

SISTEM KENDALI LINTAS TERBANG PESAWAT JET BISNIS BERBASIS *DEEP LEARNING*

TUGAS SARJANA

Disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik dari
Institut Teknologi Bandung

Oleh

Adi Kelvianto

13618043

Pembimbing:

Rianto Adhy Sasongko, S.T, M.Sc., Ph.D.

Dr. -Ing. Ir. Javensius Sembiring, S.T, M.T.

Seno Sahisnu S.T, M.T.



**PROGRAM STUDI SARJANA TEKNIK DIRGANTARA
FAKULTAS TEKNIK MESIN DAN DIRGANTARA
INSTITUT TEKNOLOGI BANDUNG**

2022

LEMBAR PENGESAHAN

SISTEM KENDALI LINTAS TERBANG PESAWAT JET BISNIS BERBASIS *DEEP LEARNING*

Oleh

Adi Kelvianto

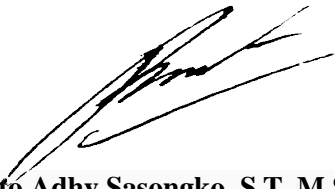
NIM: 13618043

Telah diperiksa dan disetujui sebagai Tugas Akhir Sarjana Program
Studi Teknik Dirgantara Institut Teknologi Bandung

Bandung, 19 September 2022

Pembimbing I

Pembimbing II



Rianto Adhy Sasongko, S.T, M.Sc., Ph.D.

NIP. 19740215 199903 1 002

Dr. -Ing. Ir. Javensius Sembiring, S.T, M.T.

NIP. 19810121 201404 1 001

Pembimbing III



Seno Sahisnu Rawikara, S.T, M.T.

NIP. 1911029 201903 1 008

ABSTRAK

SISTEM KENDALI LINTAS TERBANG PESAWAT JET BISNIS BERBASIS *DEEP LEARNING*

Dalam pelaksanaan *Area Navigation* (RNAV), pilot perlu memanfaatkan metode *fly-by* dan *fly-over waypoints* untuk mencapai destinasi yang dituju. Sistem yang dikembangkan pada Tugas Akhir ini berfokus pada sistem kendali *fly-over waypoints* atau lintas terbang. Strategi *PID controller* umum dijumpai dalam membangun sistem kendali ini. Namun, dengan melihat bahwa penggunaan *Deep Learning* (DL) dapat mengatasi permasalahan kendali yang muncul pada strategi PID. Maka, Tugas Akhir ini bertujuan mengembangkan sistem kendali lintas terbang berbasis DL dengan studi kasus pesawat Cirrus Vision SF50 dan mempelajari pengaruh karakter data pembelajaran terhadap performa kendali yang dihasilkan. Dalam rangka mencapai hal tersebut, dilakukan rekonstruksi data misi penerbangan melalui simulator penerbangan yang terintegrasi dengan sistem kendali lintas terbang berbasis PID. Hasil yang didapat melalui penelitian ini yaitu beberapa sistem kendali lintas terbang berbasis DL mampu memberikan keseimbangan nilai metrik kedekatan dengan *waypoint* dan *cross track distance* yang lebih baik dibandingkan metode *PID controller*. Didapati juga bahwa karakter pengendalian yang dihasilkan model DL berkaitan erat dengan karakter data yang digunakan untuk melatih model.

Kata Kunci: Sistem kendali, Lintas terbang *waypoint*, Cirrus Vision, *Deep Learning*, *PID controller*, *Bayesian optimization*.

ABSTRACT

DEEP LEARNING BASED FLY-OVER WAYPOINTS CONTROL SYSTEM FOR BUSINESS JET AIRCRAFT

In order to reach the intended destination, Area Navigation (RNAV) requires pilot to use fly-by and fly-over waypoints method. System that developed in this thesis is focusing on implementing fly-over waypoints method. PID controller strategies are commonly found in building this control system. However, by seeing the advantages offered by Deep Learning (DL) which can overcome problems that occurs in PID strategy. Hence, this thesis aims to develop DL-based fly over waypoints control system for Cirrus Vision SF50 aircraft and study the relationship between learning data characteristic with the result of control performance. Reconstruction of flight mission data through flight simulator that integrated with PID-based fly-over waypoints method is firstly done before creating DL model. The result obtained through this research are several DL-based fly-over control systems are able to provide a better balance of minimum distance to waypoints and cross track distances than the PID controller method. It was also found that the control characteristic of the DL model is closely related to the characteristic of the data used to train the model.

Keywords: *Control system, Fly-over waypoint, Cirrus Vision, Deep Learning, PID controller, Bayesian optimization.*

KATA PENGANTAR

Puji Syukur kepada Tuhan yang Maha Esa, yang telah memberikan kekuatan-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir dengan judul “Sistem Kendali Lintas Terbang pada Pesawat Jet Bisnis Berbasis *Deep Learning*”. Laporan Tugas akhir ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik dari Institut Teknologi Bandung.

Pada kesempatan ini, penulis mengucapkan banyak terima kasih pada semua pihak yang telah turut membantu dan berjasa selama penyelesaian laporan Tugas Akhir ini, antara lain:

1. Ayah, Ibu, Nenek serta Kakak saya yang memberikan doa, dukungan dan semangat selama penulis menjalani masa studi di Institut Teknologi Bandung.
2. Rianto Adhy Sasongko, S.T., M.Sc., Ph.D., Dr. -Ing. Ir. Javensius Sembiring, S.T, M.T., dan Seno Sahisnu Rawikara, S.T, M.T. selaku dosen pembimbing yang telah memberikan ilmu, nasihat dan saran kepada penulis.
3. Ony Arifianto, Ph.D. dan Ir. Toto Indriyanto, M.Sc., Ph.D. selaku penguji pada sidang Tugas Akhir yang telah memberikan banyak masukan, nasihat, dan saran untuk penulis.
4. Teman seperbimbingan, Ikhwan dan Iqbal yang telah menjadi tempat untuk berdiskusi dan berbagi *insight*.
5. Teman-teman Teknik Dirgantara 2018, Raven, yang menjadi teman seperjuangan dalam mempelajari ilmu dirgantara.
6. Teman-teman KMPN serta UATM ITB yang memberikan warna bagi penulis dalam menjalani perkuliahan di ITB.
7. Serta teman-teman penulis yang tidak dapat disebutkan satu per satu, yang telah menjadi tempat cerita, bersenang-senang dan berbagi saran.
8. Pak Yadi dan *staff* tata usaha FTMD yang telah membantu urusan administrasi penulis.

Penulis menyadari bahwa laporan Tugas Akhir ini masih jauh dari kata sempurna dan masih banyak kekurangan. Oleh karena itu, Penulis sangat mengharapkan kritik dan saran dari pembaca. Penulis berharap laporan ini dapat menambah wawasan serta memberikan manfaat bagi pembaca.

Bandung, September 2022

A handwritten signature in black ink, appearing to be 'Adi Kelvianto', written in a cursive style.

Adi Kelvianto

DAFTAR ISI

LEMBAR PENGESAHAN	i
ABSTRAK.....	ii
<i>ABSTRACT</i>	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah.....	3
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan	4
BAB II DASAR TEORI	6
2.1 <i>Deep Learning</i>	6
2.2 <i>Hyperparameter Optimization</i>	8
2.3 TensorFlow dan KerasTuner.....	9
2.4 Dinamika Terbang Pesawat.....	10
2.4.1 Tata Acuan Koordinat	10
2.4.2 Gerak Pesawat dalam Ruang 3 Dimensi	12
2.4.3 Matra Gerak Pesawat.....	13
2.4.4 Perangkat Kendali Terbang	15
2.5 <i>Area Navigation</i>	19
BAB III REKONSTRUKSI DATA MISI PENERBANGAN.....	22

3.1	Simulator Sistem Kendali dan Lintasan Terbang	22
3.1.1	Subsitem Dinamika	30
3.1.2	Subsistem <i>Monitoring</i>	32
3.1.3	Subsistem <i>Controller</i>	35
3.2	Proses Pengumpulan Data pembelajaran	41
3.2.1	Persiapan Perangkat Lunak MATLAB	42
3.2.2	Persiapan Perangkat Lunak X-Plane 11	44
3.2.3	Rekonstruksi Data Misi Penerbangan	45
BAB IV SISTEM KENDALI LINTAS TERBANG BERBASIS <i>DEEP LEARNING</i>		49
4.1	Pra-pemrosesan Data.....	49
4.1.1	<i>Resample Frequency</i>	49
4.1.2	Pembersihan Data.....	50
4.1.3	<i>Data Splitting</i>	52
4.2	Pengembangan Model <i>Deep Learning</i>	53
4.2.1	Pemilihan <i>Features</i> dan <i>Targets</i>	55
4.2.2	Normalisasi Data	58
4.2.3	<i>Bayesian Optimization</i> Menggunakan KerasTuner.....	58
4.2.4	Pembelajaran Model Menggunakan TensorFlow.....	60
4.2.5	Integrasi Model DL dengan X-Plane.....	64
BAB V HASIL SIMULASI DAN ANALISIS		68
5.1	Skenario Simulasi	68
5.2	Hasil Simulasi Misi Terbang A.....	73
5.3	Hasil Simulasi Misi Terbang B	88
5.4	Analisis.....	103
5.4.1	Misi Terbang A	103
5.4.2	Misi Terbang B.....	114

BAB VI KESIMPULAN DAN SARAN	127
6.1 Kesimpulan	127
6.2 Saran.....	128
DAFTAR PUSTAKA	129

DAFTAR GAMBAR

Gambar 1.1 <i>Fly-by</i> dan <i>fly-over waypoints</i>	1
Gambar 2.1 Diagram blok <i>neural network</i>	6
Gambar 2.2 Tata acuan koordinat benda	10
Gambar 2.3 Tata acuan koordinat angin	11
Gambar 2.4 Tata acuan koordinat horison lokal (NED)	12
Gambar 2.5 Sumbu rotasi pesawat	12
Gambar 2.6 Variabel sikap pada matra gerak longitudinal	13
Gambar 2.7 Variabel sikap pada matra gerak lateral-direksional	14
Gambar 2.8 Pemodelan sistem kendali lingkaran terbuka pada pesawat	17
Gambar 2.9 Pemodelan sistem kendali kontrol otomatis pada pesawat	18
Gambar 2.10 Terminologi navigasi	20
Gambar 3.1 Diagram komunikasi antar subsistem UAVSIM-A	23
Gambar 3.2 Tampilan utama file "MAIN_UAVSIM_Simulink.slx"	29
Gambar 3.3 Subsistem <i>data splitting</i>	29
Gambar 3.4 Pesawat Cirrus Vision SF50	31
Gambar 3.5 <i>Sidestick</i> pada Cirrus Vision SF50	32
Gambar 3.6 <i>User interface</i> subsistem <i>monitoring</i>	32
Gambar 3.7 <i>User interface</i> bagian <i>mission planner</i>	34
Gambar 3.8 Subsistem <i>controller</i>	35
Gambar 3.9 Implementasi sistem kendali <i>wing leveler</i> pada Simulink.....	36
Gambar 3.10 Implementasi sistem kendali <i>roll limiter</i> pada Simulink	36
Gambar 3.11 Strategi kendali <i>waypoint following</i>	37
Gambar 3.12 Implementasi sistem kendali <i>yaw hold</i> pada Simulink	38
Gambar 3.13 Implementasi sistem kendali <i>waypoint following</i> pada Simulink.....	39
Gambar 3.14 <i>Pseudocode controller data selection</i>	39
Gambar 3.15 Implementasi sistem kendali <i>pitch hold</i> pada Simulink	40
Gambar 3.16 Alur persiapan perangkat lunak MATLAB	42
Gambar 3.17 Contoh pendefinisian misi pada <i>mission planner</i>	43
Gambar 3.18 Alur persiapan perangkat lunak X-Plane 11	44
Gambar 3.19 Tampilan <i>flight configuration</i> X-Plane 11	45
Gambar 3.20 Alur kerja rekonstruksi data misi.....	46
Gambar 3.21 Sampel lintasan simulasi	48
Gambar 4.1 <i>Pseudocode resample frequency</i>	49
Gambar 4.2 <i>Pseudocode remove outliers</i>	50
Gambar 4.3 <i>Pseudocode data smoothing</i>	51
Gambar 4.4 <i>Pseudocode data slicing</i>	52

Gambar 4.5 Alur pengembangan model DL	54
Gambar 4.6 Nilai rata-rata <i>yaw error</i> dari keempat <i>set data</i>	55
Gambar 4.7 Nilai rata-rata <i>yaw error rate</i> dari keempat <i>set data</i>	56
Gambar 4.8 Nilai rata-rata sudut <i>roll</i> dari keempat <i>set data</i>	56
Gambar 4.9 Nilai rata-rata rasio <i>joystick roll</i> dari keempat <i>set data</i>	56
Gambar 4.10 <i>Pseudocode base model</i>	59
Gambar 4.11 Grafik <i>loss vs epochs</i>	61
Gambar 4.12 Plot prediksi <i>joystick roll</i> model DL vs data validasi	62
Gambar 4.13 Pengaturan blok <i>predict</i>	65
Gambar 4.14 Pengimplementasian model DL pada Simulink	65
Gambar 4.15 Modifikasi subsistem <i>waypoint controller</i> pada Simulink	66
Gambar 4.16 Diagram pada block "PID & Model DL"	66
Gambar 5.1 Bentuk lintasan misi terbang A	68
Gambar 5.2 Bentuk lintasan misi terbang B	68
Gambar 5.3 <i>Pseudocode minimum distance to waypoint</i>	70
Gambar 5.4 Grafik hasil simulasi metode PID percobaan 1 pada misi A	74
Gambar 5.5 Grafik hasil simulasi metode PID percobaan 2 pada misi A	75
Gambar 5.6 Grafik hasil simulasi model DL <i>set 1</i> percobaan 1 pada misi A	76
Gambar 5.7 Grafik hasil simulasi model DL <i>set 1</i> percobaan 2 pada misi A	77
Gambar 5.8 Grafik hasil simulasi model DL <i>set 2</i> percobaan 1 pada misi A	78
Gambar 5.9 Grafik hasil simulasi model DL <i>set 2</i> percobaan 2 pada misi A	79
Gambar 5.10 Grafik hasil simulasi model DL <i>set 3</i> percobaan 1 pada misi A	80
Gambar 5.11 Grafik hasil simulasi model DL <i>set 3</i> percobaan 2 pada misi A	81
Gambar 5.12 Grafik hasil simulasi model DL <i>set 4</i> percobaan 1 pada misi A	82
Gambar 5.13 Grafik hasil simulasi model DL <i>set 4</i> percobaan 2 pada misi A	83
Gambar 5.14 Grafik hasil simulasi metode PID percobaan 1 pada misi B	89
Gambar 5.15 Grafik hasil simulasi metode PID percobaan 2 pada misi B	90
Gambar 5.16 Grafik hasil simulasi model DL <i>set 1</i> percobaan 1 pada misi B	91
Gambar 5.17 Grafik hasil simulasi model DL <i>set 1</i> percobaan 2 pada misi B	92
Gambar 5.18 Grafik hasil simulasi model DL <i>set 2</i> percobaan 1 pada misi B	93
Gambar 5.19 Grafik hasil simulasi model DL <i>set 2</i> percobaan 2 pada misi B	94
Gambar 5.20 Grafik hasil simulasi model DL <i>set 3</i> percobaan 1 pada misi B	95
Gambar 5.21 Grafik hasil simulasi model DL <i>set 3</i> percobaan 2 pada misi B	96
Gambar 5.22 Grafik hasil simulasi model DL <i>set 4</i> percobaan 1 pada misi B	97
Gambar 5.23 Grafik hasil simulasi model DL <i>set 4</i> percobaan 2 pada misi	98
Gambar 5.24 Grafik nilai rata-rata jarak min. terhadap seluruh <i>waypoint</i> pada misi terbang A ...	103
Gambar 5.25 Grafik nilai rata-rata CTD terhadap seluruh <i>waypoint</i> pada misi terbang A	104
Gambar 5.26 Grafik nilai rata-rata sudut roll terhadap seluruh <i>waypoint</i> pada misi terbang A	104
Gambar 5.27 TAS dan ketinggian terhadap waktu pada misi terbang A	111

Gambar 5.28 Grafik nilai rata-rata jarak min. terhadap seluruh <i>waypoint</i> pada misi terbang B ...	114
Gambar 5.29 Grafik nilai rata-rata CTD terhadap seluruh <i>waypoint</i> pada misi terbang B.....	114
Gambar 5.30 Grafik nilai rata-rata sudut roll terhadap seluruh <i>waypoint</i> pada misi terbang B	115
Gambar 5.31 TAS dan ketinggian terhadap waktu pada misi terbang B.....	123

