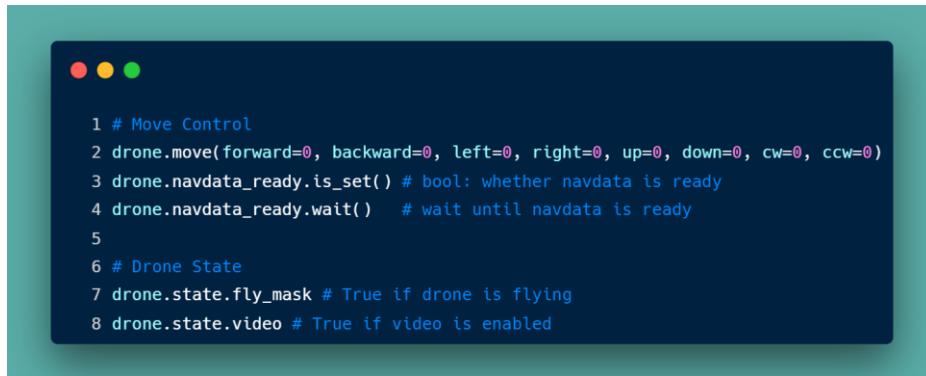
Karena fokus pengerjaan tugas akhir untuk saat ini adalah tahap Akuisisi dan Analisis data, maka dari seluruh program berdasarkan penelitian yang dilakukan oleh Pratama, G.V. (2021), akan dicari kode yang dapat dimanfaatkan untuk tahap tersebut. Setelah dipelajari, didapatkan kode dasar yang dapat digunakan untuk tahap Akuisisi dan Analisis data ini diberikan pada Gambar 3.2.8 di bawah ini.



Gambar 3.2.8 Program dasar untuk berinteraksi dengan *quadrotor*

Berdasarkan Gambar 3.2.8, kode tersebut merupakan program dasar yang digunakan untuk mengontrol *quadrotor* via laptop/komputer menggunakan *library* PyArDrone. Pada baris pertama dan kedua dilakukan *import* dependensi yang diperlukan. Baris ketiga merupakan inisiasi objek *quadrotor*. Pada baris yang

keempat inisiasi untuk pengiriman NavData dari *quadrotor*. Pada baris kelima dan keenam, bila *state* dari *quadrotor* tidak sedang terbang maka *quadrotor* dapat *take off* dengan *command* `drone.takeoff()`. Pada baris ketujuh diberikan waktu untuk *quadrotor* terbang. Pada baris kedelapan dan ke sembilan, bila *quadrotor* sedang dalam *state* terbang, maka *quadrotor* dapat dibuat *landing* dengan *command* `drone.land()`.

A screenshot of a code editor window with a dark blue background and light blue text. The code is as follows:

```
1 # Move Control
2 drone.move(forward=0, backward=0, left=0, right=0, up=0, down=0, cw=0, ccw=0)
3 drone.navdata_ready.is_set() # bool: whether navdata is ready
4 drone.navdata_ready.wait()   # wait until navdata is ready
5
6 # Drone State
7 drone.state.fly_mask # True if drone is flying
8 drone.state.video # True if video is enabled
```

Gambar 3.2.9 Program Dasar untuk mengontrol dan mengetahui state *quadrotor*

Berdasarkan Gambar 3.2.9, kode tersebut merupakan program dasar yang dapat digunakan untuk mengontrol pergerakan dari *quadrotor*. Dari program tersebut, dapat diketahui *state* dari *quadrotor* saat ini. Baris kesatu hingga keempat merupakan bagian untuk mengontrol pergerakan *quadrotor*, sedangkan baris keenam hingga kedelapan merupakan bagian untuk mengetahui *state* dari *quadrotor*. Sebenarnya, terdapat banyak *state* dari *quadrotor* yang dapat diambil oleh *user*. Namun, untuk pengerjaan tahap akuisisi dan analisis data, *state* terbang dan kamera dinilai sudah cukup untuk digunakan. Berdasarkan kode pada baris kedua, kecepatan *quadrotor* akan bernilai minimum nol dan bernilai satu untuk kecepatan gerak maksimumnya.

3.2.4 Uji Terbang

Berdasarkan program dasar yang telah dijelaskan sebelumnya kemudian dilakukan uji terbang dengan menggunakan program tersebut. Tes terbang dilakukan untuk mengetahui apakah *quadrotor* dapat terhubung dengan laptop melalui kode yang dibuat atau tidak. Pada Gambar 3.2.10 di bawah merupakan dokumentasi dari tes terbang yang dilakukan pada sebuah *quadrotor*. Berdasarkan hasil tes, *quadrotor*

terbukti dapat melakukan koneksi Wi-Fi dengan laptop dan dapat menerima perintah dari laptop.

