

**TUGAS AKHIR - EE184801**

**KONTROL FORMASI KOOPERATIF DAN  
PENGHINDARAN RINTANGAN PADA MULTIPLE  
UNMANNED AERIAL VEHICLE DENGAN GUIDANCE  
ROUTE DAN ARTIFICIAL POTENTIAL FIELD**

**VINCENTIUS CHARLES MAYNAD**

**NRP 07111940000038**

Dosen Pembimbing

**Mochammad Sahal, S.T., M.Sc.**

**NIP 197011191998021002**

**Yusuf Bilfaqih, S.T., M.T.**

**NIP 197203251999031001**

**Program Studi Sarjana Teknik Elektro**

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2022



**TUGAS AKHIR - EE184801**

**KONTROL FORMASI KOOPERATIF DAN  
PENGHINDARAN RINTANGAN PADA MULTIPLE  
UNMANNED AERIAL VEHICLE DENGAN GUIDANCE  
ROUTE DAN ARTIFICIAL POTENTIAL FIELD**

**VINCENTIUS CHARLES MAYNAD**

**NRP 07111940000038**

Dosen Pembimbing

**Mochammad Sahal, S.T., M.Sc.**

**NIP 197011191998021002**

**Yusuf Bilfaqih S.T., M.T.**

**NIP 197203251999031001**

**Program Studi Sarjana Teknik Elektro**

Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2022



**FINAL PROJECT - EE184801**

**COOPERATIVE FORMATION AND OBSTACLE  
AVOIDANCE CONTROL ON MULTIPLE UNMANNED  
AERIAL VEHICLE WITH GUIDANCE ROUTE AND  
ARTIFICIAL POTENTIAL FIELD**

**VINCENTIUS CHARLES MAYNAD**

**NRP 07111940000038**

Advisor

**Mochammad Sahal, S.T., M.Sc.**

**NIP 197011191998021002**

**Yusuf Bilfaqih S.T., M.T.**

**NIP 197203251999031001**

**Study Program Bachelor of Electrical Engineering**

Department of Electrical Engineering

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2022

## LEMBAR PENGESAHAN

### KONTROL FORMASI KOOPERATIF DAN PENGHINDARAN RINTANGAN PADA MULTIPLE UNMANNED AERIAL VEHICLE DENGAN GUIDANCE ROUTE DAN ARTIFICIAL POTENTIAL FIELD

#### TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat  
memperoleh gelar Sarjana Teknik pada  
Program Studi S-1 Teknik Elektro  
Departemen Teknik Elektro  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh : **VINCENTIUS CHARLES MAYNAD**  
NRP. 07111940000038

Disetujui oleh Tim Penguji Tugas Akhir :

1. Mochammad Sahal S.T., M.Sc.

Pembimbing



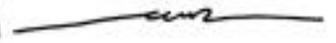
2. Yusuf Bilfaqih S.T., M.T.

Ko-pembimbing



3. Prof.Ir.H.Abdullah Alkaff, M.Sc, Ph.D.

Penguji



4. Zulkifli Hidayat, S.T., M.Sc.

Penguji



5. Mohamad Abdul Hady, S.T., M.T.

Penguji



**SURABAYA**

**Desember, 2022**

*Halaman ini sengaja dikosongkan*

## APPROVAL SHEET

### COOPERATIVE FORMATION AND OBSTACLE AVOIDANCE CONTROL ON MULTIPLE UNMANNED AERIAL VEHICLE WITH GUIDANCE ROUTE

#### FINAL PROJECT

Submitted to fulfill one of the requirements  
for obtaining an undergraduate degree at  
Undergraduate Study Program of Electrical Engineering  
Department of Electrical Engineering  
Faculty of Intelligent Electrical and Informatics Technology  
Institut Teknologi Sepuluh Nopember

By: **VINCENTIUS CHARLES MAYNAD**

NRP. 07111940000038

Approved by Final Project Examiner Team:

1. Mochammad Sahal S.T., M.Sc.

Advisor



2. Yusuf Bilfaqih S.T., M.T.

Co-Advisor



3. Prof.Ir.H.Abdullah Alkaff, M.Sc, Ph.D.

Examiner



4. Zulkifli Hidayat, S.T., M.Sc.

Examiner



5. Mohamad Abdul Hady, S.T., M.T.

Examiner



**SURABAYA**

**December, 2022**

*Halaman ini sengaja dikosongkan*

## PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP : Vincentius Charles Maynad / 07111940000038  
Program studi : Teknik Elektro  
Dosen Pembimbing / NIP : Mochammad Sahal, S.T., M.Sc. / 197011191998021002  
Yusuf Bilfaqih S.T., M.T. / 197203251999031001

dengan ini menyatakan bahwa Tugas Akhir dengan judul “Kontrol Formasi Kooperatif dan Penghindaran Rintangan pada *Multiple Unmanned Aerial Vehicle* dengan *Guidance Route* dan *Artificial Potential Field*” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

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

Surabaya, 13 Desember 2022

Mengetahui  
Dosen Pembimbing



Mochammad Sahal, ST., M.Sc.  
NIP. 197011191998021002

Mengetahui  
Dosen Ko-Pembimbing



Yusuf Bilfaqih S.T., M.T.  
NIP. 197203251999031001

Mahasiswa



Vincentius Charles Maynad  
NRP. 07111940000038

*Halaman ini sengaja dikosongkan*

## ABSTRAK

### KONTROL FORMASI KOOPERATIF DAN PENGHINDARAN RINTANGAN PADA MULTIPLE UNMANNED AERIAL VEHICLE DENGAN GUIDANCE ROUTE DAN ARTIFICIAL POTENTIAL FIELD

Nama Mahasiswa / NRP : Vincentius Charles Maynad / 0711194000038  
Departemen : Teknik Elektro FTEIC - ITS  
Dosen Pembimbing : 1. Mochammad Sahal S.T, M.Sc.  
2. Yusuf Bilfaqih S.T., M.T.

#### Abstrak

Dalam beberapa tahun terakhir, kontrol kooperatif sistem multi-UAV (*Unmanned Aerial Vehicle*) telah menjadi topik penelitian yang hangat di bidang kontrol penerbangan. Diantaranya, pengendalian formasi dan penghindaran rintangan adalah salah dua tema yang penting untuk diteliti karena kompleksitas kondisi permasalahan yang ingin diselesaikan selalu meningkat seiring waktu. Problema riil ini dapat dimodelkan sebagai permasalahan kontrol penghindaran rintangan pada formasi *quadcopter*. Sekelompok *quadcopter* ditugaskan untuk membentuk formasi (berupa bentuk V), bergerak dalam formasi menuju titik tujuan, menghindari tabrakan antar robot, dan menghindari tabrakan dengan rintangan. Model *quadcopter* yang digunakan adalah Quanser Qdrone dengan enam derajat kebebasan. *Quadcopter* dikontrol menggunakan *fuzzy state feedback controller* untuk melacak tujuan. Pada tugas akhir ini dirancang suatu sistem pengaturan formasi menggunakan pendekatan *guidance route* dengan penghindaran rintangan menggunakan metode *Artificial Potential Field* (APF). Selain itu, akan dibandingkan dua strategi penghindaran, penghindaran total dan penghindaran minimal. Berdasarkan hasil simulasi, algoritma kontrol yang dikembangkan berhasil melaksanakan tugas pengaturan formasi dan penghindaran rintangan pada sekelompok *quadcopter*. Hal ini dibuktikan dengan rata-rata indeks performansi formasi bernilai 0.800025 untuk strategi penghindaran total dan 1.2227125 untuk strategi penghindaran minimal serta trayektori masing-masing *quadcopter* yang bebas tabrakan.

**Kata kunci:** *Artificial Potential Field, Formation Control, Guidance Route, Obstacle Avoidance*

*Halaman ini sengaja dikosongkan*

## ABSTRACT

### COOPERATIVE FORMATION AND OBSTACLE AVOIDANCE CONTROL ON MULTIPLE UNMANNED AERIAL VEHICLE WITH GUIDANCE ROUTE AND ARTIFICIAL POTENTIAL FIELD

**Student Name / NRP** : Vincentius Charles Maynad / 07111940000038  
**Department** : Teknik Elektro FTEIC - ITS  
**Advisor** : 1. Mochammad Sahal S.T, M.Sc.  
2. Yusuf Bilfaqih S.T., M.T.

#### Abstract

In recent years, cooperative control of multi-UAV (Unmanned Aerial Vehicle) systems has become a hot research topic in the field of flight control. Among them, formation control and obstacle avoidance are two important themes to study because the complexity of the problem conditions to be solved always increases with time. This real problem can be modeled as an obstacle avoidance control problem in a quadcopter formation. A group of quadcopters is assigned to form a formation (in the form of a V shape), move in formation towards a destination point, avoid collisions between robots, and avoid collisions with obstacles. The quadcopter model used is the Quanser Qdrone with six degrees of freedom. The quadcopter is controlled using a fuzzy state feedback controller to track objectives. In this final project, a formation management system is designed using the guidance route approach with obstacle avoidance using the Artificial Potential Field (APF) method. Moreover, two avoidance strategies will be compared, total avoidance and minimum avoidance. Based on the simulation results, the developed control algorithm successfully performs the task of setting formation and obstacle avoidance on a group of quadcopters. This is evidenced by the average formation performance index of 0.800025 for total avoidance strategy and 1.2227125 for minimum avoidance strategy with the collision-free trajectories of each quadcopter.

**Keywords:** *Artificial Potential Field, Formation Control, Guidance Route, Obstacle Avoidance*

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas rahmat-Nya, penulis dapat menyelesaikan penelitian dan tugas akhir dengan judul “Kontrol Formasi Kooperatif dan Penghindaran Rintangan pada *Multiple Unmanned Aerial Vehicle* dengan *Guidance Route* dan *Artificial Potential Field (Cooperative Formation and Obstacle Avoidance Control on Multiple Unmanned Aerial Vehicle with Guidance Route and Artificial Potential Field)*” dengan baik.

Selama menyelesaikan penelitian dan tugas akhir ini, terdapat banyak hambatan dan rintangan yang penulis hadapi. Tetapi, banyak pula pihak yang membantu penulis dalam menyelesaikan berbagai permasalahan-permasalahan tersebut. Dengan adanya dukungan mereka, penulis dapat terus bergerak maju dalam menyelesaikan laporan tugas akhir ini. Sehingga, penulis ingin menyampaikan terima kasih kepada:

1. Tuhan Yang Maha Esa yang selalu memberikan kekuatan, kesehatan, kelancaran, dan kemudahan bagi hamba-Nya.
2. Keluarga penulis yang selalu memberikan dukungan, doa, dan semangat kepada penulis.
3. Bapak Mochammad Sahal, ST., M.Sc dan Yusuf Bilfaqih S.T., M.T. selaku dosen pembimbing I dan II yang telah memberikan banyak ilmu, saran, dan motivasi dalam pengerjaan tugas akhir ini.
4. Prof.Ir.H.Abdullah Alkaff, M.Sc, Ph.D., Zulkifli Hidayat, S.T., M.Sc., dan Mohamad Abdul Hady, S.T., M.T. selaku dosen penguji yang telah memberikan banyak saran terhadap penelitian tugas akhir ini.
5. Keluarga besar asisten Laboratorium Kontrol dan Otomasi serta asisten Laboratorium Sistem dan Sibernetika yang selalu mendukung, menemani, dan membantu penulis dalam menyelesaikan laporan tugas akhir ini.
6. Teman – teman e59 yang senantiasa memberikan dukungan dan motivasi.
7. Semua pihak yang tidak dapat ditulis secara satu-persatu yang telah membantu dalam menyelesaikan pengerjaan tugas akhir ini.

Dalam menyelesaikan tugas akhir ini, penulis sadar bahwa hasil yang telah dikerjakan masih belum sempurna dan masih banyak hal yang bisa diperbaiki. Saran, kritik, dan masukan dari seluruh pihak sangat diharapkan penulis untuk kemungkinan mengembangkan penelitian ini lebih lanjut.

**Surabaya, Desember 2022**

**Penulis**

*Halaman ini sengaja dikosongkan*

## DAFTAR ISI

LEMBAR PENGESAHAN .....	i
APPROVAL SHEET .....	iii
PERNYATAAN ORISINALITAS .....	v
ABSTRAK.....	vii
ABSTRACT .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xiv
DAFTAR TABEL.....	xvii
DAFTAR SIMBOL .....	xviii
BAB 1 PENDAHULUAN.....	19
1.1 Latar Belakang.....	19
1.2 Rumusan Masalah .....	20
1.3 Batasan Masalah .....	20
1.4 Tujuan.....	20
1.5 Manfaat.....	20
BAB 2 TINJAUAN PUSTAKA .....	21
2.1 Hasil Penelitian Terdahulu .....	21
2.2 Dasar Teori .....	22
BAB 3 METODOLOGI .....	39
3.1 Metode yang digunakan.....	39
3.2 Bahan dan peralatan yang digunakan .....	40
3.3 Urutan pelaksanaan penelitian.....	40
BAB 4 Hasil dan Pembahasan .....	51
4.1 Hasil penelitian .....	51
4.2 Pembahasan .....	71
BAB 5 Kesimpulan dan Saran .....	75
5.1 Kesimpulan.....	75
5.2 Saran .....	75
DAFTAR PUSTAKA .....	77
LAMPIRAN.....	81
BIODATA PENULIS .....	93

## DAFTAR GAMBAR

Gambar 2.1 Graf Tidak Berarah Dengan 5 Simpul dan 6 Busur.....	23
Gambar 2.2 (a) Graf Terhubung, (b) Graf Tak-terhubung.....	23
Gambar 2.3 Graf Lengkap $K_4$ .....	24
Gambar 2.4 Konfigurasi <i>Quadcopter</i> .....	24
Gambar 2.5 Ilustrasi Konsep <i>Thrust</i> dari <i>Quadcopter</i> .....	25
Gambar 2.6 Ilustrasi Konsep <i>Roll</i> dari <i>Quadcopter</i> .....	25
Gambar 2.7 Ilustrasi Konsep <i>Pitch</i> dari <i>Quadcopter</i> .....	25
Gambar 2.8 Ilustrasi Konsep <i>Yaw</i> dari <i>Quadcopter</i> .....	26
Gambar 2.9 Ilustrasi Gerak <i>Roll</i> pada <i>Quadcopter</i> .....	26
Gambar 2.10 Ilustrasi Gerak <i>Pitch</i> pada <i>Quadcopter</i> .....	27
Gambar 2.11 Ilustrasi Gerak <i>Yaw</i> pada <i>Quadcopter</i> .....	28
Gambar 2.12 Skema Kerja Logika <i>Fuzzy</i> .....	33
Gambar 2.13 Ilustrasi Cara Kerja Model Sugeno.....	34
Gambar 2.14 Ilustrasi Pengaturan Formasi Berdasarkan Posisi.....	34
Gambar 2.15 Ilustrasi GRFC.....	35
Gambar 2.16 Ilustrasi dari GRFC yang Distributif.....	36
Gambar 2.17 Ilustrasi APF.....	37
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	39
Gambar 3.2 Quanser Qdrone.....	40
Gambar 3.3 Blok Diagram untuk Keseluruhan Kontrol <i>Quadcopter</i> .....	42
Gambar 3.4 Blok Diagram Kontrol Posisi X serta Y dan Attitude $\phi$ serta $\theta$ .....	44
Gambar 3.5 <i>Membership Function</i> untuk Masukan Jarak Pada <i>Fuzzy</i> .....	45
Gambar 3.6 <i>Membership Function</i> untuk Masukan Kecepatan Pada <i>Fuzzy</i> .....	45
Gambar 3.7 Spesifikasi Bentuk Formasi V.....	46
Gambar 3.8 Komponen Gaya Pada <i>Quadcopter</i> Berdasarkan Konsep APF.....	48
Gambar 3.9 Skema Keseluruhan Sistem.....	50
Gambar 4.1 Trayektori Gerak <i>Quadcopter</i> Dalam Formasi.....	51
Gambar 4.2 Trayektori Gerak <i>Quadcopter</i> Tampak Atas.....	52
Gambar 4.3 Trayektori Gerak <i>Quadcopter</i> Tampak Samping.....	52
Gambar 4.4 Trayektori Gerak <i>Quadcopter</i> dengan Ukuran Diperbesar dan Menunjukkan Nilai Saat Itu.....	52
Gambar 4.5 Jarak Antara <i>Quadcopter</i> 1 Terhadap <i>Follower</i> -nya.....	53
Gambar 4.6 Jarak Antara <i>Quadcopter</i> 2 Terhadap <i>Follower</i> Lain.....	53
Gambar 4.7 Jarak <i>Quadcopter</i> 4 Terhadap <i>Quadcopter</i> 5.....	54
Gambar 4.8 Indeks Performansi Formasi pada Pengujian Formasi.....	54
Gambar 4.9 Pengujian Algoritma <i>Optimized APF</i> .....	55
Gambar 4.10 Pengujian Algoritma <i>Optimized APF</i> Tampak Atas.....	56
Gambar 4.11 Pengujian Algoritma <i>Optimized APF</i> Tampak Samping.....	56
Gambar 4.12 Pengujian Algoritma <i>Optimized APF</i> pada Lintasan yang Berdekatan.....	56
Gambar 4.13 Pengujian Kasus 1 – Penghindaran Total.....	57
Gambar 4.14 Pengujian Kasus 1 – Penghindaran Minimal.....	58

Gambar 4.15 Pengujian Kasus 1 Tampak Atas – Penghindaran Total.....	58
Gambar 4.16 Pengujian Kasus 1 Tampak Atas – Penghindaran Minimal.....	59
Gambar 4.17 Pengujian Kasus 1 Tampak Samping – Penghindaran Total dan Minimal.....	59
Gambar 4.18 Jarak Antara <i>Quadcopter</i> 1 Terhadap <i>Follower</i> -nya pada Kasus 1. (a) Penghindaran Total, (b) Penghindaran Minimal.....	60
Gambar 4.19 Jarak Antara <i>Quadcopter</i> 2 Terhadap <i>Follower</i> Lain pada Kasus 1. (a) Penghindaran Total, (b) Penghindaran Minimal.....	60
Gambar 4.20 Jarak <i>Quadcopter</i> 4 Terhadap <i>Quadcopter</i> 5 pada Kasus 1 – Penghindaran Total dan Minimal.....	60
Gambar 4.21 Indeks Performansi Formasi Kasus 1. (a) Penghindaran Total, (b) Penghindaran Minimal.....	61
Gambar 4.22 Pengujian Kasus 2 – Penghindaran Total.....	61
Gambar 4.23 Pengujian Kasus 2 – Penghindaran Minimal.....	62
Gambar 4.24 Pengujian Kasus 2 Tampak Atas – Penghindaran Total.....	62
Gambar 4.25 Pengujian Kasus 2 Tampak Atas – Penghindaran Minimal.....	62
Gambar 4.26 Pengujian Kasus 2 Tampak Samping – Penghindaran Total dan Minimal.....	63
Gambar 4.27 Jarak Antara <i>Quadcopter</i> 1 Terhadap <i>Follower</i> -nya pada Kasus 2. (a) Penghindaran Total, (b) Penghindaran Minimal.....	63
Gambar 4.28 Jarak Antara <i>Quadcopter</i> 2 Terhadap <i>Follower</i> Lain pada Kasus 2. (a) Penghindaran Total, (b) Penghindaran Minimal.....	63
Gambar 4.29 Jarak <i>Quadcopter</i> 4 Terhadap <i>Quadcopter</i> 5 pada Kasus 2 – Penghindaran Total dan Minimal.....	64
Gambar 4.30 Indeks Performansi Formasi Kasus 2. (a) Penghindaran Total, (b) Penghindaran Minimal.....	64
Gambar 4.31 Pengujian Kasus 3 – Penghindaran Total.....	65
Gambar 4.32 Pengujian Kasus 3 – Penghindaran Minimal.....	65
Gambar 4.33 Pengujian Kasus 3 Tampak Atas – Penghindaran Total.....	65
Gambar 4.34 Pengujian Kasus 3 Tampak Atas – Penghindaran Minimal.....	66
Gambar 4.35 Pengujian Kasus 3 Tampak Samping – Penghindaran Total dan Minimal.....	66
Gambar 4.36 Jarak Antara <i>Quadcopter</i> 1 Terhadap <i>Follower</i> -nya pada Kasus 3. (a) Penghindaran Total, (b) Penghindaran Minimal.....	66
Gambar 4.37 Jarak Antara <i>Quadcopter</i> 2 Terhadap <i>Follower</i> Lain pada Kasus 3. (a) Penghindaran Total, (b) Penghindaran Minimal.....	67
Gambar 4.38 Jarak <i>Quadcopter</i> 4 Terhadap <i>Quadcopter</i> 5 pada Kasus 3 – Penghindaran Total dan Minimal.....	67
Gambar 4.39 Indeks Performansi Formasi Kasus 3. (a) Penghindaran Total, (b) Penghindaran Minimal.....	67
Gambar 4.40 Simulasi dengan $K_{att} = 1$ dan $K_{rep} = 40$ . (a) Penghindaran Total, (b) Penghindaran Minimal.....	68
Gambar 4.41 Simulasi dengan $K_{att} = 1$ dan $K_{rep} = 10$ . (a) Penghindaran Total, (b) Penghindaran Minimal.....	68
Gambar 4.42 Simulasi dengan $K_{att} = 1$ dan $K_{rep} = 1$ . (a) Penghindaran Total, (b) Penghindaran Minimal.....	69

Gambar 4.43 Simulasi dengan $K_{att} = 0.01$ dan $K_{rep} = 0.4$ . (a) Penghindaran Total, (b) Penghindaran Minimal.....	69
Gambar 4.44 Simulasi dengan Halangan yang Ditemui Terlebih Dahulu. (a) Penghindaran Total, (b) Penghindaran Minimal.....	70
Gambar 4.45 Simulasi dengan Halangan yang Ditemui Terlebih Dahulu Tampak Atas – Penghindaran Total.....	71
Gambar 4.46 Simulasi dengan Halangan yang Ditemui Terlebih Dahulu Tampak Atas – Penghindaran Minimal.....	71

## DAFTAR TABEL

Tabel 2.1 Hasil Penelitian Terdahulu yang Digunakan Sebagai Bahan Rujukan Penelitian.....	22
Tabel 3.1 Parameter-Parameter pada Plant <i>Quadcopter</i> .....	41
Tabel 3.2 Nilai Parameter-Parameter <i>State Feedback Controller</i> .....	44
Tabel 3.3 Jarak Antar Robot pada Formasi V.....	46
Tabel 4.1 Posisi Awal Masing-Masing <i>Quadcopter</i> .....	51
Tabel 4.2 Posisi Awal <i>Quadcopter</i> , Posisi Target, dan Posisi Rintangan.....	55
Tabel 4.3 Posisi Awal <i>Quadcopter</i> dan Target.....	57
Tabel 4.4 Spesifikasi Rintangan pada Kasus 1.....	57
Tabel 4.5 Spesifikasi Rintangan pada Kasus 2.....	71
Tabel 4.6 Spesifikasi Rintangan untuk Kasus 3.....	64
Tabel 4.7 Posisi Awal Masing-Masing <i>Quadcopter</i> .....	70
Tabel 4.8 Posisi Halangan Tambahan dengan Posisi yang Lebih Dekat.....	70
Tabel 4.9 Pengaruh Nilai <i>Gain</i> APF pada Simulasi.....	73
Tabel 4.10 Perbandingan Penghindaran Total dan Penghindaran Minimal.....	74

## DAFTAR SIMBOL

<b>Simbol</b>	<b>Keterangan</b>
$V(\mathcal{G})$	Himpunan simpul dari graf $\mathcal{G}$
$E(\mathcal{G})$	Himpunan busur dari graf $\mathcal{G}$
$K_n$	Graf lengkap dengan $n$ simpul
$F$	Gaya
$U_1$	Gaya <i>thrust</i>
$U_2$	Gaya <i>roll</i>
$U_3$	Gaya <i>pitch</i>
$U_4$	Gaya <i>yaw</i>
$\xi$	Posisi linier <i>quadcopter</i>
$\Theta$	Posisi angular <i>quadcopter</i>
$\phi, \theta, \Psi$	Posisi sudut euler: <i>roll, pitch, yaw</i>
$p, q, r$	Kecepatan sudut Euler
$J$	Momen Inersia
$\tau$	Torsi
$\omega$	Kecepatan angular
$x, y, z$	Posisi linier koordinat Kartesian
$V$	Kecepatan linier
$U_{att}$	Medan atraktif
$U_{rep}$	Medan repulsif
$F_{att}$	Gaya atraktif
$F_{rep}$	Gaya repulsif
$K_{att}$	<i>Gain</i> atraktif
$K_{rep}$	<i>Gain</i> repulsif
$U_{quad}$	Medan pada <i>quadcopter</i>
$F_{quad}$	Gaya tolakan dari <i>quadcopter</i> lain
$I(t)$	Indeks performansi formasi

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

*Unmanned Aerial Vehicle* (UAV) merupakan kendaraan udara tanpa awak yang memanfaatkan gaya aerodinamika untuk menghasilkan gaya angkat (*thrust*) dan dapat terbang secara *autonomous* atau dikendalikan dari jarak jauh menggunakan teknologi kendali radio. Jenis dari UAV yang banyak digunakan saat ini adalah *quadcopter*. *Quadcopter* merupakan UAV jenis sayap berputar (*rotary wing*) yang menggunakan baling-baling atau *propeller* dalam proses penerbangannya. Sesuai namanya, *quadcopter* memiliki 4 buah *propeller* dengan masing – masing sudut *pitch* yang tetap. Apabila ditinjau dari sistemnya, *quadcopter* memiliki 6 derajat kebebasan (*6 Degree-of-Freedom*) yang terdiri dari 3 derajat kebebasan untuk gerakan translasi terhadap sumbu  $X, Y$ , dan  $Z$  serta 3 derajat kebebasan untuk gerak rotasi terhadap posisi sudut Euler  $\phi, \theta$ , dan  $\psi$  (Asti et al., 2020).

Dalam beberapa tahun terakhir, kontrol kooperatif sistem multi-UAV telah menjadi topik penelitian yang hangat di bidang kontrol penerbangan (Guo et al., 2020). Diantaranya, pengendalian formasi adalah salah satu tema yang penting untuk diteliti (CAI et al., 2020) karena sering digunakan pada bidang militer dan sipil. Namun, dengan meningkatnya kompleksitas lingkungan medan tempur modern dan persyaratan tugas, telah menjadi sangat mendesak untuk menyelesaikan tugas tertentu dengan multi-UAV, yang akan membuat kelompok UAV menghadapi tantangan yang lebih besar saat melakukan tugas yang kompleks (Cao & Sun, 2018). Oleh karena itu, sangat penting untuk mempelajari masalah yang bisa menghalangi pembentukan formasi pada sistem multi-UAV.

Pada kondisi dimana UAV tidak hanya membentuk formasi tetapi juga menuju ke suatu titik tujuan tertentu, penghindaran rintangan perlu dipertimbangkan. Penghindaran rintangan pada sebuah sistem multi-UAV memerlukan strategi kontrol yang berbeda dengan metode penghindaran rintangan biasa. Jumlah UAV yang lebih dari satu membuat kompleksitas permasalahan menjadi lebih tinggi. Penghindaran rintangan pada satu subjek menugaskan UAV untuk menghindari rintangan yang terdeteksi dan kembali mencari jalur menuju titik tujuan. Pada kumpulan UAV yang bergerak dalam formasi, penghindaran rintangan harus mempertimbangkan penghindaran tabrakan dengan UAV lain dan pembentukan kembali formasi setelah rintangan tidak terdeteksi. Kemudian UAV kembali menuju titik tujuan yang diinginkan dengan tetap mempertahankan formasi.

Pada tugas akhir ini akan disusun sebuah sistem pengaturan yang memungkinkan beberapa *quadcopter* untuk membentuk formasi dan menghindari rintangan-rintangan. Pembentukan formasi memanfaatkan topologi jaringan komunikasi antar *quadcopter* berdasarkan *guidance route*. Jaringan komunikasi ini juga akan digunakan untuk penghindaran tabrakan antar *quadcopter*. Sedangkan penghindaran rintangan dilakukan dengan memanfaatkan metode *Artificial Potential Field* (APF). Metode ini didasarkan pada data yang diperoleh dari sensor-sensor *quadcopter* terhadap rintangan statis maupun *quadcopter* lain. Kelemahan-kelemahan pada metode yang digunakan diperbaiki dengan berbagai pendekatan yang ada. Adapun untuk kasus seperti ini, terdapat dua macam strategi kontrol, yaitu strategi dimana formasi lebih penting daripada penghindaran rintangan dan strategi yang lebih mementingkan penghindaran diatas formasi. Kedua strategi akan dicoba dan dibandingkan pengaruhnya terhadap *behavior quadcopter*.

## 1.2 Rumusan Masalah

Pengaturan formasi memiliki potensi aplikasi yang luas pada berbagai bidang seperti pemindahan objek, pengawasan, pemetaan, dan eksplorasi lokasi tertentu bahkan lokasi bencana, serta membantu proses penyelamatan. Penerapan pengaturan formasi pada tugas-tugas tersebut seringkali mengharuskan robot bekerja pada medan baru yang tidak diketahui sebelumnya. Umumnya, terdapat objek-objek yang bersifat sebagai rintangan sehingga menghalangi robot untuk bergerak dengan bebas.

Sekumpulan *quadcopter* dengan posisi awal tertentu ditugaskan untuk membentuk dan mempertahankan formasi. Mereka memiliki tujuan untuk mencapai titik yang sudah ditentukan. Jika terdapat rintangan, mereka harus menghindarinya sambil mempertahankan formasi. Rumusan masalah pada penelitian ini adalah bagaimana merancang strategi kontrol bagi kelompok *quadcopter* untuk memenuhi tugas-tugas tersebut.

## 1.3 Batasan Masalah

Dalam melaksanakan tugas akhir ini, terdapat beberapa batasan masalah yang perlu diperhatikan, diantaranya sebagai berikut:

- a. UAV yang digunakan yaitu berupa *quadcopter* dengan 6 derajat kebebasan atau 6 *Degree of Freedom* (6-DoF).
- b. Pemodelan *quadcopter* yang digunakan adalah tipe Quanser-Qdrone.
- c. Implementasi simulasi dan pengujian dengan menggunakan aplikasi MATLAB.
- d. Diasumsikan bahwa komunikasi antar *quadcopter* adalah baik tanpa *delay*.
- e. Rintangan yang akan dihadapi formasi *quadcopter* berupa rintangan statis (tidak bergerak).

## 1.4 Tujuan

Pada penelitian ini, akan dikembangkan algoritma yang memanfaatkan *guidance route* untuk pengaturan formasi dan *Artificial Potential Field* (APF) untuk penghindaran rintangan pada sekumpulan *quadcopter*. Selanjutnya, akan dilakukan perbandingan antara strategi kontrol yang mementingkan formasi diatas penghindaran atau disebut penghindaran total dan strategi yang mementingkan penghindaran diatas formasi atau penghindaran minimal.

Hasilnya diharapkan dapat menjadi solusi dan referensi untuk penerapan pembentukan formasi dan penghindaran rintangan yang lebih efektif serta efisien pada berbagai bentuk kondisi medan.

## 1.5 Manfaat

Penelitian ini diharapkan bisa diterapkan dan dijadikan prototipe pada permasalahan riil. Manfaat dari penelitian ini adalah meningkatkan pemanfaatan teknologi dan peningkatan efektifitas serta efisiensi proses pengaturan formasi dan penghindaran rintangan pada kumpulan UAV.

## BAB 2 TINJAUAN PUSTAKA

### 2.1 Hasil Penelitian Terdahulu

Selama dua dekade terakhir, topik pembentukan formasi terkoordinasi telah banyak dipelajari dengan aplikasi pada sistem multi robot (Kwon & Chwa, 2012)(Saradagi et al., 2018), kendaraan udara (Mahmood & Kim, 2015)(R. Wang & Liu, 2017), kendaraan bawah air (Millan et al., 2014)(Park, 2015), dan pesawat luar angkasa (Shahbazi et al., 2017) untuk berbagai bidang. Tujuan utama dari kontrol formasi adalah menyusun strategi kontrol yang menggerakkan seluruh agen untuk membentuk formasi yang diinginkan sekaligus menjamin akurasi pelacakan dan sinkronisasi *attitude*. Namun, ada banyak tantangan teoritikal dan masalah riil untuk kontrol formasi multi agen, antara lain presisi kontrol, informasi yang terbatas, waktu *delay*, dan gangguan.

Pada literatur-literatur saat ini, berbagai metode lanjutan, seperti pendekatan *leader-follower* (Ghommam & Saad, 2018), *behavior-based* (Lin et al., 2014), *consensus-based* (Rezaee & Abdollahi, 2015), dan *virtual-structure* (Ren & Beard, 2004) telah diusulkan untuk kontrol formasi. Diantara teknik-teknik ini, pendekatan *leader-follower* telah banyak dipelajari untuk masalah kontrol formasi. Metode ini menekankan peran pemimpin dalam kelompok, yang mengetahui lintasan kelompok secara global. Agen yang lain dianggap sebagai *follower* dan memiliki akses ke informasi gerak *leader* melalui komunikasi internal. Kemudian, *follower* diarahkan untuk mempertahankan formasi bersama *leader* dengan protokol kontrol. Metode *leader-follower* telah dibahas dalam kontrol formasi mobile robot (Xiao et al., 2016), kendaraan permukaan (Sun et al., 2018), dan kendaraan udara. Tapi, karya-karya ini masih terbatas ke bidang dua dimensi.