DAFTAR TABEL

Tabel 2.1 Deskripsi terminologi navigasi	20
Tabel 3.1 Aliran data <i>monitoring to dynamics</i>	23
Tabel 3.2 Aliran data <i>dynamics to monitoring</i>	24
Tabel 3.3 Aliran data <i>monitoring to controller</i>	25
Tabel 3.4 Aliran data <i>controller to monitoring</i>	26
Tabel 3.5 Aliran data <i>dynamics to controller</i>	27
Tabel 3.6 Aliran data <i>controller to dynamics</i>	28
Tabel 3.7 Spesifikasi teknis Cirrus Vision SF50	31
Tabel 3.8 Keterangan <i>interface</i> subsistem <i>monitoring</i>	33
Tabel 3.9 Keterangan <i>interface</i> bagian <i>mission planner</i>	34
Tabel 3.10 Variabel data tambahan yang akan disimpan	41
Tabel 4.1 Kategori pembagian data pembelajaran	53
Tabel 4.2 <i>Hyperparameter space</i>	59
Tabel 4.3 Set <i>hyperparameter</i> terbaik	60
Tabel 4.4 Nilai metrik dari setiap model DL	63
Tabel 5.1 Nilai konstanta setiap sistem kendali pada kedua misi uji	69
Tabel 5.2 Detil koordinat <i>waypoint</i> dan perubahan sudut lintasan pada misi terbang A	72
Tabel 5.3 Detil koordinat <i>waypoint</i> dan perubahan sudut lintasan pada misi terbang B	72
Tabel 5.4 Detil jarak minimum pesawat terhadap <i>waypoint</i> metode PID pada misi terbang A	84
Tabel 5.5 Detil jarak minimum pesawat terhadap <i>waypoint</i> model <i>set 1</i> pada misi terbang A	84
Tabel 5.6 Detil jarak minimum pesawat terhadap <i>waypoint</i> model <i>set 2</i> pada misi terbang A	84
Tabel 5.7 Detil jarak minimum pesawat terhadap <i>waypoint</i> model <i>set 3</i> pada misi terbang A	85
Tabel 5.8 Detil jarak minimum pesawat terhadap <i>waypoint</i> model <i>set 3</i> pada misi terbang A	85
Tabel 5.9 Detil nilai <i>cross track distance</i> setiap <i>section</i> metode PID pada misi terbang A	85
Tabel 5.10 Detil nilai <i>cross track distance</i> setiap <i>section</i> model <i>set 1</i> pada misi terbang A	86
Tabel 5.11 Detil nilai <i>cross track distance</i> setiap <i>section</i> model <i>set 2</i> pada misi terbang A	86
Tabel 5.12 Detil nilai <i>cross track distance</i> setiap <i>section</i> model <i>set 3</i> pada misi terbang A	87
Tabel 5.13 Detil nilai <i>cross track distance</i> setiap <i>section</i> model <i>set 4</i> pada misi terbang A	87
Tabel 5.14 Detil jarak minimum pesawat terhadap <i>waypoint</i> metode PID pada misi terbang B	99
Tabel 5.15 Detil jarak minimum pesawat terhadap <i>waypoint</i> model <i>set 1</i> pada misi terbang B	99
Tabel 5.16 Detil jarak minimum pesawat terhadap <i>waypoint</i> model <i>set 2</i> pada misi terbang B	99
Tabel 5.17 Detil jarak minimum pesawat terhadap <i>waypoint</i> model <i>set 3</i> pada misi terbang B ...	100
Tabel 5.18 Detil jarak minimum pesawat terhadap <i>waypoint</i> model <i>set 4</i> pada misi terbang B ...	100
Tabel 5.19 Detil nilai <i>cross track distance</i> setiap <i>section</i> metode PID pada misi terbang B	100
Tabel 5.20 Detil nilai <i>cross track distance</i> setiap <i>section</i> model <i>set 1</i> pada misi terbang B	101
Tabel 5.21 Detil nilai <i>cross track distance</i> setiap <i>section</i> model <i>set 2</i> pada misi terbang B	101

Tabel 5.22 Detil nilai <i>cross track distance</i> setiap <i>section</i> model <i>set 3</i> pada misi terbang B	102
Tabel 5.23 Detil nilai <i>cross track distance</i> setiap <i>section</i> model <i>set 4</i> pada misi terbang B	102
Tabel 5.24 Perbedaan jarak min. dengan setiap <i>waypoint</i> dari metode PID pada misi terbang A	106
Tabel 5.25 Perbedaan <i>cross track distance</i> dari metode PID	108
Tabel 5.26 Perbedaan rentang sudut <i>roll</i> dari metode PID pada misi terbang A.....	109
Tabel 5. 28 Nilai bobot total metriks pada misi terbang A.....	113
Tabel 5.29 Perbedaan jarak min. dengan setiap <i>waypoint</i> dari metode PID pada misi terbang B	117
Tabel 5.30 Perbedaan <i>cross track distance</i> dari metode PID pada misi terbang B	119
Tabel 5.31 Perbedaan rentang sudut <i>roll</i> dari metode PID pada misi terbang B.....	120
Tabel 5.32 Nilai bobot total metriks pada misi terbang B	124

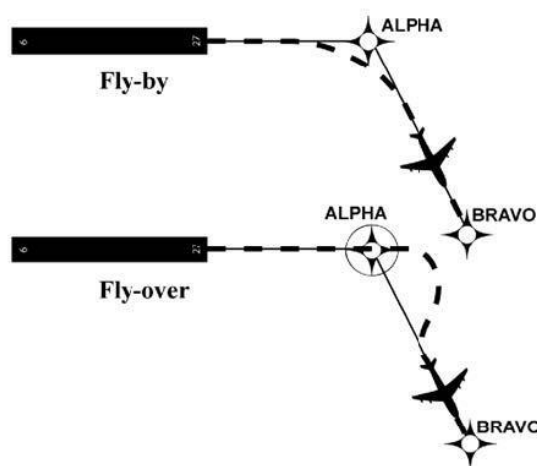
BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada masa kini, setiap pesawat terbang modern baik komersial ataupun bisnis telah dilengkapi dengan sistem bernama *Automatic Flight Control System* (AFCS), yaitu sebuah sistem terintegrasi yang terdiri berbagai macam *autopilot* untuk melaksanakan fungsi-fungsi tertentu. AFCS tidak dikembangkan terbatas pada stabilitas pesawat, namun juga dikembangkan untuk menjalankan fungsi navigasi seperti mengikuti lintasan atau koordinat *longitude* dan *latitude* yang telah didefinisikan atau sering disebut sebagai *waypoints* [1].

Dalam pengoperasian pesawat untuk mengikuti *waypoint* tersebut, terdapat dua metode dalam mencapainya. Metode pertama yang bernama *fly-by waypoints* adalah metode yang digunakan ketika pesawat harus mulai berbelok ke jalur selanjutnya sebelum mencapai *waypoint* yang memisahkan dua segmen rute atau disebut sebagai *turn anticipation* dengan kelebihan yaitu pesawat akan lebih dekat dengan lintasan yang dibentuk dari dua buah *waypoint*. Sedangkan prosedur *fly-over waypoints* atau lintas terbang digunakan ketika pesawat harus melintasi setiap *waypoint* yang didefinisikan [2]. Gambar berikut ini menunjukkan ilustrasi perbedaan dari kedua prosedur yang disebutkan di atas.



Gambar 1.1 *Fly-by* dan *fly-over waypoints* [2]

Penelitian-penelitian sebelumnya [3], [4], [5] dan [6] menunjukkan kesuksesan pengembangan suatu sistem kendali lintas terbang memanfaatkan mekanisme *proportional integral derivative* (PID) *controller* yang disimulasikan melalui simulator penerbangan. Perlu diketahui bahwa lebih dari separuh *controller* yang digunakan di dunia industri memanfaatkan mekanisme PID *controller* beserta modifikasinya, dikarenakan kemampuannya untuk mengatasi berbagai permasalahan kendali dengan mudah dan efisien [7].

Dalam rangka mengatasi kekurangan dari metode PID *controller* pada permasalahan sistem kendali, referensi [8], [9], dan [10] mengusulkan untuk menggunakan *Deep Learning* (DL) dalam mengatasi permasalahan kendali, yang kemudian dibandingkan dengan metode PID *controller*. Hasil yang didapatkan yaitu, dengan menggunakan metode DL, sistem yang dikendalikan dapat menghasilkan performa yang lebih baik terkhusus pada lingkungan yang memiliki banyak gangguan dengan stabilitas yang lebih baik dibandingkan dengan metode PID.

Dengan tersedianya simulator sistem kendali dan lintasan terbang yang telah dikerjakan sebelumnya pada pekerjaan [3]. Maka, informasi dalam proses pengendalian lintas terbang pesawat dapat digunakan untuk mengembangkan sistem kendali lintas terbang berbasis *deep learning* pada pesawat jet bisnis dengan studi kasus pesawat yaitu Cirrus Vision SF50 dan melihat keunggulan atau kekurangan yang dihasilkan dari sistem kendali menggunakan metode *deep learning* dalam melaksanakan misi lintas terbang.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disebutkan, masalah yang menjadi dasar penelitian tugas akhir ini yaitu kebutuhan untuk mengembangkan sistem kendali lintas terbang pesawat jet bisnis dengan metode *deep learning*, memanfaatkan data pengendalian yang terbatas sebagai dasar pembelajaran.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini yaitu:

1. Membangun skema pengembangan sistem kendali dengan metode *deep learning*.
2. Membangun data pembelajaran berbasis simulasi.
3. Melakukan proses pembelajaran dan pembentukan sistem kendali.
4. Mengimplementasikan dan menganalisis karakter pengendalian dari sistem kendali lintas terbang berbasis *deep learning* pada simulator penerbangan.

1.4 Batasan Masalah

Batasan masalah dan ruang lingkup pada penelitian ini terdiri atas:

1. Pesawat yang digunakan sebagai basis pengembangan sistem kendali lintas terbang menggunakan metode *deep learning* adalah pesawat Cirrus Vision SF50.
2. Data pembelajaran diperoleh melalui proses simulasi pada simulator sistem kendali dan lintasan terbang yang telah dikerjakan sebelumnya pada pekerjaan [3] yang dibangun menggunakan kendali linier sederhana (PID) dengan kondisi tidak ada gangguan angin.
3. Proses pembelajaran model dilakukan menggunakan *platform* TensorFlow 2.x.
4. Pemodelan sistem kendali lintas terbang berbasis DL terbatas hanya pada matra lateral-direksional pesawat dan berfungsi hanya sebagai pemberi perintah defleksi *joystick* pada pesawat untuk menggerakkan *aileron*.

1.5 Metodologi Penelitian

Metodologi penelitian yang diterapkan tersusun atas:

1. Kajian Literatur

Kajian literatur dilakukan untuk mempelajari literatur yang berkaitan dengan metode pengendalian lintas terbang dan konsep serta pengimplementasian *deep learning* pada simulator terbang.

2. Melakukan Rekonstruksi Data Misi Penerbangan

Data misi penerbangan untuk proses pelatihan model DL akan direkonstruksi melalui perangkat lunak yang dikembangkan oleh pekerjaan sebelumnya [3] yang mengintegrasikan MATLAB, Simulink, Python dan X-Plane dalam melaksanakan pengendalian lintas terbang berbasis *PID controller*. Pengumpulan data pembelajaran dilakukan dengan memvariasikan bentuk lintasan misi terbang dan juga nilai konstanta PID untuk merepresentasikan pilot yang berbeda dalam menyelesaikan suatu misi penerbangan yang telah ditetapkan.

3. Mengembangkan Sistem Kendali Lintas Terbang Berbasis DL

Data pembelajaran yang telah terkumpul akan melewati tahap pra-pemrosesan data dan selanjutnya akan dipecah menjadi 4 buah *set* data pembelajaran. Selanjutnya, dilakukan *hyperparameter tuning* sebelum model dilatih dan divalidasi menggunakan gabungan *set* data pembelajaran. Model yang telah melewati tahapan validasi akan disimpan.

4. Mengintegrasikan Model DL dengan Simulator Terbang

Model DL yang telah dikembangkan akan diintegrasikan dengan *simulator* penerbangan untuk mengevaluasi karakter pengendalian dari setiap model DL yang dibentuk.

5. Analisis Hasil Simulasi

Analisis akan dibatasi pada 2 jenis lintasan misi penerbangan, dimana keempat model DL dan pengendalian berbasis PID akan diuji pada kedua lintasan tersebut untuk melihat keunggulan dan kelemahan setiap model DL dibandingkan dengan metode PID menggunakan metrik berupa kedekatan pesawat dengan *waypoint* yang dituju.

1.6 Sistematika Penulisan

Penulisan tugas sarjana ini tersusun atas:

1. Bab I Pendahuluan

Berisikan latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah dan metodologi penelitian dari judul tugas sarjana ini.

2. Bab II Dasar Teori

Bab ini membahas dasar teori yang menunjang penelitian ini, yang meliputi *deep learning*, *hyperparameter optimization*, TensorFlow dan KerasTuner, Dinamika Terbang Pesawat dan *area navigation*.

3. Bab III Rekonstruksi Data Misi Penerbangan

Pada bab ini akan dibahas secara detil perangkat lunak serta strategi yang digunakan dalam pengendalian lintas terbang berbasis PID *controller* serta bagaimana proses pengumpulan data pembelajaran yang diterapkan.

4. Bab IV Sistem Kendali Lintas Terbang Berbasis *Deep Learning*

Berisikan proses pra pemrosesan data, pengembangan model *deep learning*, penggunaan metode *Bayesian Optimization* dalam melakukan *hyperparameter tuning*, dan integrasi model dengan X-Plane

5. Bab V Hasil Simulasi dan Analisis

Membahas mengenai hasil simulasi model DL yang telah dikembangkan serta analisis mengenai pengaruh data pembelajaran terhadap karakter pengendalian dari kontroler dan bagaimana keunggulan serta kelemahan model berbasis DL dibandingkan dengan pengendalian berbasis PID *controller*.

6. Bab VI Kesimpulan dan Saran

Berisikan jawaban atas rumusan masalah serta saran yang dapat diusulkan agar hasil penelitian mengenai topik ini kedepannya dapat lebih baik lagi.

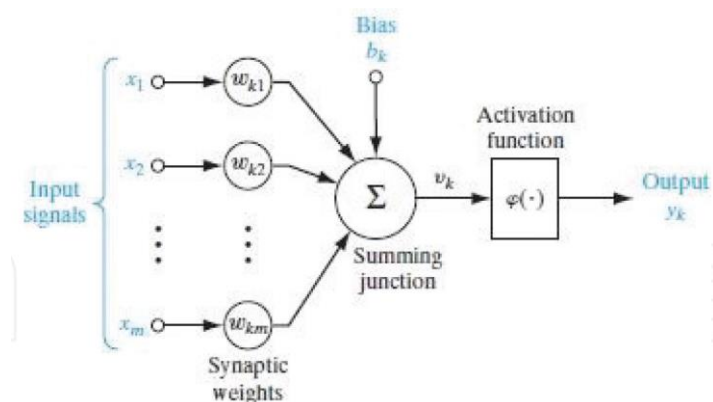
BAB II DASAR TEORI

Pada bab ini, akan dibahas mengenai dasar teori yang menunjang penelitian ini, dimulai dari *deep learning* pada subbab pertama, dilanjutkan dengan *hyperparameter optimization*, TensorFlow dan KerasTuner, dinamika terbang pesawat dan ditutup dengan *area navigation*.

2.1 *Deep Learning*

Deep Learning (DL) merupakan subbidang spesifik dari *machine learning*, dengan representasi pembelajaran dari data yang menekankan pada pembelajaran menggunakan lapisan yang semakin meningkat untuk mencapai representasi yang bermakna. Kata *deep* pada *deep learning* merepresentasikan banyaknya lapisan penyusun model, yang mana pada DL modern, dapat ditemukan 10 hingga ratusan *layer* dalam suatu model [11].

Lapisan pada *deep learning* hampir selalu belajar melalui model *neural network* yang merupakan sebuah teknik komputasi yang menstimulasi bagaimana cara otak manusia bekerja dalam melakukan pemecahan masalah. Arsitektur dari *neural network* dibuat dengan mengimitasi struktur otak manusia yang terdiri dari neuron yang berfungsi untuk melakukan pertukaran dan penampungan informasi dari cabang-cabangnya yang disebut sinapse [12].



Gambar 2.1 Diagram blok *neural network* [13]

Gambar di atas menunjukkan diagram blok dari *neural network*, yang mana sinapse direpresentasikan sebagai garis penghubung berisikan *weight* yang

akan menentukan seberapa kuat sinyal yang diterima, sedangkan neuron akan melakukan penjumlahan dari *input weight* dari sinapse, yang kemudian melalui fungsi aktivasi, jika nilai dari *input* yang diberikan melebihi suatu nilai batas tertentu, maka *output* akan dihasilkan dari *neuron* tersebut. Hal yang ingin dicapai dengan menggunakan DL adalah adanya keseimbangan antara kemampuan mengingat dan kemampuan generalisasi data yang dipelajari [13].

Proses kalibrasi dari nilai *weight* dan *bias* dari struktur *neural network* untuk mencapai fungsi permodelan yang tepat dinamakan proses *training* atau pelatihan. Berbagai algoritme telah dikembangkan untuk melakukan perubahan nilai *weight*, *bias* dan *learning rate* untuk mencapai nilai *error* yang lebih rendah atau dikenal sebagai *optimizers*. Beberapa *optimizers* yang populer digunakan diantaranya adalah *stochastic gradient descent* (SGD), RMSprop, dan Adam.

Metode pembelajaran dalam proses *training* dapat diklasifikasikan menjadi *supervised learning* dan *unsupervised learning*. Hal yang membedakan kedua metode ini ialah pada pembelajaran *supervised*, data akan direpresentasikan dalam pasangan *input* dan *output*. Sedangkan, pada pembelajaran *unsupervised*, data tidak akan memiliki label atau *output* yang diharapkan. Data tersebut kemudian akan diolah melalui algoritme pembelajaran untuk mengadaptasi nilai *weight* dan *bias* yang menghasilkan *loss* sekecil mungkin [12].

Loss adalah suatu besaran untuk mengkuantifikasi seberapa baik atau buruk model dalam melakukan sebuah prediksi dalam satu iterasi pembelajaran. Pada permasalahan kasus regresi, fungsi *loss* yang paling umum digunakan adalah *mean squared error* (MSE) dan *mean absolute error* (MAE). Fungsi *loss* MSE merupakan nilai rata-rata dari kuadrat selisih antara nilai sesungguhnya dengan nilai prediksi dari keseluruhan *dataset*. Sedangkan, fungsi *loss* MAE merupakan nilai rata-rata dari selisih nilai hasil prediksi dengan nilai observasi yang dimutlakkan untuk keseluruhan *dataset* [12]. Kedua fungsi *loss* tersebut dapat dituliskan melalui persamaan berikut, dimana \hat{y} mewakili hasil prediksi dan y mewakili nilai observasi.

$$MSE = \sum_{i=1}^n \frac{1}{n} (y_i - \hat{y}_i)^2 \quad (2.1)$$

$$MAE = \sum_{i=1}^n \frac{1}{n} |y_i - \hat{y}_i| \quad (2.2)$$

2.2 *Hyperparameter Optimization*

Dalam rangka mencapai nilai *loss* yang serendah mungkin, diperlukan arsitektur DL yang tepat dimulai dari jumlah *hidden layer*, berapa banyak neuron pada setiap *layer*, fungsi aktivasi yang digunakan, dan sebagainya yang disebut sebagai *hyperparameter*. Bila hal ini dikerjakan oleh manusia secara manual, tentunya akan memakan banyak waktu dikarenakan kemungkinan *hyperparameter* yang begitu banyak serta proses *training* yang juga membutuhkan waktu. Maka, dikembangkanlah suatu metode untuk melakukan proses optimisasi dari *hyperparameter* secara otomatis dengan mendefinisikan terlebih dahulu arsitektur serta ruang kemungkinan *hyperparameter* yang akan diubah-ubah.