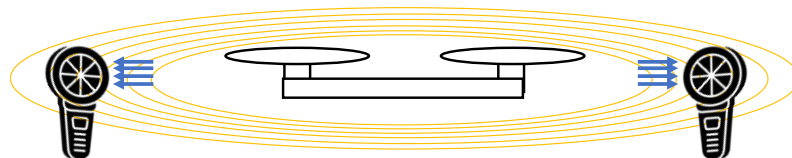


Gambar 3.2.10 Dokumentasi uji terbang menggunakan pemrograman

Setelah dapat berinteraksi dengan *quadrotor* melalui pemrograman, dapat dimulai pula langkah untuk akuisisi data. Terdapat dua *set* akuisisi data yang perlu dilakukan, yaitu pengukuran gangguan udara menggunakan anemometer pada variasi radius dari *quadrotor* dan pengukuran respons pergerakan tiap *quadrotor* atas variasi *input* yang diberikan.

3.2.5 Pengukuran Gangguan Udara

Pengukuran gangguan udara, baru dilakukan pada satu buah *quadrotor* dan beberapa variasi radius. Dilakukan pengukuran menggunakan *quadrotor* dengan nomor seri 018167 dengan menggunakan anemometer digital. Pengukuran coba dilakukan dengan skema seperti pada Gambar 3.2.11.

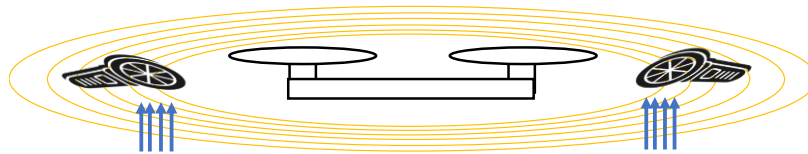


Gambar 3.2.11 Skema pertama pengukuran gangguan angin

Anemometer digital mengukur kecepatan angin dengan arah tegak lurus terhadap *trust* putaran *propeller* dari *quadrotor*. Didapatkan hasil pengukuran dengan menggunakan anemometer digital seperti pada Tabel 3.2.1 berikut ini.

Tabel 3.2.1 Hasil pengukuran kecepatan angin dengan skema pertama

r	Kiri	Kanan
10 cm	0 m/s	0 m/s
20 cm	0 m/s	0 m/s
30 cm	0 m/s	0 m/s
40 cm	0 m/s	0 m/s
50 cm	0 m/s	0 m/s
60 cm	0 m/s	0 m/s
70 cm	0 m/s	0 m/s
80 cm	0 m/s	0 m/s
90 cm	0 m/s	0 m/s
100 cm	0 m/s	0 m/s



Gambar 3.2.12 Skema kedua pengukuran gangguan angin

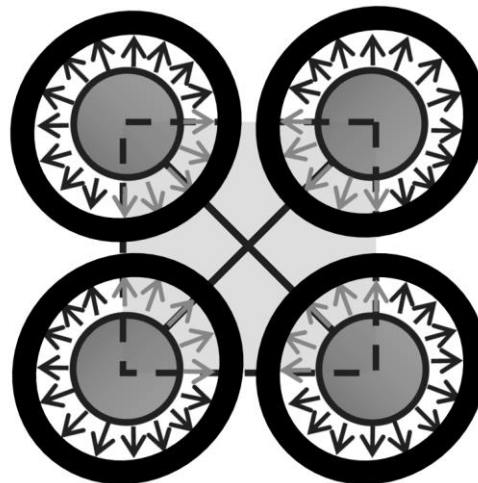
Selanjutnya, pengukuran coba dilakukan dengan skema pada Gambar 3.2.12. Anemometer digital mengukur kecepatan angin dengan arah sejajar terhadap *trust* putaran *propeller* dari *quadrotor*. Didapatkan hasil pengukuran dengan menggunakan anemometer digital seperti pada Tabel 3.2.2 berikut ini.