Baru-baru ini, masalah konsensus pada multi agen telah dieksplorasi secara masif (Ren & Beard, 2004)(Li et al., 2022). Konsensus berarti semua agen mencapai identitas pada variabel dengan kepentingan tertentu. (Li et al., 2022) mengajukan metode formasi yang bervariasi terhadap waktu berdasarkan protokol konsensus untuk menggerakkan kendaraan orde dua yang berjumlah  $m$  ke formasi yang diinginkan melalui harmonisasi posisi dan kecepatan. Beberapa karya dengan sama-sama menggunakan teknik ini bisa ditemukan di (Andreasson et al., 2014). Hanya saja, literatur-literatur ini memecahkan masalah formasi menggunakan model titik massa atau model dinamika linier. Mereka mengabaikan dampak dari karakteristik dinamika yang kompleks dari objek kontrol tertentu seperti *quadcopter*.

Pendekatan berdasarkan *virtual-structure* adalah metode lain yang banyak digunakan untuk kontrol formasi. Pada struktur ini, keseluruhan formasi diperlakukan sebagai satu struktur yang utuh. Metode ini bisa diturunkan menjadi tiga tahap (Beard et al., 2001). Pertama, tentukan dinamika *virtual-structure* yang diinginkan. Kedua, terjemahkan gerak *virtual-structure* menjadi gerak untuk tiap agen yang diinginkan. Terakhir, dapatkan kontrol pelacakan untuk tiap agen. Salah dua referensi pendekatan ini kita dapatkan dari (Benzerrouk et al., 2014). Keunggulan *virtual-structure* adalah kemudahan dalam merancang perilaku kooperatif untuk kelompok (Beard et al., 2001). Akan tetapi, struktur itu sendirilah yang akan membatasi aplikasi potensial dari formasi.

Sebagian besar metode untuk kontrol formasi dapat diklasifikasikan menjadi tiga kategori (Oh et al., 2015). Kelas pertama fokus pada kontrol formasi yang berdasar pada posisi (*Position-based Formation Control (PFC)*). Dalam teknik ini, tiap agen secara aktif mengatur penyimpangan dari posisi waktu itu dari posisi yang diinginkan untuk membentuk formasi yang diinginkan. Kelas ini lebih cocok diterapkan pada model yang berkecepatan rendah dan *omni-*

*directional*. Kelas lainnya adalah *Displacement-based Formation Control* (DFC). Tiap agen merasakan posisi relatif dari tetangganya dan mengatur penyimpangan dari perpindahannya terhadap perpindahan tetangga untuk mencapai formasi yang diinginkan. Referensinya bisa dilihat di (Padhi et al., 2014)(Dehghani & Menhaj, 2016). Limitasi dari DFC adalah tiap agen harus mengetahui posisi relatif dari tetangga-tetangganya melalui interaksi dalam topologi secara *real time*. Biasanya, agen memiliki jangkauan sensor yang terbatas sehingga diperlukan biaya yang cukup besar. Kelas terakhir digambarkan dengan mengatur jarak antar agen untuk mencapai formasi yang ditentukan oleh jarak tertentu antar agen (Kownacki & Ambroziak, 2017). Tiap agen diasumsikan dapat merasakan posisi relatif agen-agen tetangganya sehubungan dengan sistem koordinat lokal mereka. Keunggulan dari pendekatan *distance-based* adalah tiap agen memerlukan lebih sedikit informasi *global* jika dibandingkan dengan PFC dan DFC. Sayangnya, metode ini tidak bisa mencapai kontrol formasi yang teliti dan penelitian yang ada umumnya fokus pada model *single-integrator*.

Sebagai adisi untuk kontrol formasi, penerapan formasi pada lingkungan yang kompleks banyak dipelajari akhir-akhir ini. Contohnya adalah penghindaran rintangan dan keterbatasan komunikasi. Beberapa metode yang sering digunakan untuk penghindaran rintangan adalah metode *geometry-based* (Goss et al., 2004), *consensus-based* (Zhang, Yan, Zhang, et al., 2018), dan *potential-based* (Zhang, Yan, & Zhang, 2018). Untuk metode *potential-based*, ada algoritma yang dikenal sebagai *Artificial Potential Field* (APF) (Yu & Lu, 2021)(Chen et al., 2016).

Tabel 2.1 Hasil Penelitian Terdahulu yang Digunakan Sebagai Bahan Rujukan Penelitian.

	2022	2019	2021
Judul Penelitian	<i>Cooperative Position-based Formation-pursuit of Moving Target by Multi-UAVs with Collision Avoidance</i>	<i>Cooperative Formation Control of Multiple Aerial Vehicle Based on Guidance Route in a Complex Task Environment</i>	<i>UAV Formation Obstacle Avoidance Control Algorithm Based on Improved Artificial Potential Field and Consensus</i>
Penulis	Siti Nurjanah, Trihastuti, Agustinah, Muhammad Fuad	Guibin Sun, Rui Zhou, Kun Xu, Zhi Weng, Yuhang Zhang, Zhuoning Dong, Yingxun Wang	Ning Wang, Jiyang Dai, dan Jin Ying
Keterangan	Referensi yang digunakan yaitu pemodelan dinamika dan kinematika serta kontrol <i>quadcopter</i> .	Referensi yang digunakan yaitu metode <i>Guidance Route</i> pada kontrol formasi.	Referensi yang digunakan yaitu improvisasi pada <i>Artificial Potential Field</i> (APF).

## 2.2 Dasar Teori

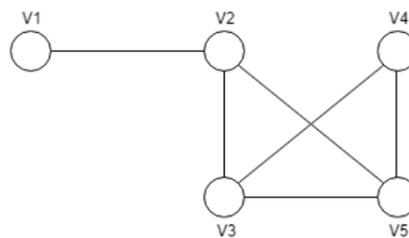
### 2.2.1 Teori Graf

Sebuah graf adalah sebuah himpunan yang terdiri atas himpunan simpul (*vertex*) dan himpunan busur (*edge*). Himpunan simpul dari sebuah graf  $\mathcal{G}$  dilambangkan dengan  $V(\mathcal{G})$ . Himpunan simpul dengan  $n$  anggota dapat dituliskan sebagai;

$$V(G) = \{v_1, v_2, \dots, v_n\}$$

Himpunan busur dilambangkan dengan  $E(G)$ . Busur adalah pasangan tak terurut dari simpul. Busur dapat dilambangkan dengan  $\{v_i, v_j\}$  atau  $v_i v_j$  dengan  $i, j = 1, 2, \dots, n$  dan  $i \neq j$ . Kardinalitas (banyaknya anggota) dari himpunan simpul pada suatu graf  $G$  disebut sebagai orde (*order*) graf  $G$  sedangkan kardinalitas dari himpunan busur disebut sebagai ukuran (*size*) graf  $G$ .

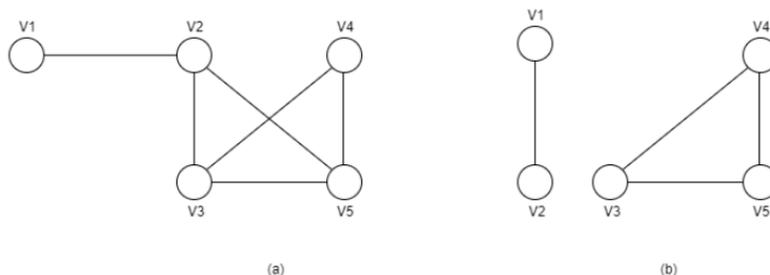
Graf dapat digambarkan dengan menggunakan titik untuk simpul dan garis untuk busur. Garis untuk sebuah busur  $v_i v_j$  digambar antara simpul  $v_i$  dan  $v_j$ . Gambar 2.1 menunjukkan sebuah contoh graf tak-berarah (*undirected graph*),  $G = (V, E)$  dengan  $V = \{v_1, v_2, \dots, v_5\}$  dan  $E = \{v_1 v_2, v_2 v_3, v_3 v_4, v_3 v_5, v_2 v_5, v_4 v_5\}$ .



Gambar 2.1 Graf Tidak Berarah dengan 5 Simpul dan 6 Busur

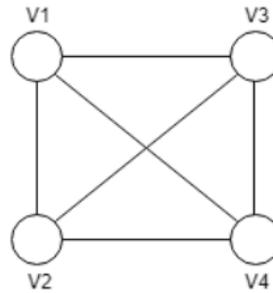
Terdapat beberapa istilah yang sering digunakan dalam analisa graf diantaranya yaitu ketetanggaan (*adjacency*) dan persekitaran (*neighborhood*). Simpul  $v_i$  dan  $v_j$  disebut bertetanggaan (*adjacent*) apabila terdapat busur diantara keduanya, disimbolkan dengan  $v_i \sim v_j$ . Dalam hal ini, busur  $v_i v_j$  dikatakan bersisian (*incident*) dengan simpul  $v_i$  dan  $v_j$ . Persekitaran  $N(i) \subseteq V$  dari simpul  $v_i$  didefinisikan sebagai himpunan  $\{v_j \in V | v_i v_j \in E\}$ , dengan kata lain, himpunan semua simpul yang bertetanggaan dengan  $v_i$ . Dalam graf tak-berarah jika  $v_j \in N(i)$ , maka  $v_i \in N(j)$ .

Jalur (*path*) dengan panjang  $n$  dari simpul  $v_i$  ke  $v_j$  adalah barisan dengan  $n + 1$  simpul berbeda yang berawal dari  $v_i$  dan berakhir di  $v_j$  sedemikian sehingga setiap simpul yang berurutan bertetanggaan. Suatu graf  $G$  dikatakan terhubung (*connected*) apabila selalu terdapat jalur yang menghubungkan dua simpul dalam graf tersebut. Graf dikatakan tak-terhubung (*disconnected*) jika keadaan tersebut tidak terpenuhi. Contoh graf terhubung dan graf tak-terhubung dapat dilihat pada Gambar 2.2.



Gambar 2.2 (a) Graf Terhubung, (b) Graf Tak-terhubung

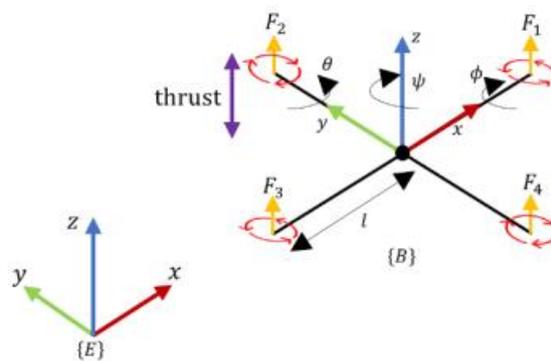
Suatu graf  $G$  dengan  $n$  simpul dikatakan sebagai graf lengkap (*complete graph*) apabila setiap pasang simpulnya bertetanggaan. Graf lengkap dengan  $n$  simpul diberi nama  $K_n$ .



Gambar 2.3 Graph Lengkap  $K_4$

### 2.2.2 Quadcopter (Nurjanah & Fuad, 2022)

*Quadcopter* merupakan suatu robot terbang kecil memiliki model mekanik yang terdiri dari empat rotor yang dipasang pada sumbu silang simetris. Bentuk ini diharapkan tipis dan kaku, sehingga diperoleh friksi udara yang kecil dan komponen yang bergerak pada *quadcopter* hanyalah putaran *propeller*. Setiap *propeller* pada *quadcopter* ini diputar oleh satu motor elektrik, sehingga terdapat empat motor sebagai aktuator untuk menghasilkan gaya angkat pada *quadcopter*. Dengan batasan menggunakan karakteristik motor dan *propeller* yang relatif sama, maka kondisi melayang yang stabil akan diperoleh kecepatan motor yang sama di tiap *propeller*. Namun pada tiap *propeller* yang berputar akan menghasilkan torsi yang bergerak medan horizontal, maka konfigurasi *propeller* terdiri dari dua yang bergerak searah jarum jam dan dua yang berlawanan arah jarum jam dan dalam tiap satu sumbu, propeller berputar dengan arah yang sama. Pada Gambar 2.4, ditunjukkan bahwa terdapat empat gerakan yang terdapat pada struktur *quadcopter*, yaitu *roll*, *pitch*, *yaw* dan *thrust*.

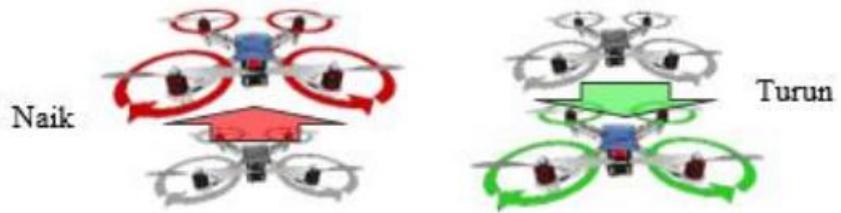


Gambar 2.4 Konfigurasi *Quadcopter*

Dengan melakukan pengaturan kecepatan putaran *propeller* maka dihasilkan beberapa komando input diantaranya:

a. *Thrust* (U1)

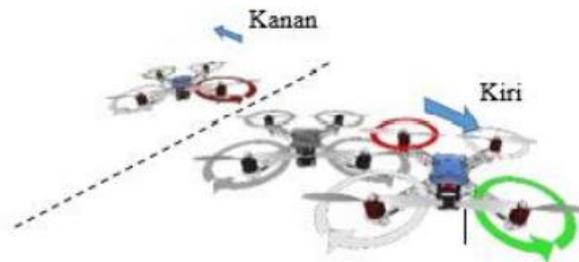
*Thrust* adalah gerakan naik dan turun dari *quadcopter*. Gerak *thrust* dicapai dengan mempercepat (warna merah) atau memperlambat (warna hijau) kecepatan motor dari seluruh *propeller* secara bersamaan sehingga menghasilkan percepatan vertikal yang ditunjukkan pada Gambar 2.5.



Gambar 2.5 Ilustrasi Konsep *Thrust* dari *Quadcopter*

b. *Roll* (U2)

*Roll* adalah salah satu gerak rotasi dari *quadcopter*, yakni rotasi terhadap sumbu  $x$  sehingga *quadcopter* bergerak ke kanan atau kiri. Gerak *roll* dicapai dengan mempercepat atau memperlambat *propeller* pada sisi kiri dan secara bersamaan memperlambat atau mempercepat *propeller* di sisi kanan akan diperoleh rotasi dengan kecepatan sudut yang tersaji pada Gambar 2.6.



Gambar 2.6 Ilustrasi Konsep *Roll* dari *Quadcopter*

c. *Pitch* (U3)

*Pitch* adalah salah satu gerak rotasi pada *quadcopter*, yakni rotasi terhadap sumbu  $y$  sehingga *quadcopter* bergerak ke depan atau belakang. Pada Gambar 2.7 ditunjukkan mekanisme yang identic dengan gerak *roll*, namun dilakukan pada *propeller* di sisi depan dan belakang akan diperoleh percepatan sudut *pitch*.



Gambar 2.7 Ilustrasi Konsep *Pitch* dari *Quadcopter*

d. *Yaw* (U4)

*Yaw* adalah gerak rotasi ketiga dari *quadcopter*, yakni rotasi terhadap sumbu  $z$  sehingga *quadcopter* dapat merubah arah secara horizontal. Dengan mempercepat atau memperlambat *propeller* yang berputar berlawanan arah

jarum jam, dan memperlambat atau mempercepat propeller yang berputar searah jarum jam akan diperoleh gerak *yaw* dengan percepatan sudut yang ditampilkan pada Gambar 2.8.



Gambar 2.8 Ilustrasi Konsep *Yaw* dari *Quadcopter*

Kemudian dari empat gerakan yang sudah dijelaskan sebelumnya, dilakukan proses pemodelan dari *quadcopter*. Pada bagian ini akan dijelaskan analisis penurunan persamaan matematika dari *quadcopter*, dimana analisis ini diperoleh dari analisis kinematika dan dinamika *quadcopter*. Hasil penurunan persamaan matematika ini akan menggambarkan semua pergerakan *quadcopter* sehingga bisa digunakan sebagai acuan dalam menyusun kontroler sesuai dengan yang diinginkan.

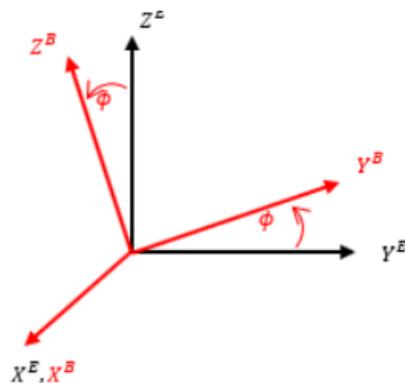
a. Kinematika *quadcopter*

Kinematika merupakan cabang ilmu mekanika yang membahas gerak sebuah benda dengan mengabaikan gaya penyebab gerakan. Pada *quadcopter* analisa gerak kinematika meliputi gerak *roll*, *pitch*, dan *yaw* yang mengacu pada koordinat *body frame* (B-frame). Untuk menentukan posisi linier terhadap bumi diperlukan sebuah matriks transformasi dari koordinat B-frame ke koodinat *earth frame* (E-frame). Sehingga posisi linier ( $\xi$ ) maupun posisi angular ( $\Theta$ ) dapat didefinisikan sebagai berikut:

$$\begin{aligned} \xi^E &= [X \quad Y \quad Z]^T \\ \Theta^E &= [\phi \quad \theta \quad \psi]^T \end{aligned} \tag{2.1}$$

- Gerak *Roll*

Gerak *roll* adalah gerak *quadcopter* yang berotasi pada sumbu *x* sebesar  $\phi$  atau dilambangkan  $R(\phi, x)$ . Gerak tersebut dapat dilihat pada Gambar 2.9.



Gambar 2.9 Ilustrasi Gerak *Roll* pada *Quadcopter*

Berdasarkan gerak *roll* pada *quadcopter* didapatkan persamaan kinematika seperti berikut

$$\begin{aligned} X^E &= X^B \cos 0^\circ + Y^B \cos 90^\circ + Z^B \cos 90^\circ \\ Y^E &= X^B \cos 90^\circ + Y^B \cos \phi + Z^B \cos(90^\circ + \phi) \\ Z^E &= X^B \cos 90^\circ + Y^B \cos(90^\circ - \phi) + Z^B \cos \phi \end{aligned} \quad (2.2)$$

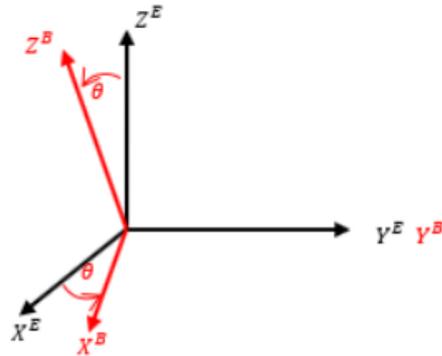
Persamaan (2.2) dapat diubah dalam bentuk matriks seperti (2.3) dan (2.4).

$$\begin{bmatrix} X^E \\ Y^E \\ Z^E \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} X^B \\ Y^B \\ Z^B \end{bmatrix} \quad (2.3)$$

$$R^{BE}(\phi, x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}, \phi = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \quad (2.4)$$

- Gerak *Pitch*

Gerak *pitch* adalah gerak *quadcopter* yang berotasi pada sumbu *y* sebesar  $\theta$  atau dilambangkan  $R(\theta, y)$ . Gerak tersebut dapat dilihat pada Gambar 2.10.



Gambar 2.10 Ilustrasi Gerak *Pitch* pada *Quadcopter*

Berdasarkan gerak *pitch* pada *quadcopter* didapatkan persamaan kinematika seperti berikut.

$$\begin{aligned} X^E &= X^B \cos \theta + Y^B \cos 90^\circ + Z^B \cos(90^\circ - \theta) \\ Y^E &= X^B \cos 90^\circ + Y^B \cos 0^\circ + Z^B \cos 90^\circ \\ Z^E &= X^B \cos(90^\circ + \theta) + Y^B \cos 90^\circ + Z^B \cos \theta \end{aligned} \quad (2.5)$$

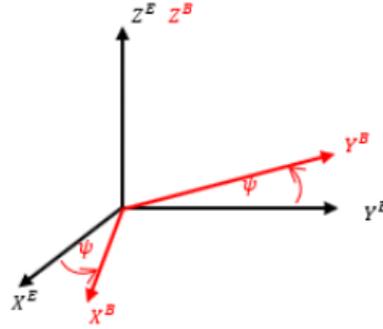
Persamaan (2.5) dapat diubah dalam bentuk matriks seperti (2.6) dan (2.7).

$$\begin{bmatrix} X^E \\ Y^E \\ Z^E \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} X^B \\ Y^B \\ Z^B \end{bmatrix} \quad (2.6)$$

$$R^{BE}(\theta, \gamma) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}, \theta = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \quad (2.7)$$

- Gerak *Yaw*

Gerak *yaw* adalah gerak *quadcopter* yang berotasi pada sumbu  $z$  sebesar  $\psi$  atau dilambangkan  $R(\psi, z)$ . Gerak tersebut dapat dilihat pada Gambar 2.11.



Gambar 2.11 Ilustrasi Gerak *Yaw* pada *Quadcopter*

Berdasarkan gerak *yaw* pada *quadcopter* didapatkan persamaan kinematika seperti pada (2.8).

$$\begin{aligned} X^E &= X^B \cos \psi + Y^B \cos(90^\circ + \psi) + Z^B \cos 90^\circ \\ Y^E &= X^B \cos(90^\circ - \psi) + Y^B \cos \psi + Z^B \cos 90^\circ \\ Z^E &= X^B \cos 90^\circ + Y^B \cos 90^\circ + Z^B \cos 0^\circ \end{aligned} \quad (2.8)$$

Persamaan (2.8) dapat diubah dalam bentuk matriks seperti persamaan (2.9) dan (2.10).

$$\begin{bmatrix} X^E \\ Y^E \\ Z^E \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X^B \\ Y^B \\ Z^B \end{bmatrix} \quad (2.9)$$

$$R^{BE}(\psi, z) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \psi = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \quad (2.10)$$

Lalu untuk mendapatkan matriks transformasi  $R_{\xi}^{BE}$  dari koordinat B-frame ke koordinat E-frame maka analisa kinematika dari gerak *roll*, *pitch*, *yaw* dapat digabungkan menjadi satu matriks. Matriks  $R_{\xi}^{BE}$  diperoleh dengan mengalikan matriks rotasi dari tiap gerak *quadcopter* seperti pada persamaan berikut.

$$\begin{aligned} R_{\xi}^{BE} &= R(\phi, x) \cdot R(\theta, y) \cdot R(\psi, z) \\ R_{\xi}^{BE} &= \begin{bmatrix} c_{\theta} c_{\psi} & -c_{\phi} s_{\psi} + s_{\phi} s_{\theta} c_{\psi} & s_{\phi} s_{\psi} + c_{\phi} s_{\theta} c_{\psi} \\ c_{\theta} s_{\psi} & c_{\phi} c_{\psi} + c_{\phi} s_{\theta} s_{\psi} & -s_{\phi} c_{\psi} + c_{\phi} s_{\theta} s_{\psi} \\ -s_{\theta} & c_{\theta} s_{\phi} & c_{\theta} c_{\phi} \end{bmatrix} \end{aligned} \quad (2.11)$$

Dimana  $c$  melambangkan  $\cos$  dan  $s$  melambangkan  $\sin$ . Selain matriks transformasi  $R_{\xi}^{BE}$ , ada pula matriks transformasi  $T_{\Theta}^{BE}$  untuk mengubah besaran angular dari koordinat B-frame ke koordinat E-frame. Matriks transformasi  $T_{\Theta}^{BE}$  mengacu pada kecepatan Euler dalam B-frame dengan membalik pola perputaran sudut *roll*, *pitch*, dan *yaw*.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = I \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R(\phi, x)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R(\phi, x)^{-1} R(\theta, y)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.12)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T_{\Theta}^{BE-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.13)$$

Berdasarkan (2.25) maka didapatkan matriks transformasi  $T_{\Theta}^{BE}$ .

$$T_{\Theta}^{BE-1} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix}$$

$$T_{\Theta}^{BE} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \quad (2.14)$$

#### b. Dinamika *quadcopter*

Kebalikan dari kinematika, dinamika merupakan analisis gerak suatu benda dengan memperhatikan gaya yang berpengaruh pada benda tersebut. Analisis dinamika dilakukan dengan menggunakan hukum Newton Euler tentang gerak suatu benda. Gaya yang muncul akibat gerak dari *quadcopter* dapat ditulis sebagai berikut:

$$\begin{bmatrix} F \text{ thrust} \\ F \text{ roll} \\ F \text{ pitch} \\ F \text{ yaw} \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} F_1 + F_2 + F_3 + F_4 \\ F_2 - F_4 \\ F_1 - F_3 \\ F_1 - F_2 + F_3 - F_4 \end{bmatrix} \quad (2.15)$$

dimana gaya  $F_i$  merupakan gaya angkat yang dihasilkan oleh tiap motor *propeller* yang didefinisikan sebagai berikut:

$$F_i = K \frac{\omega}{s+\omega} u_i \quad (2.16)$$

dengan  $K$  merupakan konstanta gaya tolak,  $\omega$  adalah *bandwidth* motor, dan  $u$  adalah sinyal kontrol dari kontroller. Untuk memperoleh konstanta  $K$  maka dicari dengan menerbangkan *quadcopter* pada posisi *hover*. Nilai  $K$  akan sebanding dengan total gaya angkat dari keempat motor *quadcopter* saat *hover*.

Pemodelan *quadcopter* dilakukan dengan kombinasi koordinat *frame* atau disebut *Hybrid-Frame* (H-frame). Penurunan model gerak translasi akan diturunkan terhadap

koordinat bumi (E-frame) karena berkaitan dengan posisi dan kecepatan *quadcopter* terhadap bumi. Sedangkan penurunan gerak rotasi diturunkan terhadap koordinat *body* (B-frame) karena gerak rotasi mempengaruhi gerakan *quadcopter* itu sendiri.

- Gerak translasi

Berdasarkan aksioma pertama Euler dari hukum II Newton didapatkan persamaan gerak translasi sebagai berikut:

$$F_i = K \frac{\omega}{s+\omega} u_i$$

$$\sum F^E = \ddot{\xi}^E m \quad (2.17)$$

dengan  $\sum F^E$  merupakan resultan gaya yang bekerja pada *quadcopter*,  $m$  merupakan massa total *quadcopter* dan  $\ddot{\xi}^E$  merupakan percepatan gerak *quadcopter*. Sehingga (2.17) dapat diubah menjadi (2.18).

$$F_f^E + F_g = \ddot{\xi} m \quad (2.18)$$

dimana  $\xi = [X \ Y \ Z]^T$  merupakan posisi *quadcopter* terhadap bumi,  $F_f^E$  adalah resultan gaya pada tiap sumbu koordinat, dan  $F_g = [0 \ 0 \ -mg]^T$  mewakili gaya gravitasi. Jika didefinisikan  $F_f^B = [F_X \ F_Y \ F_Z]^T$  sebagai resultan gaya pada koordinat B-frame,  $F_f^B$  harus ditransformasikan ke dalam koordinat E-frame. Karena gaya yang bekerja hanya pada sumbu z maka resultan gaya yang muncul hanya  $F_Z$  atau gaya *thrust* (U1).

$$F_f^E = R_{\xi}^{BE} \cdot F_f^B \quad (2.19)$$

$$F_f^E = \begin{bmatrix} F_Z(s_\phi s_\psi + c_\phi s_\theta c_\psi) \\ F_Z(-s_\phi c_\psi + c_\phi s_\theta s_\psi) \\ F_Z(c_\phi c_\theta) \end{bmatrix} = \begin{bmatrix} U1(s_\phi s_\psi + c_\phi s_\theta c_\psi) \\ U1(-s_\phi c_\psi + c_\phi s_\theta s_\psi) \\ U1(c_\phi c_\theta) \end{bmatrix} \quad (2.20)$$

substitusikan (2.20) ke (2.18), diperoleh

$$\begin{bmatrix} U1(s_\phi s_\psi + c_\phi s_\theta c_\psi) \\ U1(-s_\phi c_\psi + c_\phi s_\theta s_\psi) \\ U1(c_\phi c_\theta) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} = \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} m \quad (2.21)$$

dari persamaan (2.21) diperoleh persamaan gerak translasi *quadcopter* seperti pada (2.22).

$$\ddot{X} = (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \frac{U_1}{m}$$

$$\ddot{Y} = (-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi) \frac{U_1}{m} \quad (2.22)$$

$$\ddot{Z} = -g + (\cos \phi \cos \theta) \frac{U_1}{m}$$

- Gerak rotasi

Berdasarkan aksioma kedua Euler pada hukum II Newton didapatkan persamaan gerak rotasi sebagai berikut:

$$\tau^E = J\ddot{\Theta}^E \quad (2.23)$$

Dimana  $\tau = [U_2l \quad U_3l \quad U_4d]^T$  adalah torsi yang bekerja pada *quadcopter*,  $l$  adalah jarak motor terhadap pusat massa,  $d$  adalah konstanta *drag quadcopter*, dan  $J = \text{diag}[J_{xx} \quad J_{yy} \quad H_{zz}]$  adalah matriks diagonal yang berisi momen inersia tiap sumbu. Persamaan (2.23) merupakan persamaan pada koordinat E-frame, sehingga perlu ditransformasi ke dalam koordinat B-frame. Persamaan tersebut jika dibawa ke dalam koordinat B-frame akan menjadi:

$$\begin{aligned} T_{\Theta}\tau^B &= J(T_{\Theta}\omega^B) \\ T_{\Theta}\tau^B &= J(T_{\Theta}\dot{\omega}^B + T_{\Theta}\omega^B) \end{aligned} \quad (2.24)$$

$\omega$  adalah kecepatan sudut quadcopter yang bernilai  $\omega = [p \quad q \quad r]^T = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$ . Jika turunan dari matriks transformasi  $T_{\Theta}$  adalah  $T_{\Theta}S(\omega^B)$ , dengan  $S(\omega^B)$  adalah matriks *skew-symmetric*, maka persamaan (2.24) dapat diubah menjadi persamaan (2.25).

$$\begin{aligned} T_{\Theta}\tau^B &= J(T_{\Theta}S(\omega^B)\omega^B + T_{\Theta}\dot{\omega}^B) \\ T_{\Theta}\tau^B &= T_{\Theta}(S(\omega^B)J\omega^B + J\dot{\omega}^B) \\ T_{\Theta}\tau^B &= T_{\Theta}(\omega^B \times J\omega^B + J\dot{\omega}^B) \end{aligned} \quad (2.25)$$

Karena pada kedua ruas terdapat matriks tranformasi  $T_{\Theta}$ , maka dapat dieliminasi atau persamaan (2.25) dapat dianggap sudah ditransformasikan ke dalam koordinat B-frame.

$$\begin{aligned} \tau^B &= \omega^B \times J\omega^B + J\dot{\omega}^B \\ J\dot{\omega}^B &= -(\omega^B \times J\omega^B) + \tau^B \end{aligned} \quad (2.26)$$

untuk mempermudah proses penghitungan, diselesaikan dulu perkalian *cross product* didalam kurung.

$$\begin{aligned} -(\omega^B \times J\omega^B) &= J\omega^B \times \omega^B \\ -(\omega^B \times J\omega^B) &= \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} (J_{yy} - J_{zz})qr \\ (J_{zz} - J_{xx})pr \\ (J_{xx} - J_{yy})pq \end{bmatrix} \end{aligned} \quad (2.27)$$

dengan menyelesaikan persamaan (2.27) maka diperoleh persamaan gerak rotasi *quadcopter* seperti pada persamaan (2.29).

$$\begin{aligned} J\dot{\omega}^B &= -(\omega^B \times J\omega^B) + \tau^B \\ \begin{bmatrix} J_{xx}\dot{p} \\ J_{yy}\dot{q} \\ J_{zz}\dot{r} \end{bmatrix} &= \begin{bmatrix} (J_{yy} - J_{zz})qr \\ (J_{zz} - J_{xx})pr \\ (J_{xx} - J_{yy})pq \end{bmatrix} + \begin{bmatrix} U_2l \\ U_3l \\ U_4d \end{bmatrix} \end{aligned} \quad (2.28)$$

sehingga

$$\dot{p} = \frac{J_{yy} - J_{zz}}{J_{xx}}qr + \frac{U_2l}{J_{xx}} \quad (2.29)$$

$$\begin{aligned}\dot{q} &= \frac{J_{zz}-J_{xx}}{J_{yy}} pr + \frac{U_3 l}{J_{yy}} \\ \dot{r} &= \frac{J_{xx}-J_{yy}}{J_{zz}} pq + \frac{U_4 d}{J_{zz}}\end{aligned}$$

Dengan mengambil persamaan gerak translasi (2.22) dan persamaan gerak rotasi (2.29), maka keseluruhan model dinamika *quadcopter* dapat dinyatakan sebagai:

$$\left\{ \begin{aligned}\ddot{X} &= (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \frac{U_1}{m} \\ \ddot{Y} &= (-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi) \frac{U_1}{m} \\ \ddot{Z} &= -g + (\cos \phi \cos \theta) \frac{U_1}{m} \\ \dot{p} &= \frac{J_{yy}-J_{zz}}{J_{xx}} qr + \frac{U_2 l}{J_{xx}} \\ \dot{q} &= \frac{J_{zz}-J_{xx}}{J_{yy}} pr + \frac{U_3 l}{J_{yy}} \\ \dot{r} &= \frac{J_{xx}-J_{yy}}{J_{zz}} pq + \frac{U_4 d}{J_{zz}}\end{aligned} \right. \quad (2.30)$$

### 2.2.3 State Feedback Controller (Nurjanah & Fuad, 2022)

*State feedback controller* merupakan metode yang digunakan dalam teori sistem kontrol *feedback* (umpan balik) untuk menempatkan *pole* dari *loop* tertutup suatu plant ke lokasi yang telah ditentukan di bidang  $s$ . Sedangkan *feedback linearization* merupakan pendekatan umum yang digunakan untuk mengendalikan sistem non linier. Pendekatan ini melibatkan transformasi sistem non linier menjadi sistem linier yang ekuivalen melalui perubahan variabel dan input kontrol yang sesuai. *Feedback linearization* dapat diterapkan pada sistem non linier dengan bentuk

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= hx\end{aligned} \quad (2.31)$$

dimana  $x \in \mathbb{R}^n$  adalah vektor *state*,  $u \in \mathbb{R}^p$  adalah vektor masukan (*input*), dan  $y \in \mathbb{R}^m$  adalah vektor keluaran (*output*). Tujuannya adalah mengembangkan *control input*

$$\begin{aligned}u &= a(x) + b(x)v \\ v &= -Kx\end{aligned} \quad (2.32)$$

yang membuat peta *input-output* linier antara input baru  $v$ . Perubahan variabel

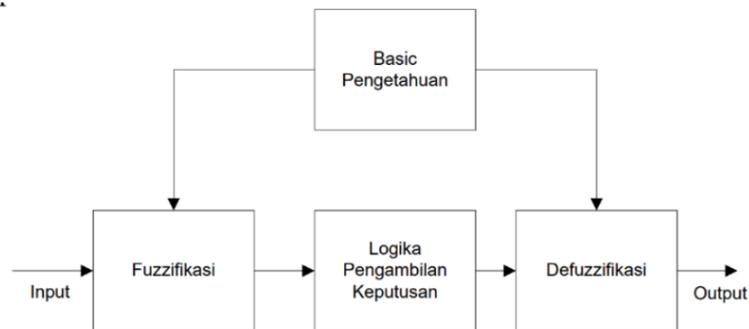
$$z = T(x) \quad (2.33)$$

mengubah sistem non linier menjadi sistem linier ekuivalen. Strategi kontrol *outer loop* untuk sistem kontrol linier yang dihasilkan kemudian dapat diterapkan.