Pada proses optimisasi tersebut, *set* dari *hyperparameter* akan dipilih secara otomatis dan model akan dibangun berdasarkan pilihan tersebut, kemudian model tersebut akan dicocokkan dengan data pembelajaran yang menghasilkan performa dari model. Selanjutnya, algoritme akan memilih *set* berikutnya untuk diuji dan proses ini dilakukan berulang hingga jumlah percobaan yang didefinisikan. Salah satu teknik yang populer digunakan dalam melakukan optimasi *hyperparameter* adalah *Bayesian Optimization* dikarenakan keefisienannya dalam memilih *set hyperparameter* berikutnya dengan memanfaatkan informasi dari percobaan sebelumnya [14].

Bayesian optimization (BO) merupakan sebuah metode yang cocok digunakan dalam permasalahan optimisasi global yang *expensive*, fungsi *black-box* yang telah sukses diaplikasikan pada berbagai macam permasalahan dalam bidang *science*, *engineering*, dan juga *hyperparameter tuning* pada model *machine learning*. BO bekerja secara iteratif dengan menentukan data poin selanjutnya menuju label untuk memaksimalkan *sampling efficiency* dan mengurangi jumlah data poin yang dibutuhkan [14].

Pada permasalahan *hyperparameter tuning* model *machine learning*, objek yang berlaku sebagai fungsi objektif adalah setiap lapisan pada *neural networks*, dimana dengan memberikan masukan berupa parameter-parameter, fungsi ini akan menghasilkan nilai berupa *loss* untuk menunjukkan tingkat performa dari parameter tersebut. Untuk memaksimalkan performa dari fungsi tersebut, digunakanlah *bayesian statistics*. Algoritme *bayesian* akan membantu mencari *hyperparameter* dari parameter *space* yang telah didefinisikan untuk mengevaluasi fungsi objektif menggunakan *gaussian processes*, dimana model *gaussian* berguna untuk memahami struktur dari fungsi objektif. Maka, melalui metode ini, *set hyperparameter* yang akan dipilih selanjutnya akan didasari informasi penggunaan *set hyperparameter* sebelumnya [14].

2.3 TensorFlow dan KerasTuner

TensorFlow adalah sebuah *end-to-end opensource platform* untuk *machine learning* yang komprehensif, fleksibel dan didukung dengan *library* serta komunitas yang memadai. Dengan menggunakan TensorFlow, model *machine learning* dapat dibuat dan dilatih dengan mudah menggunakan API seperti Keras. Selain itu, proses *deploy* dari model dapat dilakukan dimana saja menggunakan bahasa pemrograman apapun [15].

Dalam rangka melakukan optimisasi dari model *machine learning*, Keras mengembangkan suatu *framework* bernama “KerasTuner”, yang berfungsi untuk melakukan pencarian *hyperparameter* terbaik dari model Keras yang telah didefinisikan secara otomatis menggunakan beberapa algoritme pencarian yang disediakan meliputi *Bayesian Optimization*, *Hyperband*, dan *Random Search* [16].

Dengan memanfaatkan TensorFlow dan KerasTuner, akan dikembangkan suatu model *deep learning* yang bertujuan sebagai pemberi perintah kendali pada pesawat untuk bermanuver mengikuti *waypoint* yang telah didefinisikan. Sehingga pada subbab selanjutnya, akan dibahas mengenai dinamika terbang pesawat.

2.4 Dinamika Terbang Pesawat

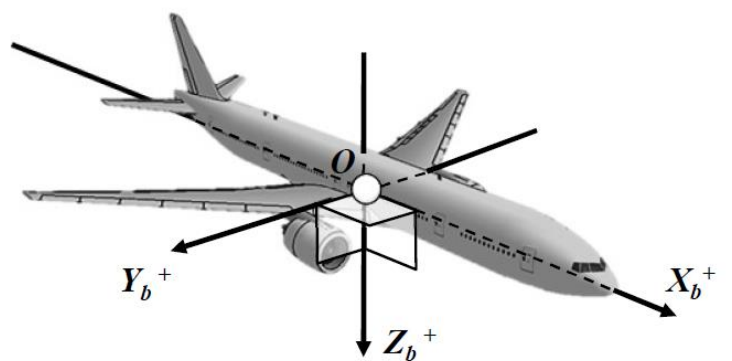
Subbab ini membahas mengenai hal-hal yang berkaitan dengan dinamika terbang pesawat dimulai dari tata acuan koordinat, gerak pesawat dalam ruang 3 dimensi, matra gerak pesawat dan perangkat kendali terbang.

2.4.1 Tata Acuan Koordinat

Tata acuan koordinat (TAK) adalah sebuah sistem koordinat ortogonal abstrak (kartesian) yang secara unik mengorientasikan vektor-vektor terhadap titik asalnya. TAK digunakan pada permasalahan dinamika terbang untuk menentukan orientasi sikap, gerak, dan posisi pesawat pada suatu ruang tertentu. Tata acuan koordinat pokok pada dinamika terbang terdiri atas TAK benda, TAK angin, dan TAK horison lokal [17].

2.4.1.1 Tata Acuan Koordinat Benda

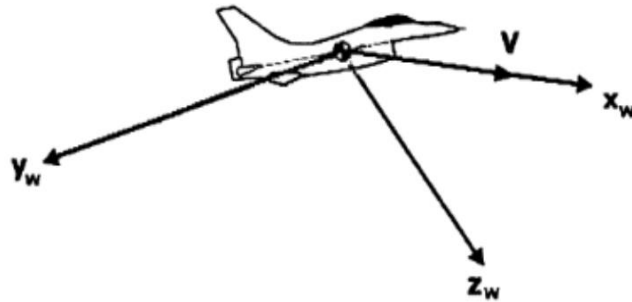
Tata acuan koordinat benda didefinisikan untuk mewakili geometri fisik dari pesawat. Sumbu ini berpusat pada *center of gravity* pesawat dan berubah bergantung pergerakan dan orientasi dari pesawat. *Thrust*, pilot dan juga sensor-sensor pada pesawat terletak pada sumbu ini. Sumbu X pada TAK benda berhimpit dengan sumbu simetris pesawat dengan nilai positif mengarah kedepan hidung pesawat, sumbu Y_b terletak tegak lurus dengan bidang simetrik pesawat dengan nilai positifnya adalah dari sisi kanan pilot dan sumbu Z_b akan mengikuti kaidah tangan kanan dari arah sumbu X_b dan sumbu Y_b yaitu mengarah ke bawah [17].



Gambar 2.2 Tata acuan koordinat benda [17]

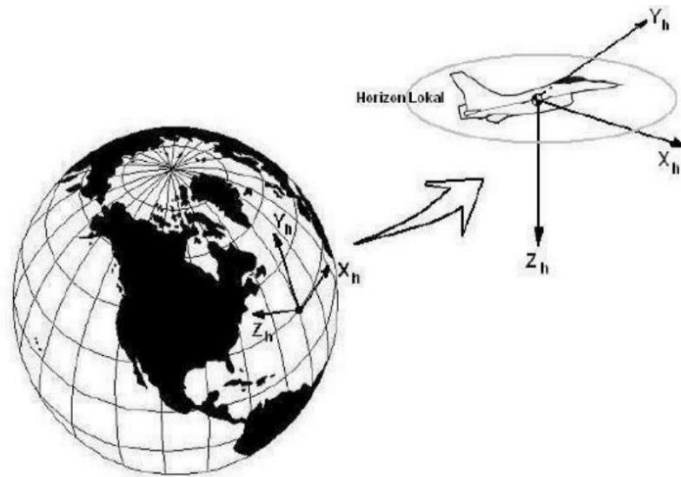
2.4.1.2 Tata Acuan Koordinat Angin

Tata acuan koordinat angin didefinisikan untuk mewakili vektor *airspeed*, penunjuk arah dari lintasan terbang dan gaya aerodinamika yang bekerja pada pesawat. TAK ini berpusat pada *center of gravity* pesawat dan independen terhadap orientasi pesawat. Sumbu X_w positif berimpit dengan vektor kecepatan udara, Sumbu Z_w positif berada pada sumbu simetris pesawat yang tegak lurus dengan sumbu X_w dengan arah kebawah, dan sumbu Y_w bernilai positif pada sisi kanan dari pilot tegak lurus sesuai dengan kaidah tangan kanan dari arah sumbu X_w dan Z_w [17].



Gambar 2.3 Tata acuan koordinat angin [17]

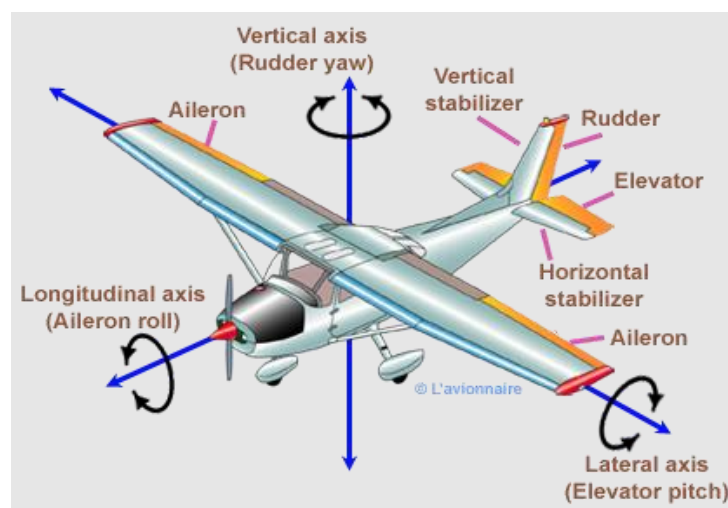
Tata acuan koordinat ini digunakan untuk merepresentasikan permukaan dari bumi, jika diasumsikan pesawat terbang pada ketinggian yang cukup rendah dibandingkan jari-jari bumi, sehingga kelengkungan permukaan bumi dapat diabaikan. TAK ini berpusat pada *center of gravity* dan independen terhadap orientasi pesawat serta arah angin. Pada umumnya, terdapat dua jenis konvensi dalam penentuan arah dari sumbu TAK ini, yaitu NED (*North-East-Down*) dan ENU (*East-North-Up*) [17]. Pada tugas akhir ini, representasi posisi pesawat dari TAK benda akan ditransformasikan menjadi TAK horison lokal berkonvensikan ENU, dengan nilai X_h positif mengarah ke timur, Y_h positif ke arah utara, dan Z_h positif ke arah atas. Gambar di bawah ini menunjukkan ilustrasi dari TAK horison lokal dengan konvensi NED.



Gambar 2.4 Tata acuan koordinat horison lokal (NED) [17]

2.4.2 Gerak Pesawat dalam Ruang 3 Dimensi

Gerak pesawat dalam ruang 3 dimensi memiliki enam derajat kebebasan yang terdiri atas 3 buah gerakan translasi dan 3 buah gerakan rotasional, sumbu rotasi dimana pesawat melakukan gerakan tersebut diilustrasikan melalui Gambar 2.1. Gerakan rotasi pada sumbu longitudinal, lateral, dan vertikal pada pesawat secara berurut dinamakan gerakan *roll*, *pitch*, dan *yaw*. Sedangkan, gerakan translasi pada sumbu longitudinal dinamakan *surge*, pada sumbu lateral dinamakan *sway* dan pada sumbu vertikal dinamakan *heave*.



Gambar 2.5 Sumbu rotasi pesawat [18]

Seluruh pergerakan pada pesawat tersebut tidak dapat terlepas dari gaya dan momen yang bekerja pada pesawat, dengan 4 aspek utama yang berkontribusi

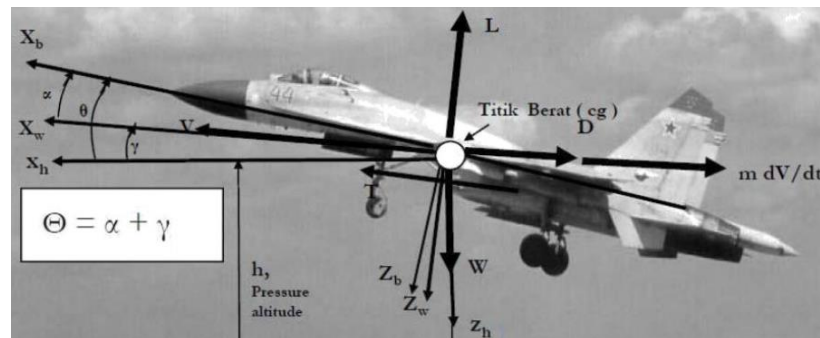
yaitu aerodinamika, propulsi, gravitasi, dan bidang kendali. Pergerakan yang dihasilkan dari pesawat juga akan mempengaruhi gaya dan momen yang bekerja pada pesawat. Dalam rangka untuk melakukan pengendalian dari gerak pesawat, digunakanlah bidang kendali yang diletakkan pada lokasi yang berbeda pada pesawat dan juga pengatur *throttle* dari mesin untuk memanipulasi gaya dan momen yang bekerja pada pesawat [17].

2.4.3 Matra Gerak Pesawat

Enam buah gerakan pesawat di udara umumnya dapat dikategorikan menjadi dua buah matra gerak yaitu matra gerak longitudinal dan matra gerak lateral-direksional.

2.4.3.1 Matra Gerak Longitudinal

Matra gerak longitudinal membagi pesawat pada bidang yang dibentuk dari sumbu longitudinal dengan sumbu vertikal. Matra gerak ini mendefinisikan gerak dan sikap pesawat pada bidang simetri yang terdiri atas 2 gerakan translasi ke arah vertikal bernama *heave* dan ke arah horizontal bernama *surge*, dengan gerakan rotasinya yaitu *pitch*. Gambar berikut ini memberikan ilustrasi dari matra gerak longitudinal pada pesawat beserta variabel sikapnya.



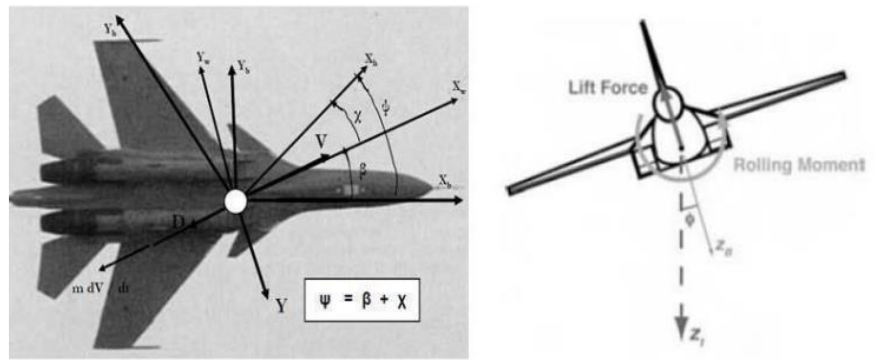
Gambar 2.6 Variabel sikap pada matra gerak longitudinal [19]

Pada gambar diatas, dapat terlihat berbagai variabel sikap pada matra gerak longitudinal. Sudut *pitch* yang diwakili dengan simbol θ menunjukkan orientasi dari badan pesawat (TAK benda) relatif terhadap bumi (TAK horison lokal), variabel yang mendefinisikan sudut serang (α) mewakili orientasi dari badan pesawat (TAK benda) relatif terhadap arah terbangnya (TAK angin), dan variabel terakhir adalah sudut arah terbang atau *climb* (γ) yang mengindikasikan arah

terbang pesawat pada bidang vertikal yang dibentuk dari TAK angin relatif terhadap TAK horison lokal. Penjumlahan dari sudut serang dengan sudut arah terbang menghasilkan sudut *pitch*.

2.4.3.2 Matra Gerak Lateral-Direksional

Matra gerak lateral-direksional digunakan untuk mengamati sikap dan gerak pesawat pada bidang lateral yaitu pada tampak depan dan bidang direksional yaitu ketika pesawat dilihat dari sisi atas, dimana pada konfigurasi pesawat konvensional, kedua bidang ini biasanya akan berpasangan. Matra gerak ini terdiri atas 2 gerakan rotasi yaitu *roll* dan *yaw* serta gerakan translasi berupa *sway*. Gambar dibawah ini menunjukkan ilustrasi beserta variabel sikap dari matra gerak lateral-direksional.



Gambar 2.7 Variabel sikap pada matra gerak lateral-direksional [19]

Gambar pada sisi kiri Gambar 2.6 menunjukkan matra gerak direksional yang terdiri dari 3 buah variabel sikap. Variabel sudut *heading* (ψ) yang menunjukkan orientasi sudut dari badan pesawat (TAK benda) relatif terhadap bidang horizontal (TAK horison lokal), selanjutnya terdapat sudut *side slip* yang diwakili dengan simbol β yang mewakili orientasi sudut dari TAK benda relatif terhadap arah terbang pesawat di bidang horizontal (TAK angin), dan sudut χ adalah sudut arah terbang pada bidang horizontal yang merupakan sudut relatif antara TAK angin dan TAK horison lokal. Sedangkan pada matra gerak lateral, terdapat variabel sikap sudut *roll* (ϕ) yang menunjukkan orientasi badan pesawat (TAK benda) terhadap permukaan bumi. Sudut ini juga dapat didefinisikan sebagai sudut *bank* (μ) jika orientasi yang dilihat adalah pada TAK angin pada bidang lateral [19].

2.4.4 Perangkat Kendali Terbang

Pesawat konvensional memiliki 3 buah bidang kendali utama yaitu *aileron*, *elevator* dan *rudder* dengan posisi penempatannya dapat dilihat melalui Gambar 2.5. Setiap bidang kendali ini akan terhubung dengan bidang kendali pada *cockpit*. Sehingga, perintah kendali yang diberikan *pilot* dapat ditransmisikan melalui mekanisme kontrol untuk menggerakkan bidang kendali yang diharapkan. Mekanisme kontrol yang dimiliki setiap pesawat tentunya berbeda-beda, diantaranya yaitu sistem mekanikal, sistem hidrolis dengan bantuan sistem mekanikal, sistem *electro-hydro mechanical*, dan mekanisme yang paling banyak digunakan pada pesawat terbang modern yaitu *fly by wire* dan *fly by light* [20].