Tabel 3.2.2 Hasil pengukuran kecepatan angin dengan skema kedua

r	Kiri	Kanan
10 cm	15.9 m/s	20 m/s
20 cm	13.8 m/s	12.7 m/s
30 cm	7.4 m/s	8.6 m/s
40 cm	4.2 m/s	2.8 m/s
50 cm	1.5 m/s	1.9 m/s
60 cm	0 m/s	0 m/s
70 cm	0 m/s	0 m/s
80 cm	0 m/s	0 m/s
90 cm	0 m/s	0 m/s

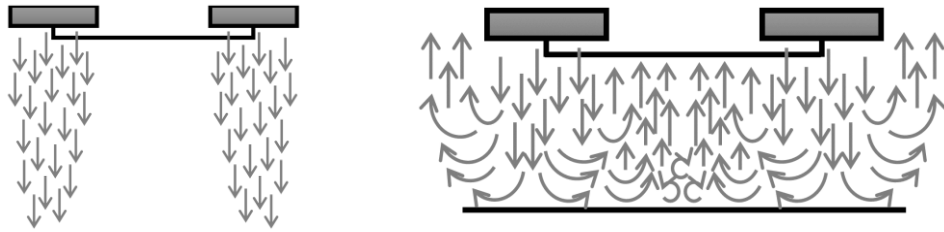
r	Kiri	Kanan
100 cm	0 m/s	0 m/s

Pengukuran kecepatan angin dengan arah tegak lurus terhadap *trust* putaran *propeller* dari *quadrotor* menghasilkan nilai nol pada semua radius pengukuran. Hal tersebut dapat dikarenakan kecepatan angin yang cukup kecil untuk dibaca oleh anemometer digital. Selain itu *quadrotor* menggunakan pelindung *propeller* yang menyelimuti *propeller* secara horizontal di sekitar *propeller*. Sesuai Gambar 3.2.13, hal tersebut dapat menyebabkan udara yang tertahan atau berbelok karena adanya pelindung tersebut. Pengukuran kecepatan angin dengan arah sejajar terhadap *trust* putaran *propeller* dari *quadrotor* menghasilkan variasi nilai pada beberapa radius pengukuran. Namun terlihat pola penurunan kecepatan angin dengan semakin besar radius pengukuran.



Gambar 3.2.13 Dinamika udara dari *quadrotor* pada arah horizontal

Berdasarkan kedua hasil tersebut, gangguan udara berasal dari turbulensi udara yang memantul dari *ground*, bukan secara langsung dari *propeller*. Hal tersebut sesuai dengan penelitian yang dilakukan oleh Nekoo dkk. (2021), turbulensi udara paling besar terdapat pada arah sejajar dengan *trust* putaran *quadrotor*. Pada Gambar 3.2.14 diperlihatkan ilustrasi turbulensi udara saat *quadrotor* sedang terbang. Saat Turbulensi udara tersebut dipantulkan oleh *ground* maka terdapat udara pantulan yang dapat mengganggu *quadrotor* terbang di sekitarnya.