### 2.2.4 Fuzzy Logic

Logika fuzzy merupakan sebuah pernyataan nilai kebenaran yang menerapkan logika atau pengetahuan manusia. Nilai kebenaran yang dinyatakan yaitu berupa nilai dalam selang yaitu 0 hingga 1, berbeda dengan logika klasik yang menyatakan kebenaran dengan nilai terbatas, yaitu 0 jika bernilai salah dan 1 jika bernilai benar. Penarikan kesimpulan nilai kebenaran yang dinyatakan dalam Logika Fuzzy dapat direpresentasikan dalam sistem

penarikan kesimpulan dari sekumpulan kaidah fuzzy yang dinamakan sebagai *Fuzzy Inference System* (FIS). Berbagai proses dalam FIS ditunjukkan pada Gambar 2.12.



Gambar 2.12 Skema Kerja Logika *Fuzzy*

Berikut penjelasan dari masing-masing proses.

- a. Fuzzifikasi  
Proses pemetaan nilai numerik atau nilai tegas ke dalam himpunan *fuzzy*. Masukan-masukan tersebut akan dibentuk sebagai masukan *fuzzy* yang berupa variabel linguistik. Variabel linguistik yang dimiliki sistem dibuat dan ditentukan berdasarkan fungsi keanggotaan (*membership functions*) yang digunakan. Pengalaman dari para ahli (perancang) sangat penting dalam membuat basis aturan (*rule base*) dalam perancangan *fuzzy control* sehingga dapat mencapai nilai *error* dan *delta error* sama dengan nol.
- b. Operasi logika fuzzy  
Proses perhitungan derajat kebenaran dari sekumpulan anggota himpunan fuzzy yang merupakan input atau masukan dengan Operator Logika *AND*, *OR*, atau *NOT* dalam menyambungkan kedua atau lebih himpunan fuzzy yang berperan sebagai masukan.
- c. Implikasi  
Sebuah proses untuk mendapatkan kesimpulan dari nilai predikat konsekuen dari hasil operasi logika fuzzy.
- d. Defuzzifikasi  
Setelah mendapatkan nilai kesimpulan dari hasil operasi logika fuzzy, maka ditentukan besaran nilai *crisp* dari himpunan fuzzy tersebut, yang berarti mengubah nilai logika menjadi besaran numerik.

#### 2.2.4.1 Model Fuzzy Sugeno (Nurjanah & Fuad, 2022)

Terdapat dua macam metode defuzzifikasi, salah satunya adalah model Sugeno. Defuzzifikasi dalam model Sugeno mudah diterapkan karena hasil dari setiap aturan sudah jelas dan hasil total ditentukan oleh bobot dan jumlah hasil dari setiap aturan. Aturan umum dalam model fuzzy Sugeno dapat dilihat dalam bentuk berikut:

$$\text{if Input 1} = x \text{ and Input 2} = y, \text{ then Output is } z = ax + by + c$$

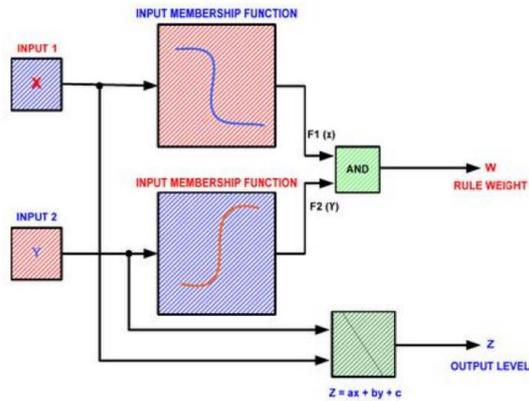
Untuk model Sugeno orde nol, keluaran  $z$  adalah konstan ( $a = b = 0$ ). Tingkat keluaran  $z_i$  dari setiap aturan berdasar pada pembobotan dari aturan tersebut. Sebagai contoh, aturan *AND* dengan Input 1 =  $x$  and Input 2 =  $y$ , maka besar bobotnya adalah

$$w_i = \text{AndMethod}(F_1(x), F_2(y))$$

dimana  $F_1(x)$  dan  $F_2(y)$  adalah fungsi keanggotaan dari Input 1 dan Input 2. Keluaran akhir dari sistem adalah rata-rata bobot dari semua keluaran aturan, dihitung dengan menggunakan

$$\text{Output} = \frac{\sum_{i=1}^N w_i z_i}{\sum_{i=1}^N w_i}$$

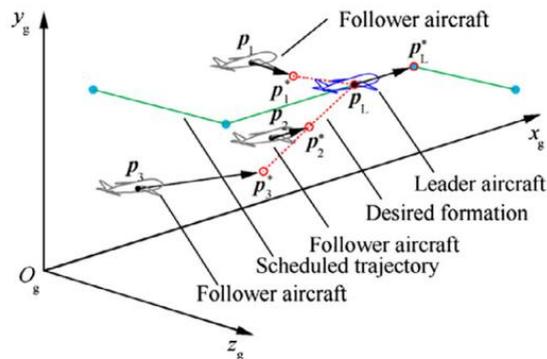
$N$  adalah jumlah aturan. Bagaimana aturan Sugeno beroperasi ditunjukkan pada gambar 2.13.



Gambar 2.13 Ilustrasi Cara Kerja Model Sugeno

### 2.2.5 Position-based Formation Control (PFC) (SUN et al., 2020)

Kita pertimbangkan sebuah sistem multi agen dengan agen berupa *quadcopter* berjumlah  $n$ . Dalam formasi, terdapat 1 *leader* dengan  $n - 1$  *follower*. *Leader* diarahkan untuk mengikuti lintasan yang sudah ditentukan, sedang masing-masing *follower* melacak posisi yang diinginkan. Anggap  $\rho_i = [x_i, y_i, z_i]^T$  dan  $\rho_i^* = [x_i^*, y_i^*, z_i^*]^T$  sebagai posisi sekarang dan posisi yang diinginkan oleh *quadcopter* ke- $i$ . Secara khusus,  $\rho_L$  dan  $\rho_L^*$  adalah posisi sekarang dan posisi yang diinginkan untuk *leader*.



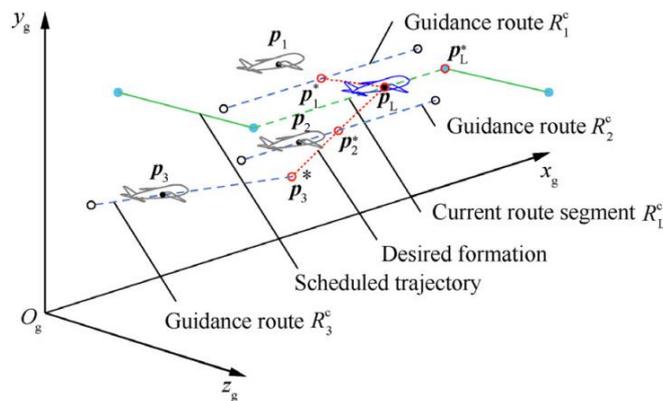
Gambar 2.14 Ilustrasi Pengaturan Formasi Berdasarkan Posisi

Seperti gambar 2.14, *leader* bergerak mengikuti lintasan sedangkan posisi *follower* yang diinginkan ditentukan oleh posisi pemimpin dan bentuk formasi yang diinginkan. Permasalahan dari PFC adalah masalah pelacakan *waypoint*. Oleh karena itu, objektif dari PFC adalah mencapai  $\rho_i \rightarrow \rho_i^*$  sambil memenuhi  $\rho_j - \rho_i = \rho_j^* - \rho_i^*$  untuk  $i, j = 1, 2, \dots, n$ .

Pendekatan PFC adalah salah satu pendekatan yang paling populer dan sering digunakan. PFC dapat dengan efektif menerapkan kontrol formasi dalam aplikasi praktis. Namun, pendekatan ini lebih cocok bila diterapkan pada agen yang bergerak dengan kecepatan rendah dan *omni-directional*.

### 2.2.5.1 Guidance-route based Formation Control (GRFC)

Metode GRFC adalah pengembangan dari PFC. Pertama-tama, *quadcopter leader* akan bergerak mengikuti rute yang sudah ditentukan. Secara bersamaan, rute panduan (*guidance route*) akan dibuat untuk tiap *follower* berdasarkan bentuk formasi yang diinginkan. Kemudian, para *follower* mengikuti rute panduan untuk menjaga formasi. Dari konsep ini, metode GRFC dapat diformulasikan.



Gambar 2.15 Ilustrasi GRFC

Kita pertimbangkan sebuah sistem multi agen dengan agen berupa *quadcopter* berjumlah  $n$  ( $1$  *leader* dengan  $n - 1$  *follower*). Pertukaran informasi antar agen ditentukan oleh sebuah graf tidak berarah. Sebuah formasi dapat ditulis dengan  $d = [d_1^T, d_2^T, \dots, d_n^T]^T$  dengan  $d_i = [R_i^c, V_i^c]^T$  dimana  $R_i^c$  adalah rute panduan dan  $V_i^c$  adalah kecepatan yang diharapkan pada rute. Protokol dirancang untuk kontrol formasi terkoordinasi yang memenuhi pernyataan:

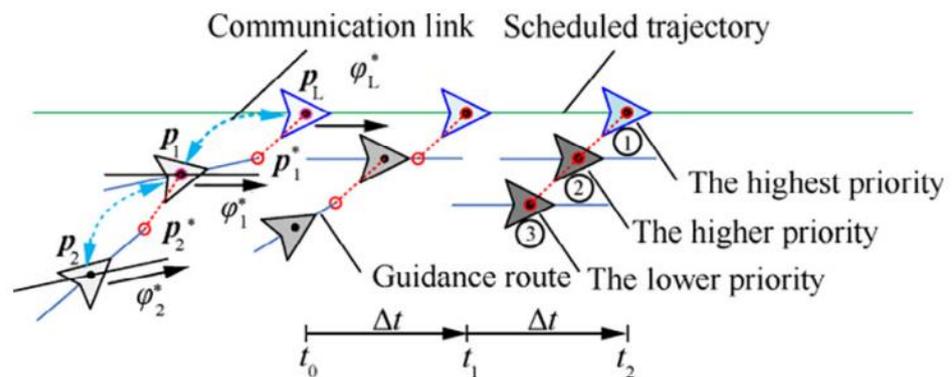
1. Pencocokan jalur penerbangan: setiap *quadcopter* dikontrol untuk melacak rute panduannya sendiri tanpa penyimpangan atau  $\lim_{t \rightarrow \infty} (R_i^c - R_i) = 0$ .  $R_i$  adalah jalur penerbangan saat ini untuk *quadcopter*  $i$ . Perlu diingat bahwa rute panduan *leader* adalah lintasan tugas tempat *leader* berada.
2. Pencocokan kecepatan maju: setiap *quadcopter* dikontrol untuk mencapai kecepatan yang diinginkan atau  $\lim_{t \rightarrow \infty} (V_i^c - V_i) = 0$ .  $V_i$  adalah kecepatan *quadcopter* ke  $i$  saat ini.

3. Penghindaran tabrakan: Diasumsikan tidak ada tabrakan yang terjadi antar *quadcopter*.

dari pertanyaan-pernyataan diatas, bisa kita simpulkan kalau GRFC terdiri dari dua permasalahan, yakni *Guidance-Route Generation (GRG)* dan *Guidance-Route Tracking (GRT)*. Keuntungan dari konstruksi ini adalah GRFC dapat memanfaatkan keunggulan dari perancangan rute untuk memenuhi pembentukan formasi pada tugas yang kompleks secara penuh, antara lain mempertahankan formasi, penghindaran rintangan, *formation rendezvous*, dan lain-lain.

Perbedaan dari rute panduan dengan rute biasa adalah rute panduan dibuat secara dinamis dalam real-time melalui interaksi antar *quadcopter* yang berdekatan untuk membentuk formasi. Pada penelitian ini, formasi yang ingin dibentuk sudah ditentukan terlebih dahulu. Namun, metode GRFC juga bisa diterapkan pada kasus yang mana formasi belum ditentukan terlebih dahulu atau berubah-ubah selama posisi yang diinginkan bisa diberikan.

Selain itu, GRFC juga bisa diterapkan pada algoritma kontrol formasi yang distributif karena GRG dan GRT setiap *quadcopter* dihitung dan dikendalikan secara independen. Hal ini berarti masing-masing *quadcopter* hanya akan berkomunikasi dan mendapat informasi dari tetangganya.



Gambar 2.16 Ilustrasi dari GRFC yang Distributif

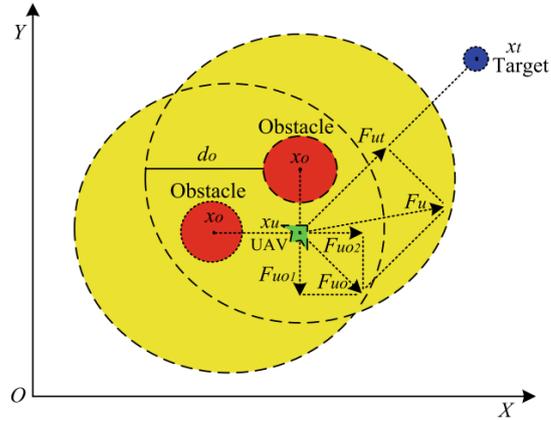
terlihat dari gambar 2.16, saat waktu  $t_0$ , *follower 1* mendapat informasi dari *leader* lalu menghitung posisi yang diinginkan dan rute panduannya. Dengan cara yang sama, *follower 2* mendapat informasi dari *follower 1* lalu menghitung posisi yang diinginkan dan rute panduannya.

### 2.2.6 Artificial Potential Field (N. Wang et al., 2021)

Perencanaan jalur (*path planning*) dan penghindaran rintangan (*obstacle avoidance*) merupakan salah dua masalah utama dalam kontrol UAV. Masalah *path planning* diklasifikasikan menjadi dua, yaitu statis dan dinamis. Pada permasalahan statis, seluruh lingkungan kerja UAV sudah diketahui atau *a priori* dan tidak mengalami perubahan. Pada permasalahan dinamis, lingkungan kerja UAV hanya diketahui sebagian atau bahkan tidak diketahui sama sekali.

Pada penelitian ini akan digunakan pendekatan APF (*Artificial Potential Field*). Dalam metode APF, robot dianggap sebagai sebuah titik yang berada pada suatu medan gaya

buatan (*artificial force field*). Medan buatan tersebut terdiri atas medan atraktif yang berasal dari target dan medan repulsif yang berasal dari rintangan. Fungsi potensial (*potential function*) merupakan jumlah dari potensial atraktif yang menarik robot ke arah tujuan (target) yang diinginkan dan total potensial repulsif yang mendorong robot menjauh dari rintangan.



Gambar 2.17 Ilustrasi APF

Konsep APF dapat dituliskan dengan persamaan:

$$U_{ut}(x_u, x_t) = 0.5\alpha_{ut}||x_u - x_t||^2$$

$$F_{ut}(x_u, x_t) = -\nabla U_{ut}(x_u, x_t) = -\alpha_{ut}||x_u - x_t||$$

$$U_{uo}(x_u, x_o) = \begin{cases} 0.5\beta_{uo} \left( \frac{1}{||x_u - x_o||} - \frac{1}{d_o} \right)^2, & ||x_u - x_o|| \leq d_o \\ 0, & ||x_u - x_o|| > d_o \end{cases}$$

$$F_{uo}(x_u, x_o) = -\nabla U_{uo}(x_u, x_o) = \begin{cases} \frac{\beta_{uo}}{||x_u - x_o||^2} \left( \frac{1}{||x_u - x_o||} - \frac{1}{d_o} \right), & ||x_u - x_o|| \leq d_o \\ 0, & ||x_u - x_o|| > d_o \end{cases}$$

dimana  $U_{ut}(x_u, x_t)$  adalah medan potensial atraktif,  $U_{uo}(x_u, x_o)$  adalah medan potensial repulsive,  $\nabla U_{ut}(x_u, x_t)$  dan  $\nabla U_{uo}(x_u, x_o)$  adalah gradien  $U_{ut}(x_u, x_t)$  dan  $U_{uo}(x_u, x_o)$  pada  $x_u$  secara berurutan.  $F_{ut}(x_u, x_t)$  adalah gaya atraktif dan  $F_{uo}(x_u, x_o)$  adalah gaya repulsif.  $||x_u - x_t||$  adalah jarak UAV ke tujuan atau target. Sedangkan  $||x_u - x_o||$  adalah jarak UAV ke rintangan.  $d_o$  adalah pengaruh jarak medan potensial repulsif rintangan terhadap UAV. Terakhir,  $\alpha_{ut}$  dan  $\beta_{uo}$  adalah koefisien *gain* dari gaya atraktif dan daya repulsif. Resultan gaya yang diterima UAV adalah...

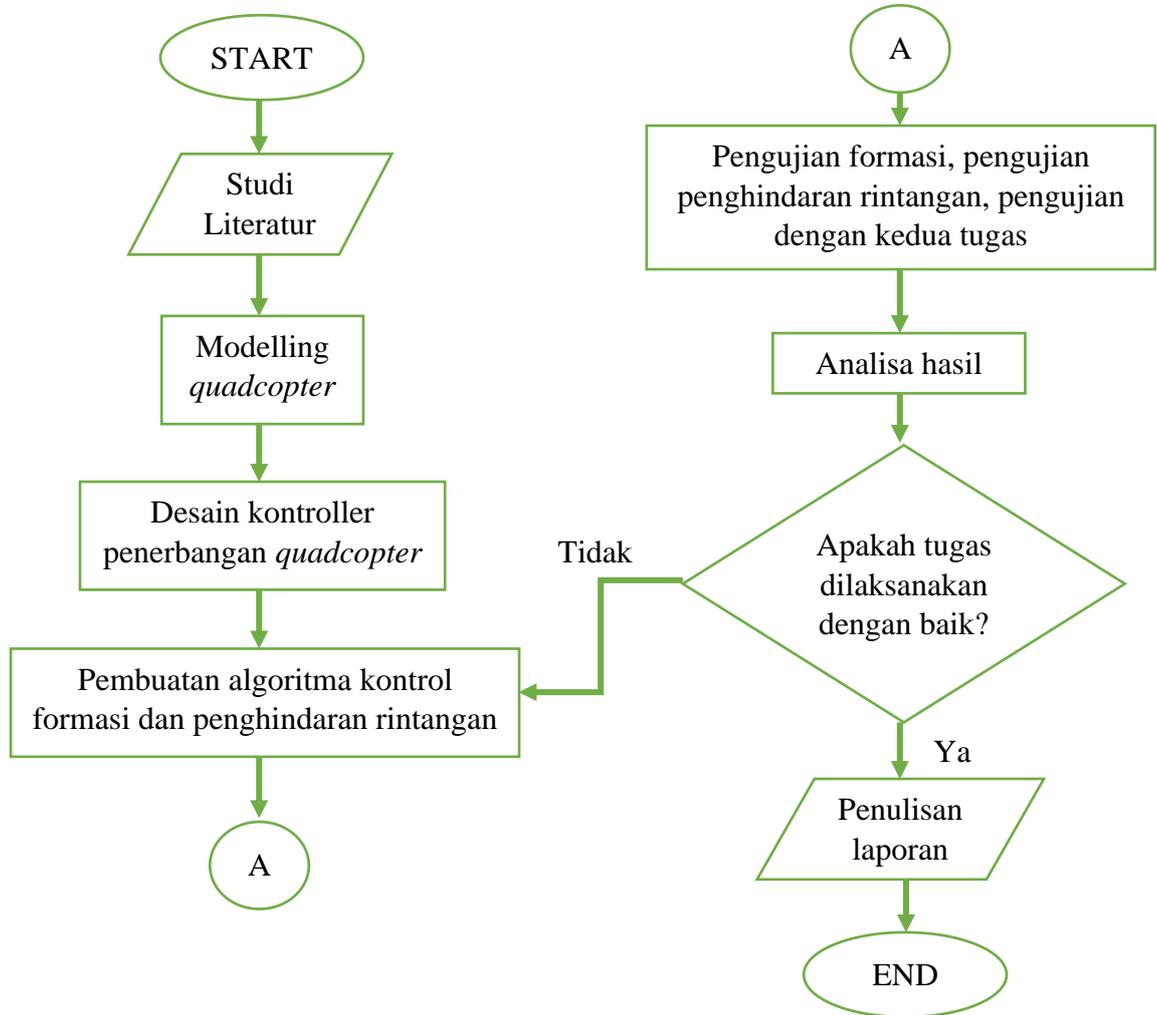
$$F_u(x_u) = F_{ut}(x_u, x_t) + F_{uo}(x_u, x_o)$$

dimana  $F_u(x_u)$  adalah resultan gaya.

*Halaman ini sengaja dikosongkan*

## BAB 3 METODOLOGI

### 3.1 Metode yang digunakan



Gambar 3.1 Diagram Alir Metodologi Penelitian

Dari gambar 3.1, bisa dilihat prosedur penilitan yang dilakukan pada tugas akhir ini. Adapun penjelasan secara garis besarnya adalah sebagai berikut:

#### 1. Studi literatur

Tahap studi literatur dilaksanakan untuk mengkaji teori-teori serta mengumpulkan data penelitian mengenai perancangan sistem multi agen untuk pembentukan formasi dan penghindaran rintangan pada *quadcopter*. Selain itu, dipelajari juga konsep *modelling* dan kontrol pada masing-masing individu *quadcopter*. Sumber-sumber yang digunakan pada studi literatur ini dipastikan memiliki kredibilitas yang terpercaya, seperti jurnal atau *paper* dari IEEE, Elsevier, dan lain-lain.

## 2. Perancangan sistem

Pada tahap perancangan sistem, pertama dilakukan pemodelan *quadcopter* yang akan dijadikan objek kontrol. Didesain pula kombinasi *state feedback controller* dan *fuzzy logic* untuk penerbangan *quadcopter*. Dilanjutkan dengan membuat algoritma kontrol formasi berdasarkan *guidance route* dan penghindaran rintangan menggunakan *Artificial Potential Field (APF)*. Terakhir, dilakukan proses penetapan parameter-parameter kontrol untuk kebutuhan simulasi.

## 3. Pengujian sistem dan evaluasi

Sistem yang sudah dirancang akan diuji menggunakan simulasi pada program MATLAB. Hasil simulasi nantinya akan diuji dan dianalisa. Salah satu pengujian yang dilakukan adalah melihat jarak antar *quadcopter* selama melakukan penerbangan. Bila terdapat ketidakcocokan dengan hasil yang diharapkan, dilakukan evaluasi dan perancangan ulang pada algoritma.

## 4. Penyusunan laporan tugas akhir

Pada tahap ini dilakukan penyusunan laporan dari hasil penelitian untuk Tugas Akhir. Bentuk akhir dari laporan adalah buku Tugas Akhir.

### 3.2 Bahan dan peralatan yang digunakan

Adapun alat dan bahan yang digunakan untuk menunjang penelitian Tugas Akhir ini adalah:

1. Laptop untuk *programming* dan penulisan laporan.
2. Program MATLAB untuk perhitungan dan simulasi.
3. Model *quadcopter* Quanser Qdrone sebagai *plant* yang dikontrol.
4. Algoritma kontrol formasi dan penghindaran rintangan untuk *quadcopter* dengan dua strategi kontrol.

### 3.3 Urutan pelaksanaan penelitian

#### 3.3.1 Modelling quadcopter

Pada tugas akhir ini, tipe *quadcopter* yang digunakan adalah Quanser Qdrone. Plant ini telah banyak digunakan untuk penelitian yang dilakukan di luar ruangan. Quanser Qdrone mampu diaplikasikan meskipun memiliki risiko kerusakan yang tinggi dan mudah bermanuver karena memiliki kerangka badan dari serat karbon. Kerangka dengan dimensi 40 cm × 40 cm × 15 cm tersebut tahan lama dan ringan.



Gambar 3.2 Quanser Qdrone

Pemodelan kinematika dan dinamika dari *quadcopter* yang digunakan telah dijelaskan secara rinci pada bab 2. Secara keseluruhan, model dinamika *quadcopter* yang digunakan adalah:

$$\dot{X} = (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \frac{U_1}{m} \quad (3.1)$$

$$\dot{Y} = (-\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi) \frac{U_1}{m} \quad (3.2)$$

$$\dot{Z} = -g + (\cos \phi \cos \theta) \frac{U_1}{m} \quad (3.3)$$

$$\dot{p} = \frac{J_{yy} - J_{zz}}{J_{xx}} qr + \frac{U_2 l}{J_{xx}} \quad (3.4)$$

$$\dot{q} = \frac{J_{zz} - J_{xx}}{J_{yy}} pr + \frac{U_3 l}{J_{yy}} \quad (3.5)$$

$$\dot{r} = \frac{J_{xx} - J_{yy}}{J_{zz}} pq + \frac{U_4 d}{J_{zz}} \quad (3.6)$$

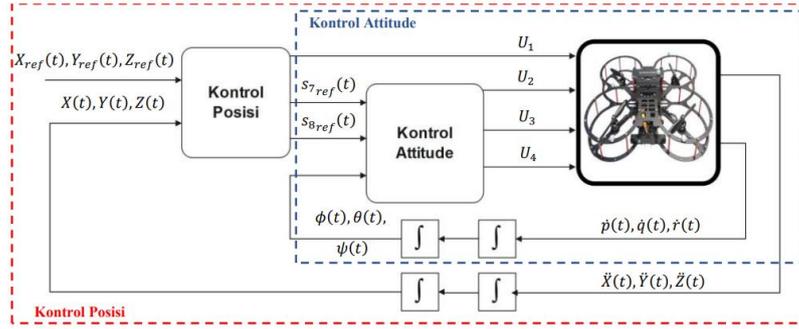
dimana  $X, Y, Z$  merupakan posisi *quadcopter* dan  $p, q, r$  merupakan kecepatan roll ( $\phi$ ), pitch ( $\theta$ ) dan yaw ( $\psi$ ). Untuk parameter *quadcopter* yang digunakan dapat dilihat pada Tabel 3.1.

Tabel 3.1 Parameter-Parameter pada Plant *Quadcopter* (Nurjanah & Fuad, 2022)

Parameter (satuan)	Simbol	Nilai
Massa ( $kg$ )	$m$	1
Percepatan gravitasi ( $\frac{m}{s^2}$ )	$g$	9.81
Momen inersia pada sumbu X ( $kg.m^2$ )	$J_{xx}$	0.03
Momen inersia pada sumbu Y ( $kg.m^2$ )	$J_{yy}$	0.03
Momen inersia pada sumbu Z ( $kg.m^2$ )	$J_{zz}$	0.04
Jarak rotor dari pusat massa ( $m$ )	$l$	0.2
Konstanta <i>drag</i> ( $N$ )	$d$	$3.13 \times 10^{-5}$

### 3.3.2 Perancangan kontrol penerbangan *quadcopter*

Skema kontrol *quadcopter* yang digunakan sebagai kontroler pelacakan dibagi menjadi dua, yaitu kontrol posisi sebagai *outer loop* dan kontrol *attitude* sebagai *inner loop* yang dapat dilihat pada Gambar 3.3. Pada kontrol posisi, yang menjadi masukan adalah nilai kesalahan dari posisi referensi  $[X_{ref}(t) \ Y_{ref}(t) \ Z_{ref}(t)]^T$  terhadap posisi *quadcopter* yang sesungguhnya  $[X(t) \ Y(t) \ Z(t)]^T$ . Keluaran dari kontrol posisi adalah  $U_1, s_{7_{ref}}(t)$ , dan  $s_{8_{ref}}(t)$ . Selanjutnya,  $s_{7_{ref}}(t), s_{8_{ref}}(t)$ , dan posisi sudut *quadcopter*  $[\phi(t) \ \theta(t) \ \psi(t)]^T$  menjadi masukan bagi kontrol *attitude* yang menghasilkan keluaran  $U_2, U_3$ , dan  $U_4$ . *State feedback controller* digunakan untuk mengontrol ketinggian ( $Z$ ) dan heading ( $\psi$ ). Kombinasi *state feedback controller* dan *fuzzy logic controller* tipe Sugeno digunakan untuk mengontrol posisi  $X$  serta  $Y$  dan *attitude*  $\phi$  serta  $\theta$ . Sistem *quadcopter* (1, 2, ...,  $n$ ) yang digunakan untuk semua *quadcopter* adalah identik.



Gambar 3.3 Blok Diagram untuk Keseluruhan Kontrol *Quadcopter*

Secara keseluruhan *fuzzy state feedback controller* digunakan untuk mengontrol gerakan *quadcopter* dari posisi mulai bergerak menuju ke posisi referensi yang diinginkan. Hal ini akan sangat bermanfaat untuk metode yang digunakan pada kontrol formasi nantinya.

### 3.3.2.1 Kontrol ketinggian

Untuk melakukan kontrol ketinggian menggunakan *state feedback controller*, persamaan (3.3) harus dimodifikasi dengan menambahkan gaya gesek udara sehingga persamaan *state* menjadi:

$$\ddot{Z} = -g + (\cos \phi \cos \theta) \frac{U_1}{m} - d \cdot \dot{Z} \quad (3.7)$$

dimana

$$U_1 = \frac{m}{\cos \phi \cos \theta} (g + K(Z_{ref} - Z) + L(\dot{Z}_{ref} - \dot{Z}) + \ddot{Z}_{ref} + d \cdot \dot{Z}) \quad (3.8)$$

jika persamaan (3.8) disubstitusikan ke persamaan (3.7), persamaan *state*  $\ddot{Z}$  akan menjadi

$$\ddot{Z} = K_1(Z_{ref} - Z) + L_1(\dot{Z}_{ref} - \dot{Z}) + \ddot{Z}_{ref} \quad (3.9)$$

dengan menyederhanakan persamaan (3.9), proses pengontrolan akan menjadi lebih mudah karena hanya perlu mencari nilai parameter  $K_1$  dan  $L_1$  yang menghasilkan output terbaik.