2.4.4.1 Aileron

Aileron adalah bidang kendali yang memberikan kontrol gerakan *roll* terhadap bidang *longitudinal*, yang terletak pada bagian luar dari *trailing edge* pada setiap sisi sayap, yang berdefleksi secara berlawanan arah satu sama lain. *Aileron* akan menerima masukan dari bidang kendali pada *cockpit* yang dapat berupa *joystick* atau *control wheel* melalui mekanisme

Ketika *pilot* menggerakkan *control wheel* atau *joystick* ke arah kanan, maka *aileron* kanan akan terdefleksi ke arah atas, sedangkan *aileron* kiri akan terdefleksi ke arah bawah. Dengan defleksi *aileron* kiri yang mengarah ke bawah, *camber* dari *airfoil* akan bertambah, sehingga menyebabkan gaya angkat pada sisi kiri pesawat akan meningkat, sedangkan dikarenakan defleksi *aileron* kanan mengurangi *camber*, maka gaya angkat yang dihasilkan pada sisi kanan sayap akan berkurang. Resultan perubahan gaya akibat *pilot* menggerakkan *control wheel* ke arah kanan akan menyebabkan pesawat melakukan gerakan *roll* ke arah kanan dari sudut pandang *pilot*.

Dikarenakan defleksi *aileron* ke arah bawah akan menghasilkan gaya angkat yang lebih besar, tentunya *induced drag* yang dihasilkan akan lebih besar dibandingkan dengan *aileron* yang terdefleksi ke arah atas. Maka daripada itu, sisi sayap yang memiliki gaya hambat lebih besar akan bergerak lebih lambat dibanding sisi sayap satunya, yang mengakibatkan sisi sayap yang memiliki gaya angkat dan hambat yang lebih besar akan mengalami gerakan *yaw* yang berlawanan dari arah

berbelok pesawat. Sebagai contoh ketika pesawat akan melakukan *roll* ke arah kanan, akan terjadi gerakan *yaw* ke arah kiri dari sudut pandang pilot [1] [21].

2.4.4.2 Elevator

Elevator adalah bidang kendali primer yang terletak pada *horizontal stabilizer* dengan fungsi untuk mengontrol sikap *pitch* dari pesawat atau rotasi terhadap sumbu lateral, serta berperan membantu *horizontal stabilizer* untuk mempertahankan kestabilan longitudinal [21]. *Elevator* akan menerima masukan dari *control column* pada *cockpit* yang dapat digerakan maju dan mundur.

Ketika pilot mendorong *control column* ke arah dalam, *elevator* akan terdefleksi ke arah bawah, yang mana akan menyebabkan bertambahnya *camber* dari *elevator* dan berimbas pada meningkatnya gaya angkat pada *horizontal stabilizer*. Dikarenakan posisi dari *horizontal stabilizer* selalu berada di belakang *center of gravity (c.g)* dari pesawat, maka penambahan *camber* tersebut akan menimbulkan momen yang membuat hidung dari pesawat akan menghadap ke bawah atau disebut juga dengan *pitch down* [1] [21].

2.4.4.3 Rudder

Rudder berfungsi untuk mengontrol pergerakan pesawat terhadap sumbu vertikal yakni gerakan *yaw*, yang mana posisinya diletakkan pada *vertical stabilizer*. *Rudder* dapat didefleksikan oleh pilot dengan cara menginjak *rudder pedals* yang terpisah menjadi pedal kiri dan pedal kanan.

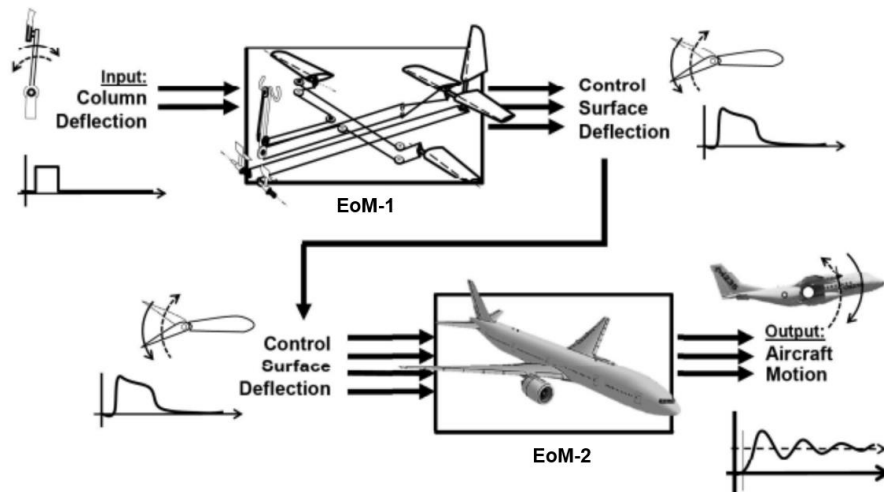
Ketika pedal kiri ditekan dengan kaki, *rudder* akan terdefleksi ke arah kiri, yang menyebabkan adanya gaya horizontal yang mendorong *vertical stabilizer* ke arah kanan, dan menghasilkan momen *yaw* yang akan menggerakkan hidung pesawat ke arah kiri dari sudut pandang pilot [1] [21].

2.4.4.4 Kendali Terbang

Dalam melakukan pengendalian pesawat terbang, terdapat dua metode kendali yang dapat ditemui, yakni pengendalian lingkaran terbuka dan lingkaran tertutup. Pada sistem kendali lingkaran terbuka, aksi pengendalian yang dihasilkan akan independen terhadap *output* yang dihasilkan dikarenakan proses yang searah dari *input* menuju model sistem. Pada sistem kendali lingkaran tertutup, konsep

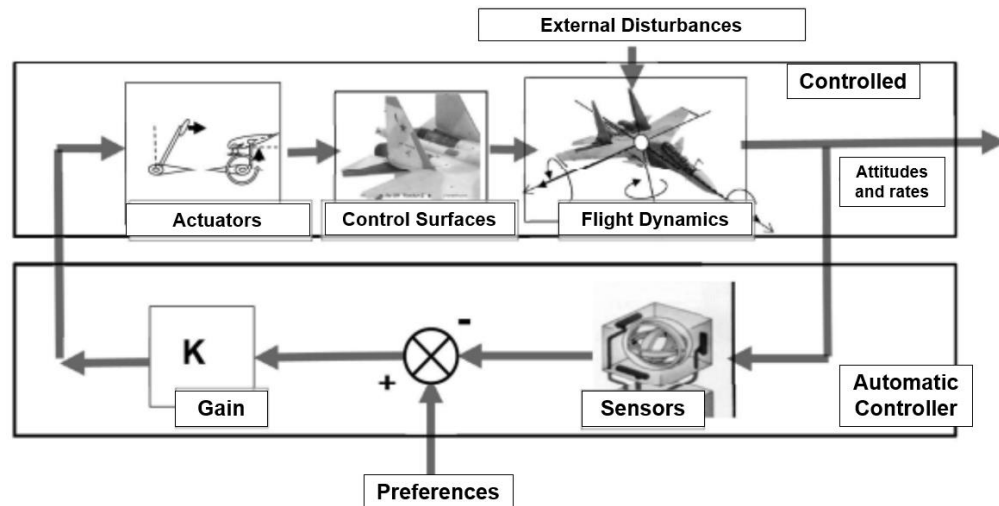
pengendalian yang digunakan serupa dengan sistem kendali lingkaran terbuka, namun terdapat tambahan *feedback loops* yang akan mengumpanbalikkan sinyal keluaran untuk dibandingkan terhadap sinyal masukan.

Sistem kendali lingkaran terbuka pada pesawat terbang dapat ditemukan pada sistem kendali aktuator bidang kendali pesawat. Pada permodelan sistem kendali ini, sinyal masukan yang diterima dari pergerakan *control column* akan diproses melalui model mekanisme kontrol, menghasilkan defleksi dari bidang kendali yang diinginkan. Defleksi dari bidang kendali ini kemudian akan diproses pada model dinamik pesawat atau persamaan gerak pesawat yang kemudian mempengaruhi gerak atau sikap dari pesawat terbang. Gambar berikut mengilustrasikan proses permodelan kendali lingkaran terbuka yang dimaksud.



Gambar 2.8 Pemodelan sistem kendali lingkaran terbuka pada pesawat [17]

Pengimplementasian sistem kendali lintas tertutup pada pesawat dapat ditemukan pada proses pengendalian pesawat oleh pilot ataupun pada kontroler otomatis. Gambar di bawah ini menunjukkan pemodelan sistem kendali lingkaran tertutup dalam penerapan kontroler otomatis.



Gambar 2.9 Pemodelan sistem kendali kontrol otomatis pada pesawat [17]

Melalui sistem kendali kontroler otomatis atau *automatic flight control system* (AFCS) yang ditampilkan pada gambar di atas, dapat dilihat adanya *feedback loops* yang mengumpanbalikkan sikap serta perubahan sikap pesawat pada sensor yang kemudian dibandingkan pada nilai preferensi. Perbedaan nilai ini selanjutnya dikalikan dengan *gain* tertentu yang kemudian diteruskan pada aktuator yang selanjutnya menggerakkan bidang kendali pesawat yang menyebabkan perubahan sikap dari pesawat.

Dengan dikembangkannya AFCS, beban fisik serta psikologis dari pilot dapat dikurangi. Terdapat 4 buah kategori AFCS, yaitu sebagai *stability augmentation system* dengan tujuan meningkatkan stabilitas pesawat, *trimmed flight holding system* atau *autopilot* yang bertujuan untuk mempertahankan suatu kondisi terbang ataupun sistem panduan terbang, *command augmentation system* untuk mengoptimasi perintah kontrol dari pilot, dan *stability generator system* yang menciptakan kestabilan buatan pada pesawat yang tidak stabil [20]

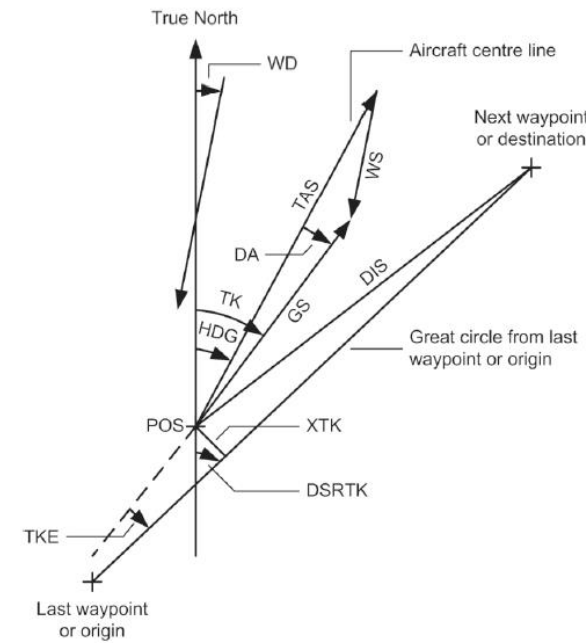
2.5 *Area Navigation*

Area navigation atau yang sering disebut RNAV adalah suatu metode navigasi yang memungkinkan pesawat beroperasi pada lintasan terbang apapun yang dikehendaki dalam jangkauan alat bantu navigasi baik berbasis darat maupun udara atau dalam batasan kapabilitas dari alat bantu mandiri atau kombinasi dari alat bantu tersebut. Penerapan RNAV dalam sistem navigasi pesawat memiliki kelebihan yaitu penggunaan waktu serta bahan bakar yang lebih hemat, mengurangi ketergantungan pada radar, ketinggian dan kecepatan terbang sehingga dapat mengurangi transmisi radio yang dibutuhkan serta memungkinkan penggunaan ruang udara yang lebih efisien [2].

Dalam melaksanakan navigasi berbasis metode ini, pilot perlu memiliki pemahaman dari berbagai tipe dari *waypoint* dan tipe *leg* yang digunakan pada prosedur RNAV. *Waypoint* adalah posisi geografis yang telah ditentukan yang diwakili dalam koordinat lintang dan bujur. Penggunaan RNAV dilakukan menggunakan komputer navigasi, dimana koordinat dari *waypoints* akan dimasukkan kedalam komputer baik secara manual oleh pilot atau secara otomatis menggunakan *database* terintegrasi untuk membentuk sebuah rencana penerbangan. Selanjutnya, komputer akan mengestimasi posisi dari pesawat menggunakan sensor navigasi pada pesawat untuk dibandingkan dengan posisi *waypoint* yang akan dituju, perbedaan posisi tersebutlah yang akan membentuk informasi panduan terbang [22].

Dalam pelaksanaan prosedur RNAV pada komputer navigasi, pilot dapat memilih diantara kedua metode yang disediakan, yakni *fly-over* dan *fly-by waypoints*. Metode *fly-by waypoints* adalah metode yang digunakan ketika pesawat harus mulai berbelok ke jalur selanjutnya sebelum mencapai *waypoint* yang memisahkan dua segmen rute atau disebut sebagai *turn anticipation*. Sedangkan metode *fly-over waypoints* atau lintas terbang digunakan ketika pesawat harus melintasi setiap *waypoint* yang didefinisikan [2]. Ilustrasi perbedaan dari kedua metode ini dapat dilihat pada Gambar 1.1 di atas.

Berikut ini ditunjukkan terminologi umum yang digunakan pada navigasi pesawat.



Gambar 2.10 Terminologi navigasi [23]

Dengan keterangan dari setiap istilah yang ada pada gambar dijelaskan melalui tabel berikut.

Tabel 2.1 Deskripsi terminologi navigasi [23]

Istilah	Singkatan	Deksripsi
<i>Cross track distance</i>	XTK	Jarak terdekat antara POS dengan lintasan yang dikehendaki
<i>Desired track angle</i>	DSRTK	Sudut antara <i>true north</i> dengan lintasan yang dikehendaki
<i>Distance</i>	DIS	<i>Great circle distance</i> terhadap <i>waypoint</i> berikutnya
<i>Drift angle</i>	DA	Sudut antara <i>heading</i> pesawat dengan <i>ground track</i>
<i>Ground track angle</i>	TK	Sudut antara <i>true north</i> dengan lintasan terbang pesawat
<i>Heading</i>	HDG	Sudut horizontal antara sumbu longitudinal pesawat dengan koordinat referensi
<i>Present position</i>	POS	Koordinat lintang dan bujur dari posisi pesawat
<i>Track angle error</i>	TKE	Sudut antara lintasan saat ini dengan lintasan yang dikehendaki
<i>Wind direction</i>	WD	Sudut antara <i>true north</i> dengan vektor arah angin
<i>True airspeed</i>	TAS	Kecepatan pesawat relatif terhadap udara
<i>Wind speed</i>	WS	Kecepatan udara
<i>Ground speed</i>	GS	Kecepatan pesawat relatif terhadap permukaan bumi

Teori dasar yang menunjang penelitian ini telah dipaparkan melalui bab ini. Pada bab selanjutnya, akan dibahas mengenai proses rekonstruksi data misi penerbangan menggunakan simulator sistem kendali dan lintasan terbang yang telah dilakukan sebelumnya pada pekerjaan [3] serta proses pengumpulan data pembelajaran yang dilakukan.

BAB III

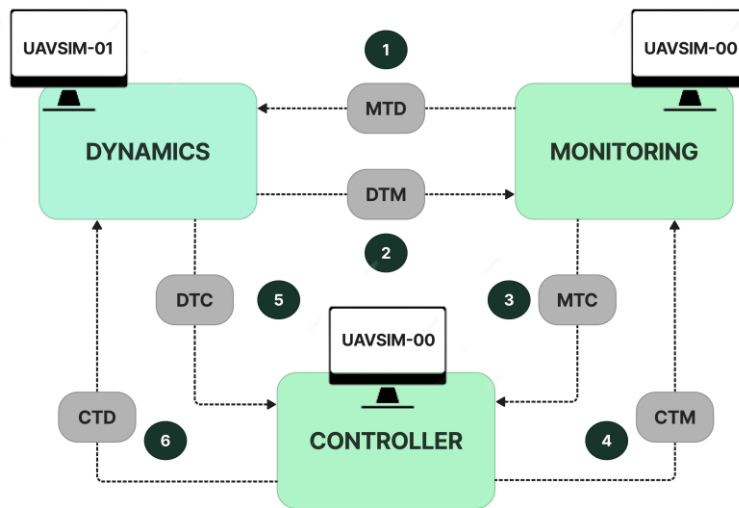
REKONSTRUKSI DATA MISI PENERBANGAN

Pada bab ini, akan dibahas mengenai simulator sistem kendali dan lintasan terbang yang telah dikembangkan sebelumnya pada pekerjaan [3] secara detil dari setiap subsistem penyusunnya serta variabel data yang dikirim dan diterima. Pada subbab selanjutnya, dijelaskan bagaimana tahapan proses pengumpulan data pembelajaran dilakukan dimulai dari persiapan perangkat lunak MATLAB, X-Plane 11, yang ditutup dengan penjelasan langkah dalam melaksanakan rekonstruksi data misi penerbangan dan variabel data yang disimpan sebagai basis pembelajaran model *deep learning*.

3.1 Simulator Sistem Kendali dan Lintasan Terbang

Simulator sistem kendali dan lintasan terbang pekerjaan [3] yang bernamakan “UAVSIM-A” terdiri dari tiga buah subsistem yaitu subsistem *dynamics*, *monitoring* dan subsistem *controller*. Ketiga subsistem tersebut saling berkomunikasi satu dengan yang lainnya dengan cara mengirim atau menerima data. Perangkat lunak yang digunakan pada subsistem *monitoring* dan *controller* adalah MATLAB dan Simulink. Sedangkan, untuk subsistem dinamika, digunakan perangkat lunak program simulator terbang X-Plane. Sedangkan, antar-muka antara subsistem atau yang dikenal dengan istilah *plugin* dikembangkan menggunakan bahasa pemrograman Python. Dalam pengintegrasian pada perangkat keras, akan digunakan dua unit komputer, yaitu komputer yang dinamakan UAVSIM-00 yang akan digunakan untuk menjalankan subsistem kontroler dan subsistem monitoring, dan satu unit komputer lagi yang dinamakan UAVSIM-01 untuk menjalankan perangkat lunak simulator penerbangan sebagai subsistem dinamika.