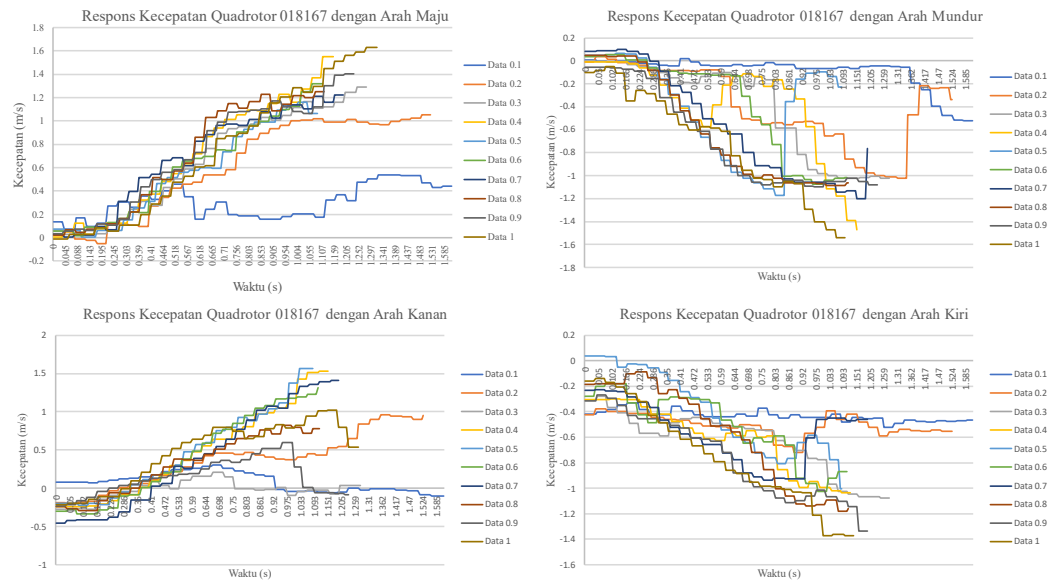


Gambar 3.2.14 Turbulensi udara yang dihasilkan *quadrotor*

3.2.6 Pengukuran Respons Pergerakan *Quadrotor*

Pengukuran respons pergerakan tiap *quadrotor* belum dilakukan untuk saat ini. Namun berdasarkan penelitian yang dilakukan oleh Pratama, G.V. (2021), telah dilakukan pengukuran respons pergerakan tiap *quadrotor* yang sama dengan yang dipakai pada tugas akhir ini. Pengukuran respons pergerakan tiap *quadrotor* dapat didekati melalui identifikasi pergerakan dengan *First Order Plus Time Delay* (FOPTD).

Pada pemrograman kontrol pergerakan yang dijelaskan pada bagian *Set Up Aplikasi, Library* dan Pemrograman sebelumnya, kecepatan gerak *quadrotor* dapat divariasikan nol hingga satu. Pada penelitian sebelumnya dilakukan karakterisasi respons pergerakan *quadrotor* mulai dari *input* kecepatan 0,1 hingga 1 dengan delta 0,1. Pada Gambar 3.2.15 ditunjukkan hasil pengukuran respons pergerakan *quadrotor* dengan nomor seri 018167 sebelumnya.



Gambar 3.2.15 Hasil pengukuran respons kecepatan *quadrotor* 018167 berbagai arah. Setelah itu dilakukan *fitting* data dengan model FOPTD untuk setiap arah dan setiap *input* kecepatan. Didapatkan hasil parameter FOPTD untuk berturut-turut arah pergerakan maju, mundur, kiri, dan kanan diberikan pada Tabel 3.2.3, Tabel 3.2.4, Tabel 3.2.5, dan Tabel 3.2.6.

Tabel 3.2.3 Hasil parameter FOPTD untuk arah maju

Speed Input	Gain K	Tau	TD
0.1	6.093	1.088	0.000
0.2	7.123	0.601	0.241
0.3	5.455	0.613	0.271
0.4	6.765	1.186	0.232
0.5	4.511	1.168	0.195
0.6	3.889	1.061	0.213
0.7	2.028	0.486	0.142
0.8	1.793	0.313	0.251
0.9	1.924	0.638	0.204
1	4.618	2.330	0.217
rata-rata	4.420	0.948	0.197
minimum	1.793	0.313	0.000
maksimum	7.123	2.330	0.271

Tabel 3.2.4 Hasil parameter FOPTD untuk arah mundur

<i>Speed Input</i>	<i>Gain K</i>	Tau	TD
0.1	33.207	6.122	1.019
0.2	13.302	1.947	0.389
0.3	17.571	2.605	0.518
0.4	44.289	3.768	0.754
0.5	3.146	0.371	0.301
0.6	1.865	0.131	0.549
0.7	1.514	0.216	0.322
0.8	1.462	0.245	0.282
0.9	1.237	0.235	0.238
1	608.443	381.968	0.090
rata-rata	72.604	39.761	0.446
minimum	1.237	0.131	0.090
maksimum	608.443	381.968	1.019

Tabel 3.2.5 Hasil parameter FOPTD untuk arah kanan

<i>Speed Input</i>	<i>Gain K</i>	Tau	TD
0.1	0.000	0.001	0.001
0.2	2.656	0.280	0.356
0.3	0.000	0.001	0.001
0.4	11.160	1.815	0.336
0.5	8.642	1.852	0.371
0.6	8.550	2.086	0.347
0.7	3.386	0.748	0.442
0.8	1.389	0.551	0.336
0.9	0.245	0.001	0.362
1	0.837	0.191	0.304
rata-rata	3.686	0.753	0.286
minimum	0.000	0.001	0.001
maksimum	11.160	2.086	0.442

Tabel 3.2.6 Hasil parameter FOPTD untuk arah kiri

<i>Speed Input</i>	<i>Gain K</i>	Tau	TD
0.1	4.720	0.003	0.001
0.2	2.479	0.004	0.000
0.3	4.384	0.910	0.000
0.4	3.388	0.838	0.000