### 3.3.2.2 Kontrol heading ( $\psi$ )

*Heading* dari *quadcopter* akan dikontrol dengan *state feedback controller* secara regulator agar nilai  $\psi \rightarrow 0$ , dengan:

$$U_4 = \frac{J_{zz}}{d} (-K_4\psi - L_4\dot{\psi}) \quad (3.10)$$

persamaan (3.10) disubstitusikan ke persamaan (3.6) menjadi:

$$\dot{r} = \frac{J_{xx} - J_{yy}}{J_{zz}} pq - K_4\psi - L_4\dot{\psi} \quad (3.11)$$

karena pada tabel 3.1 diketahui bahwa nilai  $J_{xx} = J_{yy}$ , maka persamaan (3.11) dapat disederhanakan menjadi:

$$\dot{r} = -K_4\psi - L_4\dot{\psi} \quad (3.12)$$

### 3.3.2.3 Kontrol posisi pada sumbu X dan Y serta attitude $\phi$ dan $\theta$

Ketika *quadcopter* mendapat posisi yang diinginkan atau target maka kontrol pelacakan target akan aktif. *Fuzzy logic controller* digunakan untuk mengontrol posisi X dan Y untuk mendapatkan *output* yang akan menjadi input referensi untuk kontrol *attitude* dari  $\phi$  dan  $\theta$  (sesuai gambar 3.4). Karena *heading* ( $\psi$ ) dikontrol oleh *state feedback controller* secara regulator ( $\psi \rightarrow 0$ ), persamaan (3.1) dapat disederhanakan menjadi:

$$\begin{aligned}\ddot{X} &= (\sin \phi \sin 0 + \cos \phi \sin \theta \cos 0) \frac{U_1}{m} \\ \ddot{X} &= (\cos \phi \sin \theta) \frac{U_1}{m}\end{aligned}\quad (3.13)$$

dimana

$$\cos \phi \sin \theta = s_7 \quad (3.14)$$

Dari persamaan (3.14),  $s_7$  merupakan variabel yang akan dikontrol oleh *state feedback controller* untuk menghasilkan nilai  $U_2$ . Turunan dari  $s_7$  akan diasumsikan sama dengan  $p$ , sehingga

$$\begin{aligned}s_7 &\approx p \\ \dot{s}_7 &\approx \dot{p} = \frac{J_{yy}-J_{zz}}{J_{xx}} qr + \frac{U_2 l}{J_{xx}}\end{aligned}\quad (3.15)$$

Selanjutnya  $U_2$  akan dimodifikasi menjadi

$$U_2 = \frac{J_{xx}}{l} (K_2 s_{7ref} - K_2 s_7 - L_2 \dot{s}_{7ref} - L_2 \dot{s}_7 + \ddot{s}_{7ref}) \quad (3.16)$$

Jika persamaan (3.16) disubstitusikan ke persamaan (3.4) maka

$$\dot{s}_7 \approx \dot{p} = \frac{J_{yy}-J_{zz}}{J_{xx}} qr + K_2 s_{7ref} - K_2 s_7 - L_2 \dot{s}_{7ref} - L_2 \dot{s}_7 + \ddot{s}_{7ref} \quad (3.17)$$

Dari Tabel 3.1, diketahui bahwa nilai  $J_{yy}$  dan  $J_{zz}$  memiliki selisih yang kecil maka dapat dianggap menjadi nol. Oleh karena itu, persamaan (3.17) dapat disederhanakan menjadi:

$$\dot{s}_7 \approx \dot{p} = K_2 s_{7ref} - K_2 s_7 - L_2 \dot{s}_{7ref} - L_2 \dot{s}_7 + \ddot{s}_{7ref} \quad (3.18)$$

Sama seperti sebelumnya, persamaan (3.2) dapat disederhanakan menjadi:

$$\begin{aligned}\ddot{Y} &= (-\sin \phi \cos 0 + \cos \phi \sin \theta \sin 0) \frac{U_1}{m} \\ \ddot{Y} &= (-\sin \phi) \frac{U_1}{m}\end{aligned}\quad (3.19)$$

dimana

$$\sin \phi = S_8 \quad (3.20)$$

Dari persamaan (3.20),  $S_8$  merupakan variabel yang akan dikontrol oleh *state feedback controller* untuk mendapatkan nilai  $U_3$ . Turunan dari  $S_8$  diasumsikan sama dengan  $q$  sehingga

$$\dot{S}_8 \approx q$$

$$\dot{s}_8 \approx \dot{q} = \frac{J_{zz} - J_{xx}}{J_{yy}} pr + \frac{U_3 l}{J_{yy}} \quad (3.21)$$

Selanjutnya  $U_3$  akan dimodifikasi menjadi

$$U_3 = \frac{J_{yy}}{l} (K_3 s_{8ref} - K_3 s_8 - L_3 \dot{s}_{8ref} - L_3 \dot{s}_8 + \ddot{s}_{8ref}) \quad (3.22)$$

Jika persamaan (3.22) disubstitusikan ke persamaan (3.5) maka

$$\ddot{s}_8 \approx \dot{q} = \frac{J_{zz} - J_{xx}}{J_{yy}} pr + K_3 s_{8ref} - K_3 s_8 - L_3 \dot{s}_{8ref} - L_3 \dot{s}_8 + \ddot{s}_{8ref} \quad (3.23)$$

Dari Tabel 3.1, diketahui bahwa nilai  $J_{xx}$  dan  $J_{zz}$  memiliki selisih yang kecil maka dapat dianggap menjadi nol. Oleh karena itu, persamaan (3.17) dapat disederhanakan menjadi:

$$\ddot{s}_8 \approx \dot{q} = K_3 s_{8ref} - K_3 s_8 - L_3 \dot{s}_{8ref} - L_3 \dot{s}_8 + \ddot{s}_{8ref} \quad (3.24)$$

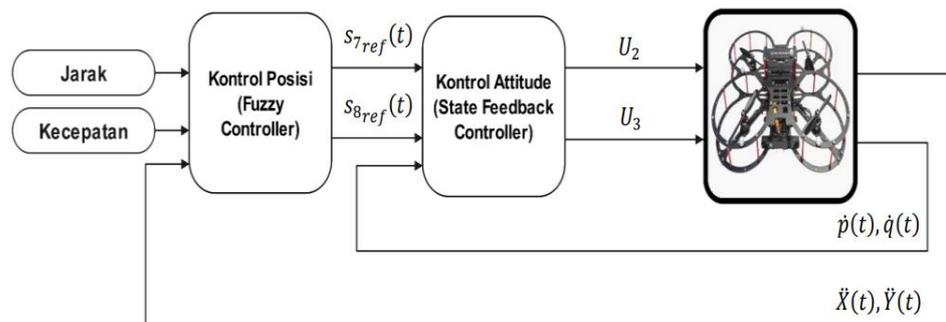
Nilai parameter *state feedback controller* didapatkan melalui proses *trial and error* dengan memberikan titik-titik target atau *waypoint*, pada saat *quadcopter* berhasil mencapai semua titik target yang ditentukan, maka parameter dapat digunakan. Nilai parameter yang digunakan tertulis pada tabel 3.2.

Tabel 3.2 Nilai Parameter-Parameter *State Feedback Controller*

Parameter	Nilai	Parameter	Nilai
$K_1$	16	$L_1$	9
$K_2$	100	$L_2$	21
$K_3$	100	$L_3$	21
$K_4$	0.09	$L_4$	0.61

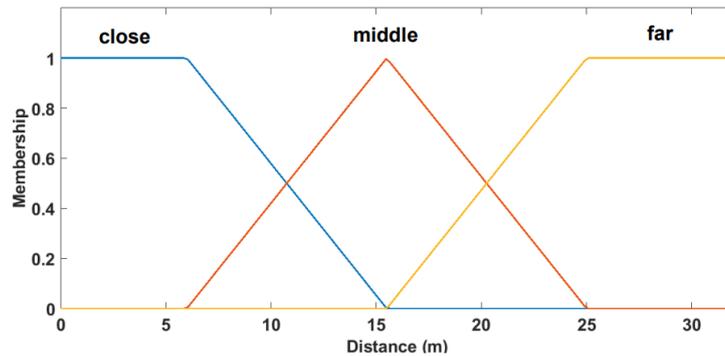
Kontrol *attitude*  $\phi$  dan  $\theta$  menggunakan *state feedback controller* memerlukan *input* referensi ( $s_{7ref}, \dot{s}_{7ref}, \ddot{s}_{7ref}, s_{8ref}, \dot{s}_{8ref}, \ddot{s}_{8ref}$ ) agar bisa menghasilkan *output*  $U_2$  dan  $U_3$  yang diinginkan. Nilai referensi ini merupakan hasil keluaran dari *fuzzy logic controller*.

*fuzzy logic controller* yang dipakai memiliki dua masukan, yaitu jarak dan kecepatan dari *quadcopter* menuju target. Skemanya dapat dilihat pada Gambar 3.4. Sedangkan keluarannya merupakan referensi untuk sudut *quadcopter*.



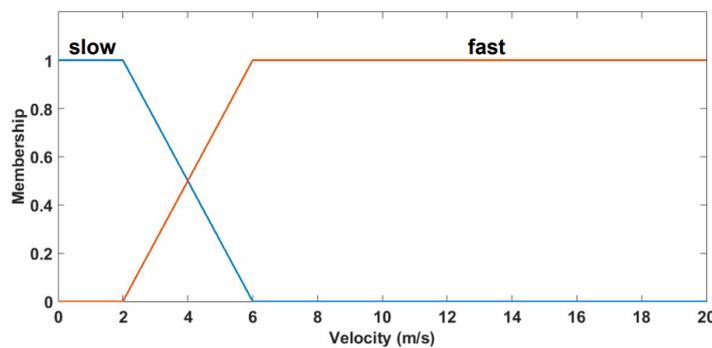
Gambar 3.4 Blok Diagram Kontrol Posisi X serta Y dan *Attitude*  $\phi$  serta  $\theta$

Diasumsikan jarak adalah selisih antara posisi ( $X$  dan  $Y$ ) *quadcopter* saat ini terhadap posisi targetnya. Gambar 3.5 merupakan ilustrasi fungsi keanggotaan (*membership function*) untuk masukan jarak. Jika selisih antara posisi *quadcopter* dan target lebih besar dari 25 meter maka masuk ke kategori *far* (jauh). Untuk kategori *middle* (sedang), jika jarak antara *quadcopter* dan target berada diantara 6 meter dan 25 meter. Untuk jarak yang lebih kecil dari 6 meter akan masuk ke kategori *close* (dekat).



Gambar 3.5 *Membership Function* untuk Masukan Jarak pada *fuzzy*

Sedangkan kecepatan adalah selisih kecepatan ( $\dot{X}$  dan  $\dot{Y}$ ) *quadcopter* saat ini terhadap kecepatan targetnya. Jika selisih antara kecepatan *quadcopter* dan target lebih besar dari 4 m/s maka termasuk kategori *fast* (kencang). Sedangkan untuk kategori *slow* (pelan), jika selisih kecepatan *quadcopter* dan target lebih kecil dari 4 m/s.



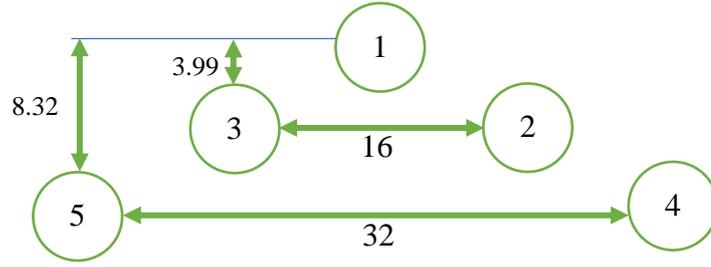
Gambar 3.6 *Membership Function* untuk Masukan Kecepatan pada *Fuzzy*

Terakhir, nilai keluaran dari controller *fuzzy* adalah  $-0.5 < ref < 0.5$ . Dengan asumsi kemiringan maksimal UAV sebesar  $30^\circ$ . Rule base yang berlaku adalah

- Jika jarak adalah jauh,  $ref = 0.5$
- Jika jarak adalah sedang dan kecepatan adalah kencang,  $ref = 0.3$
- Jika jarak adalah dekat dan kecepatan adalah kencang,  $ref = -0.5$
- Jika jarak adalah dekat dan kecepatan adalah pelan,  $ref = 0$

### 3.3.3 Perancangan kontrol formasi

Untuk tugas akhir ini, bentuk formasi yang diinginkan adalah *Vformation*. Gambar 3.7 dan Tabel 3.3 mendeskripsikan bentuk V dengan lebih jelas dan lengkap. Gambar 3.7 menunjukkan pola formasi yang diinginkan dengan jarak antar robot pada sumbu x dan sumbu y. Tabel 3.3 menunjukkan jarak antar robot secara keseluruhan.



Gambar 3.7 Spesifikasi Bentuk Formasi V

Tabel 3.3 Jarak Antar Robot pada Formasi V

Robot	1	2	3	4	5
1	0	8.9443	8.9443	17.8904	17.8904
2	8.9443	0	16.0034	8.9443	24.3311
3	8.9443	16.0034	0	24.3311	8.9443
4	17.8904	8.9443	24.3311	0	32.0017
5	17.8904	24.3311	8.9443	32.0017	0

Konsep yang digunakan untuk kontrol formasi adalah *distributed guidance route*, quadcopter *leader* memberikan rute panduan kepada dua *follower* tetangganya (*follower* 2 dan 3) lalu kedua *follower* tersebut akan memberikan rute panduan ke dua *follower* lainnya lagi (*follower* 4 dan 5). *Leader* sendiri akan bergerak mengikuti rute ke titik target utama kelompok yang bernilai konstan dan sudah ditentukan dari awal. Rute panduan yang diberikan oleh *leader* kepada *follower*-nya berupa titik tujuan yang berubah secara dinamis berdasarkan posisi *leader*. Didefinisikan  $X_{target_i}(t) = [x_{target} \ y_{target} \ z_{target}]$  merupakan titik tujuan *quadcopter* ke-*i* tiap satuan waktu *t*. Titik tujuan masing-masing *quadcopter follower* dapat dituliskan sebagai berikut:

$$X_{target_2}(t) = [x_{leader} + 2l \sin v \quad y_{leader} - 6l \cos v \quad z_{leader}] \quad (3.25)$$

$$X_{target_3}(t) = [x_{leader} - 6l \sin v \quad y_{leader} + 2l \cos v \quad z_{leader}] \quad (3.26)$$

$$X_{target_4}(t) = [x_{follower_2} + 2l \sin v \quad y_{follower_2} - 6l \cos v \quad z_{follower_2}] \quad (3.27)$$

$$X_{target_5}(t) = [x_{follower_3} - 6l \sin v \quad y_{follower_3} + 2l \cos v \quad z_{follower_3}] \quad (3.28)$$

*l* adalah panjang atau jarak formasi yang kita inginkan, *v* adalah sudut dari formasi. Terdapat pula sebuah konstanta bernilai 2 dan 6 disana yang sebenarnya juga bisa diubah-ubah sesuai dengan bentuk formasi yang diinginkan (dan bergantung pada nilai *l* dan *v*). Bisa dilihat dari persamaan (3.25) dan persamaan (3.26) kalau target *quadcopter* 2 dan 3 bergantung pada posisi *leader*. Kombinasi nilai  $x_{target}$  dan  $y_{target}$  antara keduanya disesuaikan agar berada di kanan dan kiri *leader*. Sedangkan  $z_{target}$  sama dengan ketinggian *leader*. Begitu pula dengan *quadcopter* 4 dan 5, namun karena mereka adalah *follower* tingkat dua maka mereka mendapat panduan dari *quadcopter* 2 dan 3 secara berurutan. Bila formasi ingin diubah, yang perlu dilakukan hanyalah merubah perumusan posisi target pada persamaan (3.25) hingga (3.28). Dengan kata lain, metode ini adalah metode yang sangat fleksibel.

Titik target dari masing-masing *quadcopter* ini kemudian dijadikan masukan ke algoritma penghindaran rintangan (dijelaskan pada sub bab 3.3.4) sehingga *quadcopter* akan bergerak untuk mencapai formasi yang diinginkan sekaligus menghindari rintangan. Selain itu, algoritma penghindaran rintangan ditambah dengan mekanisme baru untuk memastikan formasi terbentuk dan terjaga dengan lebih baik.

### 3.3.4 Perancangan algoritma penghindaran rintangan

Konsep perancangan algoritma *Artificial Potential Field* (APF) dalam ruang tiga dimensi adalah target baik diam maupun bergerak menghasilkan gaya tarik, sedangkan halangan statis (seperti bangunan atau pohon) dan dinamis (*quadcopter* lainnya) akan menghasilkan gaya tolak. Dengan kata lain, target akan menghasilkan medan potensial tarik. Sebaliknya, halangan akan menghasilkan medan potensial tolak. Adapun APF yang kita gunakan adalah metode APF yang sudah dimodifikasi, yakni *optimized APF*. Kita kembangkan algoritma kontrol melalui sebuah scenario bahwa satu *quadcopter* terbang dalam ruang tiga dimensi dan posisinya adalah  $X = (x, y, z)^T$ . Kedua medan, potensial tarik  $U_{att}(X)$  dan tolak  $U_{rep}(X)$  akan membentuk sebuah medan total yang bisa direpresentasikan sebagai berikut:

$$U_{APF}(X) = U_{att}(X) + U_{rep}(X) \quad (3.29)$$

Didefinisikan

$$U_{att}(X) = \frac{1}{2}k_{att}(X - X_{target})^2 \quad (3.30)$$

dimana  $k_{att}$  adalah konstanta *gain* atraktif,  $X$  adalah vektor posisi *quadcopter*, dan  $X_{target}$  adalah vektor posisi dari target. Sedangkan untuk medan gaya tolaknya

$$U_{rep}(X) = \begin{cases} \frac{1}{2}k_{rep} \left( \frac{1}{\rho(X)} - \frac{1}{\rho_0} \right)^2 (X - X_{target})^n, & \text{untuk } \rho(X) \leq \rho_0 \\ 0, & \text{untuk } \rho(X) > \rho_0 \end{cases} \quad (3.31)$$

dimana  $n$  adalah bilangan *real* sembarang yang nilainya lebih besar dari nol,  $k_{rep}$  adalah konstanta *gain* repulsif,  $\rho(X)$  adalah jarak antara posisi *quadcopter* saat ini terhadap halangan,  $\rho_0$  adalah jarak benturan maksimum, bergantung pada kecepatan gerak dan perlambatan *quadcopter*. Jika jarak antara *quadcopter* dan halangan lebih besar dari  $\rho_0$ , medan potensial tolak tidak akan mempengaruhi gerakan dari UAV.  $(X - X_{target})^n = |(x - x_{target})^n| + |(y - y_{target})^n| + |(z - z_{target})^n|$  adalah jarak relatif antara *quadcopter* dan titik target.

Untuk mendapatkan gaya tolak dan gaya tarik, dihitung nilai negatif gradien dari medan potensial tolak dan tarik.

$$F_{att}(X) = -grad(U_{att}) = -k_{att}(X - X_{target}) \quad (3.32)$$

sedangkan

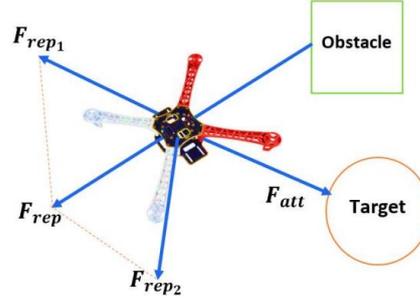
$$F_{rep}(X) = -grad(U_{rep}) = \begin{cases} F_{rep1}(X) + F_{rep2}(X), & \text{untuk } \rho(X) \leq \rho_0 \\ 0, & \text{untuk } \rho(X) > \rho_0 \end{cases} \quad (3.33)$$

yang mana

$$F_{rep1}(X) = k_{rep} \left( \frac{1}{\rho(X)} - \frac{1}{\rho_0} \right) \frac{1}{\rho(X)^2} (X - X_{target})^n \frac{\partial \rho(X)}{\partial(X)} \quad (3.34)$$

$$F_{rep2}(X) = -\frac{n}{2} k_{rep} \left( \frac{1}{\rho(X)} - \frac{1}{\rho_0} \right)^2 (X - X_{target})^{n-1} \frac{\partial(X - X_{target})}{\partial(X)} \quad (3.35)$$

Dari persamaan (3.34) dan persamaan (3.35) terlihat bahwa  $F_{rep}(X)$  diuraikan menjadi  $F_{rep1}(X)$  dan  $F_{rep2}(X)$  dimana  $F_{rep1}(X)$  adalah komponen gaya dalam arah garis antara *quadcopter* dan halangan, dan  $F_{rep2}(X)$  adalah komponen gaya dalam arah garis antara *quadcopter* dan target, seperti gambar 3.8.



Gambar 3.8 Komponen Gaya pada *Quadcopter* Berdasarkan Konsep APF

Salah satu konsep penting lain yang perlu diperhatikan adalah sistem penghindaran rintangan berdasarkan ketinggian. Rintangan yang akan digunakan pada tugas akhir ini memiliki bentuk tabung. Konsep APF yang diterapkan sekarang hanya dapat melakukan penghindaran secara horisontal. Jadi, meskipun penghindaran dengan menambah ketinggian dirasa lebih efektif dan efisien, hal itu tidak dapat dilakukan. Oleh karena itu, algoritma APF repulsif akan bekerja bila  $\rho(X) \leq \rho_0$  dan ketinggian *quadcopter* dikurangi 0.075 meter lebih kecil daripada tinggi rintangan ( $z - 0.075 \leq h_{rintangan}$ ). Pengurangan 0.075 (0.075 adalah setengah dari tinggi Quanser Qdrone dalam meter) diterapkan karena umumnya pada praktik di dunia nyata, ketinggian *quadcopter* dihitung dari pusat massanya yang biasanya berada tepat di tengah *body quadcopter*. Dengan kata lain, bila kita hanya menggunakan ( $z \leq h_{rintangan}$ ) maka *body quadcopter* bisa saja menabrak rintangan.

Seperti yang sudah dijelaskan pada bab I, terdapat dua strategi kontrol yang akan digunakan dan dibandingkan. Pertama, strategi dimana membentuk dan mempertahankan formasi lebih penting daripada penghindaran rintangan. Pada strategi ini, kelompok *quadcopter* akan berusaha untuk menjaga formasi sepanjang waktu meskipun bertemu dengan halangan. Oleh karena itu, bisa saja jarak yang ditempuh untuk masing-masing *quadcopter* lebih panjang daripada yang seharusnya. Strategi pertama disebut penghindaran total. Sebaliknya, strategi kedua akan mementingkan penghindaran rintangan daripada formasi dan disebut penghindaran minimal. Karena tidak harus selalu mempertahankan formasi, bisa saja satu atau lebih *quadcopter* bergerak ke arah yang berbeda dari formasi sehingga memecah formasi. Untuk membuat kedua strategi, radius penghindaran  $\rho_0$  pada APF akan dimodifikasi.

Pada penghindaran total, formasi akan diperlakukan sebagai satu *body*, bergerak lebih kompak, dan tidak mudah terpecah, terutama saat penghindaran rintangan. Oleh karena itu, diberikan modifikasi terhadap radius penghindaran ( $\rho_0$ ), terutama milik *leader*. Nilai  $\rho_0$

diambil dari jarak minimum *quadcopter* terluar terhadap halangan sehingga formasi akan menghindari dengan jarak aman untuk seluruh *quadcopter*. Untuk mencapai itu, dimanfaatkan logika *if* pada program, bila radius atau jari-jari dari halangan berukuran 10 atau kurang, diambil  $\rho_0 = 20$ . Bila radius halangan berukuran lebih dari 10, diambil  $\rho_0 = \min(2 * radius, 20)$ . Nilai 20 diambil karena dalam formasi yang dibentuk, *quadcopter leader* memiliki jarak sekitar 16 meter terhadap *follower* terluarnya.

Sedangkan untuk penghindaran minimal, tentu formasi tetap akan terbentuk dan dipertahankan namun bisa terpecah dengan lebih mudah karena penghindaran tidak mempertimbangkan ukuran dari formasi. Pada kasus tertentu, beberapa *quadcopter* harus melepaskan diri sebelum kembali ke formasi setelah menghindari rintangan. Tiap *quadcopter* memiliki radius penghindaran yang berubah-ubah relatif terhadap ukuran dari halangan.  $\rho_0 = a * radius$  dan nilainya tetap sepanjang simulasi.  $a$  adalah konstanta yang nilainya bisa disesuaikan dengan keinginan.

Selain rintangan statis, ada pula rintangan yang sifatnya dinamis, yaitu *quadcopter* lain. Karena dianggap sebagai rintangan, *quadcopter* lain juga akan menghasilkan tolakan. Dalam praktiknya, jarak aman *quadcopter* biasanya jauh lebih besar daripada volume *quadcopter* itu sendiri. Medan potensial tolak yang dihasilkan oleh UAV lainnya adalah:

$$U_{quad}(i) = \begin{cases} \sum_{j=1}^m \frac{1}{2} k_{rep} \left( \frac{1}{\rho(X_{ij})} - \frac{1}{\rho_0} \right)^2 (X_i - X_{target})^n, & \text{untuk } \rho(X_{ij}) \leq \rho_{uav} \\ 0, & \text{untuk } \rho(X_{ij}) > \rho_{uav} \end{cases} \quad (3.36)$$

$\rho(X_{ij})$  adalah jarak *euclidean* antara *quadcopter* ke- $i$  dan *quadcopter* ke- $j$ ,  $m$  adalah jumlah *quadcopter*,  $\rho_{uav}$  adalah radius pengaruh medan potensial *repulsive* terhadap *quadcopter*. Jarak antar UAV dapat dihitung dengan persamaan berikut:

$$\rho(X_{ij}) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (3.37)$$

Sehingga dapat dihitung gaya tolak yang dihasilkan oleh *quadcopter* lain,  $F_{quad}(i)$ , adalah nilai negatif dari gradien medan potensial tolak pada persamaan (3.36).

Keluaran akhir dari algoritma penghindaran rintangan adalah gaya total kombinasi dari gaya atraktif dan repulsif.

$$F_{total}(X) = F_{att}(X) + F_{rep}(X) + F_{quad}(i) \quad (3.38)$$

Gaya total ini memiliki tiga nilai, yaitu terhadap sumbu  $x$ ,  $y$ , dan  $z$ . Kemudian gaya dijumlahkan dengan posisi *quadcopter* sekarang dan menjadi referensi posisi bagi controller penerbangan. Adapun untuk mendukung proses mempertahankan formasi, nilai  $F_{total}(X)$  akan diubah-ubah bila memenuhi kriteria tertentu

- Untuk *quadcopter leader*:  
Jika jarak antara *quadcopter leader* dan salah satu *follower* terlampaui jauh melebihi 4 meter (untuk *follower* 2 dan 3) atau 8 meter (untuk *follower* 4 dan 5) maka nilai  $F_{total}((x, y)) = \frac{F_{total}((x, y))}{100}$

- Untuk *quadcopter follower*:  
Jika jarak antara posisi *quadcopter* sekarang dengan titik tujuan terlampaui jauh melebihi dari 0 meter maka nilai  $F_{total}((x, y)) = F_{total}((x, y)) \cdot 15$ .

Dengan begitu, *quadcopter leader* akan melambat untuk menunggu *quadcopter follower*-nya yang tertinggal jauh. Di saat yang sama, *follower* akan mempercepat dirinya bila masih jauh dari posisi yang diharapkan. Mekanisme ini akan menghasilkan pergerakan yang lebih mulus bagi semua *quadcopter*.

### 3.3.5 Indeks Performansi Formasi

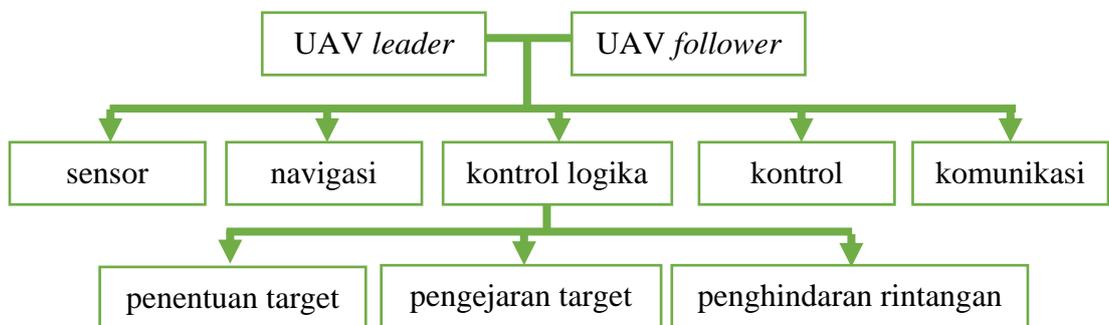
Untuk melihat bagaimana performa *quadcopter* menjaga formasi, dibuat sebuah indeks performansi formasi modifikasi, yang didefinisikan sebagai berikut:

$$I(t) = \sum_{i=2}^5 \left| \left( \|X_{leader} - X_i\| - \left\| X_{leader} - X_{target_i} \right\| \right) \right| \quad (3.39)$$

dimana  $I(t)$  adalah indeks performansi tiap satuan waktu,  $X_{leader}$  adalah posisi *leader*,  $X_i$  adalah posisi *follower* ke- $i$ ,  $X_{target_i}$  adalah posisi *follower* ke- $i$  yang diinginkan. Konsep indeks ini adalah membandingkan formasi yang terbentuk dengan formasi yang idealnya terbentuk berdasarkan jarak masing-masing *quadcopter* terhadap *leader*. Indeks akan bernilai nol jika dan hanya jika jarak antar *quadcopter* pada formasi yang terbentuk sama dengan jarak antar *quadcopter* yang diinginkan seperti pada spesifikasi formasi.

### 3.3.6 Skema Keseluruhan Sistem

Skema keseluruhan sistem dapat dilihat dari gambar 3.9. Sistem yang digunakan pada seluruh *quadcopter* adalah identik.



Gambar 3.9 Skema Keseluruhan Sistem

## BAB 4 Hasil dan Pembahasan

### 4.1 Hasil penelitian

Sub-bab ini akan menunjukkan dan menjelaskan hasil simulasi yang dilakukan pada program MATLAB berdasarkan metode kontrol yang sudah dibentuk. Terdapat beberapa pengujian yang dilakukan agar kemampuan dari tiap algoritma bisa terlihat dengan jelas.

Untuk keseluruhan pengujian kecuali penghindaran rintangan, terdapat sekelompok *quadcopter* yang berjumlah 5 buah. Masing-masing *quadcopter* memiliki titik awal yang berbeda-beda seperti yang tertulis pada tabel 4.1.

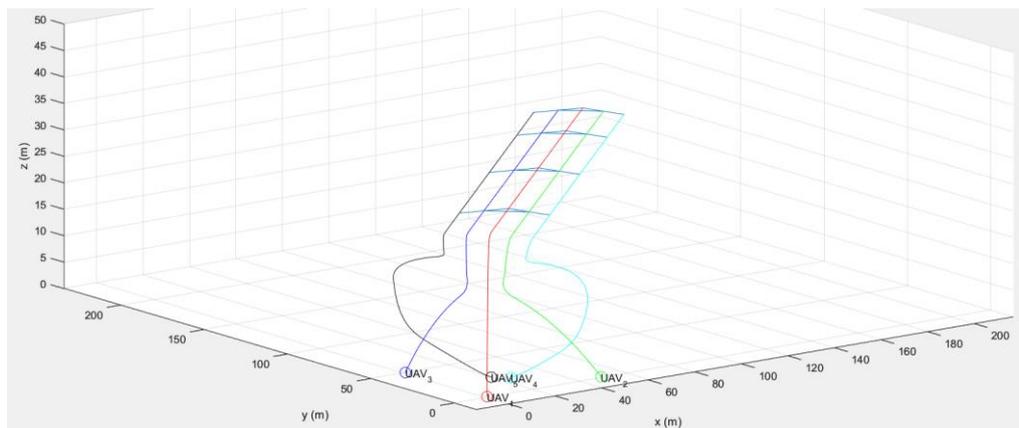
Tabel 4.1 Posisi Awal Masing-Masing *Quadcopter*

<i>quadcopter</i>	1	2	3	4	5
x (m)	1	50	1	25	20
y (m)	1	1	50	20	25
z (m)	0	0	0	0	0

Target atau tujuan dari *quadcopter leader* (*quadcopter* 1) selalu berada di titik  $X_{target1} = (500, 500, 30)^T$  yang mana juga menjadi titik tujuan kelompok.