Proses pertukaran data antara subsistem pada UAVSIM-A dapat dibagi menjadi 6 buah aliran data, yang direpresentasikan melalui Gambar 3.1.



Gambar 3.1 Diagram komunikasi antar subsistem UAVSIM-A

Dengan penjelasan masing-masing aliran data yang telah dinomori pada Gambar 3.1 tersebut dijabarkan sebagai berikut.

1. Aliran Data MTD (*Monitoring to Dynamics*)

Aliran ini terjadi antara subsistem *monitoring* dengan subsistem dinamika, dengan data yang dikirim pada aliran ini disajikan pada tabel berikut ini.

Tabel 3.1 Aliran data *monitoring to dynamics*

Pengirim/Penerima		UAVSIM-00/UAVSIM-01
Laju transfer		50 Hz
No.	Data	Keterangan
01.	<i>Wind speed</i>	Kecepatan angin
02.	<i>Wind direction</i>	Arah angin diukur 0° dari utara (CW)
03.	<i>Pause simulation</i>	Data untuk memberhentikan simulasi
04.	<i>View</i>	Data untuk mengubah tampilan pada subsistem dinamika

2. Aliran Data DTM (*Dynamics to Monitoring*)

Data yang dialirkan dari subsistem dinamika menuju subsistem *monitoring* ditampilkan melalui Tabel 3.2.

Tabel 3.2 Aliran data *dynamics to monitoring*

Pengirim/Penerima		UAVSIM-01/UAVSIM-00
Laju transfer		50 Hz
No.	Data	Keterangan
01.	<i>Latitude</i>	Posisi latitude pesawat dalam koordinat WSG84
02.	<i>Longitude</i>	Posisi longitude pesawat dalam koordinat WSG84
03.	<i>Altitude</i>	Posisi altitude pesawat dalam koordinat WSG84
04.	X	Latitude yang dikonversi menjadi TAK horizon lokal (ENU)
05.	Y	Longitude yang dikonversi menjadi TAK horizon lokal (ENU)
06.	Z	<i>Altitude</i> yang dikonversi menjadi TAK horizon lokal (ENU)
07.	<i>Yaw angle</i>	Sudut yaw pesawat
08.	<i>Pitch angle</i>	Sudut <i>pitch</i> pesawat
09.	<i>Roll angle</i>	Sudut <i>roll</i> pesawat
10.	<i>True Airspeed</i>	Kecepatan pesawat relatif terhadap udara
11.	<i>Yoke heading ratio</i>	Rasio pergerakan <i>joystick</i> arah horizontal (-1 hingga 1)
12.	<i>Yoke pitch ratio</i>	Rasio pergerakan <i>joystick</i> arah vertikal (-1 hingga 1)
13.	<i>Yoke roll ratio</i>	Rasio pergerakan <i>joystick</i> arah horizontal (-1 hingga 1)
14.	<i>Throttle</i>	Masukan <i>throttle</i> menuju <i>engine</i>
15.	<i>Thrust</i>	<i>Thrust</i> yang dihasilkan <i>engine</i>
16.	<i>Fuel flow</i>	Laju aliran <i>fuel</i>
17.	<i>Rudder</i>	Sudut defleksi <i>rudder</i>
18.	<i>Elevator</i>	Sudut defleksi <i>elevator</i>
19.	<i>Left aileron</i>	Sudut defleksi <i>aileron</i> kiri
20.	<i>Right aileron</i>	Sudut defleksi <i>aileron</i> kanan
21.	<i>Ground speed</i>	Kecepatan pesawat relatif terhadap <i>ground</i>
22.	<i>Wind speed</i>	Kecepatan angin
23.	<i>Roll rate</i>	Laju perubahan sudut <i>roll</i>

3. Aliran Data MTC (*Monitoring to Controller*)

Aliran data MTC yaitu berupa pemberian data misi *waypoint* yang telah didefinisikan serta kondisi awal yang didapat dan disimpan pada subsistem *monitoring*, dengan penjabaran lebih detil ditampilkan pada tabel berikut.

Tabel 3.3 Aliran data *monitoring to controller*

Pengirim/Penerima		UAVSIM-00/UAVSIM-00
Laju transfer		-
No.	Data	Keterangan
01.	<i>AC initial position</i>	Posisi awal pesawat
02.	Seluruh koordinat	Seluruh koordinat <i>waypoint</i> dalam arah X
	<i>waypoint</i> dalam arah X	yang didefinisikan pada tampilan <i>mission planner</i>
03.	Seluruh koordinat	Seluruh koordinat <i>waypoint</i> dalam arah Y
	<i>waypoint</i> dalam arah Y	yang didefinisikan pada tampilan <i>mission planner</i>
04.	Seluruh koordinat	Seluruh koordinat <i>waypoint</i> dalam arah Z
	<i>waypoint</i> dalam arah Z	yang didefinisikan pada tampilan <i>mission planner</i>

4. Aliran Data CTM (*Controller to Monitoring*)

Aliran data ini berupa data koordinat *waypoint* yang akan dituju beserta status dari sistem *autopilot* yang dikirimkan oleh subsistem kontroler menuju subsistem *monitoring* agar keberhasilan sistem *autopilot* dapat dipantau. Secara lebih detil mengenai data yang dikirimkan pada aliran data CTM disajikan pada Tabel 3.4.

Tabel 3.4 Aliran data *controller to monitoring*

Pengirim/Penerima		UAVSIM-00/UAVSIM-00
Laju transfer		50 Hz
No.	Data	Keterangan
01.	<i>Next waypoint ID</i>	Menunjukkan pesawat sedang mengarah menuju <i>waypoint</i> ke berapa
02.	<i>Next waypoint X position</i>	Posisi koordinat X dari <i>waypoint</i> yang akan dituju
03.	<i>Next waypoint Y position</i>	Posisi koordinat Y dari <i>waypoint</i> yang akan dituju
04.	<i>Next waypoint Z position</i>	Posisi koordinat Z dari <i>waypoint</i> yang akan dituju
05.	<i>Distance to waypoint</i>	Jarak pesawat terhadap <i>waypoint</i> yang dituju
06.	<i>Heading reference</i>	Sudut <i>heading</i> referensi yang harus dicapai pesawat
07.	<i>WP following status</i>	Status <i>autopilot</i> <i>WP following</i> (1 atau 0)
08.	<i>Pitch Hold status</i>	Status <i>autopilot</i> <i>pitch hold</i> (1 atau 0)
09.	<i>Wing Leveler Status</i>	Status <i>autopilot</i> <i>wing leveler</i> (1 atau 0)

5. Aliran Data DTC (*Dynamics to Controller*)

Aliran data ini mengirimkan data kondisi kini dari pesawat agar dapat dibandingkan kondisi kini dengan kondisi data misi yang didefinisikan sehingga subsistem kontroler dapat memberikan *input* manipulasi bidang kendali kepada subsitem dinamika agar misi terbang dapat tercapai. Tabel 3.5 menunjukkan data yang dikirim dari subsitem dinamika menuju subsitem kontroler.

Tabel 3.5 Aliran data *dynamics to controller*

Pengirim/Penerima		UAVSIM-01/UAVSIM-00
Laju transfer		50 Hz
No.	Data	Keterangan
01.	X	<i>Longitude</i> yang dikonversi menjadi TAK horizon lokal (ENU)
02.	Y	<i>Latitude</i> yang dikonversi menjadi TAK horizon lokal (ENU)
03.	Z	<i>Altitude</i> yang dikonversi menjadi TAK horizon lokal (ENU)
04.	<i>Yaw angle</i>	Sudut yaw pesawat
05.	<i>Pitch angle</i>	Sudut <i>pitch</i> pesawat
06.	<i>Roll angle</i>	Sudut <i>roll</i> pesawat
07.	<i>True Airspeed</i>	Kecepatan pesawat relatif terhadap udara
08.	<i>Yoke heading ratio</i>	Rasio pergerakan <i>joystick</i> arah horizontal (-1 hingga 1)
09.	<i>Yoke pitch ratio</i>	Rasio pergerakan <i>joystick</i> arah vertikal (-1 hingga 1)
10.	<i>Yoke roll ratio</i>	Rasio pergerakan <i>joystick</i> arah horizontal (-1 hingga 1)
11.	<i>Throttle</i>	Masukan <i>throttle</i> menuju <i>engine</i>

6. Aliran Data CTD (*Controller to Dynamics*)

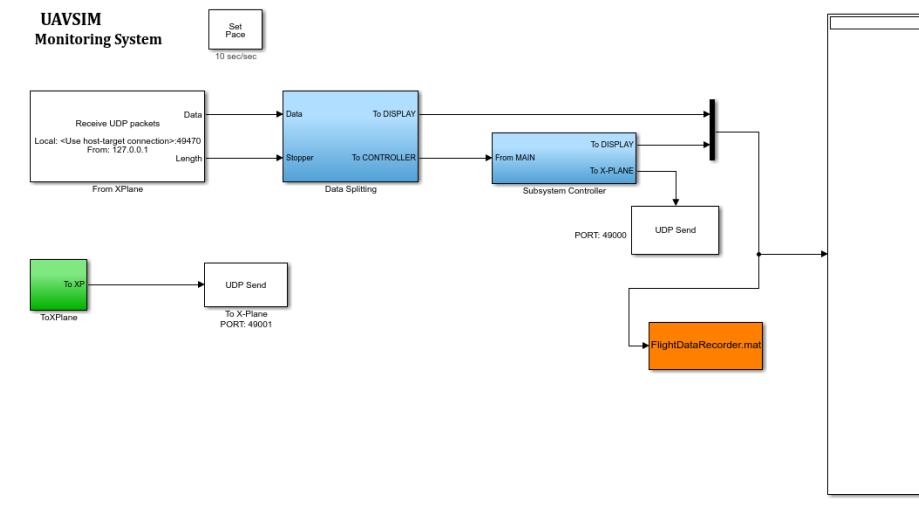
Data yang dikirim dari subsistem kontroler menuju subsitem dinamika berupa hasil koreksi perhitungan setiap subsistem yang menyusun subsistem kontroler, serta status dari *controller* yang digunakan apakah menyala atau tidak yang diwakili dengan nilai 1

untuk menyala dan 0 untuk tidak menyala. Tabel berikut ini menunjukkan data yang dikirim pada aliran data CTD.

Tabel 3.6 Aliran data *controller to dynamics*

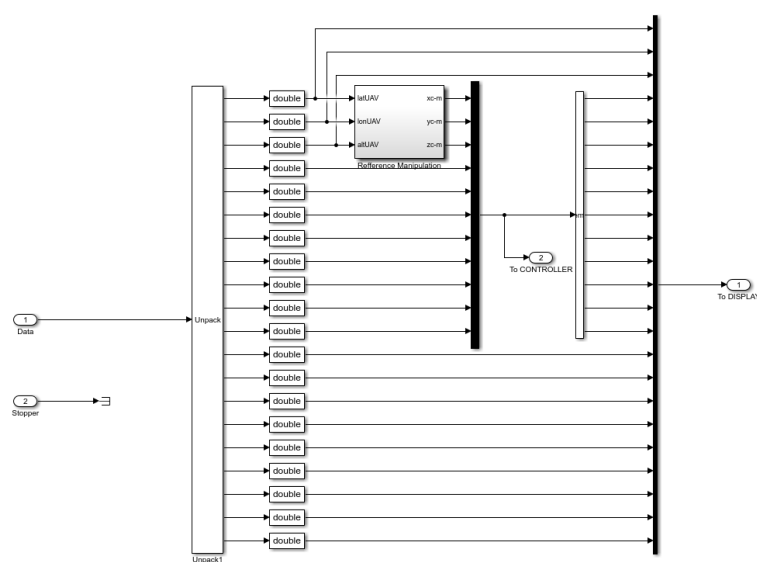
Pengirim/Penerima		UAVSIM-00/UAVSIM-01
Laju transfer		50 Hz
No.	Data	Keterangan
01.	<i>Data input joystick (yoke roll ratio)</i>	Keluaran dari subsistem <i>waypoint controller</i> sebagai perintah mengendalikan <i>joystick</i> arah horizontal
02.	<i>WP Following Status</i>	Status <i>autopilot WP following</i> (1 atau 0)
03.	<i>Data input joystick (yoke roll ratio)</i>	Keluaran dari subsistem <i>wing leveler</i> sebagai perintah mengendalikan <i>joystick</i> arah horizontal
04.	<i>Wing Leveler Status</i>	Status <i>autopilot wing leveler</i> (1 atau 0)
05.	<i>Data input joystick (yoke_pitch_ratio)</i>	Keluaran dari subsistem <i>pitch hold</i> sebagai perintah mengendalikan <i>joystick</i> arah vertikal
06.	<i>Pitch Hold Status</i>	Status <i>autopilot pitch hold</i> (1 atau 0)

Proses penerimaan dan pengiriman data yang telah disebutkan di atas dapat dieksekusi melalui *file* Simulink yang berjudul “MAIN_UAVSIM_Simulink.slx” yang akan menjembatani pengiriman data dari subsistem dinamika dengan memanfaatkan *block* “UDP Receive” sehingga seluruh data yang berasal dari subsistem dinamik dapat diterima dan diproses oleh subsistem *monitoring* maupun kontroler. Begitu juga halnya dengan pengiriman data kembali menuju subsistem dinamika dapat dilakukan dengan memanfaatkan *block* “UDP Send”. Gambar 3.2 di bawah ini menunjukkan tampilan utama dari *file* Simulink tersebut, yang berguna untuk memproses data yang akan ditampilkan pada GUI subsistem *monitoring* dan proses perhitungan deviasi nilai target dengan nilai kini dari pesawat melalui subsistem kontroler.



Gambar 3.2 Tampilan utama file "MAIN_UAVSIM_Simulink.slx"

Subsistem “*data splitting*” yang terlihat pada gambar diatas akan menerima aliran data yang disebutkan pada Tabel 3.5 dan melakukan pemecahan aliran data mana yang akan menuju subsistem kontroler dan *monitoring*. Tampilan dari subsistem *data splitting* dapat dilihat melalui Gambar 3.3.



Gambar 3.3 Subsistem data *splitting*

Dapat dilihat dari Gambar 3.3 bahwa tiga data pertama yang dialirkan dari subsistem dinamika menuju *monitoring* yaitu *latitude*, *longitude*, dan *altitude* akan melewati *block* yang bernama “*Reference Manipulation*”. Pada subsistem ini, ketiga data tersebut yang semula berada pada tata acuan koordinat *geodetic* perlu diubah terlebih dahulu menjadi TAK *Geocentric* atau yang juga dikenal *Earth*

Center Earth Fixed sebagai perantara untuk kemudian dikonversi menjadi tata acuan koordinat lokal dengan konvensi *East North Up* (ENU).

3.1.1 Subsitem Dinamika

X-Plane merupakan perangkat lunak *flight simulator* yang dikembangkan oleh Laminar Research, yang dalam pengembangannya menggunakan “*Blade Element Theory*”. Perangkat lunak X-Plane dipilih untuk mewakili dinamika gerak dari pesawat yang akan disimulasikan dikarenakan dengan menggunakan perangkat lunak ini, telah disediakan model matematik atau model dinamik dari berbagai pesawat udara yang diklaim ‘mendekati’ dinamika pesawat sebenarnya [24].

Selain itu, dengan menggunakan X-Plane proses ekstraksi data dari hasil simulasi dapat dilakukan dengan mudah dan cepat melalui protokol UDP yang dapat digunakan sebagai jalur untuk mengirim data dari X-Plane ke perangkat lunak lainnya atau sebaliknya. Tidak hanya itu, X-Plane juga memberikan fasilitas berupa *plugin*. Dengan *plugin* ini, *user* dapat mengakses hampir semua data yang dimiliki oleh X-Plane.

3.1.1.1 Cirrus Vision SF50

Pesawat yang digunakan sebagai studi kasus dalam melaksanakan misi kendali lintas terbang adalah pesawat Cirrus Vision SF50. Pesawat ini dipilih dengan alasan yaitu pesawat ini merupakan satu-satunya pesawat jet bisnis yang disediakan langsung oleh X-Plane, sehingga dapat dipastikan model dinamik dari pesawat ini merepresentasikan gerak pesawat yang sebenarnya.

Cirrus Vision SF50 merupakan pesawat jet yang sangat ringan dan secara orisinal didesain dan dimanufaktur oleh Cirrus Aircraft yang berbasis di Minnesota, Amerika Serikat. Namun, setelah penerbangan perdananya pada tahun 2008, pengembangan pesawat ini dihentikan karena kekurangan dana. Pada tahun 2011, Cirrus dibeli oleh *China Aviation Industry General Aircraft* (CAIGA) dan pengembangannya dilanjutkan hingga mendapatkan *type certificate* oleh FAA pada Oktober 2016 [25].



Gambar 3.4 Pesawat Cirrus Vision SF50 [25]

Pesawat ini didesain untuk membawa maksimum enam penumpang dengan sistem propulsi yang terdiri dari satu unit mesin *turbofan* jenis Williams FJ33 yang mampu memberikan *thrust* sebesar 8 kN. Beberapa spesifikasi teknis dari pesawat ini ditampilkan melalui tabel berikut.