<i>Speed Input</i>	<i>Gain K</i>	Tau	TD
0.5	2.652	0.651	0.233
0.6	5.000	2.758	0.000
0.7	0.956	0.204	0.000
0.8	2.760	1.043	0.181
0.9	2.033	0.998	0.000
1	3.129	1.986	0.000
rata-rata	3.150	0.939	0.042
minimum	0.956	0.003	0.000
maksimum	5.000	2.758	0.233

Berdasarkan hasil data yang ditunjukkan di atas, untuk *input* kecepatan lebih dari 0.5 tampaknya masih belum relevan. Pada Gambar 3.2.15 *input* kecepatan lebih dari 0,5 masih berada pada keadaan *transien* belum mencapai keadaan tunak. Hal tersebut menyebabkan hasil parameter FOPTD yang didapatkan untuk *input* kecepatan lebih dari 0,5 memiliki variasi yang berbeda-beda.

Masalah tersebut disebabkan keterbatasan panjang lintasan yang digunakan pada penelitian sebelumnya, sehingga *quadrotor* tidak dapat mencapai kecepatan tunaknya. Untuk itu, pada penelitian tugas akhir ini perlu dilakukan pengukuran respons pergerakan *quadrotor* dengan lintasan yang lebih panjang untuk memastikan pergerakan *quadrotor* mencapai keadaan tunak. Data karakteristik tersebut, dapat digunakan sebagai acuan dalam melakukan pengukuran respons pergerakan tiap *quadrotor* kedepannya. Selain itu data tersebut juga dapat digunakan untuk data awal dalam mencoba simulasi. Sampai saat ini masih dilakukan proses pembelajaran simulasi simulink dengan memanfaatkan data karakteristik tersebut.

BAB IV

RENCANA SELANJUTNYA DAN PENUTUP

4.1 Rencana Selanjutnya

Rencana selanjutnya untuk proses pengerjaan tugas akhir, akan tetap mengerjakan tahap pertama yaitu tahap akuisisi dan analisis data dan mulai mempelajari langkah awal pengerjaan tahap kedua yaitu simulasi. Pengukuran gangguan udara menggunakan anemometer akan dilakukan pada dua buah *quadrotor* yang lain. Pengukuran respons pergerakan *quadrotor* dengan lintasan yang lebih panjang untuk memastikan pergerakan *quadrotor* mencapai keadaan tunak. Selanjutnya melakukan karakterisasi sistem dinamik untuk setiap *quadrotor* dengan variasi *input* yang diberikan. Pada pengerjaan tahap kedua, mulai mempelajari simulasi simulink dengan memanfaatkan data karakteristik *quadrotor* penelitian sebelumnya.

4.2 Penutup

Demikian Laporan Kemajuan penelitian tugas akhir dengan judul “Perancangan Sistem Kontrol Pergerakan Untuk Mengatasi Pengaruh Gangguan Udara Dalam Formasi *Multiquadrotor*” telah disusun berdasarkan kegiatan penelitian yang telah dilakukan penulis selama satu semester dan berisi rencana penelitian selanjutnya.

Dengan demikian, Laporan Kemajuan ini diharapkan dapat memberikan gambaran terkait pendahuluan yang berisi latar belakang, rumusan masalah, hingga metode penelitian ditulis berdasarkan studi literatur yang sudah dilakukan oleh penulis dari beberapa jurnal, tugas akhir dan tesis. Laporan Kemajuan ini pun diharapkan dapat menjelaskan terkait metodologi dan metode yang dilakukan pada penelitian tugas akhir. Oleh karena itu, penulis berharap agar mendapatkan kritik dan saran terhadap penelitian yang telah dilakukan ataupun rencana penelitian selanjutnya.