#### 4.1.1 Uji pembentukan formasi

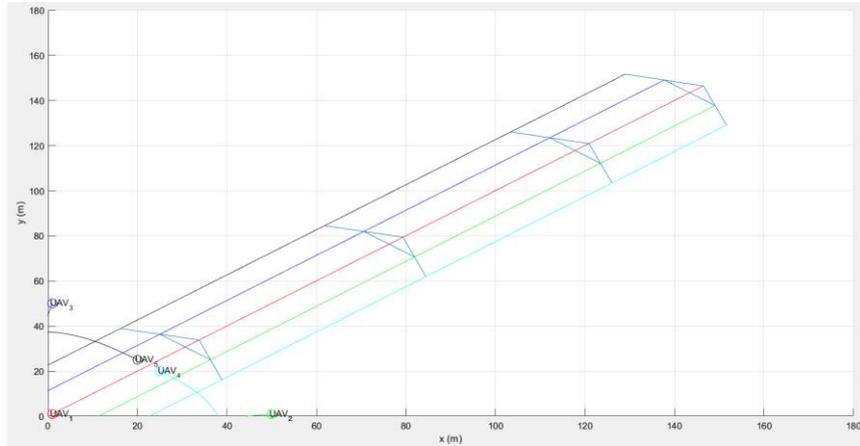
Pertama, kita lakukan pengujian untuk membentuk formasi pada lingkungan yang ideal atau tanpa rintangan apapun. Dipilih nilai  $K_{rep} = 0.04$ ,  $K_{att} = 0.3$ ,  $l = 2$ , dan  $v = \frac{\pi}{2}$ . Meskipun tidak ada rintangan,  $K_{rep}$  dan  $K_{att}$  tetap diperlukan untuk menggerakkan *quadcopter* menuju titik tujuan. Sedangkan  $l$  dan  $v$  digunakan untuk membentuk formasi yang diinginkan. Perlu diingat bahwa formasi yang digunakan adalah formasi V. Selain itu, dipakai konsep iterasi untuk menggambarkan waktu yang berjalan pada program. Dipilih  $Iterasi = 50000$ , semakin tinggi nilainya maka simulasi akan berjalan lebih lama.



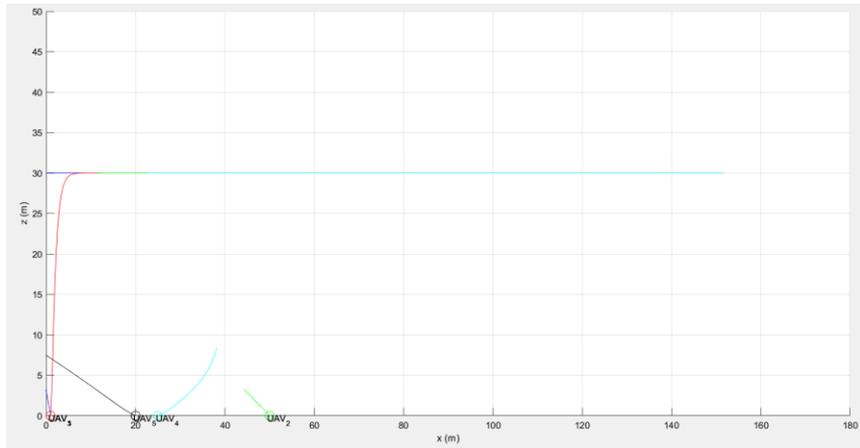
Gambar 4.1 Trayektori Gerak *Quadcopter* Dalam Formasi

Dapat dilihat melalui gambar 4.1 kalau kelompok *quadcopter* telah berhasil membentuk formasi sejak awal iterasi. Mereka bergerak naik terlebih dahulu hingga mencapai tinggi target lalu bergerak pada sumbu X dan Y ke posisi target. Diberikan pula

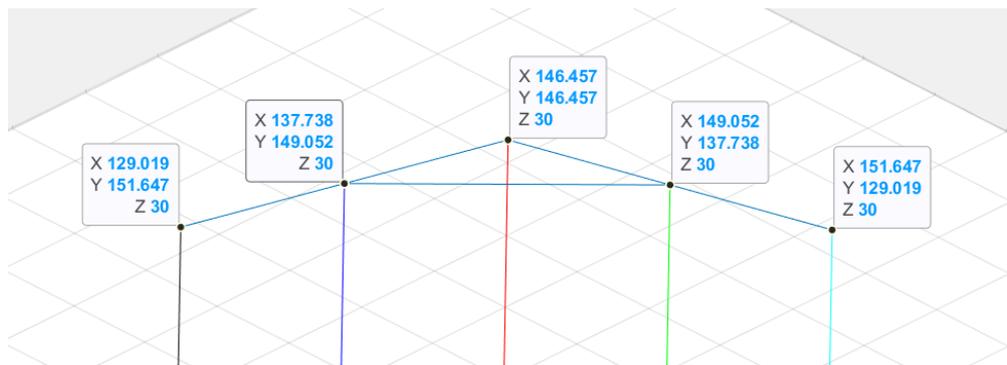
garis antar *quadcopter* untuk menunjukkan formasi seperti apa yang terbentuk pada beberapa waktu iterasi. Bentuknya konstan selama iterasi.



Gambar 4.2 Trayektori Gerak *Quadcopter* Tampak Atas



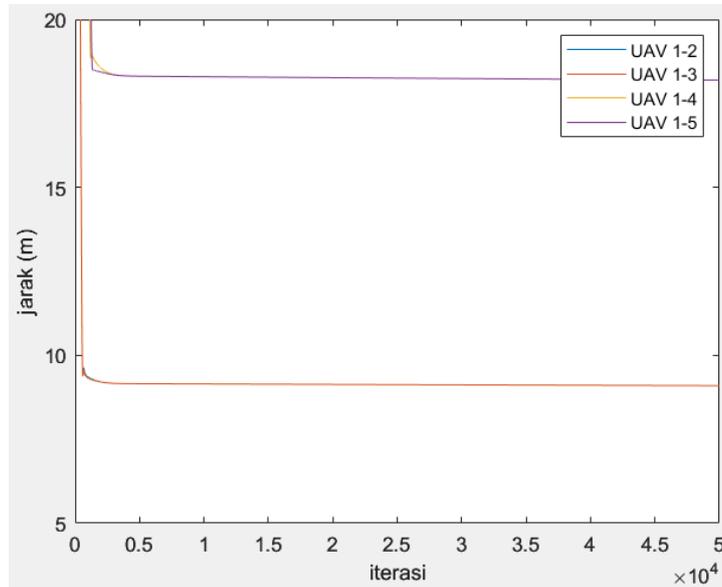
Gambar 4.3 Trayektori Gerak *Quadcopter* Tampak Samping



Gambar 4.4 Trayektori Gerak *Quadcopter* dengan Ukuran Diperbesar dan Menunjukkan Nilai Saat Itu

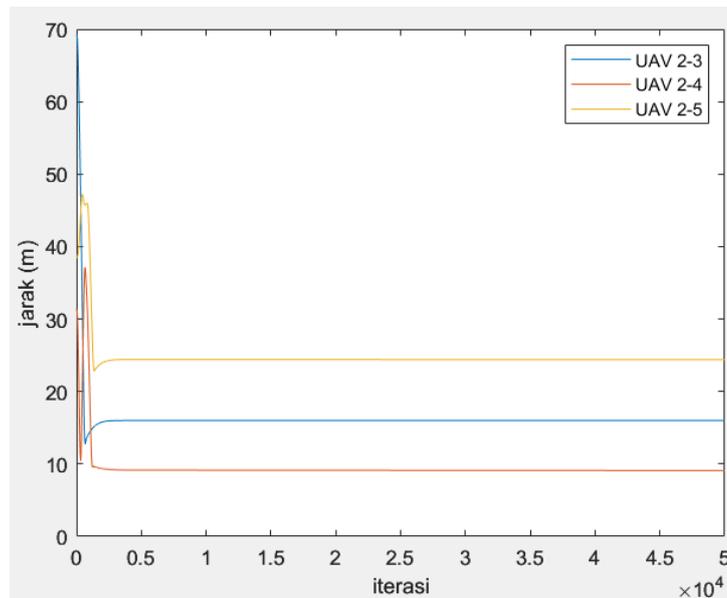
Gambar 4.4 memberikan penampakan yang lebih jelas untuk formasi yang terbentuk. Karena bergerak di sepanjang garis  $x = y$ , posisi X dan Y *quadcopter* 2 adalah kebalikan dari posisi X dan Y *quadcopter* 3. Hal ini berarti jarak *quadcopter* 1 ke *quadcopter* 2 sama

dengan *quadcopter* 1 ke *quadcopter* 3. Pernyataan-pernyataan ini juga berlaku pada *quadcopter* 4 dan 5.



Gambar 4.5 Jarak Antara *Quadcopter* 1 Terhadap *Follower*-nya

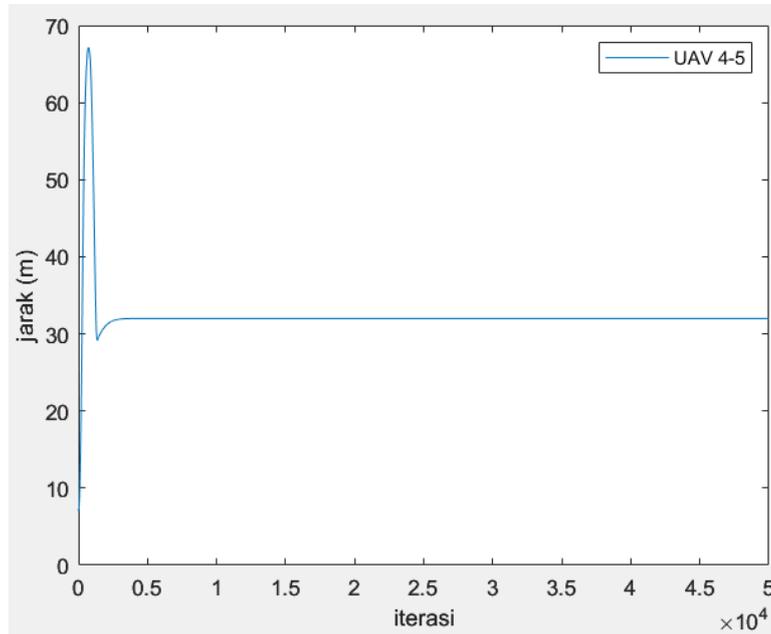
Jarak *quadcopter* 1 ke *quadcopter* 2 dan 3 adalah sekitar 9.1 meter. Sedangkan jarak *quadcopter* 1 ke *quadcopter* 4 dan 5 adalah sekitar 18.2 meter. Nilainya selalu konstan sepanjang waktu. Terjadi penambahan jarak pada awal iterasi untuk *quadcopter* 4 dan 5 karena kedua *quadcopter* bergerak menjauh terlebih dahulu baru menuju formasi yang diinginkan. Hal ini terjadi akibat posisi awal *quadcopter* 2 dan 3. Hasil berbeda bisa didapatkan dengan penempatan awal yang berbeda.



Gambar 4.6 Jarak *Quadcopter* 2 Terhadap *Quadcopter Follower* Lain

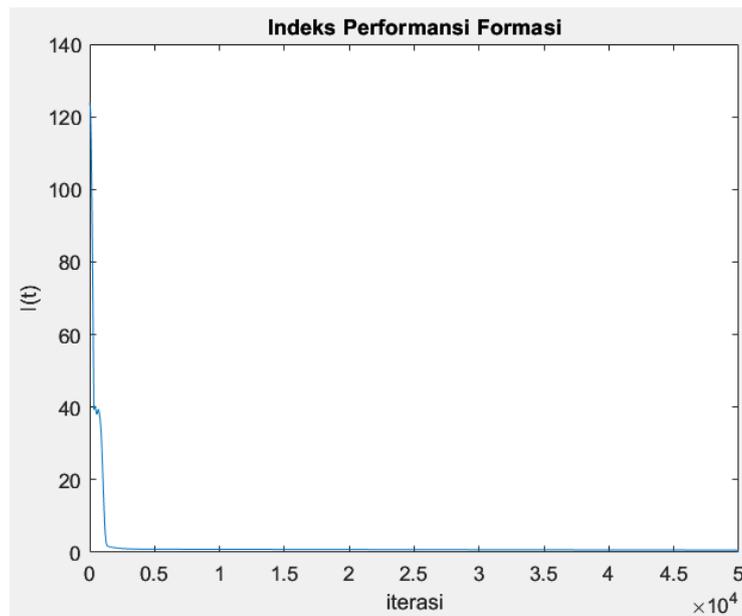
Untuk *quadcopter* 2, jarak ke *quadcopter* 4 tentu adalah yang paling kecil dengan nilai sekitar 9 meter. Jarak ke *quadcopter* 3 selalu sama, yaitu 16 meter, sesuai dengan formasi

yang diinginkan (pada sub-bab 3.3.3). Terlihat kalau *quadcopter* 4 dan 5 juga mengalami kenaikan jarak terhadap *quadcopter* 2 pada awal iterasi seperti terhadap *quadcopter* 1. Gambar 4.7 menunjukkan jarak *quadcopter* 4 dan 5 yang konstan di 32 meter.



Gambar 4.7 Jarak *Quadcopter* 4 Terhadap *Quadcopter* 5

Jarak antar *quadcopter* yang terjadi sudah cukup mirip dari yang seharusnya (titik target) dengan sedikit *error* sekitar 0.2 meter pada *steady state*-nya. Bila kita lihat menggunakan indeks performansi formasi, didapatkan *steady state* pada sekitar nilai 0. Hanya terdapat sedikit *overshoot* diawal karena penyesuaian *quadcopter* dari titik awal ke formasi. Nilai ini bisa dikatakan cukup rendah dan menunjukkan kalau formasi terjaga sepanjang waktu. Rerata dari indeks performansi di gambar 4.8 adalah 1,8888 karena.



Gambar 4.8 Indeks Performansi Formasi pada Pengujian Formasi

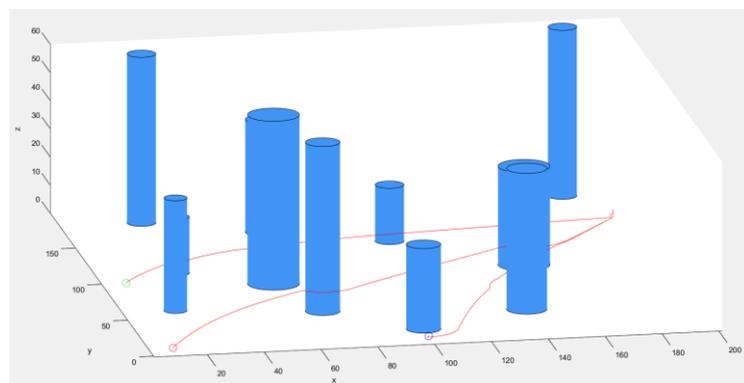
### 4.1.2 Uji penghindaran rintangan

Dengan formasi yang sudah cukup bagus, pengujian selanjutnya adalah penghindaran rintangan berdasarkan algoritma *optimized Artificial Potential Field (APF)*. Diasumsikan terdapat tiga *quadcopter* yang bergerak secara individu tanpa membentuk formasi menuju ke satu titik tujuan yang sama. Mereka bergerak di lingkungan yang kompleks dengan banyak halangan. Tiap halangan memiliki posisi serta ketinggian yang berbeda-beda. Informasi soal tinggi menjadi penting karena bila tinggi rintangan dibawah ketinggian *quadcopter* maka hanya perlu dilewati saja. Ada pula radius rintangan untuk menggambarkan luas rintangan.

Tabel 4.2 Posisi Awal *Quadcopter*, Posisi Target, dan Posisi Rintangan

Nama Objek	x (m)	y (m)	z (m)	Radius (m)
<i>Quadcopter 1</i>	10	10	0	-
<i>Quadcopter 2</i>	100	10	0	-
<i>Quadcopter 3</i>	10	100	0	-
Target (bersama)	185	120	10	-
Halangan 1	70	50	60 (ketinggian)	6
Halangan 2	20	60	40 (ketinggian)	4
Halangan 3	60	90	60 (ketinggian)	9
Halangan 4	140	40	50 (ketinggian)	7
Halangan 5	180	190	60 (ketinggian)	5
Halangan 6	30	180	60 (ketinggian)	5
Halangan 7	100	20	30 (ketinggian)	6
Halangan 8	30	110	20 (ketinggian)	4
Halangan 9	150	100	35 (ketinggian)	9
Halangan 10	70	160	40 (ketinggian)	7
Halangan 11	110	140	20 (ketinggian)	5

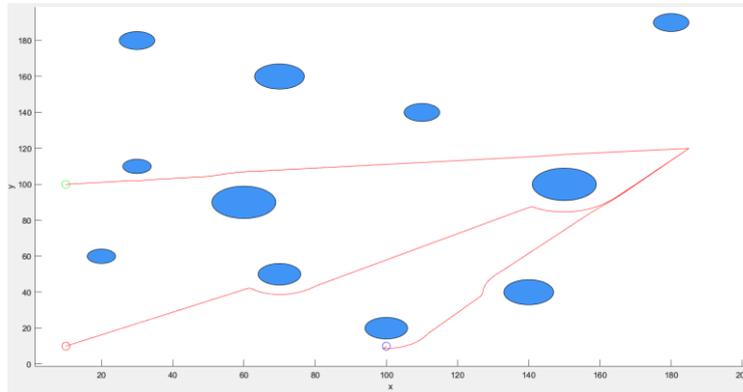
Ditetapkan nilai  $K_{rep} = 0.1$  dan  $K_{att} = 0.04$ . Selain itu, ditentukan kalau  $\rho_0 = radius\ dari\ obstacle$  dan  $n = 2$ .



Gambar 4.9 Pengujian Algoritma *Optimized APF*

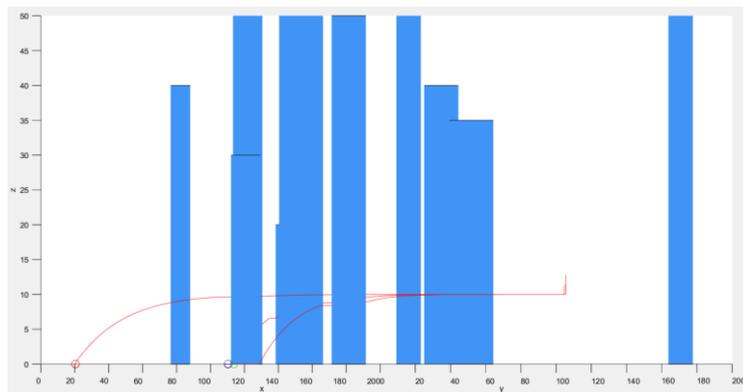
Gambar 4.9 menunjukkan bahwa semua *quadcopter* berhasil mencapai titik tujuan sambil menghindari rintangan. Lintasan yang ditempuh masing-masing *quadcopter* berbeda bergantung pada posisi awal dan rintangan yang ada pada rute masing-masing *quadcopter*. Proses penghindaran rintangan yang terjadi adalah *quadcopter* memutari

permukaan rintangan sejauh radius rintangan tersebut. Gerakan penghindarannya pun cukup halus.



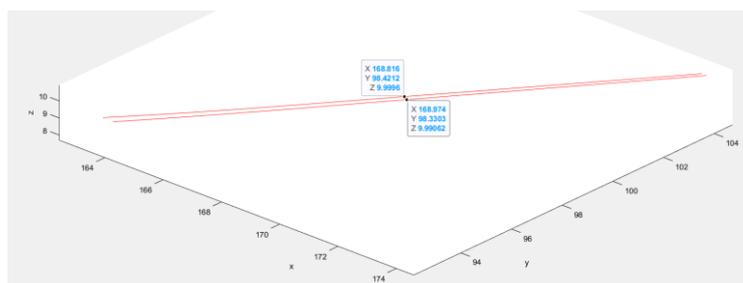
Gambar 4.10 Pengujian Algoritma *Optimized APF* Tampak Atas

Meski begitu, terdapat hal yang cukup menarik pada gambar 4.10, yaitu ada dua *quadcopter* yang memiliki lintasan cukup dekat bahkan segaris. Terletak di sekitar  $x = 160$  sampai  $180$  dan  $y = 80$  hingga  $100$ .



Gambar 4.11 Pengujian Algoritma *Optimized APF* Tampak Samping

Dilihat dari samping pun, ternyata kedua lintasan memiliki ketinggian yang identik. Namun, ternyata tidak terjadi tabrakan diantara kedua *quadcopter*. Ada dua hal yang memperkuat pernyataan ini. Pertama, waktu atau iterasi saat *quadcopter* 1 melintas berbeda dari *quadcopter* 2 akibat lintasan sebelumnya yang tidak sama. *Quadcopter* 2 melintas terlebih dahulu. Kedua, lintasan yang terlihat identik ternyata tidak benar-benar sama. Gambar 4.12 menunjukkan kalau lintasan diperbesar, terdapat selisih diantara keduanya sehingga tidak mungkin keduanya menabrak satu sama lain.



Gambar 4.12 Pengujian Algoritma *Optimized APF* pada Lintasan yang Berdekatan

### 4.1.3 Uji keseluruhan sistem dengan variasi kasus

Pengujian berikutnya akan menggabungkan kedua tugas pada lingkungan yang berbeda dari sebelumnya. Terdapat 5 *quadcopter* yang bergerak dalam formasi menuju sebuah titik tujuan. Pada lintasannya, berdiri satu atau beberapa rintangan (bergantung jenis kasus) yang harus dihindari. Titik tujuan hanya diketahui oleh *leader*, sedang *follower* akan mengikuti *leader* berdasarkan algoritma pembentukan formasi pada sub-bab 3.3.3.

Tabel 4.3 Posisi Awal *Quadcopter* dan Target

Nama Objek	x (m)	y (m)	z (m)
<i>Quadcopter 1</i>	1	1	0
<i>Quadcopter 2</i>	50	1	0
<i>Quadcopter 3</i>	1	50	0
<i>Quadcopter 4</i>	25	20	0
<i>Quadcopter 5</i>	20	25	0
Target ( <i>leader</i> )	500	500	30

Dipilih nilai  $K_{rep} = 0.04$ ,  $K_{att} = 0.3$ ,  $l = 2$ , dan  $v = \frac{\pi}{2}$ , sama seperti pengujian formasi di sub-bab 4.1.1. Ditentukan kalau  $n = 3,1$ ,  $\rho_0$  penghindaran minimal =  $2 * radius$ , dan  $\rho_0$  penghindaran total =  $\max(2 * radius, 20)$  untuk rintangan berukuran kurang dari sama dengan 10 serta =  $\min(2 * radius, 20)$  untuk rintangan berukuran lebih dari 10. Tiap kasus akan dicoba menggunakan strategi penghindaran total dan penghindaran minimal.

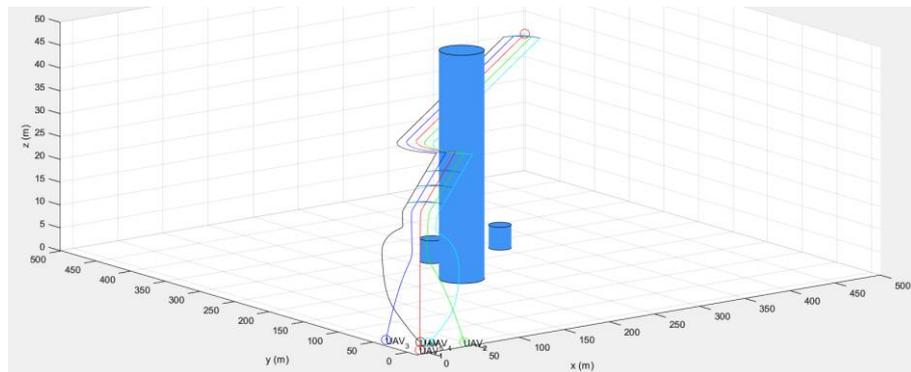
#### 4.1.3.1 Kasus 1: 1 rintangan berukuran besar

Untuk kasus pertama, formasi akan dihadapkan ke satu rintangan yang berukuran besar. Berikut spesifikasi rintangan yang dimaksud

Tabel 4.4 Spesifikasi Rintangan untuk Kasus 1

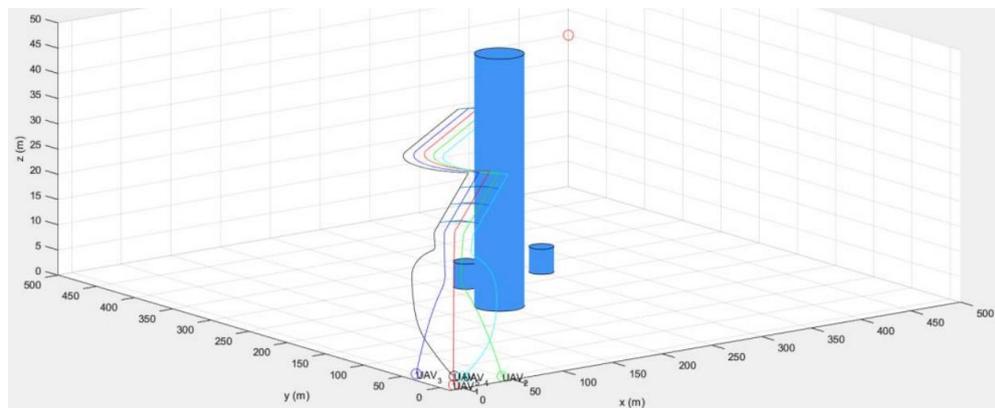
Nama objek	x (m)	y (m)	z (m)	Radius
Halangan 1	200	200	50	20

Pengujian ini dilakukan untuk melihat bagaimana pengaruh rintangan yang berukuran besar terhadap gerak kelompok *quadcopter*. Banyak iterasi yang digunakan adalah 900000.



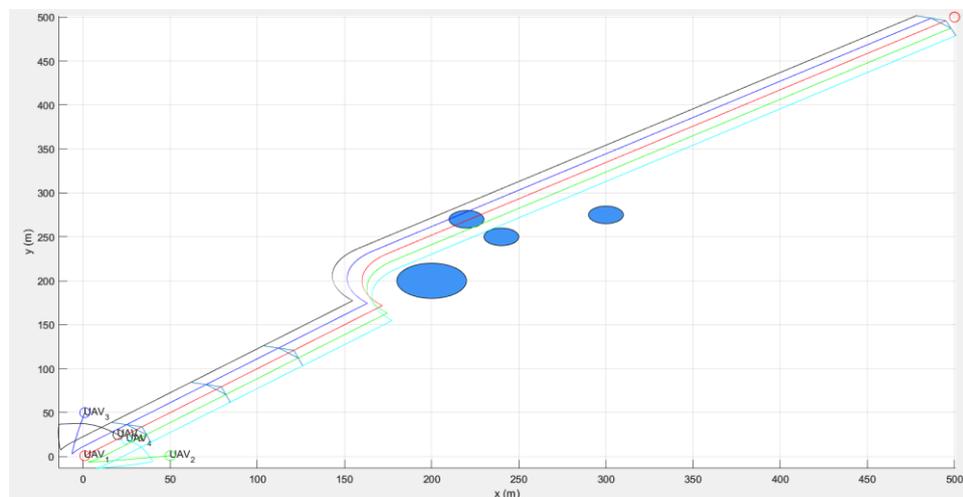
Gambar 4.13 Pengujian Kasus 1 – Penghindaran Total

Gambar 4.13 menunjukkan kalau formasi *quadcopter* berhasil melewati halangan dengan baik tanpa memecah formasi. Meskipun terdapat beberapa halangan lain pada simulasi, ketinggian mereka sangat rendah sehingga bisa diabaikan oleh formasi. Terlihat kalau tidak ada tabrakan antar *quadcopter* ataupun *quadcopter* yang terpisah dari formasi. Seluruh formasi akan ikut bergerak melalui sisi kiri halangan meskipun penghindaran ke kanan merupakan opsi yang seharusnya cukup menarik bagi *quadcopter* 2 dan *quadcopter* 4. Penghindaran dilakukan pada jarak sekitar 20 meter dari titik pusat halangan, sesuai dengan radius penghindaran untuk penghindaran total. Jarak tersebut membuat jarak *quadcopter* 4 (paling kanan dalam formasi) cukup dekat dengan halangan, tapi masih masih berada pada jarak aman. Lebih jelasnya bisa dilihat pada gambar 4.15.

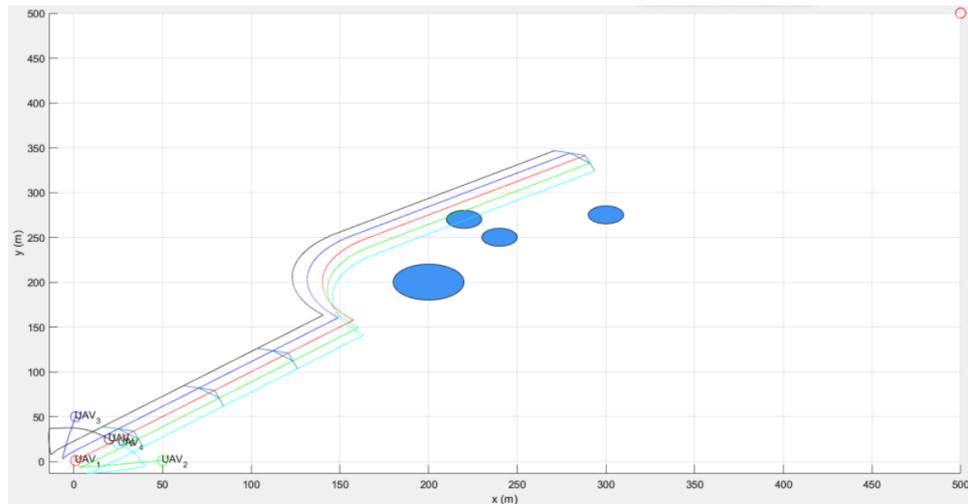


Gambar 4.14 Pengujian Kasus 1 – Penghindaran Minimal

Lalu dari gambar 4.14, bisa dilihat kalau hasil yang didapat dari penghindaran minimal cukup mirip dengan penghindaran total. Hanya saja, formasi melakukan belokan pada jarak yang lebih jauh dari halangan, dimana dua kali radius berarti sekitar 40 meter dari titik pusat halangan. Hal ini terjadi karena masing-masing *quadcopter* bergerak dengan mengikuti rute panduan yang diberikan. Belokan yang jauh membuat tidak ada *quadcopter follower* yang memisah dari formasi. Penampakan yang lebih jelas bisa dilihat dari gambar 4.16.

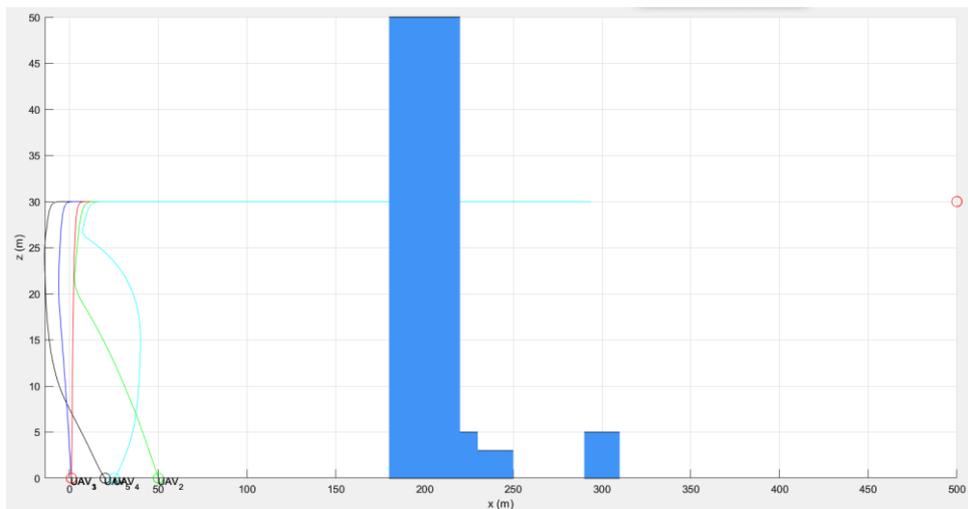


Gambar 4.15 Pengujian Kasus 1 Tampak Atas – Penghindaran Total



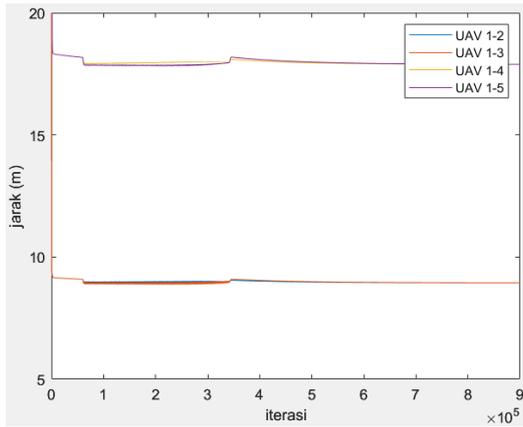
Gambar 4.16 Pengujian Kasus 1 Tampak Atas – Penghindaran Minimal

Selain itu, dengan jumlah iterasi yang sama, terlihat kalau strategi penghindaran total akan mencapai posisi yang lebih jauh dibanding strategi penghindaran minimal. Pada gambar 4.15, formasi hampir mencapai titik tujuan yang diberi tanda lingkaran merah. Sedangkan posisi formasi pada gambar 4.16 masih berada ditengah jalan. Tentu jarak penghindaran yang lebih lebar akan berakibat pada waktu tempuh yang lebih lama.