Tabel 3.7 Spesifikasi teknis Cirrus Vision SF50 [26]

Spesifikasi	Nilai
<i>Maximum Take Off Weight</i> (kg)	2721.55
<i>Maximum Payload</i> (kg)	544.31
<i>Maximum Range</i> (km)	2240.92
<i>Maximum Speed</i> (m/s)	154.33
<i>Wing Span</i> (m)	11.68

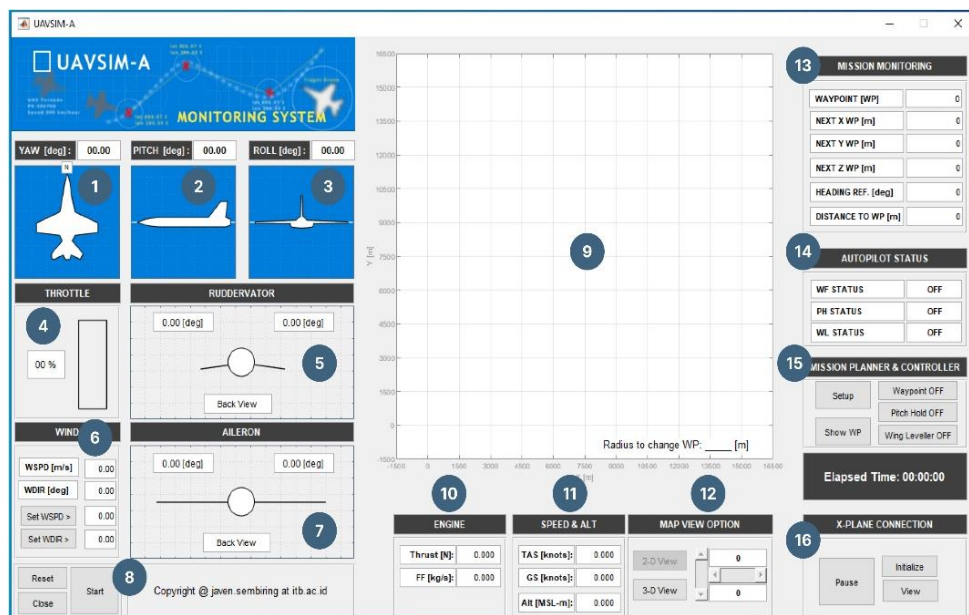
Untuk melakukan manuver *pitch* dan *roll*, pilot akan menggerakkan *joystick* atau disebut *sidestick* pada pesawat ini, untuk memberikan perintah kendali secara mekanikal. Sedangkan untuk melakukan gerakan *yaw*, seperti pada pesawat konvensional umumnya, pesawat ini menggunakan *rudder pedal*. Gambar 3.5 menunjukkan bidang kendali yang digunakan oleh *pilot* dalam memanipulasi defleksi dari bidang kendali pesawat Cirrus Vision SF50, yaitu berupa *sidestick*.



Gambar 3.5 Sidestick pada Cirrus Vision SF50

3.1.2 Subsistem *Monitoring*

Subsistem *monitoring* dibangun dengan tujuan sebagai sarana interaksi antara *user* dengan *simulator*. Interaksi tersebut berupa proses *monitoring* dan pendefinisian misi lintas terbang wahana melalui *mission planner*, dimana *user* dapat mendefinisikan misi berupa *waypoint* untuk dijalankan oleh subsistem *controller*. *Interface* pada subsistem ini seluruhnya dikembangkan menggunakan MATLAB GUIDE, termasuk antar-muka *mission planner*. Gambar 3.6 berikut ini menunjukkan tampilan dari subsistem *monitoring* yang mana akan muncul bila pengguna menjalankan *file* bernama “MAIN_UAVSIM.m”.



Gambar 3.6 User interface subsistem *monitoring* [3]

Keterangan mengenai setiap fitur pada *user interface* subsistem *monitoring* dijelaskan melalui tabel berikut.

Tabel 3.8 Keterangan *interface* subsistem *monitoring*

No.	Keterangan
1.	Menampilkan animasi sudut <i>heading</i> beserta nilai numeriknya dalam satuan derajat.
2.	Menampilkan animasi sudut <i>pitch</i> beserta nilai numeriknya dalam satuan derajat.
3.	Menampilkan animasi sudut <i>roll</i> beserta nilai numeriknya dalam satuan derajat.
4.	Menampilkan animasi posisi <i>throttle</i> beserta nilai numeriknya dalam satuan persen.
5.	Menampilkan animasi defleksi <i>elevator</i> beserta nilai numeriknya dalam satuan derajat.
6.	Menampilkan kecepatan dan arah terbang angin kondisi kini beserta tombol untuk mengatur kecepatan dan arah angin yang dikirim ke subsistem <i>Controller</i> .
7.	Menampilkan animasi defleksi <i>aileron</i> beserta dengan nilai numeriknya dalam satuan derajat.
8.	Tiga tombol dengan fungsi sebagai berikut: <ul style="list-style-type: none"> • Tombol <i>Reset</i>: me-reset UAVSIM-A kembali pada kondisi awal. • Tombol <i>Start</i>: memulai UAVSIM-A. • Tombol <i>Close</i>: keluar dari program UAVSIM-A.
9.	Menampilkan pergerakan UAV dalam koordinat lokal X, Y, Z serta misi <i>waypoint</i> yang akan dijalankan oleh UAV.
10.	Memonitor properti dari <i>engine</i> , meliputi <i>thrust</i> yang dihasilkan (dalam Newton) dan besarnya <i>fuel rate</i> dalam satuan kg/sec.
11.	Memantau kondisi terbang UAV yang meliputi tinggi terbang, <i>true airspeed</i> , dan <i>ground speed</i> .
12.	Berfungsi untuk mengatur <i>view Map</i> , baik tampilan 2D atau 3D serta opsi untuk memutar proyeksi tampilan.
13.	<i>Mission Monitoring</i> – fungsi pada bagian ini berhubungan dengan status misi yang sedang dijalankan. Data yang ditampilkan berupa <i>waypoint</i> tujuan beserta dengan koordinatnya, sudut referensi <i>yaw</i> , jarak terhadap <i>waypoint</i> tujuan, dan status dari sistem kendali (<i>ON/OFF</i>).
14.	<i>Autopilot Status</i> – memantau status autopilot. Terdapat 3 <i>autopilot</i> yaitu <i>Waypoint Following</i> , <i>Pitch Hold</i> , dan <i>Wing Leveller</i> . Data pada <i>Autopilot Status</i> ini diperoleh dari subsistem <i>Controller</i> .
15.	<i>Mission Planner & Controller</i> – Terdapat tombol <i>Setup</i> , <i>Show WP</i> , dan 3 buah fungsi <i>autopilot</i> untuk diaktifkan: <ul style="list-style-type: none"> • Tombol <i>Setup</i>: menampilkan GUI <i>Mission Planner</i> dimana melalui GUI ini <i>user</i> dapat mendefinisikan <i>waypoint</i> tujuan. • Tombol <i>Show WP</i>: menampilkan <i>waypoint</i> misi pada tampilan 9. • Tombol <i>Waypoint OFF</i>: mengaktifkan subsistem kendali <i>waypoint controller</i> ketika ditekan. • Tombol <i>Pitch hold OFF</i>: mengaktifkan subsistem kendali <i>pitch hold</i> ketika ditekan. • Tombol <i>Wing leveler OFF</i>: mengaktifkan subsistem kendali <i>wing leveler</i> ketika ditekan.
16.	<i>X-Plane Connection</i> – berhubungan dengan koneksi dan konfigurasi view pada X-Plane. <ul style="list-style-type: none"> • Tombol <i>Pause</i> berfungsi untuk menghentikan sementara subsistem Dynamics. • Tombol <i>Initialize</i> berfungsi untuk mendapatkan posisi awal UAV pada bola bumi X-Plane dan mentransfer posisi ini ke koordinat lokal UAVSIM-A. • Tombol <i>View</i> berfungsi untuk mengatur <i>view</i> pada X-Plane. <i>View</i> yang ditampilkan berupa tampak belakang (<i>chase view</i>), tampak samping, tampak depan dan <i>free view</i>.

Tampilan dari antar-muka *mission planner* yang akan muncul ketika *user* menekan tombol *setup* pada zona bernomor 15 pada Gambar 3.6 ditampilkan di bawah ini, dengan keterangan pada setiap zona nomor pada Tabel 3.8.



Gambar 3.7 User interface bagian *mission planner* [3]

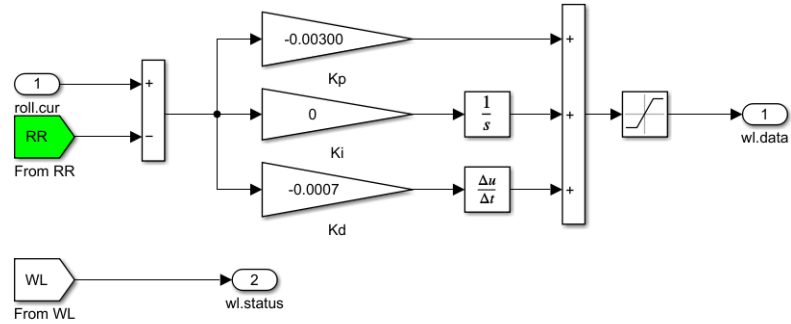
Tabel 3.9 Keterangan *interface* bagian *mission planner*

No.	Keterangan
1.	Sebagai input banyaknya <i>waypoint</i> yang akan dijalankan (maksimum 10).
2.	Berfungsi sebagai tombol awal pendefinisian <i>waypoint</i> . Melalui tombol ini <i>user</i> dapat mendefinisikan <i>waypoint</i> dengan cara mengklik poin-poin pada peta 6.
3.	Menampilkan koordinat secara numerik. Koordinat yang ditampilkan terdiri dari posisi X, Y, dan Z (tinggi relatif terhadap <i>ground</i>).
4.	Berfungsi sebagai pengontrol tampilan peta baik dalam tampilan 2 dimensi maupun 3 dimensi.
5.	Terdapat tiga tombol fungsi: <ul style="list-style-type: none"> • <i>Save</i>: menyimpan data <i>waypoint</i> dalam file berformat *.mat. • <i>Reset</i>: me-reset tampilan <i>Mission Planner</i> ke kondisi awal. • <i>Close</i>: keluar dari <i>Mission Planner interface</i>.
6.	Sebagai tempat mendefinisikan dan menampilkan <i>waypoint</i> .

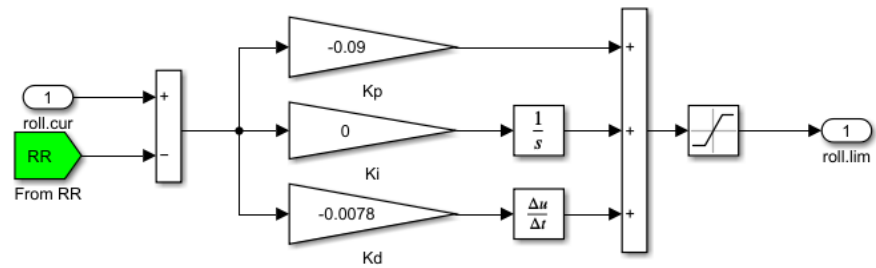
$$\phi_{error} = \phi_{cur} - \phi_{ref} \quad (3.1)$$

$$u(t)_{js_roll} = K_P \phi_{error}(t) + K_D \frac{d}{dt} \phi_{error}(t) \quad (3.2)$$

Besarnya nilai K_P dan K_D yang didapat melalui metode *trial and error* untuk kedua sistem kendali tersebut beserta penerapannya pada Simulink dapat dilihat melalui gambar berikut.



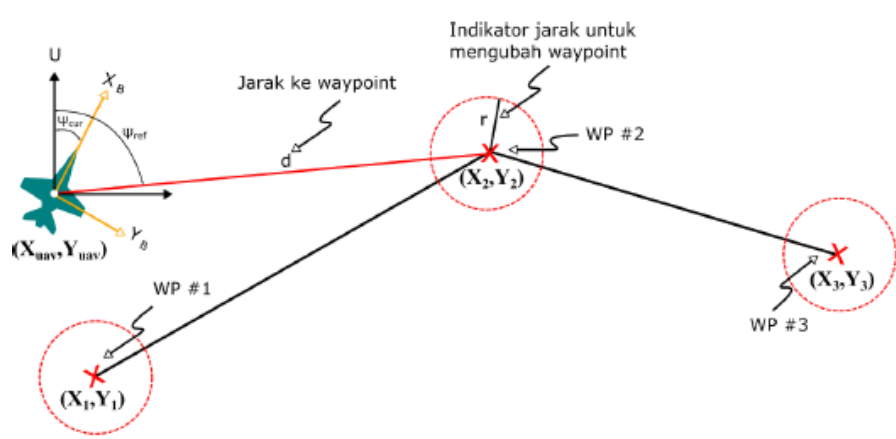
Gambar 3.9 Implementasi sistem kendali *wing leveler* pada Simulink



Gambar 3.10 Implementasi sistem kendali *roll limiter* pada Simulink

3.1.3.2 Sistem Kendali *Waypoint Following*

Pada penelitian ini, sistem kendali *waypoint following* yang digunakan merupakan gabungan dari sistem kendali *roll limiter*, *wing leveler* dan *yaw hold*, yang bertujuan untuk mengarahkan pesawat untuk mengikuti *waypoint* yang telah didefinisikan sebelumnya pada *mission planer*. Algoritme dari sistem kendali *yaw hold* adalah membandingkan sudut *heading* referensi yang adalah sudut *heading* tujuan dengan sudut *heading* pesawat saat ini. Perbedaan nilai ini akan dikalikan *KP* dan *KD* sehingga menghasilkan besaran yang akan memberikan kendali terhadap *joystick* terkhusus pada sumbu *roll*. Gambar 3.11 di bawah ini menampilkan gambaran strategi kendali yang digunakan pada sistem kendali *waypoint following*.



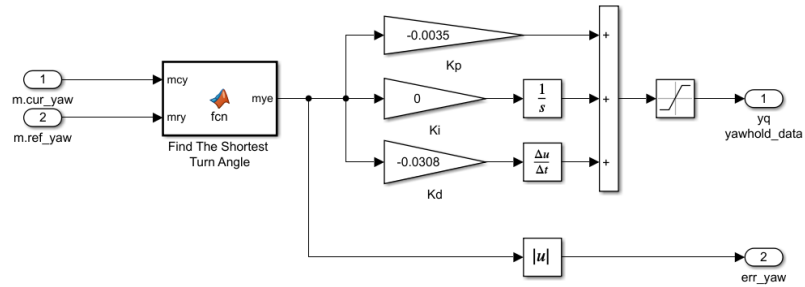
Gambar 3.11 Strategi kendali *waypoint following* [3]

Besaran ψ_{cur} menyatakan besarnya nilai sudut *heading* pesawat terkini, yang didapat langsung dari X-Plane. Nilai ψ_{ref} merupakan variabel nilai referensi dari sudut *heading* *waypoint* tujuan yang akan berubah setiap saat bergantung kepada posisi pesawat relatif terhadap letak *waypoint* tujuan. Besaran nilai sudut referensi ini dapat dihitung dengan menggunakan persamaan trigonometri berikut.

$$\psi_{ref} = \arctan\left(\frac{y_2 - y_{UAV}}{x_2 - x_{UAV}}\right) \quad (3.1)$$

Kedua besaran sudut *yaw* yang telah didapat akan dikonversi terlebih dahulu yang semula memiliki rentang nilai dari 0° hingga 360° menjadi -180° hingga 180°. Selanjutnya, nilai dari ψ_{error} dapat dicari dengan mengurangi ψ_{cur} dengan ψ_{ref} dengan memperhatikan arah terbang yang paling efisien untuk mencapai *waypoint* yang dituju.

Besaran ini akan digunakan sebagai *input* terhadap *PID controller*, yang mana besaran dari PID akan berbeda-beda bergantung terhadap pesawat yang digunakan. Untuk mendapatkan nilai konstanta tersebut, digunakan metode *trial and error* untuk mendapatkan kontroler yang mampu melaksanakan misi lintas terbang dengan gerakan osilasi seminim mungkin. Penerapan strategi kendali ini pada Simulink ditampilkan melalui Gambar 3.12.



Gambar 3.12 Implementasi sistem kendali *yaw hold* pada Simulink

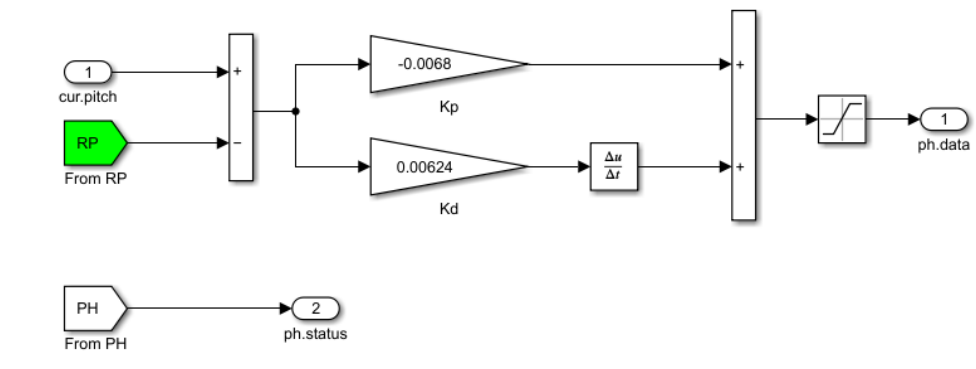
Untuk melakukan perubahan tujuan *waypoint*, ditetapkan sebuah parameter yaitu jarak pesawat terhadap *waypoint* (d). Besaran dari nilai d ditetapkan sejauh 450 m, dimana ketika jarak pesawat terhadap *waypoint* sudah mencapai nilai d , pesawat akan diarahkan menuju *waypoint* selanjutnya. Nilai dari variabel ini dapat dihitung dengan sederhana melalui persamaan di bawah ini dikarenakan nilai x dan y pesawat sudah direpresentasikan pada TAK horison lokal.

$$d = \sqrt{(x_2 - x_{UAV})^2 + (y_2 - y_{UAV})^2} \quad (3.2)$$

Penerapan dari sistem kendali *waypoint following* yang melibatkan 3 buah sistem kendali yaitu sistem kendali *roll limiter*, *yaw hold* dan *wing leveler* dapat dilihat melalui Gambar 3.13.