DAFTAR PUSTAKA

- [1] K. Martin, B. Stefan, Y. Kaloyan, C. Denis, Kambushev, and C. Nayden, "Influence of atmospheric turbulence on the control of flying robotics systems," 2019.
- [2] H. H. Hady, M. N. Elya, A. B. Azrena, and S. N. Makhtar, "Development of *Quadrotor* Control under Wind Disturbance," Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ISMSIT52890.2021.9604649.
- [3] P. Aleix, T. L. Brett, and P. H. Jonathan, "Dynamic Landing of an Autonomous *Quadrotor* on a Moving Platform in Turbulent Wind Conditions," Paris, Perancis, Aug. 2020.
- [4] G. Valerian, "Kontrol Formasi *Quadrotor* Menggunakan Metode Cyclic Leader Follower SFM Sebagai Media Pelacakan Lokasi Tesis," Bandung, 2022.
- [5] S. R. Nekoo, J. A. Acosta, G. Heredia, and A. Ollero, "Soft-Landing of Multi-Rotor Drones using a Robust Nonlinear Control and Wind Modeling," in *2021 International Conference on Unmanned Aircraft Systems, ICUAS 2021*, Institute of Electrical and Electronics Engineers Inc., Jun. 2021, pp. 1070–1079. doi: 10.1109/ICUAS51884.2021.9476763.
- [6] G. Regula and B. Lantos, "Formation Control of a Large Group of UAVs with Safe Path Planning," Budapest, 2013.
- [7] W. Zhikai, H. Deqing, H. Tianpeng, and Q. Na, "Active disturbance rejection control for a *quadrotor* UAV," Chengdu, Nov. 2020.
- [8] F. Mukhlis, E. Ekawati, M. Fakhruroji, and I. A. Prakosa, "Compensating Wind Disturbance on Gesture Commanded *Quadrotors* Flight," 2015.
- [9] M. J. Kamran, D. Hazry, S. A. Faiz, M. T. Hassan, A. W. Faizan, and Hussain, "Altitude and Horizontal Motion Control of *Quadrotor* UAV in the Presence of Air Turbulence," Dec. 2013.
- [10] M. Dounia, B. Z. Abdeslem, and H. Ramdane, "Trajectory Tracking Performance with two Nonlinear Controllers of *Quadrotor* under Wind Effect," in *Proceedings of the International Conference on Advanced Systems and Emergent Technologies, IC_ASET 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 50–55. doi: 10.1109/IC_ASET49463.2020.9318319.

- [11] MathWorks, “Dryden Wind Turbulence Model (Continuous),” <https://www.mathworks.com/help/aeroblks/drydenwindturbulencemodelcontinuous.html>.
- [12] P. Abichandani, D. Lobo, G. Ford, D. Bucci, and M. Kam, “Wind Measurement and Simulation Techniques in Multi-Rotor Small Unmanned Aerial Vehicles,” *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.2977693.
- [13] D. Kuchеров, A. Kozub, O. Sushchenko, and R. Skrynkovskyy, “Stabilizing the spatial position of a *quadrotor* by the backstepping procedure,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 1188–1199, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp1188-1199.
- [14] R. Nazir, A. A. Iaghari, K. Kumar, S. David, and M. Ali, “Survey on Wireless Network Security,” *Archives of Computational Methods in Engineering*, vol. 29, no. 3. Springer Science and Business Media B.V., pp. 1591–1610, May 01, 2021. doi: 10.1007/s11831-021-09631-5.
- [15] B. A. Forouzan, *TCP/IP Protocol Suite*, 3rd ed. McGraw-Hill, Inc., 2005.
- [16] F. Taha Al-Dhief *et al.*, “Performance Comparison between TCP and UDP Protocols in Different Simulation Scenarios,” 2018. [Online]. Available: <https://www.researchgate.net/publication/329698255>
- [17] R. Khandait, V. Kumar, V. Bhurse, V. Tiwari, and S. Khubalkar, “Quadcopter Control using Different Controllers,” in *2022 International Conference on Intelligent Controller and Computing for Smart Power, ICICCSP 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICICCSP53532.2022.9862416.
- [18] T. Huang and D. Huang, “Backstepping control for a *quadrotor* unmanned aerial vehicle,” in *Proceedings - 2020 Chinese Automation Congress, CAC 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 2475–2480. doi: 10.1109/CAC51589.2020.9326765.
- [19] M. Karahan and C. Kasnakoglu, “Nonlinear Modeling and Robust Backstepping Control of a *Quadrotor* Unmanned Aerial Vehicle,” in *2022 5th International Conference on Computer and Informatics Engineering, IC2IE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 94–99. doi: 10.1109/IC2IE56416.2022.9970136.
- [20] B. Xu and K. Mou, “Adaptive Feed-Forward Control for Seeker Based on the Disturbance Observer,” in *Proceedings of 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC 2019)*, 2019.

- [21] V. A. Budnyaev, I. F. Filippov, V. V. Vertegel, and S. Y. Dudnikov, “Simulink-based quadcopter control system model,” in *2020 1st International Conference Problems of Informatics, Electronics, and Radio Engineering, PIERE 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 246–250. doi: 10.1109/PIERE51041.2020.9314676.