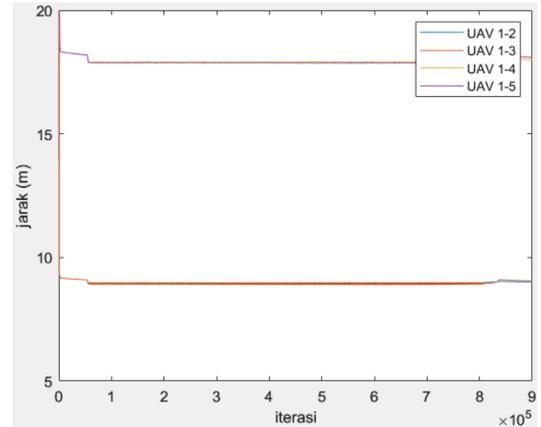


Gambar 4.17 Pengujian Kasus 1 Tampak Samping – Penghindaran Total dan Minimal

Dilihat dari gambar 4.18 (a) dan 4.18 (b), hanya terjadi sedikit penyimpangan jarak antara *quadcopter* 1 terhadap *quadcopter* lainnya karena penghindaran dilakukan dari jarak yang cukup jauh sehingga tidak merusak formasi terlalu banyak. Penyimpangan pada strategi penghindaran total terjadi lebih singkat dibanding strategi penghindaran minimal, tidak lain karena jarak penghindaran yang lebih lebar. Setelah cukup banyak iterasi, selisih jarak mencoba kembali ke nilai awalnya. Unikny, pergeseran pada grafik yang terjadi bukanlah naik melainkan turun menuju ke nilai ideal dari jarak formasi. Kehadiran halangan justru membuat pergerakan *quadcopter* melambat sehingga formasi yang diinginkan terbentuk dengan lebih baik.

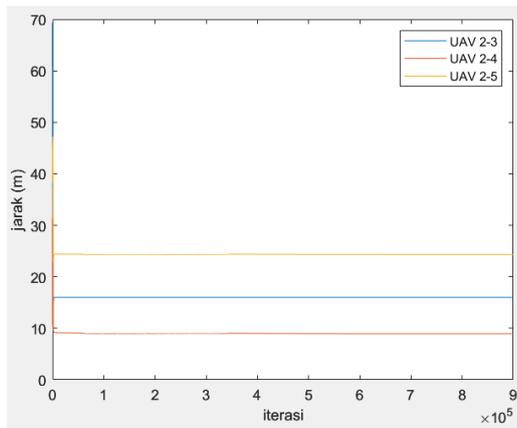


(a)

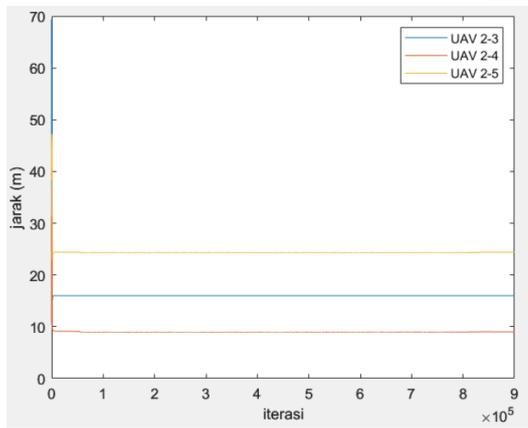


(b)

Gambar 4.18 Jarak *Quadcopter* 1 Terhadap *Follower*-nya Pada Kasus 1. (a) Penghindaran Total, (b) Penghindaran Minimal

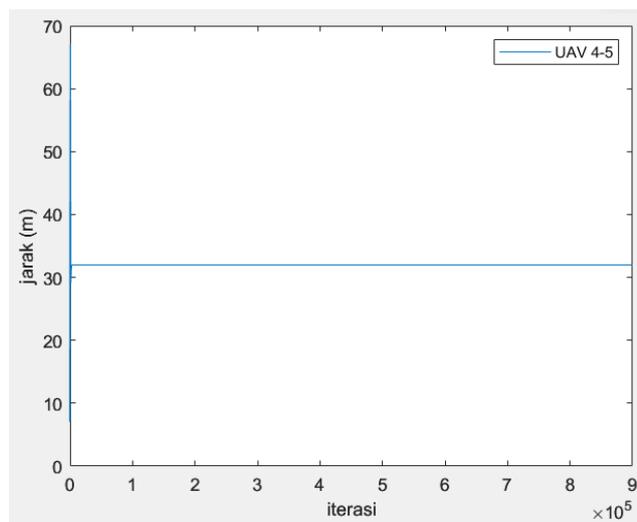


(a)



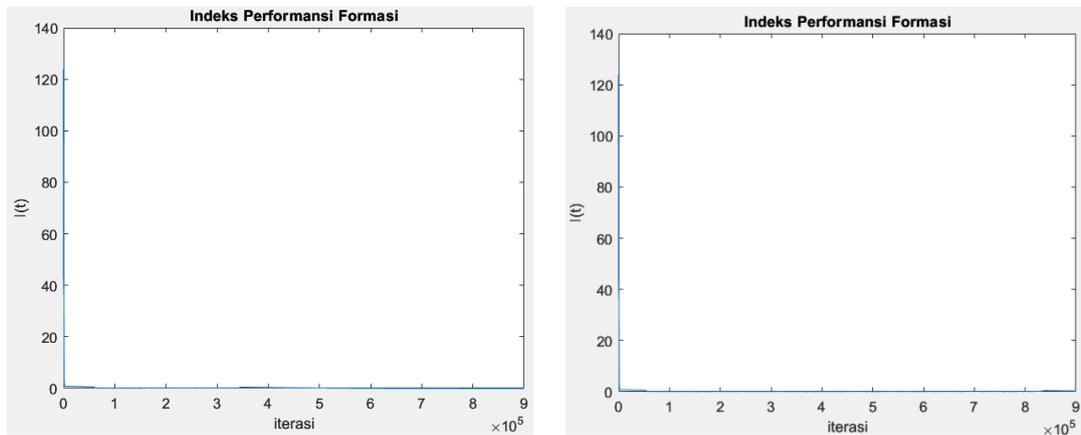
(b)

Gambar 4.19 Jarak *Quadcopter* 2 Terhadap *Quadcopter Follower* Lain pada Kasus 1. (a) Penghindaran Total, (b) Penghindaran Minimal



Gambar 4.20 Jarak *Quadcopter* 4 Terhadap *Quadcopter* 5 pada Kasus 1 – Penghindaran Total dan Minimal

Hal yang cukup menarik terjadi pada percobaan ini. Kita lihat gambar 4.21, indeks performansi menunjukkan hasil yang baik. Pada sebagian besar waktu, indeks mencapai nilai yang cukup rendah mendekati nilai 0, terutama saat formasi sedang menghindari rintangan. Rerata indeks performansi kasus 1 untuk strategi penghindaran total adalah 0.2310 sedang penghindaran minimal adalah 0.2122.



(a) (b)

Gambar 4.21 Indeks Performansi Formasi Kasus 1. (a) Penghindaran Total, (b) Penghindaran Minimal

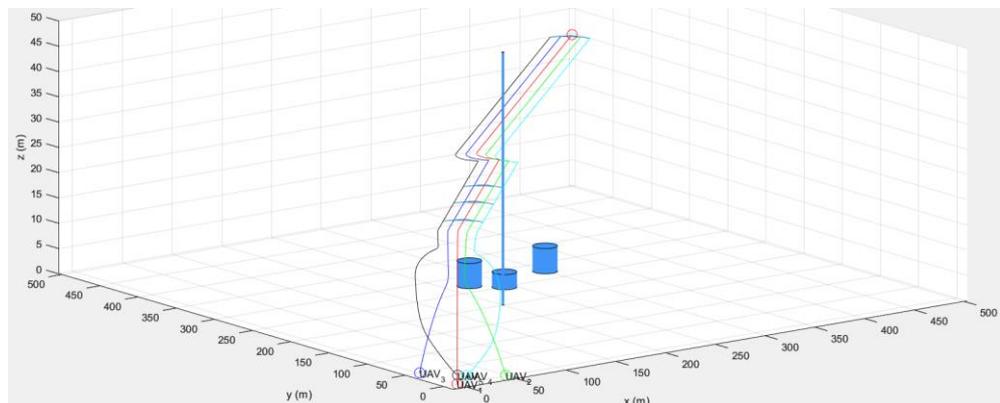
#### 4.1.3.2 Kasus 2: 1 rintangan berukuran kecil

Berkebalikan dengan kasus 1, dicoba satu rintangan yang memiliki ukuran cukup kecil bahkan lebih kecil daripada ukuran formasi itu sendiri. Spesifikasi rintangan yang dimaksud

Tabel 4.5 Spesifikasi Rintangan untuk Kasus 2

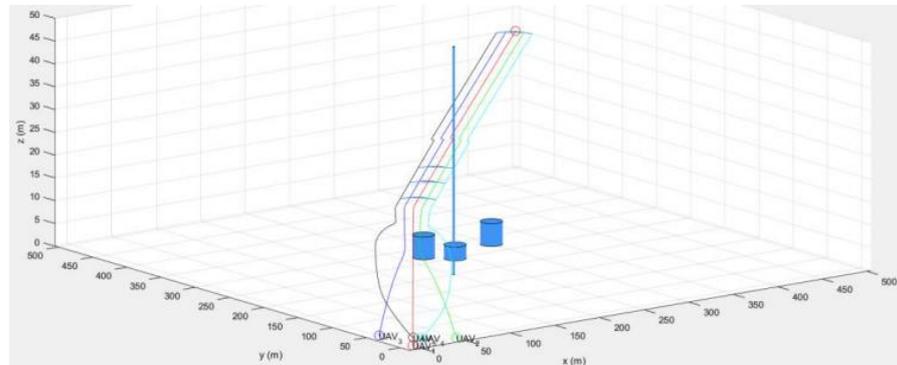
Nama objek	x (m)	y (m)	z (m)	Radius
Halangan 1	200	200	50	1

Pengujian ini dilakukan untuk melihat bagaimana respon formasi terhadap rintangan yang ada di lintasan namun bisa jadi tidak terlalu mengganggu formasi. Iterasi yang digunakan berjumlah 900000.



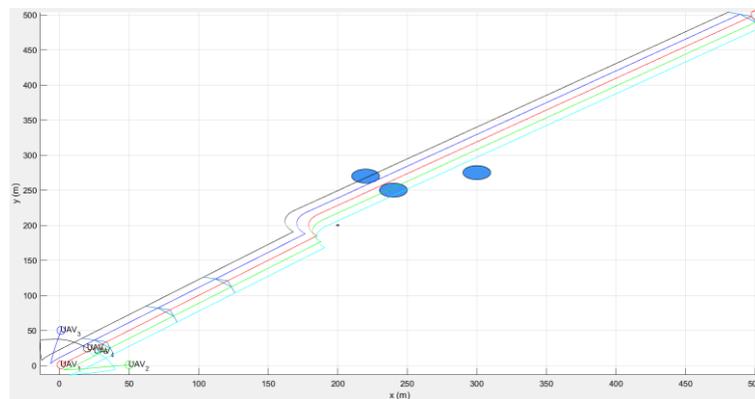
Gambar 4.22 Pengujian Kasus 2 – Penghindaran Total

Untuk penghindaran rintangan yang berukuran kecil, strategi penghindaran total tetap memberikan perlakuan yang sama dengan rintangan berukuran besar. Menggunakan radius penghindaran ( $\rho_0$ ) yang berukuran 20, formasi tetap menjadi 1 karena *leader* menghindari dari jarak sekitar 20 meter yang aman bagi keseluruhan formasi seperti pada gambar 4.22. Lebih jelasnya ada pada gambar 4.24.

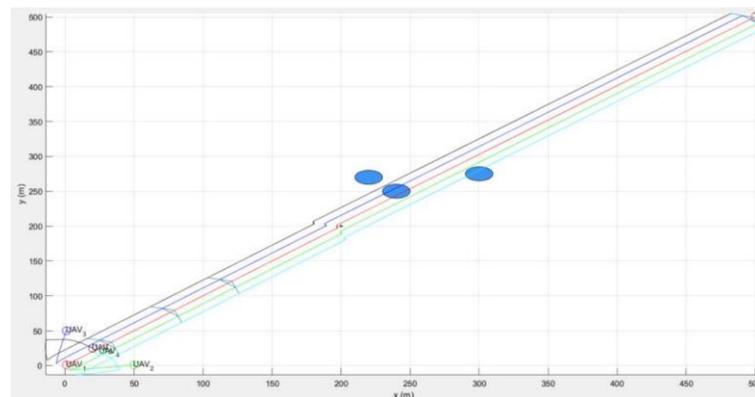


Gambar 4.23 Pengujian Kasus 2 – Penghindaran Minimal

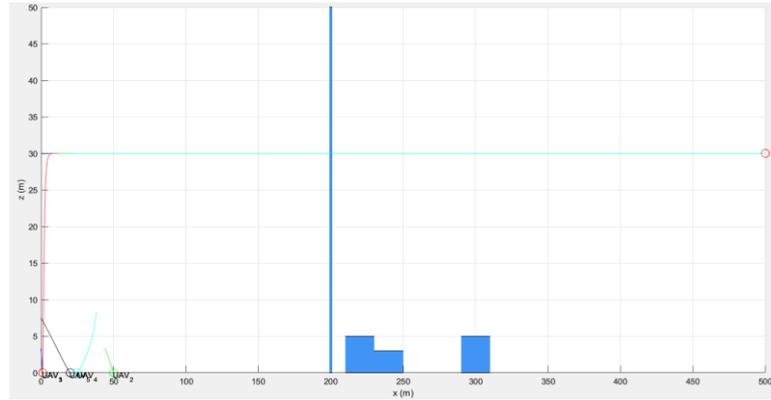
Sebaliknya untuk strategi penghindaran minimal, kelompok *quadcopter* bisa penghindaran sambil mempertahankan formasi meskipun harus terpisah ke kiri dan kanan rintangan. Ukuran rintangan cukup kecil sehingga bisa masuk diantara *quadcopter leader* dan 2. Hanya saja, karena posisi halangan yang tepat ditengah rute formasi, *quadcopter leader* perlu melakukan penghindaran sehingga membelokkan seluruh *quadcopter*. Pergerakan yang lebih jelas terlihat pada gambar 4.25.



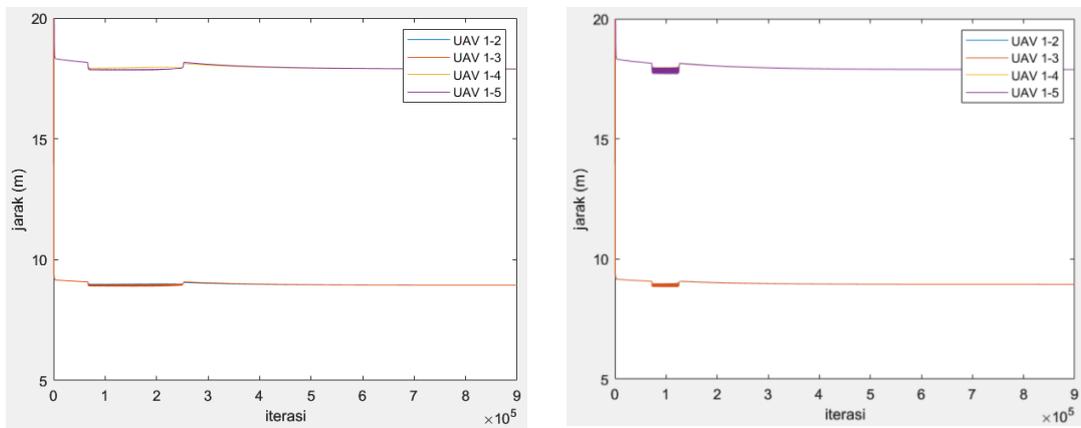
Gambar 4.24 Pengujian Kasus 2 Tampak Atas – Penghindaran Total



Gambar 4.25 Pengujian Kasus 2 Tampak Atas – Penghindaran Minimal



Gambar 4.26 Pengujian Kasus 2 Tampak Samping – Penghindaran Total dan Minimal

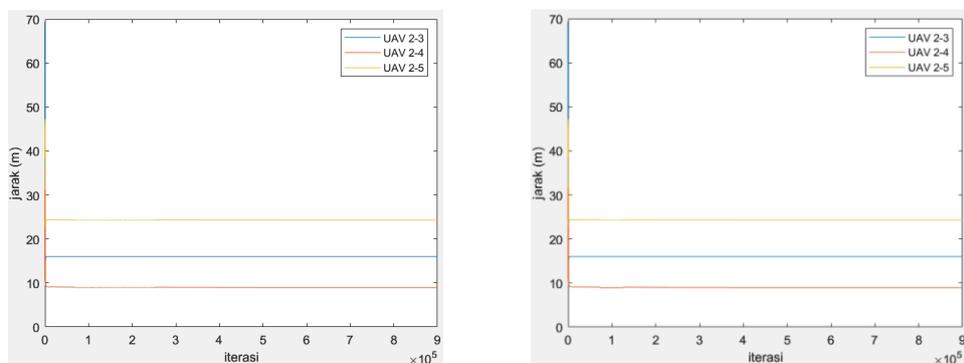


(a)

(b)

Gambar 4.27 Jarak Antara *Quadcopter* 1 Terhadap *Follower*-nya Ujian pada Kasus 2. (a) Penghindaran Total, (b) Penghindaran Minimal

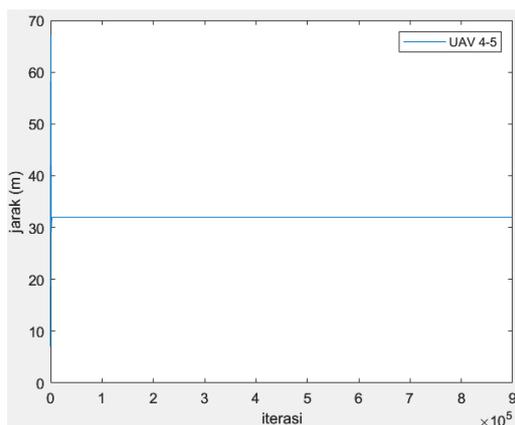
Hal yang sama terjadi seperti pada kasus 1. Terjadi penyimpangan pada grafik di sekitar iterasi 100000 hingga 200000 atau waktu saat formasi menghindari halangan. Hal ini memperkuat fenomena dimana halangan menyempurnakan formasi. Namun, berkebalikan dengan kasus 1, kali ini penyimpangan untuk strategi penghindaran minimal lebih singkat dibanding strategi penghindaran total. Tentu akibat dari jarak penghindaran yang lebih kecil.



(a)

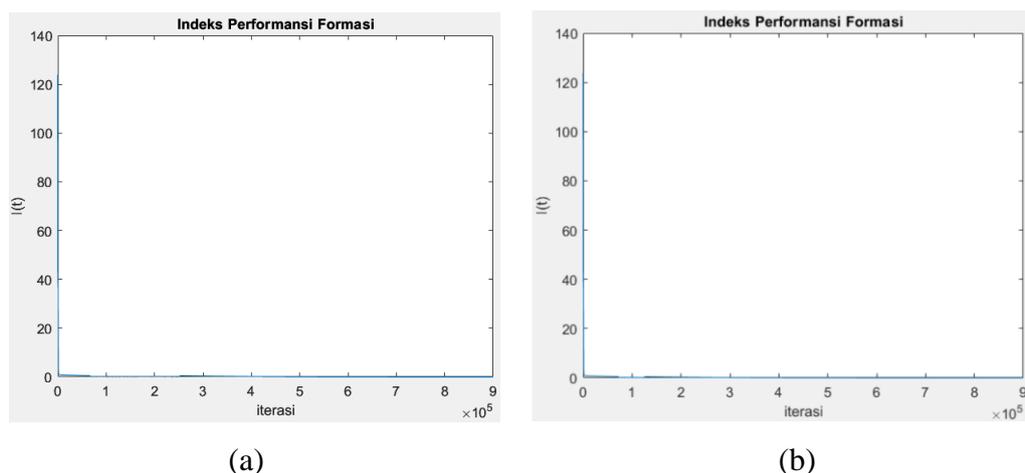
(b)

Gambar 4.28 Jarak *Quadcopter* 2 Terhadap *Quadcopter Follower* Lain pada Kasus 2. (a) Penghindaran Total, (b) Penghindaran Minimal



Gambar 4.29 Jarak *Quadcopter* 4 Terhadap *Quadcopter* 5 pada Kasus 2 – Penghindaran Total dan Minimal

Indeks performansi kasus 2 juga terlihat baik dan mirip dengan kasus 1 dimana terjadi sedikit penyimpangan pada grafik ketika formasi menghindari rintangan. Alhasil, rerata indeks performansi untuk kasus 2 untuk strategi penghindaran total adalah 0,2164 dan penghindaran minimal adalah 0,2033. Hasilnya terlihat pada gambar 4.30.



Gambar 4.30 Indeks Performansi Formasi Kasus 2. (a) Penghindaran Total, (b) Penghindaran Minimal

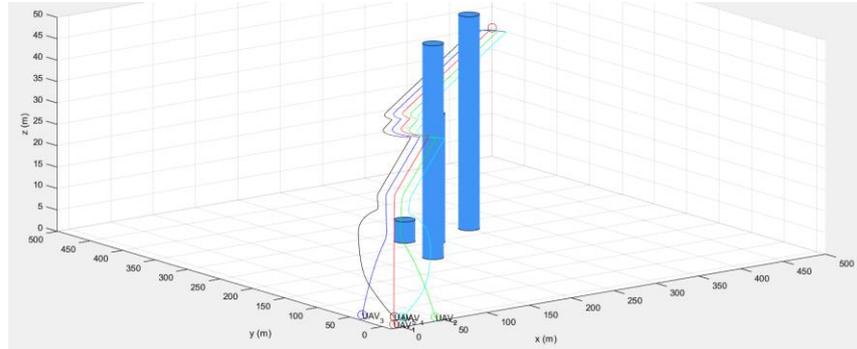
#### 4.1.3.3 Kasus 3: beberapa rintangan dengan variasi

Kasus ketiga bisa dikatakan yang paling lengkap dengan menggabungkan berbagai bentuk rintangan, termasuk variasi ketinggian. Spesifikasi rintangan yang digunakan adalah

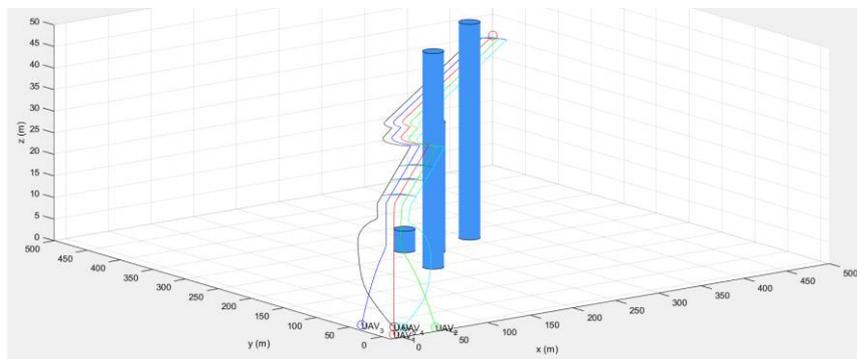
Tabel 4.6 Spesifikasi Rintangan untuk Kasus 3

Nama objek	x (m)	y (m)	z (m)	Radius
Halangan 1	200	200	50	10
Halangan 2	300	275	50	10
Halangan 3	220	270	5	10
Halangan 4	240	250	30	10

Pengujian ini dilakukan untuk melihat perbedaan respon formasi terhadap beragam posisi dan bentuk rintangan. Banyak iterasi yang digunakan adalah 900000.

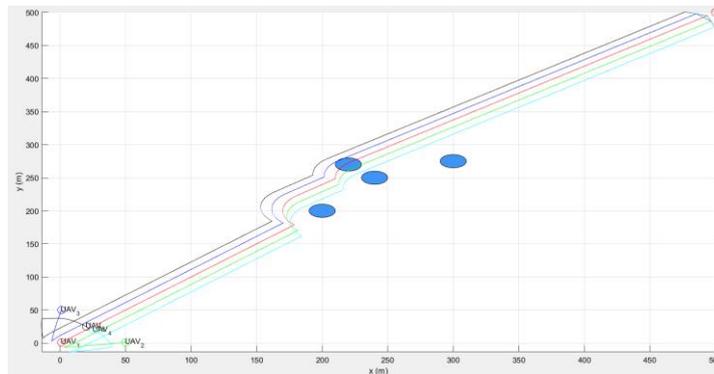


Gambar 4.31 Pengujian Kasus 3 – Penghindaran Total

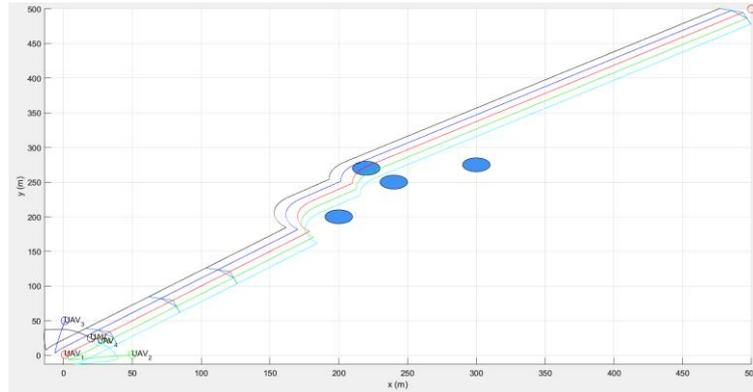


Gambar 4.32 Pengujian Kasus 3 – Penghindaran Minimal

Terlihat dari gambar 4.31 dan gambar 4.32 kalau kedua strategi penghindaran mampu membuat kelompok *quadcopter* menuju ke titik tujuan tanpa bersentuhan dengan rintangan-rintangan yang ada sembari tetap mempertahankan formasinya. Kedua gambar terlihat identik. Fenomena ini terjadi karena ukuran halangan, 10, adalah nilai yang membuat radius penghindaran kedua strategi sama sehingga *behavior* yang dihasilkan juga sama. Untuk penghindaran total, radius penghindarannya adalah 20. Sedangkan penghindaran minimal akan menentukan radius penghindaran berdasarkan 2 kali ukuran halangan, yang kebetulan juga 20 pada kasus ini. Formasi akan menghindari rintangan yang paling depan atau halangan 1 karena berada tepat di tengah lintasan dan memiliki tinggi yang melebihi ketinggian terbang *quadcopter*.

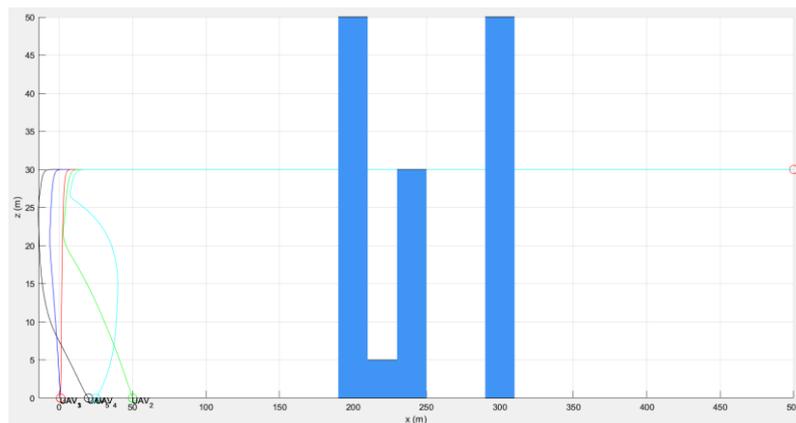


Gambar 4.33 Pengujian Kasus 3 Tampak Atas – Penghindaran Total

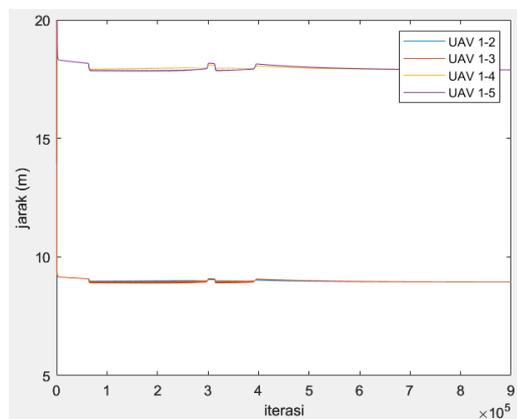


Gambar 4.34 Pengujian Kasus 3 Tampak Atas – Penghindaran Minimal

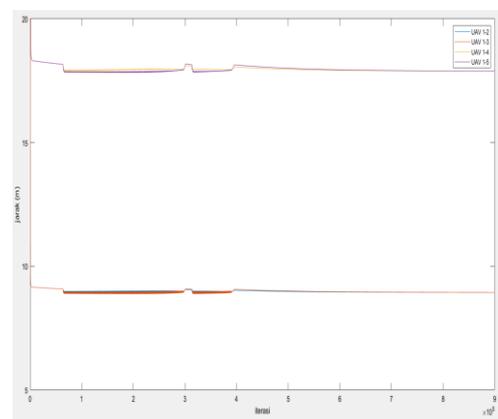
Lalu untuk halangan 4, meskipun gambar 4.31 dan gambar 4.32 menunjukkan kalau tingginya sama dengan ketinggian *quadcopter*, sesuai dengan pembahasan di sub-bab 3.3.4, formasi akan tetap melakukan penghindaran. Sedangkan halangan 3 dan halangan 2 akan dilewati oleh formasi tanpa kendala karena memiliki tinggi yang rendah lebih rendah dari ketinggian terbang formasi dan berada di jarak yang aman (terutama terhadap *leader*), secara berurutan.



Gambar 4.35 Pengujian Kasus 3 Tampak Samping – Penghindaran Total dan Minimal



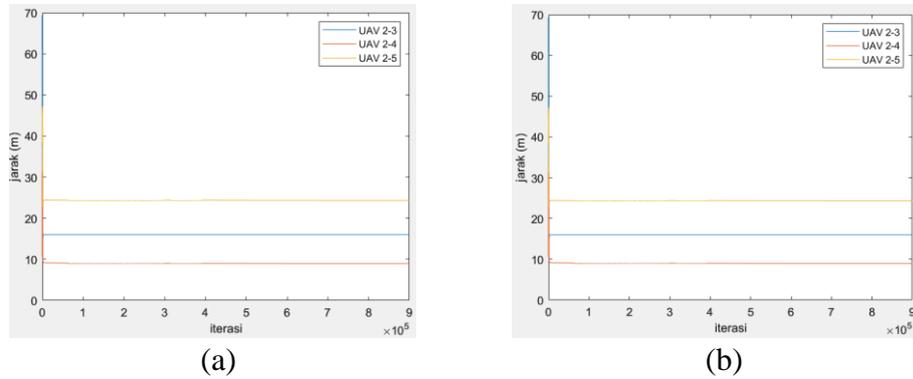
(a)



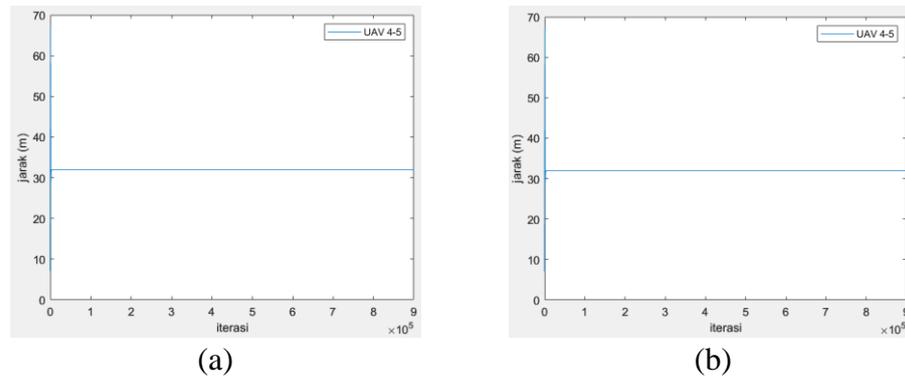
(b)

Gambar 4.36 Jarak Antara *Quadcopter* 1 Terhadap *Follower*-nya pada Kasus 3. (a) Penghindaran Total, (b) Penghindaran Minimal

Berdasarkan gambar 4.36, grafik jarak untuk percobaan ini memiliki naik turun yang lebih banyak daripada dua kasus lainnya. Hal ini dikarenakan masing-masing *quadcopter* perlu melakukan beberapa penghindaran sepanjang jalan sehingga harus memecah formasi untuk sementara.

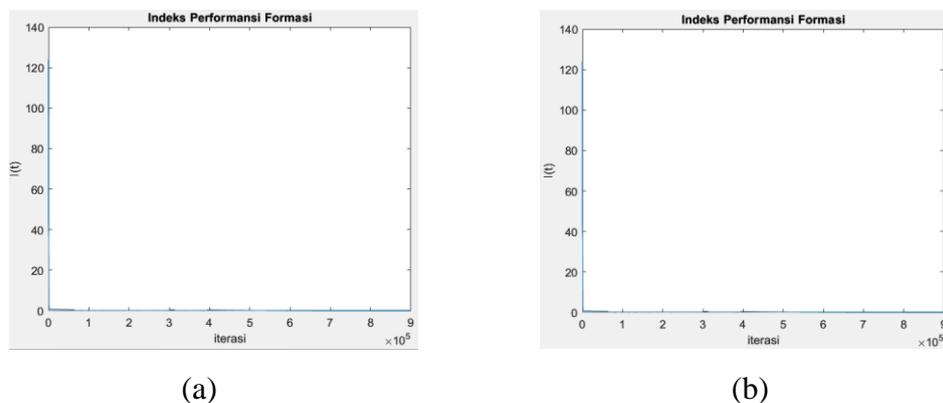


Gambar 4.37 Jarak *Quadcopter* 2 Terhadap *Quadcopter Follower* Lain pada Kasus 3. (a) Penghindaran Total, (b) Penghindaran Minimal



Gambar 4.38 Jarak *Quadcopter* 4 Terhadap *Quadcopter* 5 pada Kasus 3. (a) Penghindaran Total, (b) Penghindaran Minimal

Untuk memastikan apakah formasi terus terjaga dan melihat seberapa besar pengaruh halangan terhadap formasi, dilihat indeks performansi formasi pada gambar 4.39. Nilai indeks sebagian besar berada di sekitar nilai 0 dan terus mendekati 0 seiring berjalannya waktu. Rata-rata nilai indeks performansi untuk kedua strategi adalah sama, yaitu 0,23. Angka ini menunjukkan kalau formasi terus terjaga sepanjang waktu.