3.1.3.3 Sistem Kendali *Pitch Hold*

Sistem kendali *pitch hold* ditujukan untuk mengatur sudut *pitch* pesawat agar konstan pada suatu nilai referensi yang ditentukan. Pengaturan ini dilakukan dengan membandingkan nilai referensi sudut *pitch* dengan nilai kini sudut *pitch*. Nilai *error* ini akan dikalikan dengan nilai K_P dan K_D untuk selanjutnya dijadikan sebagai input ke *joystick* pada sumbu *pitch* untuk nantinya menggerakkan bidang kendali *elevator*. Dikarenakan pada penelitian ini matra *longitudinal* diabaikan, maka sistem kendali ini dibiarkan dalam keadaan tidak menyala. Penerapan sistem kendali ini pada Simulink dapat dilihat melalui gambar berikut.



Gambar 3.15 Implementasi sistem kendali *pitch hold* pada Simulink

3.2 Proses Pengumpulan Data pembelajaran

Proses pengumpulan data pembelajaran pesawat Cirrus Vision SF50 dalam melakukan berbagai misi terbang dilakukan dengan memanfaatkan perangkat lunak MATLAB, Simulink, X-Plane dan Python yang telah dikembangkan pada pekerjaan [3]. Proses ini secara sederhana dimulai dari persiapan perangkat lunak MATLAB, X-Plane, lalu dilanjutkan dengan rekonstruksi data misi penerbangan.

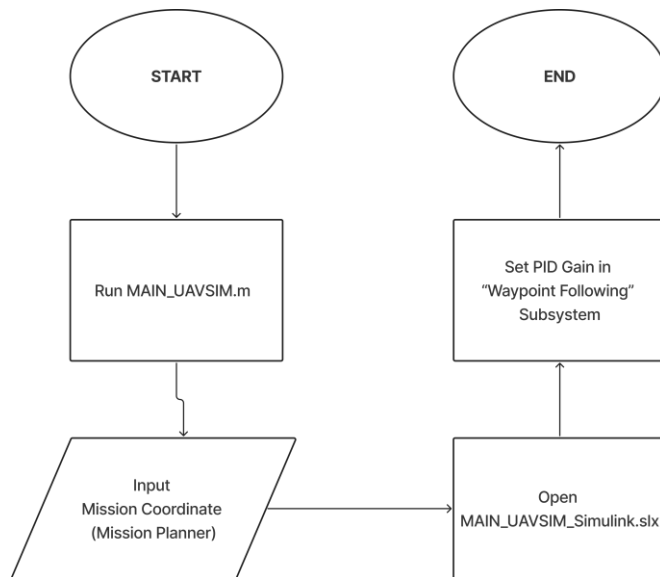
Data yang akan disimpan pada setiap misi penerbangan adalah gabungan dari aliran data *dynamics* dan *controller* menuju *monitoring* yang telah dipaparkan pada Tabel 3.2 dan Tabel 3.4, serta tambahan beberapa variabel yang akan menunjang proses pengembangan model kontroler berbasis DL yang ditampilkan melalui Tabel 3.9 berikut.

Tabel 3.10 Variabel data tambahan yang akan disimpan

Data	Keterangan
yaw_error	Nilai perbedaan sudut <i>yaw current</i> dengan <i>yaw reference</i> yang memiliki sudut putar terkecil
JSRoll_Sim	Nilai <i>joystick roll</i> keluaran dari PID <i>controller</i>
Roll_lim_stat	Status <i>roll limiter</i> dengan nilai 1 bila sudut <i>roll</i> melebihi 30°
KP	Nilai <i>yaw_error</i> yang dikalikan dengan konstanta <i>proportional</i>
KD	Nilai <i>yaw_error</i> yang dikalikan dengan konstanta <i>derivative</i>
error_rate	Laju perubahan variabel <i>yaw_error</i>

3.2.1 Persiapan Perangkat Lunak MATLAB

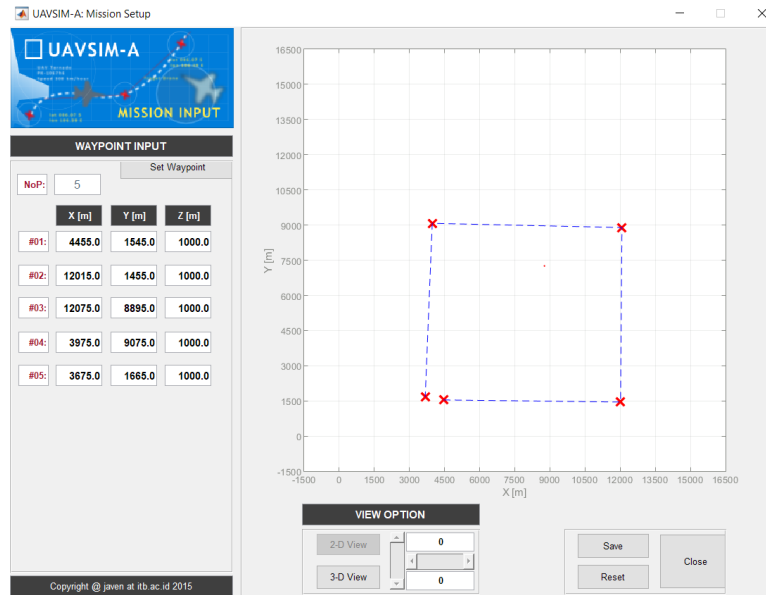
Tahapan pertama dalam proses pengumpulan data adalah dengan mempersiapkan perangkat lunak MATLAB hasil pekerjaan [3]. Dengan alur persiapan sebagai berikut.



Gambar 3.16 Alur persiapan perangkat lunak MATLAB

Detil mengenai alur persiapan perangkat lunak MATLAB dipaparkan dibawah ini.

1. Diawali dengan menjalankan perangkat lunak MATLAB dan mengubah lokasi *folder* pada *folder* bernamakan “uavms”. Kemudian, *file* yang bernamakan “MAIN_UAVSIM.m” akan dibuka dan dijalankan, yang menghasilkan keluaran berupa *Graphical User Interface* dari subsistem *monitoring* seperti yang ditunjukkan pada Gambar 3.3.
2. Selanjutnya, dengan menekan tombol *setup* pada GUI tersebut, akan muncul *interface* dari *mission planner*. Dengan mendefinisikan jumlah *waypoint* yang diinginkan pada zona 1 dan menekan tombol “*set waypoint*”, penulis dapat mendefinisikan koordinat misi pada koordinat kartesian seperti yang dapat dilihat pada Gambar 3.15. Setelah misi yang dibentuk sudah sesuai keinginan, maka tombol “*save*” ditekan untuk menyimpan misi tersebut.

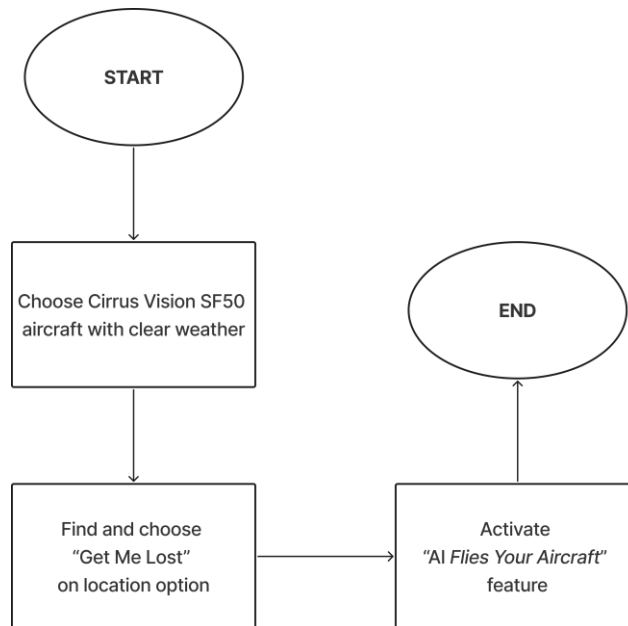


Gambar 3.17 Contoh pendefinisian misi pada *mission planner*

3. Kemudian, akan dibuka *file* Simulink pada *folder* “uavms” yang bernamakan “MAIN_UAVSIM_Simulink.slx”
4. Nilai konstanta PID pada subsistem “*Yaw Hold*” perlu diubah-ubah untuk merepresentasikan pilot yang berbeda dalam menyelesaikan misi terbang tertentu.

3.2.2 Persiapan Perangkat Lunak X-Plane 11

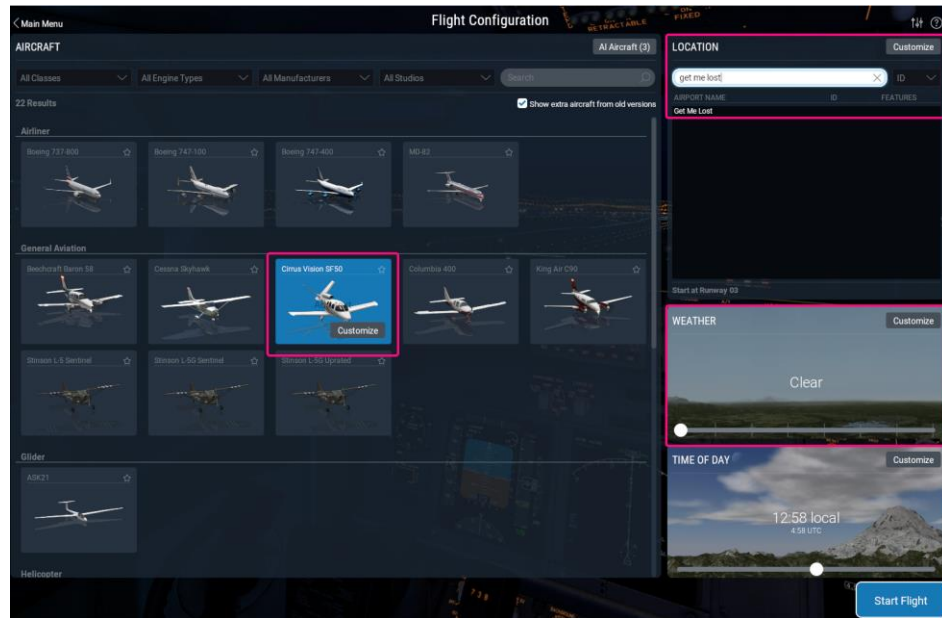
Alur persiapan perangkat lunak X-Plane 11 dapat dilihat melalui gambar di bawah ini.



Gambar 3.18 Alur persiapan perangkat lunak X-Plane 11

Detil mengenai setiap proses yang terpapar pada Gambar 3.17 dipaparkan dibawah ini.

1. Pada tampilan awal X-Plane 11, dipilih menu “*new flight*”, yang akan membawa pengguna pada halaman *flight configuration*. Disini akan dipilih Pesawat Cirrus Vision SF50 untuk disimulasikan dengan kondisi cuaca “*clear*” untuk meniadakan gangguan angin selama proses simulasi,
2. Lokasi penerbangan ditetapkan menjadi “*get me lost*”, yang akan menempatkan pesawat di udara dengan koordinat, ketinggian dan arah terbang secara acak. Hal ini dilakukan untuk mempercepat proses pengambilan data, dibandingkan dengan melakukan *take-off* secara *manual* hingga ketinggian tertentu. Pengaturan konfigurasi terbang pada X-Plane 11 ditampilkan melalui Gambar 3.19

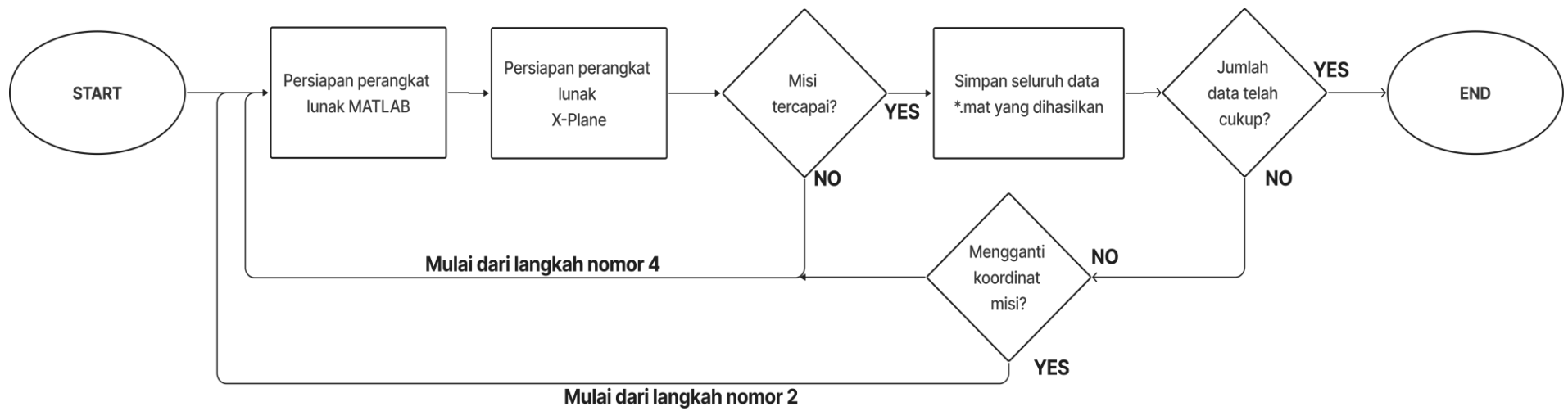


Gambar 3.19 Tampilan *flight configuration* X-Plane 11

3. Ketika pesawat sudah berada di udara, fitur bernama “AI *Flies your Aircraft*” diaktifkan, yang mana letak fitur ini terdapat pada *toolbar* bernama “*flight*”. Kegunaan dari fitur ini yaitu pesawat dapat diterbangkan dan distabilkan secara otomatis, sehingga ketika simulasi dijalankan pesawat sudah dalam kondisi stabil.

3.2.3 Rekonstruksi Data Misi Penerbangan

Proses rekonstruksi data misi penerbangan berdasarkan misi yang telah didefinisikan dapat dilakukan setelah persiapan perangkat lunak MATLAB dan X-Plane sudah dilaksanakan. Perlu dicatat bahwa proses pembukaan *file* hanya cukup dilakukan satu kali. Alur kerja rekonstruksi data misi secara umum dapat digambarkan melalui Gambar 3.20.

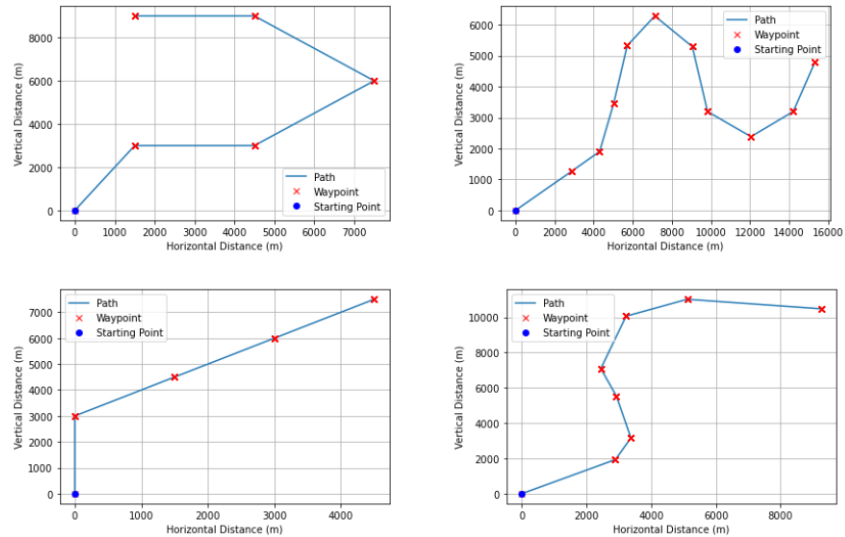


Gambar 3.20 Alur kerja rekonstruksi data misi

Detil dari alur kerja di atas adalah sebagai berikut.

1. Akan dijalankan terlebih dahulu tahapan nomor 2 pada persiapan perangkat lunak MATLAB berdasarkan misi yang ingin dieksekusi. Bila tidak ada perubahan misi maka tahap ini dapat dilewati.
2. Nilai konstanta PID akan diatur seperti yang terpapar pada langkah nomor 4 pada persiapan perangkat lunak MATLAB.
3. Setiap proses pada persiapan perangkat lunak X-Plane dilaksanakan.
4. Akan dipastikan terlebih dahulu bahwa pesawat telah mencapai keadaan stabil. Setelah itu, akan ditekan tombol “*Initialize*” pada GUI subsistem *monitoring*, yang akan mengambil posisi awal pesawat pada bola bumi X-Plane dan mentransfer posisi ini ke koordinat lokal UAVSIM-A, yaitu pada koordinat (0,0).
5. Tombol “*Start*” ditekan untuk menjalankan program Simulink, lalu tombol yang bertuliskan “*Waypoint OFF*” ditekan untuk mengaktifkan kendali lintas terbang. Dikarenakan penelitian ini hanya berfokus pada matra gerak lateral-direksional, maka tombol kontrol yang lain tidak akan diaktifkan.
6. Kembali lagi pada tampilan X-Plane, fitur “*AI Flies your Aircraft*” perlu dinonaktifkan agar pesawat dapat sepenuhnya diambil alih oleh subsistem kontroler dari Simulink.
7. Jika pesawat mampu menyelesaikan misi yang telah didefinisikan serta tidak melakukan gerakan *swaying* atau berosilasi terlalu banyak, seluruh data *.mat yang dihasilkan akan disimpan dalam suatu *folder* dengan format *comma separated value (CSV)*.

Langkah nomor 1 hingga 7 dilakukan hingga didapati 10 buah *files* CSV yang kemudian akan digabungkan menjadi satu buah *file* CSV. Bentuk lintasan misi yang disimulasikan terdiri dari banyak *waypoint*, dengan harapan dengan melakukan satu kali simulasi, didapatkan karakter kriteria rentang sudut *roll* yang berbeda-beda. Gambar 3.21 menunjukkan sampel dari lintasan yang disimulasikan.



Gambar 3.21 Sampel lintasan simulasi

Melalui *file* gabungan tersebut, akan dilakukan karakterisasi data pembelajaran berdasarkan kriteria sudut *roll* pesawat pada setiap *section* dengan detailnya disampaikan pada bab 4. Maksud dari *section* ini adalah rentang dimana pesawat bergerak menuju $waypoint_{(n)}$ dari $waypoint_{(n-1)}$. Tidak seperti kasus *machine learning* biasanya, dimana data pembelajaran yang digunakan sebisa mungkin adalah hal yang serupa dengan tujuan utama model tersebut, dengan dilakukannya karakterisasi data pembelajaran, lintasan dari setiap *set* data pembelajaran tidak akan serupa dengan lintasan yang nantinya akan diuji pada subsistem dinamika.