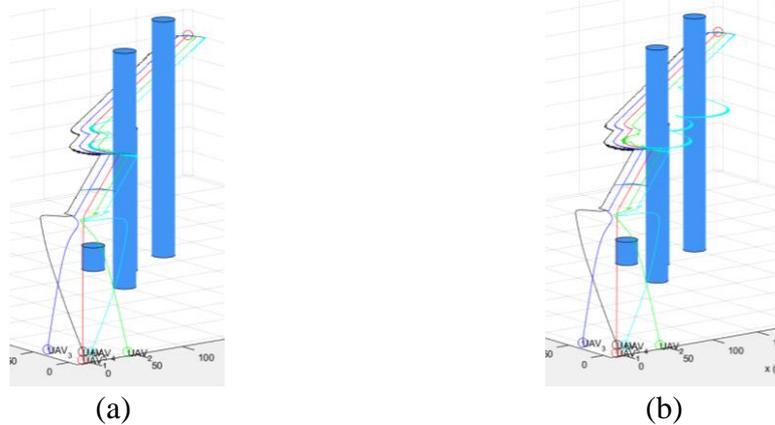


Gambar 4.39 Indeks Performansi Pormasi Kasus 3. (a) Penghindaran Total, (b) Penghindaran Minimal

#### 4.1.4 Perbandingan nilai gain pada kasus 3

Nilai *gain* atraktif  $K_{att}$  dan *gain* repulsif  $K_{rep}$  yang digunakan pada pengujian sebelumnya merupakan nilai yang sudah cukup baik untuk keperluan simulasi. Pada sub-bab ini, ditunjukkan bagaimana pemilihan nilai  $K_{att}$  dan  $K_{rep}$  mempengaruhi hasil simulasi. Beberapa nilai *gain* dipilih dan akan dilakukan analisa terkait hasil yang didapat. Nilai parameter lain dibiarkan sama dengan yang digunakan pada kasus 3. Semua simulasi dibawah dilakukan dengan menggunakan kedua strategi penghindaran.

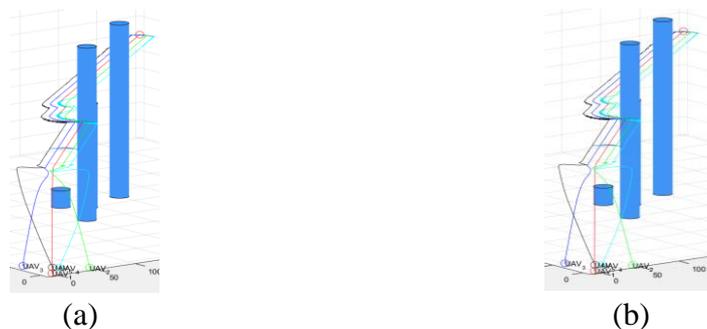
1. Nilai  $K_{att} = 1$  dan  $K_{rep} = 40$



Gambar 4.40 Simulasi dengan  $K_{att} = 1$  dan  $K_{rep} = 40$ . (a) Penghindaran Total, (b) Penghindaran Minimal

Bisa terlihat dari gambar 4.40 (a) dan (b), *quadcopter* bergerak dengan lebih berlebih dari sebelumnya. *Quadcopter* 4 dan 5 terbang keatas kanan dan kiri lalu masuk ke formasi. Pergerakan dari masing-masing *quadcopter* terutama *follower* membentuk *chattering* atau gerigi, termasuk trayektori tebal oleh *quadcopter* 4 saat melakukan penghindaran sendiri pada gambar 4.40 (b). Pada gambar 4.40 (a), nilai *gain* yang besar membuat *quadcopter* 4 harus terpisah karena tertinggal dari formasi. Indeks performansinya adalah 1,0009. Sedangkan hasil untuk strategi penghindaran minimal pada gambar 4.40 (b) cukup berbeda dari penghindaran total. Tidak hanya *quadcopter* 4, *quadcopter* 2 juga harus menghindari rintangan dengan tolakan yang lebih kuat dibanding kasus-kasus sebelumnya sehingga mereka harus terpisah dari formasi untuk iterasi yang cukup banyak. Indeks performansinya adalah 4,4143.

2. Nilai  $K_{att} = 1$  dan  $K_{rep} = 10$



Gambar 4.41 Simulasi dengan  $K_{att} = 1$  dan  $K_{rep} = 10$  (a). Penghindaran Total, (b) Penghindaran Minimal

Hasil yang didapat hampir mirip dengan percobaan sebelumnya, hanya saja efek *chattering* menurun dan pemisahan hanya dilakukan oleh *quadcopter* 4 untuk kedua strategi. Sedangkan *quadcopter* 2 bisa mengikuti formasi. Hasil untuk kedua strategi cukup identik. Oleh karena itu, diasumsikan mereka memiliki nilai indeks performansi yang sama, yaitu 1,0009.

3. Nilai  $K_{att} = 1$  dan  $K_{rep} = 1$



Gambar 4.42 Simulasi dengan  $K_{att} = 1$  dan  $K_{rep} = 1$ . (a) Penghindaran Total, (b) Penghindaran Minimal

Hasil yang didapat dengan nilai kedua gain sama dengan 1 adalah sangat mirip terhadap percobaan sebelumnya menggunakan  $K_{rep} = 10$ . Tidak ada perbedaan signifikan, termasuk perbandingan antara dua strategi penghindaran. Namun, indeks performansinya mengalami sedikit perubahan menjadi 0,9714. Artinya, *error* formasi semakin mengecil.

4. Nilai  $K_{att} = 0.01$  dan  $K_{rep} = 0.4$



Gambar 4.43 Simulasi dengan  $K_{att} = 0.01$  dan  $K_{rep} = 0.4$ . (a) Penghindaran Total, (b) Penghindaran Minimal

Dengan pendekatan berbeda, yaitu mengurangi kedua nilai, terutama  $K_{att}$  yang turun lebih drastis, menyebabkan pergerakan *quadcopter* lebih halus dan mengurangi gerakan-gerakan berlebih seperti waktu lepas landas di kasus-kasus sebelumnya. Hanya saja, *quadcopter* 4 masih harus melakukan penghindaran. Hal itu, terjadi karena nilai  $K_{att}$  yang cukup kecil sehingga membuat *quadcopter* 4 tertinggal dari formasi. Namun, untuk mencapai posisi seperti gambar 4.43, dibutuhkan banyak sekali iterasi sehingga memakan waktu komputasi yang lama. Hal menarik yang didapatkan adalah pemilihan nilai  $K_{att}$  berperan penting untuk

menentukan *behavior quadcopter*. Selain itu, modifikasi pada nilai *goal* juga bisa mempercepat gerak *quadcopter*, tentu saja dengan pertimbangan tertentu. Berdasarkan gambar 4.41, bisa dilihat kalau percobaan menggunakan kedua strategi memiliki hasil yang identik. Nilai indeks performansi untuk percobaan ini adalah 0,8616.

5. Nilai  $K_{att} = 0.1$  dan  $K_{rep} = 0.4$

Pada percobaan terakhir dengan nilai  $K_{att} = 0.1$  dan  $K_{rep} = 0.4$ , didapatkan hasil yang cukup mirip dengan percobaan pada kasus-kasus sebelumnya. Oleh karena itu, tidak disertakan dokumentasinya. Hanya saja, seperti percobaan nomor 4, pergerakannya lebih lambat.

#### 4.1.5 Pengujian Penghindaran Sebelum Formasi

Bagaimana bila ada sebuah kondisi dimana *quadcopter* bertemu dengan halangan sebelum membentuk formasi? Kita asumsikan sebuah skenario baru dengan kondisi yang berbeda dari sebelumnya untuk memudahkan pengamatan pada hal yang ingin diuji. Tujuan formasi adalah titik (150, 150, 30). Ditentukan pula posisi awal masing-masing *quadcopter* dan halangan:

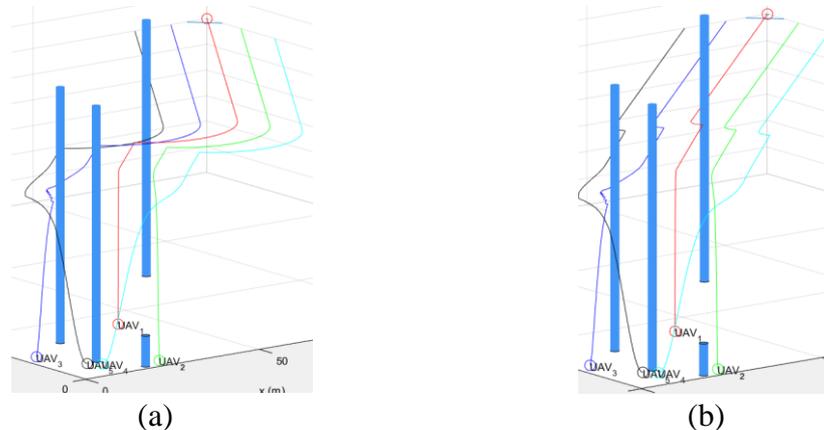
Tabel 4.7 Posisi Awal Masing-Masing *Quadcopter*

Nama	X (m)	Y (m)	Z (m)
<i>quadcopter</i> 1	30	30	0
<i>quadcopter</i> 2	25	5	0
<i>quadcopter</i> 3	5	25	0
<i>quadcopter</i> 4	12.5	10	0
<i>quadcopter</i> 5	10	12.5	0

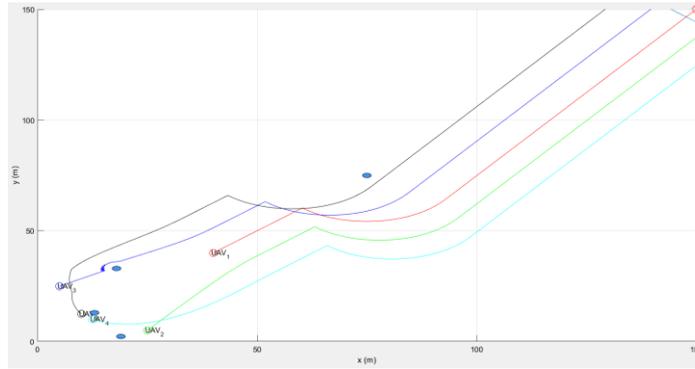
Tabel 4.8 Posisi Halangan Tambahan dengan Posisi yang Lebih Dekat

Nama	X	Y	Z	Radius
Halangan 1	13	13	60	1
Halangan 2	18	31	60	1
Halangan 3	19	2.3	6	1
Halangan 4	55	55	60	1

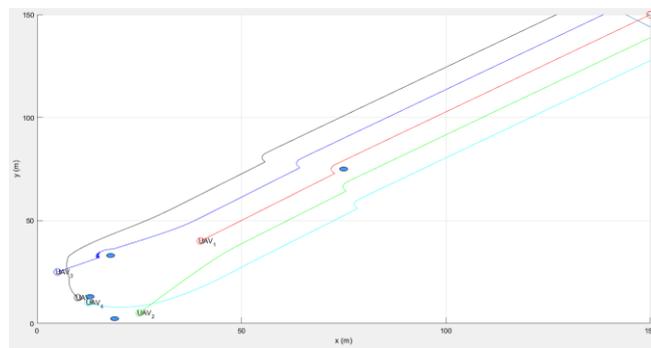
Dilakukan pengujian untuk kedua strategi penghindaran, didapatkan hasil



Gambar 4.44 Simulasi dengan Halangan yang Ditemui Terlebih Dahulu. (a) Penghindaran Total, (b) Penghindaran Minimal



Gambar 4.45 Simulasi dengan Halangan yang Ditemui Terlebih Dahulu Tampak Atas – Penghindaran Total



Gambar 4.46 Simulasi dengan Halangan yang Ditemui Terlebih Dahulu Tampak Atas – Penghindaran Minimal

Terlihat dari gambar 4.44 sampai gambar 4.46, kalau kelompok *quadcopter* mampu menghindari halangan meskipun belum masuk ke formasi. Fokus ada pada *quadcopter* 3 karena halangan benar-benar ada pada lintasannya. Penghindaran yang dilakukan oleh *quadcopter* 3 mempengaruhi bentuk rute panduan untuk *quadcopter* 5.

## 4.2 Pembahasan

### 4.2.1 Pembahasan pembentukan formasi

Hasil simulasi telah menunjukkan kalau proses membentuk dan mempertahankan formasi menggunakan metode *guidance route* berhasil dilakukan. Tiap *quadcopter* bergerak bersamaan mengikuti bentuk formasi. Namun, bila diperhatikan dengan lebih seksama, terdapat *error* berupa selisih antara posisi riil *quadcopter* terhadap posisi yang seharusnya ditempati. Hal ini kemungkinan besar terjadi karena pengaturan kecepatan yang belum sempurna mengingat metode *guidance route* adalah metode yang sangat fleksibel dimana tiap *quadcopter* diperbolehkan bergerak sebebas mungkin asalkan mengikuti rute panduan yang diberikan.

Solusi dari permasalahan ini adalah membiarkan *quadcopter* bergerak lebih lama. Seiring waktu, selisih jarak itu akan terpankas karena *quadcopter follower* secara perlahan akan bergerak mengejar posisi target panduan. Selain itu, kita juga bisa melakukan penyesuaian dengan nilai parameter-parameter seperti mengatur  $F_{total}(X)$  pada sub-bab 3.3.4.

## 4.2.2 Pembahasan penghindaran rintangan

Dari hasil simulasi, kita tahu bahwa metode *optimized Artificial Potential Field* (APF) bisa digunakan untuk melakukan penghindaran rintangan pada *quadcopter*. Tidak ada tabrakan yang terjadi baik antara halangan dan *quadcopter* maupun *quadcopter* dengan sesama *quadcopter*.

Metode APF sangat erat kaitannya dengan nilai  $k_{att}$  (*gain* atraktif) dan  $k_{rep}$  (*gain* repulsif). Perubahan nilai masing-masing *gain* bisa merubah performa seluruh sistem. Nilai  $k_{att}$  dan  $k_{rep}$  yang kecil akan membuat gerakan *quadcopter* menjadi lebih lambat namun lebih halus saat melakukan belokan. Sebaliknya, bila nilai  $k_{att}$  dan  $k_{rep}$  terlalu besar, gerakan *quadcopter* memang menjadi lebih cepat tapi sangat rentan terjadi gerakan berlebihan seperti *chattering*, terutama saat menghindari halangan.

Untuk tugas akhir ini, penghindaran yang bisa dilakukan hanyalah penghindaran secara horizontal, yaitu penghindaran dengan memutar halangan. Konsep ini sebenarnya sangat menggambarkan bagaimana konsep APF bekerja. Titik tujuan akan bertindak seperti sumber atraktif yang menarik *quadcopter* menuju posisinya. Sedangkan halangan akan memancarkan medan repulsif yang menolak *quadcopter*. Untuk menghindari halangan, *quadcopter* bergerak pada permukaan medan repulsif tersebut.

Adapun konsep penghindaran antar *quadcopter* sebenarnya tidak terlalu krusial pada kasus yang diangkat, terutama karena jarak antar *quadcopter* baik diawal maupun formasi cukup jauh. Tetapi, tidak ada salahnya memperhitungkan hal ini karena pada kasus yang berbeda seperti formasi dengan jarak yang kecil, *quadcopter* cukup rentan untuk menabrak *quadcopter* lain.

## 4.2.3 Pembahasan keseluruhan sistem

Lingkungan yang bervariasi merupakan tantangan tersendiri bagi formasi *quadcopter*, terutama untuk metode yang diterapkan pada penelitian ini. Perbedaan bentuk dari masing-masing halangan akan mempengaruhi nilai *gain* berapakah yang sebaiknya dipilih. Sebagai contoh, bila radius halangan pada kasus pertama dibuat lebih besar, diperlukan nilai  $k_{att}$  dan  $k_{rep}$  yang lebih besar pula. Halangan yang membesar juga bisa mempengaruhi seberapa besar medan repulsif sebaiknya ditentukan. Sebagai kesimpulan, lingkungan yang variatif, memerlukan *tuning* parameter lanjutan.

Hal yang cukup menarik adalah bagaimana *quadcopter follower* tidak melakukan penghindaran terhadap halangan meskipun berada di dalam radius jarak penghindaran halangan. Fenomena terlihat lebih jelas terutama pada pengujian kasus 1 dan 3. Pada percobaan, memang jarak penghindaran dan ukuran halangan sudah diatur sedemikian rupa sehingga penghindaran oleh *quadcopter leader* akan membawa *follower*-nya ke rute yang aman. Perilaku tersebut terjadi akibat konsep metode yang digunakan. Guidance route adalah metode dimana titik tujuan akhir hanya diketahui oleh *leader*, sedang *follower* hanya diberi titik-titik tujuan dinamis dengan interval pendek berdasarkan posisi *leader*. Jadi bila dilihat dari sudut pandang APF, ketika *leader* memiliki titik tujuan yang jauh, ia akan melakukan penghindaran yang dibutuhkan sesuai kaidah APF. Bersamaan dengan itu, titik tujuan dari *follower* akan berubah-ubah dan pada beberapa iterasi berada didalam

radius penghindaran halangan sehingga *quadcopter* akan tetap terbang didalam radius penghindaran karena titik tujuannya juga berada disana.

#### 4.2.4 Pembahasan perbandingan nilai gain

Dari simulasi yang sudah dilakukan pada beberapa nilai *gain*, didapatkan hasil sebagai berikut

Tabel 4.9 Pengaruh Nilai *Gain* APF pada Simulasi

No.	$K_{att}$	$K_{rep}$	Hasil
1	1	40	Efek <i>chattering</i> yang begitu tinggi, penghindaran pada radius halangan dilakukan oleh quadcopter 2 dan 4. Banyak gerakan berlebih.
2	1	10	Efek <i>chattering</i> menurun, penghindaran pada radius halangan hanya dilakukan oleh quadcopter 4. Banyak gerakan berlebih.
3	1	1	Mirip dengan hasil percobaan nomor 2.
4	0.01	0.4	Menunjukkan hasil yang paling halus namun gerakan per iterasi yang paling lambat. Tetap ada <i>quadcopter</i> yang tertinggal.
5	0.1	0.4	Mirip dengan simulasi kasus tapi dengan gerak per iterasi yang lebih lambat. Masih ada gerakan berlebih.

Hasil pada tabel 4.9 memperkuat penjelasan pada pembahasan 4.2.2.

#### 4.2.5 Pembahasan Penghindaran Sebelum Formasi

Kondisi yang disimulasikan pada sub-bab 4.1.5 adalah salah satu kondisi yang bisa saja terjadi pada kondisi nyata, misalkan pada pendaratan darurat. Dengan memanfaatkan algoritma APF, masing-masing *quadcopter* bisa melakukan penghindaran terhadap tiap halangan dengan mudah. Namun untuk mencapai itu, perintah bagi *follower* untuk mempercepat dirinya harus dihilangkan karena bisa membuat quadcopter menabrak halangan. Hal itu terjadi karena halangan yang cukup kecil dan percepatan terlalu besar.

#### 4.2.6 Pembahasan dan Perbandingan Strategi Penghindaran Total dan Penghindaran Minimal

Pembahasan pada sub-bab ini dikhususkan untuk menganalisa dan membandingkan dua strategi kontrol penghindaran, yaitu penghindaran total dan penghindaran minimal. Penghindaran total adalah strategi yang memaksa formasi menghindari pada jarak maksimum tertentu sehingga formasi bisa menghindari halangan tanpa berpisah. Untuk tugas akhir ini, dipilih jarak maksimum tersebut bernilai 20 meter karena panjang setengah formasi adalah sekitar 16 meter. Sedangkan penghindaran minimal adalah strategi penghindaran yang tidak memperhatikan jarak maksimum tertentu sehingga memungkinkan satu atau lebih *quadcopter* terpisah saat melakukan penghindaran.

Berdasarkan berbagai pengujian yang sudah dilakukan, bisa dilihat kalau strategi penghindaran total memberikan hasil yang lebih rapi dibanding strategi penghindaran minimal. Hal itu disebabkan oleh pergerakan yang lebih kompak dalam mempertahankan formasi. Tentu, keunggulan yang diberi kekompakan juga membawa kekurangan seperti

jarak tempuh yang lebih jauh karena harus menghindar lebih dari yang diperlukan. Hal yang menarik adalah kekurangan tersebut tidaklah absolut. Maksudnya adalah kekurangan itu bisa menjadi keunggulan pada kondisi tertentu seperti pada pengujian rintangan besar. Selain itu, strategi ini memberikan performa yang lebih stabil di pengujian dengan nilai *gain* yang berbeda-beda.

Untuk strategi penghindaran minimal, pembahasannya kurang lebih adalah kebalikan dari strategi penghindaran total. Strategi ini akan menghasilkan penghindaran yang bisa dikatakan minimal secara jarak untuk halangan yang berukuran kecil, namun akan menjadi kurang efektif pada halangan berukuran besar. Perubahan nilai *gain*, terutama yang bernilai besar, juga bisa memanipulasi respon strategi sehingga hasilnya sangat berbeda dari nilai idealnya.

Sebagai kesimpulannya, kondisi dari lingkungan atau skenario yang dijalankan sangat mempengaruhi bagaimana respon dari kedua strategi. Pada beberapa kondisi, strategi penghindaran total memang terasa lebih baik. Namun pada kondisi lain, berlaku sebaliknya. Ada pula kondisi dimana kedua strategi menghasilkan hasil yang sama seperti pada kasus 3. Perbandingan dari kedua strategi yang dituliskan pada tabel berikut

Tabel 4.10 Perbandingan Penghindaran Total dan Penghindaran Minimal

Kriteria	Penghindaran total	Penghindaran minimal
Panjang lintasan	Lebih panjang untuk kondisi halangan berukuran kecil.	Lebih panjang untuk kondisi halangan berukuran besar
Pemecahan formasi	Lebih kompak	Lebih mudah terpecah

## BAB 5 Kesimpulan dan Saran

### 5.1 Kesimpulan

Berdasarkan analisa data dan pembahasan yang telah dijelaskan pada bab sebelumnya, didapatkan kesimpulan sebagai berikut:

1. Pemanfaatan metode *guidance route* yang dikombinasikan dengan *Artificial Potential Field* (APF) efektif untuk melaksanakan tugas pengaturan formasi dan penghindaran rintangan pada sekelompok *quadcopter*.
2. Rata-rata semua nilai indeks performansi pada pengujian formasi dan keseluruhan sistem untuk strategi penghindaran total adalah 0.800025 dan untuk strategi penghindaran minimal adalah 1.2227125. Angka ini menunjukkan bahwa formasi yang terbentuk selalu terjaga sepanjang waktu dengan hanya sedikit *error*.
3. Jarak antar *quadcopter* pada seluruh percobaan tidak ada yang menyentuh angka 0 yang berarti tidak pernah terjadi tabrakan antar *quadcopter*.
4. Secara visual, terlihat kalau *quadcopter* bisa menghindari seluruh halangan.
5. Strategi penghindaran, baik penghindaran total maupun minimal memiliki keunggulan dan kelemahan masing-masing dan penggunaannya dapat disesuaikan dengan kondisi yang dihadapi.

### 5.2 Saran

Berdasarkan penelitian yang telah dilakukan maka diperoleh saran untuk pengembangan kedepan, yaitu sebagai berikut:

1. Memilih nilai-nilai parameter yang lebih baik sehingga hasil lebih optimal.
2. Mengembangkan metode adaptif yang memungkinkan perubahan pada formasi ataupun nilai parameter pada kondisi tertentu.
3. Melakukan pengujian dengan rute yang lebih kompleks.
4. Mempertimbangkan rintangan yang bergerak (*dynamic obstacle*).
5. Mengembangkan algoritma penghindaran yang memungkinkan *quadcopter* menghindar secara vertikal bila memenuhi syarat tertentu.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- Andreasson, M., Dimarogonas, D. V., Sandberg, H., & Johansson, K. H. (2014). Distributed control of networked dynamical systems: Static feedback, integral action and consensus. *IEEE Transactions on Automatic Control*, 59(7), 1750–1764. <https://doi.org/10.1109/TAC.2014.2309281>
- Asti, I. S., Agustinah, T., & Santoso, A. (2020). Obstacle Avoidance with Energy Efficiency and Distance Deviation Using KNN Algorithm for Quadcopter. *Proceedings - 2020 International Seminar on Intelligent Technology and Its Application: Humanification of Reliable Intelligent Systems, ISITIA 2020*, 285–291. <https://doi.org/10.1109/ISITIA49792.2020.9163788>
- Beard, R. W., Lawton, J., & Hadaegh, F. Y. (2001). *Control*. 9(6), 777–790.
- Benzerrouk, A., Adouane, L., & Martinet, P. (2014). Stable navigation in formation for a multi-robot system based on a constrained virtual structure. *Robotics and Autonomous Systems*, 62(12), 1806–1815. <https://doi.org/10.1016/j.robot.2014.07.004>
- CAI, Z., WANG, L., ZHAO, J., WU, K., & WANG, Y. (2020). Virtual target guidance-based distributed model predictive control for formation control of multiple UAVs. *Chinese Journal of Aeronautics*, 33(3), 1037–1056. <https://doi.org/10.1016/j.cja.2019.07.016>
- Cao, Y., & Sun, Y. (2018). Necessary and sufficient conditions for consensus of third-order discrete-time multi-agent systems in directed networks. *Journal of Applied Mathematics and Computing*, 57(1–2), 199–210. <https://doi.org/10.1007/s12190-017-1101-8>
- Chen, Y. B., Luo, G. C., Mei, Y. S., Yu, J. Q., & Su, X. L. (2016). UAV path planning using artificial potential field method updated by optimal control theory. *International Journal of Systems Science*, 47(6), 1407–1420. <https://doi.org/10.1080/00207721.2014.929191>
- Dehghani, M. A., & Menhaj, M. B. (2016). Integral sliding mode formation control of fixed-wing unmanned aircraft using seeker as a relative measurement system. *Aerospace Science and Technology*, 58, 318–327. <https://doi.org/10.1016/j.ast.2016.08.011>
- Ghommam, J., & Saad, M. (2018). Adaptive Leader-Follower Formation Control of Underactuated Surface Vessels Under Asymmetric Range and Bearing Constraints. *IEEE Transactions on Vehicular Technology*, 67(2), 852–865. <https://doi.org/10.1109/TVT.2017.2760367>
- Goss, J., Rajvanshi, R., & Subbarao, K. (2004). Aircraft conflict detection and resolution using mixed geometric and collision cone approaches. *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, 1(August), 670–689. <https://doi.org/10.2514/6.2004-4879>

- Guo, K., Li, X., & Xie, L. (2020). Simultaneous cooperative relative localization and distributed formation control for multiple UAVs. *Science China Information Sciences*, 63(1), 2019–2021. <https://doi.org/10.1007/s11432-018-9603-y>
- Kownacki, C., & Ambroziak, L. (2017). Local and asymmetrical potential field approach to leader tracking problem in rigid formations of fixed-wing UAVs. *Aerospace Science and Technology*, 68, 465–474. <https://doi.org/10.1016/j.ast.2017.05.040>
- Kwon, J. W., & Chwa, D. (2012). Hierarchical formation control based on a vector field method for wheeled mobile Robots. *IEEE Transactions on Robotics*, 28(6), 1335–1345. <https://doi.org/10.1109/TRO.2012.2206869>
- Li, Q., Hua, Y., Dong, X., & Ren, Z. (2022). Time-Varying Formation Tracking Control for Unmanned Aerial Vehicles: Theories and Applications. *IFAC-PapersOnLine*, 55(3), 49–54. <https://doi.org/10.1016/j.ifacol.2022.05.009>
- Lin, J. L., Hwang, K. S., & Wang, Y. L. (2014). A simple scheme for formation control based on weighted behavior learning. *IEEE Transactions on Neural Networks and Learning Systems*, 25(6), 1033–1044. <https://doi.org/10.1109/TNNLS.2013.2285123>
- Mahmood, A., & Kim, Y. (2015). Leader-following formation control of quadcopters with heading synchronization. *Aerospace Science and Technology*, 47, 68–74. <https://doi.org/10.1016/j.ast.2015.09.009>
- Millan, P., Orihuela, L., Jurado, I., & Rubio, F. R. (2014). Formation control of autonomous underwater vehicles subject to communication delays. *IEEE Transactions on Control Systems Technology*, 22(2), 770–777. <https://doi.org/10.1109/TCST.2013.2262768>
- Nurjanah, S., & Fuad, M. (2022). *Cooperative Position-based Formation-pursuit of Moving Targets by Multi-UAVs with Collision Avoidance*. 6(2), 82–90.
- Oh, K. K., Park, M. C., & Ahn, H. S. (2015). A survey of multi-agent formation control. *Automatica*, 53, 424–440. <https://doi.org/10.1016/j.automatica.2014.10.022>
- Padhi, R., Rakesh, P. R., & Venkataraman, R. (2014). Formation flying with nonlinear partial integrated guidance and control. *IEEE Transactions on Aerospace and Electronic Systems*, 50(4), 2847–2859. <https://doi.org/10.1109/TAES.2014.120719>
- Park, B. S. (2015). Adaptive formation control of underactuated autonomous underwater vehicles. *Ocean Engineering*, 96, 1–7. <https://doi.org/10.1016/j.oceaneng.2014.12.016>
- Ren, W., & Beard, R. W. (2004). Decentralized Scheme for Spacecraft Formation Flying via the Virtual Structure Approach. *Journal of Guidance, Control, and Dynamics*, 27(1), 73–82. <https://doi.org/10.2514/1.9287>

- Rezaee, H., & Abdollahi, F. (2015). Pursuit Formation of Double-Integrator Dynamics Using Consensus Control Approach. *IEEE Transactions on Industrial Electronics*, 62(7), 4249–4256. <https://doi.org/10.1109/TIE.2014.2384479>
- Saradagi, A., Muralidharan, V., Krishnan, V., Menta, S., & Mahindrakar, A. D. (2018). Formation Control and Trajectory Tracking of Nonholonomic Mobile Robots. *IEEE Transactions on Control Systems Technology*, 26(6), 2250–2258. <https://doi.org/10.1109/TCST.2017.2749563>
- Shahbazi, B., Malekzadeh, M., & Koofgar, H. R. (2017). Robust Constrained Attitude Control of Spacecraft Formation Flying in the Presence of Disturbances. *IEEE Transactions on Aerospace and Electronic Systems*, 53(5), 2534–2543. <https://doi.org/10.1109/TAES.2017.2704160>
- SUN, G., ZHOU, R., XU, K., WENG, Z., ZHANG, Y., DONG, Z., & WANG, Y. (2020). Cooperative formation control of multiple aerial vehicles based on guidance route in a complex task environment. *Chinese Journal of Aeronautics*, 33(2), 701–720. <https://doi.org/10.1016/j.cja.2019.08.009>
- Sun, Z., Zhang, G., Lu, Y., & Zhang, W. (2018). Leader-follower formation control of underactuated surface vehicles based on sliding mode control and parameter estimation. *ISA Transactions*, 72, 15–24. <https://doi.org/10.1016/j.isatra.2017.11.008>
- Wang, N., Dai, J., & Ying, J. (2021). UAV Formation Obstacle Avoidance Control Algorithm Based on Improved Artificial Potential Field and Consensus. *International Journal of Aeronautical and Space Sciences*, 22(6), 1413–1427. <https://doi.org/10.1007/s42405-021-00407-6>
- Wang, R., & Liu, J. (2017). Adaptive formation control of quadrotor unmanned aerial vehicles with bounded control thrust. *Chinese Journal of Aeronautics*, 30(2), 807–817. <https://doi.org/10.1016/j.cja.2017.01.007>
- Xiao, H., Li, Z., & Philip, C. (2016). Formation control of leader-follower mobile robots' systems using model predictive control based on neural-dynamic optimization. *IEEE Transactions on Industrial Electronics*, 63(9), 5752–5762. <https://doi.org/10.1109/TIE.2016.2542788>
- Yu, W., & Lu, Y. (2021). UAV 3D environment obstacle avoidance trajectory planning based on improved artificial potential field method. *Journal of Physics: Conference Series*, 1885(2). <https://doi.org/10.1088/1742-6596/1885/2/022020>
- Zhang, J., Yan, J., & Zhang, P. (2018). Fixed-wing UAV formation control design with collision avoidance based on an improved artificial potential field. *IEEE Access*, 6, 78342–78351. <https://doi.org/10.1109/ACCESS.2018.2885003>

Zhang, J., Yan, J., Zhang, P., & Kong, X. (2018). Collision Avoidance in Fixed-Wing UAV Formation Flight Based on a Consensus Control Algorithm. *IEEE Access*, 6, 43672–43682. <https://doi.org/10.1109/ACCESS.2018.2864169>

## LAMPIRAN

### Source code MATLAB

#### a. Program utama

```
%-----Posisi awal tiap quadcopter
%-----Bila ingin menambahkan quadcopter, tambahkan lagi variabelnya
%-----Contoh => start{6} untuk menambahkan quadcopter ke 6
start{1} = [1,1,0];
start{2} = [50,1,0];
start{3} = [1,50,0];
start{4} = [25,20,0];
start{5} = [20,25,0];
%-----State awal --> semua bernilai 0
%-----Sama seperti start, bila ada quadcopter 6 dst, tambahkan state{i}
state0= [0 0 0 0 0 0 0 0];
state{1}=[start{1} state0]';
state{2}=[start{2} state0]';
state{3}=[start{3} state0]';
state{4}=[start{4} state0]';
state{5}=[start{5} state0]';
%-----all_state adalah variabel yang akan digunakan untuk menyimpan semua
%-----nilai state. Untuk awal, nilainya sama dengan state.
all_state{1}=state{1};
all_state{2}=state{2};
all_state{3}=state{3};
all_state{4}=state{4};
all_state{5}=state{5};

%-----Memberi posisi awal quadcopter tanda berupa lingkaran dan memberi nama
%-----Bila menambah quadcopter, tambahkan juga bagian ini untuk kejelasan
plot3(start{1}(1),start{1}(2),start{1}(3), 'o', 'Markersize',10, 'color', 'r'); %draw
the initial position
text(start{1}(1)-0.5,start{1}(2)-0.5,start{1}(3)-0.5, 'UAV_1'); %Callout name
hold on
plot3(start{2}(1),start{2}(2),start{2}(3), 'o', 'Markersize',10, 'color', 'g'); %draw
the initial position
text(start{2}(1)-0.5,start{2}(2)-0.5,start{2}(3)-0.5, 'UAV_2'); %Callout name
hold on
plot3(start{3}(1),start{3}(2),start{3}(3), 'o', 'Markersize',10, 'color', 'b'); %draw
the initial position
text(start{3}(1)-0.5,start{3}(2)-0.5,start{3}(3)-0.5, 'UAV_3'); %Callout name
hold on
plot3(start{4}(1),start{4}(2),start{4}(3), 'o', 'Markersize',10, 'color', 'c'); %draw
the initial position
text(start{4}(1)-0.5,start{4}(2)-0.5,start{4}(3)-0.5, 'UAV_4'); %Callout name
hold on
plot3(start{5}(1),start{5}(2),start{5}(3), 'o', 'Markersize',10, 'color', 'k'); %draw
the initial position
text(start{5}(1)-0.5,start{5}(2)-0.5,start{5}(3)-0.5, 'UAV_5'); %Callout name
hold on

%-----Goal dari quadcopter leader atau quadcopter 1
goal{1} = [500,500,30]';
%-----Melakukan plot untuk goal leader
plot3(goal{1}(1),goal{1}(2),goal{1}(3), 'o', 'Markersize',10, 'color', 'r');
```

```

hold on

%----Membuat halangan berupa tabung, terdiri dari posisi (Cpos) dan radius
%----Bila ingin menambah halangan, tambahkan saja posisi dan radiusnya
Cpos = [200,200,50;300,275,50;220,270,5;240,250,30];
radius = [10;10;10;10];
for i=1:4 %----for untuk plot halangan, sesuaikan jumlahnya dengan jumlah
halangan
    create_cylinder(radius(i,1),Cpos(i,:),[0.25, 0.58, 0.96]); %(radius, posisi,
warna)
end

%----Untuk path planning
obstacles = transpose(Cpos); %----memasukan posisi halangan ke variabel baru
bernama obstacles
iteration = 900000; %----Jumlah iterasi, semakin banyak, simulasi akan semakin
lama.
                %Skenario kompleks yang memiliki banyak rintangan
                %umumnya memerlukan jumlah iterasi yang banyak
                %Default untuk skenario 3 adalah 900000 iterasi

current_pos{1} = transpose(start{1}); %----memasukan posisi start ke variabel
baru current_pos
current_pos{2} = transpose(start{2}); %----nantinya variabel ini akan menyimpan
nilai posisi quadcopter pada iterasi ke-i
current_pos{3} = transpose(start{3}); %----tambah bila ingin menambah quadcopter
current_pos{4} = transpose(start{4});
current_pos{5} = transpose(start{5});

previous_pos{1} = current_pos{1}; %----memasukan posisi sekarang ke variabel lain
bernama previous_pos
previous_pos{2} = current_pos{2}; %----nantinya variabel ini akan menyimpan
posisi quadcopter pada iterasi ke-(i-1)
previous_pos{3} = current_pos{3}; %----tambah bila ingin menambah quadcopter
previous_pos{4} = current_pos{4};
previous_pos{5} = current_pos{5};

Krep = 0.04; %----Nilai gain repulsif yang akan mempengaruhi penolakan dari
quadcopter lain dan halangan
Katt = 0.3; %----Nilai gain atraktif yang akan mempengaruhi kekuatan goal
menarik quadcopter

data_points{1} = zeros(iteration,3); %----menyimpan data posisi tiap iterasi
dalam variabel data_points
data_points{2} = zeros(iteration,3); %----sebagai awal, diberi semua nilai = 0
(zeros)
data_points{3} = zeros(iteration,3);
data_points{4} = zeros(iteration,3);
data_points{5} = zeros(iteration,3);

F = zeros(3,length(obstacles)); %----nilai awal untuk gaya repulsif dari
halangan. Nilai awal = 0
all_Ft{1}=zeros(3,1); %----untuk menyimpan semua nilai F total (Ft)
all_Ft{2}=all_Ft{1}; %----nilai awalnya adalah 0
all_Ft{3}=all_Ft{1};
all_Ft{4}=all_Ft{1};
all_Ft{5}=all_Ft{1};

```

```

%-----Beberapa Parameter untuk formation control
l=2; %jarak formation
v=pi/4; %sudut formation

tic
for i=1:iteration %-----for dengan jumlah iterasi untuk proses keseluruhan

    for j=1:1 %-----perulangan untuk leader
        agenlain{1}=[current_pos{mod(j,5)+1}]; %-----leader melihat posisi
        quadcopter lain (agenlain)
        agenlain{2}=[current_pos{mod(j+1,5)+1}];
        agenlain{3}=[current_pos{mod(j+2,5)+1}];
        agenlain{4}=[current_pos{mod(j+3,5)+1}];

        if i>1 %-----Jika iterasi lebih dari 1, F total previous
            (Ftp) quadcopter j adalah Ft yang sebelumnya
            Ftp{j}=Ft{j}; %-----Ftp akan berguna sebagai referensi bagi
            dinamika quadcopter pada fungsi UAV_input_Ft
        else
            Ftp{j}=[0;0;0];
        end

        %-----melakukan perhitungan Ft dengan fungsi avoidance_multiRobot2
        %-----bisa dikatakan perhitungan F dari metode APF
        %-----Ft akan berguna sebagai referensi bagi dinamika quadcopter pada
        fungsi UAV_input Ft
        [Ft{j}, rou{j}, zeta{1}] =
        avoidance_multiRobot_2(current_pos{j},goal{j},Katt,Krep,obstacles,radius,F,agenlai
        n);

        %-----Modifikasi: Jika jarak quadcopter lain terlampau jauh dari
        %leader maka leader akan memperlambat dirinya dengan mengecilkan
        %nilai Ft-nya.
        if (sqrt((current_pos{1}(1)-current_pos{2}(1))^2 + (current_pos{1}(2)-
        current_pos{2}(2))^2)>=4) || (sqrt((current_pos{1}(1)-current_pos{3}(1))^2 +
        (current_pos{1}(2)-current_pos{3}(2))^2)>=4) || (sqrt((current_pos{1}(1)-
        current_pos{4}(1))^2 + (current_pos{1}(2)-
        current_pos{4}(2))^2)>=8) || (sqrt((current_pos{1}(1)-current_pos{5}(1))^2 +
        (current_pos{1}(2)-current_pos{5}(2))^2)>=8)
            Ft{1}(1) = Ft{1}(1)*0.01;
            Ft{1}(2) = Ft{1}(2)*0.01;
        end

        previous_pos{j} = current_pos{j}; %-----update nilai previous_pos
        state{j}=UAV_input_Ft(state{j},Ft{j},Ftp{j}); %-----menghitung nilai state
        (12 variabel state) dari quadcopter
        %-----disinilah

        pers_state_uavnkontroller diakses
        current_pos{j}=state{j}(1:3); %-----posisi sekarang diambil dari state
        variabel ke 1 sampai 3 (X Y Z)
        all_state{j}(:,i+1)=state{j}'; %-----update nilai all_state dengan nilai
        state sekarang
        all_Ft{j}(:,i+1)=Ft{j}; %-----update nilai all_Ft dengan nilai Ft sekarang
        data_points{j}(i,:)=transpose(current_pos{j}); %-----update nilai
        data_points dengan posisi sekarang
    end
    for j=2:5 %-----berikut perulangan untuk quadcopter 2 sampai 5 (followers).
        %-----kodenya mirip dengan quadcopter leader, hanya ada

```

```

        %sedikit perbedaan dan akan saya beri keterangan
        agenlain{1}=[current_pos{mod(j,5)+1}];
        agenlain{2}=[current_pos{mod(j+1,5)+1}];
        agenlain{3}=[current_pos{mod(j+2,5)+1}];
        agenlain{4}=[current_pos{mod(j+3,5)+1}];

        %----penentuan nilai goal untuk quadcopter follower, quadcopter 2
        %dan 3 akan mendapat pengaruh dari leader sedangkan quadcopter 4
        %dan 5 mendapat pengaruh dari 2 dan 3 secara berurutan.
        %----disinilah bentuk formasi bisa diatur
        goal{2}=[current_pos{1,1}(1)+2*1*sin(v) current_pos{1,1}(2)-6*1*cos(v)
current_pos{1,1}(3)]'; %1 2
        goal{3}=[current_pos{1,1}(1)-6*1*sin(v) current_pos{1,1}(2)+2*1*cos(v)
current_pos{1,1}(3)]'; %2 1
        goal{4}=[current_pos{1,2}(1)+2*1*sin(v) current_pos{1,2}(2)-6*1*cos(v)
current_pos{1,2}(3)]'; %1 2
        goal{5}=[current_pos{1,3}(1)-6*1*sin(v) current_pos{1,3}(2)+2*1*cos(v)
current_pos{1,3}(3)]'; %2 1

        if i>1
            Ftp{j}=Ft{j};
        else
            Ftp{j}=[0;0;0];
        end

        [Ft{j}, rou{j}, zeta{j}] =
avoidance_multiRobot(current_pos{j},goal{j},Katt,Krep,obstacles,radius,F,agenlain)
;

        %----Modifikasi: Jika jarak quadcopter followers dengan goalnya
        %terlampau jauh maka quadcopter akan mempercepat dirinya dengan
        %memperbesar nilai Ft-nya
        if (sqrt((goal{j}(1)-current_pos{j}(1))^2 + (goal{j}(2)-
current_pos{j}(2))^2)>0)
            Ft{j}(1) = Ft{j}(1)*15;
            Ft{j}(2) = Ft{j}(2)*15;
        end

        previous_pos{j} = current_pos{j};

state{j}=UAV_input_Ft_follower(current_pos{j},goal{j},state{j},Ft{j},Ftp{j});
        current_pos{j}=state{j}(1:3);
        all_state{j}(:,i+1)=state{j}';
        all_Ft{j}(:,i+1)=Ft{j};
        data_points{j}(i,:)=transpose(current_pos{j});
    end
    %----Beberapa Parameter untuk formation control
    l=2; %formation distance
    v=pi/4; %formation angle

    %----Menghitung nilai selisihIdeal, yaitu jarak ideal antar quadcopter
    %dalam formasi
    %----Akan digunakan pada perhitungan Index performansi
    for k=1:5
        for h=1:5
            if k==h
                selisihIdeal{k,h}(i,:)= 0;
            else

```

```

                selisihIdeal{k,h}(i,:)= sqrt((data_points{k}(i,1)-
goal{h}(1))^2+(data_points{k}(i,2)-goal{h}(2))^2); %+(data_points{k}(i,3)-
goal{h}(3))^2);
            end
        end
    end

    %----Menghitung nilai selisih, yaitu jarak aktual antar quadcopter
    %dalam formasi
    %----Akan digunakan pada perhitungan Index performansi
    for k=1:5
        for h=1:5
            if k==h
                selisih{k,h}(i,:)= 0;
            else
                selisih{k,h}(i,:)= sqrt((data_points{k}(i,1)-
data_points{h}(i,1))^2+(data_points{k}(i,2)-
data_points{h}(i,2))^2); %+(data_points{k}(i,3)-data_points{h}(i,3))^2);
            end
        end
    end
end
toc

%----Setting untuk plot MATLAB
grid on
xlabel("x (m)");
ylabel("y (m)");
zlabel("z (m)");
xlim([0 500]);
ylim([0 500]);
zlim([0 50]);

%----Menunjukkan trayektori pergerakan tiap quadcopter
%----Bila menambah quadcopter, tambahkan bagian ini
plot3(data_points{1}(:,1),data_points{1}(:,2),data_points{1}(:,3), 'Color', 'r');
plot3(data_points{2}(:,1),data_points{2}(:,2),data_points{2}(:,3), 'Color', 'g');
plot3(data_points{3}(:,1),data_points{3}(:,2),data_points{3}(:,3), 'Color', 'b');
plot3(data_points{4}(:,1),data_points{4}(:,2),data_points{4}(:,3), 'Color', 'c');
plot3(data_points{5}(:,1),data_points{5}(:,2),data_points{5}(:,3), 'Color', 'k');

%----Menunjukkan garis antar posisi quadcopter pada iterasi ke-10000
%----Bila ingin melihat pada iterasi lain, tinggal diganti saja
i = 10000;
line([data_points{1}(i,1),data_points{2}(i,1)], [data_points{1}(i,2),data_points{2}
(i,2)], [data_points{1}(i,3),data_points{2}(i,3)]) %Garis dari UAV 1 ke 2
line([data_points{1}(i,1),data_points{3}(i,1)], [data_points{1}(i,2),data_points{3}
(i,2)], [data_points{1}(i,3),data_points{3}(i,3)]) %Garis dari UAV 1 ke 3
line([data_points{2}(i,1),data_points{3}(i,1)], [data_points{2}(i,2),data_points{3}
(i,2)], [data_points{2}(i,3),data_points{3}(i,3)]) %Garis dari UAV 2 ke 3
line([data_points{1}(i,1),data_points{4}(i,1)], [data_points{1}(i,2),data_points{4}
(i,2)], [data_points{1}(i,3),data_points{4}(i,3)]) %Garis dari UAV 1 ke 4
line([data_points{1}(i,1),data_points{5}(i,1)], [data_points{1}(i,2),data_points{5}
(i,2)], [data_points{1}(i,3),data_points{5}(i,3)]) %Garis dari UAV 1 ke 5

%----Menunjukkan garis antar posisi quadcopter pada iterasi ke-25000
i = 25000;

```

```

line([data_points{1}(i,1),data_points{2}(i,1)], [data_points{1}(i,2),data_points{2}(i,2)], [data_points{1}(i,3),data_points{2}(i,3)]) %Garis dari UAV 1 ke 2
line([data_points{1}(i,1),data_points{3}(i,1)], [data_points{1}(i,2),data_points{3}(i,2)], [data_points{1}(i,3),data_points{3}(i,3)]) %Garis dari UAV 1 ke 3
line([data_points{2}(i,1),data_points{3}(i,1)], [data_points{2}(i,2),data_points{3}(i,2)], [data_points{2}(i,3),data_points{3}(i,3)]) %Garis dari UAV 2 ke 3
line([data_points{1}(i,1),data_points{4}(i,1)], [data_points{1}(i,2),data_points{4}(i,2)], [data_points{1}(i,3),data_points{4}(i,3)]) %Garis dari UAV 1 ke 4
line([data_points{1}(i,1),data_points{5}(i,1)], [data_points{1}(i,2),data_points{5}(i,2)], [data_points{1}(i,3),data_points{5}(i,3)]) %Garis dari UAV 1 ke 5

%----Menunjukkan garis antar posisi quadcopter pada iterasi ke-40000
i = 40000;
line([data_points{1}(i,1),data_points{2}(i,1)], [data_points{1}(i,2),data_points{2}(i,2)], [data_points{1}(i,3),data_points{2}(i,3)]) %Garis dari UAV 1 ke 2
line([data_points{1}(i,1),data_points{3}(i,1)], [data_points{1}(i,2),data_points{3}(i,2)], [data_points{1}(i,3),data_points{3}(i,3)]) %Garis dari UAV 1 ke 3
line([data_points{2}(i,1),data_points{3}(i,1)], [data_points{2}(i,2),data_points{3}(i,2)], [data_points{2}(i,3),data_points{3}(i,3)]) %Garis dari UAV 2 ke 3
line([data_points{1}(i,1),data_points{4}(i,1)], [data_points{1}(i,2),data_points{4}(i,2)], [data_points{1}(i,3),data_points{4}(i,3)]) %Garis dari UAV 1 ke 4
line([data_points{1}(i,1),data_points{5}(i,1)], [data_points{1}(i,2),data_points{5}(i,2)], [data_points{1}(i,3),data_points{5}(i,3)]) %Garis dari UAV 1 ke 5

%----Menunjukkan garis antar posisi quadcopter pada iterasi terakhir
line([data_points{1}(iteration,1),data_points{2}(iteration,1)], [data_points{1}(iteration,2),data_points{2}(iteration,2)], [data_points{1}(iteration,3),data_points{2}(iteration,3)]) %Garis dari UAV 1 ke 2
line([data_points{1}(iteration,1),data_points{3}(iteration,1)], [data_points{1}(iteration,2),data_points{3}(iteration,2)], [data_points{1}(iteration,3),data_points{3}(iteration,3)]) %Garis dari UAV 1 ke 3
line([data_points{2}(iteration,1),data_points{3}(iteration,1)], [data_points{2}(iteration,2),data_points{3}(iteration,2)], [data_points{2}(iteration,3),data_points{3}(iteration,3)]) %Garis dari UAV 2 ke 3
line([data_points{1}(iteration,1),data_points{4}(iteration,1)], [data_points{1}(iteration,2),data_points{4}(iteration,2)], [data_points{1}(iteration,3),data_points{4}(iteration,3)]) %Garis dari UAV 1 ke 4
line([data_points{1}(iteration,1),data_points{5}(iteration,1)], [data_points{1}(iteration,2),data_points{5}(iteration,2)], [data_points{1}(iteration,3),data_points{5}(iteration,3)]) %Garis dari UAV 1 ke 5

%----Plot jarak leader terhadap followernya
figure(2)
title("Jarak UAV leader terhadap follower-nya")
plot(selisih{1,2})
hold on
plot(selisih{1,3})
plot(selisih{1,4})
plot(selisih{1,5})
ylabel("jarak (m)");
xlabel("iterasi");
legend("UAV 1-2", "UAV 1-3", "UAV 1-4", "UAV 1-5")

%----Plot jarak follower 2 terhadap quadcopter 3, 4, dan 5
figure(3)
title("Jarak UAV 2 terhadap UAV lainnya")
plot(selisih{2,3})
hold on

```

```

plot(selisih{2,4})
plot(selisih{2,5})
ylabel("jarak (m)");
xlabel("iterasi");
legend("UAV 2-3", "UAV 2-4", "UAV 2-5")

```

```

%-----Plot jarak follower 4 ke 5
figure(4)
title("Jarak UAV 4 terhadap UAV 5")
plot(selisih{4,5})
ylabel("jarak (m)");
xlabel("iterasi");
legend("UAV 4-5")

```

## b. Program penghindaran halangan

```

function [Ft, rou,
zeta]=avoidance_multiRobot(current_pos,goal,Katt,Krep,obstacles,radius,F,agenlain)

%Nilai gaya atraktif (Fatt) secara general
Fatt = potential_attraction(Katt, current_pos, goal);

%Perulangan untuk penghindaran halangan
for k = 1: length(obstacles)
    robot_height = current_pos(3,1);
    flag = 0;
    % Menghitung jarak quadcopter terhadap halangan secara horizontal (X
    % dan Y)
    rou = sqrt((current_pos(1,1)-obstacles(1,k))^2+(current_pos(2,1)-
obstacles(2,k))^2)-radius(k,1);
    % turunan dari rou
    d_rou = [current_pos(1,1)-obstacles(1,k); current_pos(2,1)-
obstacles(2,k)]/(rou+radius(k,1));

    % Parameter batas apakah quadcopter terlalu dekat dengan halangan atau
    % tidak
    n = 3.1;
    zeta=2*radius(k,1);

    if rou<=zeta
        if robot_height-0.075 <= obstacles(3,k)
            % Flag adalah variabel yang menyatakan bahwa quadcopter harus
            % bergerak pada bidang X dan Y
            % Jadi tidak ada peningkatan ketinggian
            flag = 1;
            Frep1 = Krep*((1/rou)-(1/zeta))*(1/rou^2)*dist_factor(current_pos,
goal, n, flag)*d_rou;
            Frep2 = -(n/2)*Krep*((1/rou)-(1/zeta))^2*dist_factor(current_pos,
goal, n-1, flag)*diff_distance_factor(current_pos, goal, n, flag);
            F(:,k) = vertcat(Frep1,0)+Frep2;
            Fatt = Katt*[goal(1,1)-current_pos(1,1);goal(2,1)-current_pos(2,1);
0];
            F(3:k)=0;
        else
            F(:,k) = 0;
        end
    elseif rou > zeta
        F(:,k) = 0;
    end
end

```

```

end

%Perulangan untuk penghindaran agent lain
[~,nkk]=size(agenlain);
for kk = 1: nkk
    % Menghitung jarak quadcopter terhadap quadcopter lain secara horizontal
    rouAL = sqrt((current_pos(1,1)-agenlain{kk}(1))^2+(current_pos(2,1)-
agenlain{kk}(2))^2 ...
    +(current_pos(3,1)-agenlain{kk}(3))^2);
    % turunan dari rouAL
    d_rouAL = [current_pos(1,1)-agenlain{kk}(1); current_pos(2,1)-
agenlain{kk}(2); ...
    current_pos(3,1)-agenlain{kk}(3)]/rouAL;

    % Nilai parameter untuk menentukan apakah quad terlalu dekat dengan
    % quadcopter lain atau tidak
    zetaAL = 2;
    nAL = 2;
    if rouAL<=zetaAL
        if robot_height > agenlain{kk}(3)
            % Penghindaran agen lain gerakannya naik
            Frep1AL = Krep*((1/rouAL)-
(1/zetaAL))*(1/rouAL^2)*dist_factor(current_pos, goal, n, flag)*d_rouAL;
            Frep2AL = (n/2)*Krep*((1/rouAL)-(1/zetaAL))^2*dist_factor(current_pos,
goal, n-1, flag)*diff_distance_factor(current_pos, goal, n, flag);
            F(:,k+kk) = Frep1AL+Frep2AL;
            Fatt = Katt*[goal(1,1)-current_pos(1,1);goal(2,1)-current_pos(2,1);
0];
            F(1:2,k+kk) = [0;0];
        else
            F(:,k+kk) = 0;
        end
    elseif rou > zeta
        F(:,k+kk) = 0;
    end
end
end

Frep = sum(F,2); % Menjumlahkan semua gaya repulsif (Frep)
Ft = Fatt+Frep; % Ft atau gaya akhir adalah resultan dari F atraktif dan F
repulsif
if flag == 1
    flag = 0;
    Ft(3,1) = 0;
end
end

```

### c. Program masukan bagi masing-masing *quadcopter*

```

function x=UAV_input_Ft(x,Ft,Ftp)

dt=0.01; % parameter untuk diskritisasi

% Perhitungan kecepatan tranlasi (vx, vy, dan vz)
vx=(Ft(1)-Ftp(1))/dt;
vy=(Ft(2)-Ftp(2))/dt;
vz=(Ft(3)-Ftp(3))/dt;

%Perhitungan untuk referensi pada kontroller
r=zeros(9,1);

```

```

r(1)=x(1)+Ft(1); %referensi posisi
r(2)=x(2)+Ft(2);
r(3)=x(3)+Ft(3);
r(4)=vx;          %referensi kecepatan
r(5)=vy;
r(6)=vz;
dx=pers_state_uavnkotroller(x,r); %dinamika robot + kotroller --> dimasukan ke
fungsi pers_state_uavnkotroller
x=x+dt*dx; % dinamika diskritisasi %update nilai x --> x adalah state quadcopter.
end

```

#### d. Program *plant* dan kotroller *quadcopter*

```

function dx=pers_state_uavnkotroller(x,r)
    dx=[0;0;0;0;0;0;0;0;0;0;0;0];
    %3 posisi x y z
    %3 kecepatan translasi x y z
    %3 orientasi roll pitch yaw
    %3 kecepatan sudut roll pitch yaw
    %parameter
    mm=1;
    g=10;
    Ix=0.03;
    Iy=0.03;
    Iz=0.05;
    dd=0.7; %gaya gesek udara/drag
    d=3.13e-3; %koefisien gerak yaw
    L=0.2; %koefisien gerak roll/pitch

    %Parameter Kotroller
    K2=100;
    L2=21;
    K3=K2;
    L3=L2;
    K4=0.09;
    L4=0.61;
    K1=16;
    L1=9;
    %Kotroller Ketinggian
    %xr= [1 2 3 4 5 6 7 8 9 10 11 12]
    %x = [x y z u v w roll pitch yaw p q r]
    %r = [x y z xdot ydot zdot xddot yddot zddot]
    %u1=mm*(g+K1*(r(3)-x(3))+L1*(r(6)-x(6))+r(9)+dd/mm*r(6))/cos(x(7))/cos(x(8));
    u1=mm*(g+K1*(r(3)-x(3))+L1*(r(6)-x(6))+r(9)+r(6)/mm)/cos(x(7))/cos(x(8));
    u1=saturasi(u1,0,20);
    Peb=[1 sin(x(7))*tan(x(8)) cos(x(7))*tan(x(8))
         0 cos(x(7)) -sin(x(7))
         0 sin(x(7))/cos(x(8)) cos(x(7))/cos(x(8))];
    dx(1:3)=x(4:6);
    dx(4)=(sin(x(9))*sin(x(7))+cos(x(9))*cos(x(7))*sin(x(8)))*u1/mm-dd*x(4)/mm;
    dx(5)=(-cos(x(9))*sin(x(7))+sin(x(9))*cos(x(7))*sin(x(8)))*u1/mm-dd*x(5)/mm;
    dx(6)=-g+cos(x(7))*cos(x(8))*u1/mm-dd*x(6)/mm; %tesis hlm 34
    dx(7:9)=Peb*x(10:12);

    %Kotroller orientasi
    s8=sin(x(8))*cos(x(7));
    s7=sin(x(7));
    ds8=dx(8)*cos(x(7))*cos(x(8))-dx(7)*sin(x(7))*sin(x(8));
    ds7=dx(7)*cos(x(7));

```

```

if (x(9)>0.1 || x(9)<-0.1)
    u2=Iy/L*(-K2*s7-L2*ds7);
    u3=Ix/L*(-K3*s8-L3*ds8);
else
    [r7,r8]=outer_cascade_tracking(x,r);
    u2=Iy/L*(K2*r7-K2*s7-L2*ds7);
    u3=Ix/L*(K3*r8-K3*s8-L3*ds8);
end
u4=Iz/d*(-K4*x(9)-L4*x(12));
%untuk pdot, qdot, rdot
dx(10)=((Iy-Iz)*x(11)*x(12)+u2*L)/Ix;
dx(11)=((Iz-Ix)*x(10)*x(12)+u3*L)/Iy;
dx(12)=((Ix-Iy)*x(10)*x(11)+u4*d)/Iz;
end

function [r7,r8]=outer_cascade_tracking(x,r)
s7=(x(1)-r(1))*cos(x(9))+(x(2)-r(2))*sin(x(9));
s8=-(x(1)-r(1))*sin(x(9))+(x(2)-r(2))*cos(x(9));
v7=(x(4)-r(4))*cos(x(9))+(x(5)-r(5))*sin(x(9));
v8=-(x(4)-r(4))*sin(x(9))+(x(5)-r(5))*cos(x(9));
s=sqrt(s7*s7+s8*s8);
v=sqrt(v7*v7+v8*v8);
sa=25;
sb=6;
vc=4;
if (s>sa)
    a=1;
    b=0;
elseif (s>sb)
    a=(s-sb)/(sa-sb);
    b=1-a;
else
    a=0;
    b=s/sb;
end
if (v>vc)
    c=1;
else
    c=v/vc;
end
rra=0.5;
rrb=0.3;
rrc=0.5;
if(v>0 && s>0)
    r7=s8/s*rra*a+s8/s*rrb*b*(1-c)+v8/v*rrc*c*(1-a);
    r8=-(s7/s*rra*a+s7/s*rrb*b*(1-c)+v7/v*rrc*c*(1-a));
elseif (s>0)
    r7=s8/s*rra*a+s8/s*rrb*b*(1-c);
    r8=-(s7/s*rra*a+s7/s*rrb*b*(1-c));
elseif (v>0)
    r7=v8/v*rrc*c*(1-a);
    r8=-(v7/v*rrc*c*(1-a));
else
    r7=0;
    r8=0;
end
if (r7~=0 && r8~=0)
    rrr=sqrt(r7^2+r8^2);

```

```

        if (rrr>0.5)
            r7=0.5*r7/rrr;
            r8=0.5*r8/rrr;
        end
    end
end
end

function z=saturasi(x,lb,ub)
    if (x>ub)
        z=ub;
    elseif (x<lb)
        z=lb;
    else
        z=x;
    end
end
end

```

#### e. Program untuk membuat halangan

```

% Function for creating cylinders
% The following function is referred from the Dr Lyuba ALBOUL lecture notes.
function [h1, h2, h3] = create_cylinder(Radius, pos,color)
nSides = 100;
[X,Y,Z] = cylinder(Radius, nSides);
Height = pos(3);
Z = Z*Height;
Xpos = pos(1);
Ypos = pos(2);
X = X + Xpos;
Y = Y + Ypos;
h1 = surf(X,Y,Z, 'facecolor',color, 'LineStyle','none');
h2 = fill3(X(1,:),Y(1,:),Z(1,:),color);
h3 = fill3(X(2,:),Y(2,:),Z(2,:),color);
end

```

#### f. Program penghitung jarak

```

function [d] = dist_factor(cu, tu, n,flag)
x = cu(1,1);
y = cu(2,1);
% changing the z based on the robot position if robot is near to obstracle it only
moves in xy plane.
if flag == 1
    z = 0;
    zt = 0;
else
    z = cu(3,1);
    zt = tu(3,1);
end
xt = tu(1,1);
yt = tu(2,1);
d = abs((x-xt)^n)+abs((y-yt)^n)+abs((z-zt)^n);
end

```

#### g. Program penghitung turunan jarak

```

function [diff_dist] = diff_distance_factor(cu, tu, n,flag)
x = cu(1,1);
y = cu(2,1);

```

```

if flag ==1 % changing the z based on the robot position if robot is near to
obstacle it only moves in xy plane.
    z = 0;
    zt = 0;
else
    z = cu(3,1);
    zt = tu(3,1);
end
xt = tu(1,1);
yt = tu(2,1);
diff_dist = [abs(n*(x-xt)^(n-1)); abs(n*(y-yt)^(n-1)); abs(n*(z-zt)^(n-1))]; %
differentiation of dist_factor
end

```

#### h. Program indeks performansi formasi

```

iteration = 900000; %ubah berdasarkan jumlah iterasi yang dipakai di program utama
for i=1:iteration
    for k=1:5
        for h=1:5
            if k==h
                index{k,h}(i,:)= 0;
            else
                index{k,h}(i,:)= abs(abs(selisih{k,h}(i))-
abs(selisihIdeal{k,h}(i)));
            end
        end
    end
    indexTotal(i) = index{1,2}(i) + index{1,3}(i) + index{1,4}(i) + index{1,5}(i);
end
plot(indexTotal)
title("Indeks Performansi Formasi")
ylabel("I(t)");
xlabel("iterasi");

```

## BIODATA PENULIS



Penulis dilahirkan di Surabaya, 10 Mei 2001, merupakan anak pertama dari 2 bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK Kartika Surabaya, SD Katolik Santa Clara Surabaya, SMP Katolik Santa Clara Surabaya, dan SMA Katolik St. Louis 1 Surabaya. Setelah lulus dari SMA pada tahun 2019, Penulis mendaftar dan diterima di Departemen Teknik Elektro FTEIC - ITS pada tahun 2019 melalui jalur SNMPTN dan terdaftar dengan NRP 07111940000038. Setelah melewati tahun kedua, penulis menjatuhkan pilihan bidang studi pada Teknik Sistem Pengaturan.

Selama kuliah, penulis aktif di beberapa kegiatan yang diselenggarakan oleh departemen, Keluarga Mahasiswa Katolik ITS (KMK ITS), EVOLUTION ITS, dan aktif menjadi asisten Laboratorium Kontrol dan Otomasi ITS.