Pada bab ketiga ini, telah dibahas mengenai detail dan variabel yang dikirimkan dari setiap subsistem penyusun simulator sistem kendali dan lintasan terbang yang telah dikembangkan sebelumnya pada pekerjaan [3] dan [4] dan juga proses pengumpulan data pembelajaran beserta detail variabel data yang disimpan, yang mana akan menjadi basis pembelajaran dari sistem kendali yang akan dikembangkan menggunakan metode *deep learning* yang akan dibahas pada bab selanjutnya.

BAB IV

SISTEM KENDALI LINTAS TERBANG BERBASIS *DEEP LEARNING*

Pada bab ini akan dibahas mengenai tahapan yang dilakukan dalam proses pengembangan sistem kendali lintas terbang berbasis *deep learning* yang didahului dengan pra-pemrosesan data dan diikuti dengan proses pengembangan model *deep learning* hingga proses integrasi model *deep learning* dengan subsistem dinamika.

4.1 Pra-pemrosesan Data

Pada tahapan pra-pemrosesan data, *file* yang berisi gabungan dari 10 buah data misi penerbangan akan diterapkan proses sampel ulang frekuensi data, pembersihan data, yang kemudian diikuti dengan proses *data splitting*.

4.1.1 Resample Frequency

Pra-pemrosesan data dimulai dengan menerapkan resample pada setiap data misi penerbangan hasil keluaran dari MATLAB dan Simulink. Dikarenakan laju penyimpanan data semula sebesar 50 Hz, ini akan menyebabkan meningkatnya biaya komputasi untuk memproses data tersebut. Maka, dilakukan resample pada setiap data misi penerbangan menjadi satu data sampel setiap 4 data sampel pada data mentah. Dengan melakukan hal ini, jumlah data sampel akan berkurang menjadi seperempat kali jumlah data sampel sebelumnya. *Pseudocode* dalam membentuk fungsi *resample* data disajikan melalui gambar berikut.

FUNCTION 2: RESAMPLE FREQUENCY

Input: *df*: pandas dataframe; *freq*: integer
Output: resampled dataframe based on frequency defined

```
1  DEFINE FUNCTION resample_df(df, freq)
2      new_df ← blank dataframe with same column name as df
3      FOR index IN df where column name is 'index'
4          IF index EQUAL TO 1
5              new_df AT first row ← df AT first row
6          END IF
7          ELSE IF index MOD 4 EQUAL TO 0
8              new_df AT row (index/4) ← df AT row (index- 1)
9          END IF
10     END FOR
11     RETURN new_df
12 END FUNCTION
```

Gambar 4.1 *Pseudocode resample frequency*

4.1.2 Pembersihan Data

Setelah melalui proses resample, data akan dibersihkan terlebih dahulu. Pembersihan data yang dimaksud meliputi:

4.1.2.1 Menghilangkan *Outliers*

Proses ini bertujuan untuk menghilangkan sampel data yang memiliki perintah *joystick* pada sumbu *roll* di luar nilai wajarnya atau disebut juga *outliers*. Hal ini dicapai dengan menghapus baris pada tabel dengan indeks pada kolom yang mewakili perintah *joystick* pada sumbu *roll* memiliki nilai lebih besar sama dengan 0.6 atau lebih kecil sama dengan -0.6. Nilai batas tersebut ditetapkan demikian dikarenakan keseluruhan data misi dalam keadaan normal tidak pernah menunjukkan nilai defleksi *joystick* pada sumbu *roll* yang lebih besar dari 0.4 atau lebih kecil dari -0.4. Penerapan algoritme ini dalam suatu fungsi dapat dilihat melalui Gambar 3.22.

FUNCTION 3: REMOVE OUTLIERS

Input: df: pandas dataframe
Output: dataframe without outliers

```
1  DEFINE FUNCTION remove_outliers(df)
2      js_roll ← df where column name is 'JSRoll_Sim'
      idx_list ← empty list as storage
3      FOR index, value IN js_roll
4          IF value ≥ 0.6 OR value ≤ -0.6
5              Append idx to idx_list
6          END IF
7          IF number of items in idx_list NOT EQUAL TO 0
8              FOR i IN idx_list
9                  Remove row-i from dataframe
10             END FOR
11         END IF
12     END FOR
13     RETURN df
14 END FUNCTION
```

Gambar 4.2 Pseudocode remove outliers

4.1.2.2 Penghalusan Data

Penghalusan data yang dimaksud yaitu terbatas pada nilai masukan pada *joystick* sumbu *roll* yang bernilai fluktuatif akibat perintah kendali *roll limiter*, yang akan memberikan reaksi berlawanan terhadap arah defleksi *joystick* ketika sudut *roll* dari pesawat melebihi 30°.

Hal ini dicapai dengan mengganti nilai *joystick* pada sumbu *roll* menjadi *Not a Number* (NaN) jika pada waktu yang terkait, sudut *roll* dari pesawat melebihi 30°. Selanjutnya nilai NaN ini akan diganti nilainya dengan melakukan interpolasi linier. *Pseudocode* dari tahapan melakukan penghalusan data dapat dilihat melalui Gambar 3.23.

FUNCTION 4: DATA SMOOTHING

Input: *df*: pandas dataframe; *filter_column*: string
Output: smooth data on column 'JSRoll_Sim' by applying linear interpolation method

```
1  DEFINE FUNCTION data_smoothing(df)
2      df_cutted ← df starting from 2nd index
3      roll_lim_stat ← df_cutted with column name of
        'Roll_lim_stat'
4      idx_list ← empty list as storage
5      FOR index, value IN roll_lim_stat
6          IF value EQUAL TO 1
7              Append index to idx_list
8          END FOR
9      FOR value_js IN df_cutted with column name of
        'JSRoll_Sim'
10         Convert value_js with the same index as in
            idx_list become NaN
11         Interpolate NaN value using Linear method
12         Change value df with column name 'JSRoll_Sim'
            with interpolated value
13     END FOR
14     RETURN df
15 END FUNCTION
```

Gambar 4.3 *Pseudocode data smoothing*

4.1.2.3 Pemotongan Data

Pemotongan data terkhusus sesaat sebelum pesawat berhasil melintasi *waypoint* terakhir dibutuhkan untuk mendapatkan data pembelajaran yang relevan. Hal ini dikarenakan algoritme dari pengendalian lintas terbang akan terus memerintahkan pesawat untuk tetap menuju *waypoint* terakhir jika sudah tidak ada *waypoint* selanjutnya. Hal ini menyebabkan pesawat akan melakukan manuver berbelok yang ekstrem dalam rangka mencapai *waypoint* tersebut. Pemotongan data ini dilakukan dengan menerapkan algoritme yang ditampilkan pada Gambar 3.24.

FUNCTION 5: DATA SLICING

Input: df: pandas dataframe
Output: sliced dataframe especially on the end of flight

```
1  DEFINE FUNCTION data_slicing(df)
2      num_of_wp ← unique value from df with column name of 'num_wp'
3      Last_index_list ← empty list as storage
4      FOR index, value IN df with column name of 'num_wp'
5          IF value EQUAL TO last value of num_of_wp AND
           (df with column name of 'wp_dist' at index is
            above 400 and below 500)
6              Append index to Last_index_list
7          END IF
8      END FOR
9      Slice df until the last value of Last_index_list
10     RETURN df
11 END FUNCTION
```

Gambar 4.4 Pseudocode data slicing

4.1.3 Data Splitting

Data yang telah melewati proses *resample frequency* dan pembersihan data kemudian akan dipecah menjadi empat buah *set* data pembelajaran dalam rangka mengkarakterisasi data pembelajaran untuk melihat pengaruh data pembelajaran terhadap kualitas model kontroler. Karakterisasi yang dimaksud pada penelitian ini yaitu karakter sudut *roll* dari pesawat dari setiap *section waypoint* demi *waypoint*. Karakter tersebut dikategorikan menjadi empat buah kategori yang dapat dilihat melalui Tabel 4.1.

Tabel 4.1 Kategori pembagian data pembelajaran

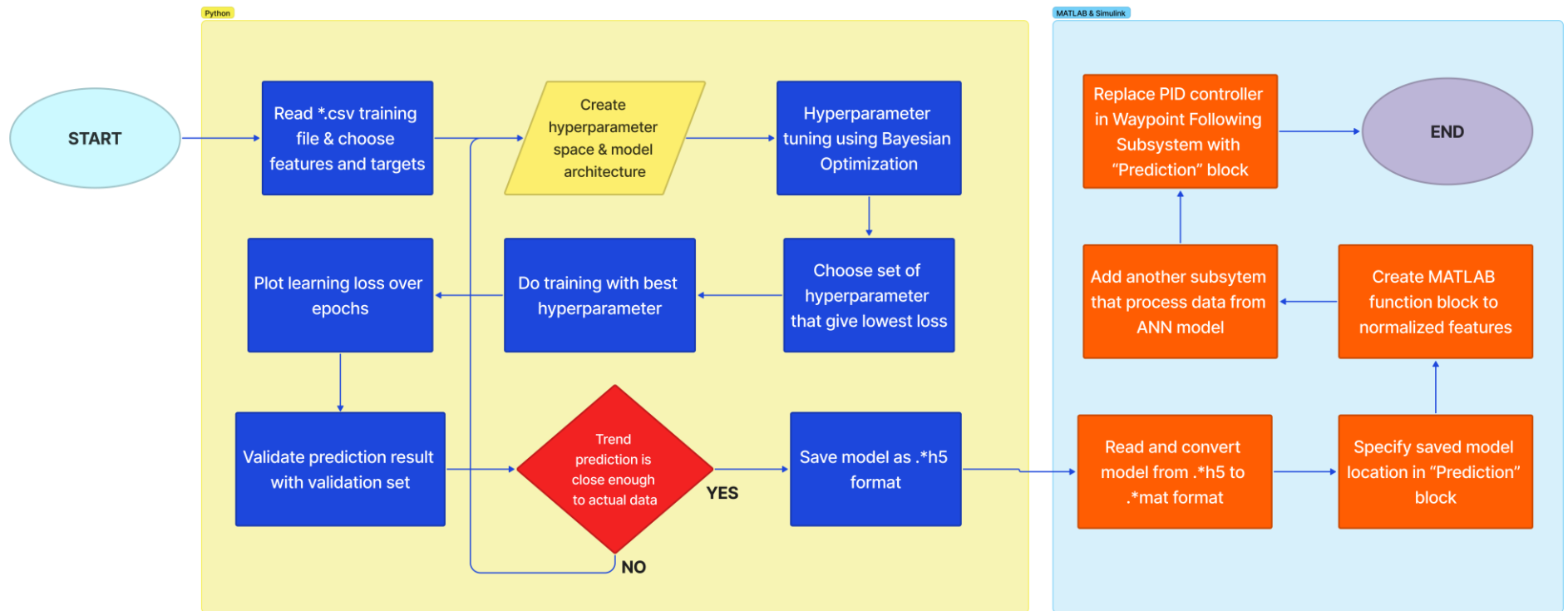
<i>Set</i>	Rentang Sudut <i>Roll</i>
1	$-10^\circ \leq \phi \leq 10^\circ$
2	$-10^\circ \leq \phi \leq 40^\circ$
3	$-40^\circ \leq \phi \leq 10^\circ$
4	$-40^\circ \leq \phi \leq 40^\circ$

Dapat dilihat melalui tabel di atas, *set* data pembelajaran pertama memiliki rentang sudut *roll* yang paling sempit yaitu dari -10° hingga 10° . Sedangkan *set* data pembelajaran dengan rentang sudut *roll* yang paling luas dipegang oleh *set* data pembelajaran keempat. Batas atas dan batas bawah yang bernilai 40° dipilih dengan alasan yaitu ketika pada kasus manuver ekstrem, sistem kendali *roll limiter* yang berguna untuk membatasi sudut *roll* pesawat agar tidak melebihi 30° tidak mampu memberikan sinyal kendali yang cukup untuk membatasi gerakan *roll* pesawat.

Set data pembelajaran kedua dibentuk demikian dengan tujuan memperkaya data pembelajaran *set* pertama dengan rentang sudut *roll* positif atau dapat diartikan manuver ke arah kanan. Sedangkan *set* data pembelajaran ketiga bertujuan memperluas *set* data pembelajaran pertama pada kondisi manuver berbelok ke arah kiri. Selanjutnya, akan dibangun model *deep learning* untuk masing-masing *set* data.

4.2 Pengembangan Model *Deep Learning*

Model DL yang akan dikembangkan berjumlah empat buah model, dengan tahapannya yaitu dimulai dari pemilihan *features* dan *targets* yang akan digunakan, proses normalisasi data, *hyperparameter optimization* menggunakan algoritme *Bayesian Optimization* pada TensorFlow dan proses mengubah model menjadi format *.mat. Gambar 4.5 menunjukkan alur dari pengembangan keempat model *deep learning* yang dilakukan.



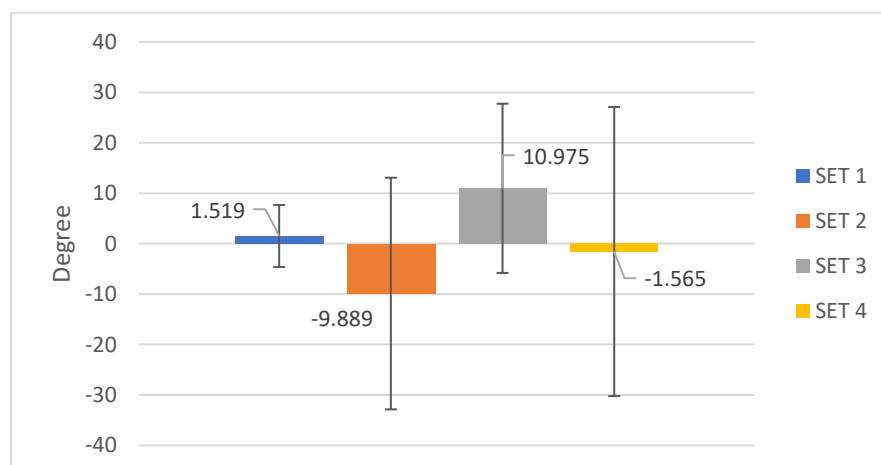
Gambar 4.5 Alur pengembangan model DL

4.2.1 Pemilihan *Features* dan *Targets*

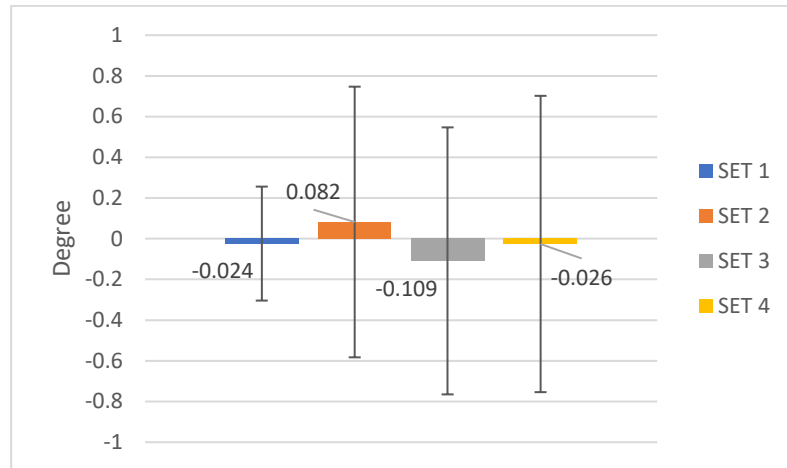
Targets yang dipilih dalam penelitian ini akan disamakan dengan keluaran dari subsistem *waypoint controller* pada pekerjaan [3] yaitu memberikan perintah defleksi *joystick* pada sumbu *roll* atau sama dengan variabel bernama “JSRoll_Sim” yang tertampil pada Tabel 3.9.

Features yang dipilih melibatkan variabel umum yang digunakan pada metode PID yaitu berupa deviasi nilai atau *error* yang pada kasus ini adalah *error* dari sudut *yaw*, dan laju perubahannya. Nilai *error* dipilih sebagai *features* mengingat bahwa sistem kendali yang ingin dikembangkan diharapkan dapat bekerja baik pada lintasan apapun, sehingga dengan digunakannya nilai *error* sebagai *features* pembelajaran, lintasan apapun yang akan diuji bukanlah suatu masalah. Selain itu, akan digunakan *features* berupa sudut *roll* mengingat karakterisasi data pembelajaran dilakukan berdasarkan rentang nilai sudut *roll* pesawat dan variabel ini memiliki korelasi bernilai 0.73 terhadap variabel “JSRoll_Sim” pada data gabungan yang telah melewati tahapan pra-pemrosesan data.

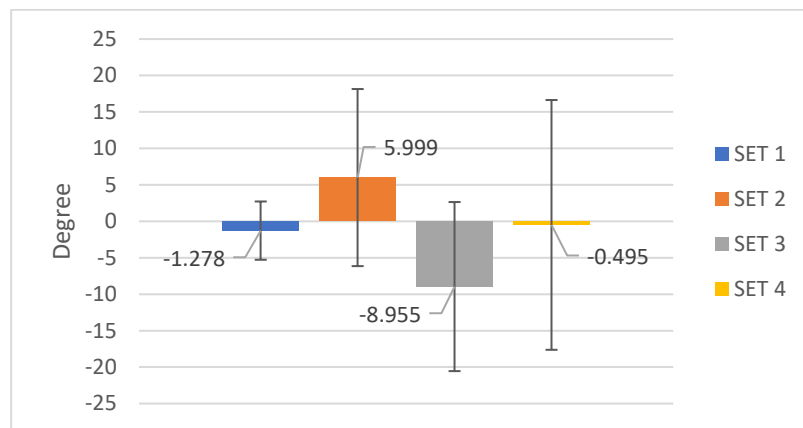
Gambar berikut ini menunjukkan nilai rata-rata beserta nilai standar deviasi dari setiap *features* dan juga *targets* dari keempat buah *set* data.



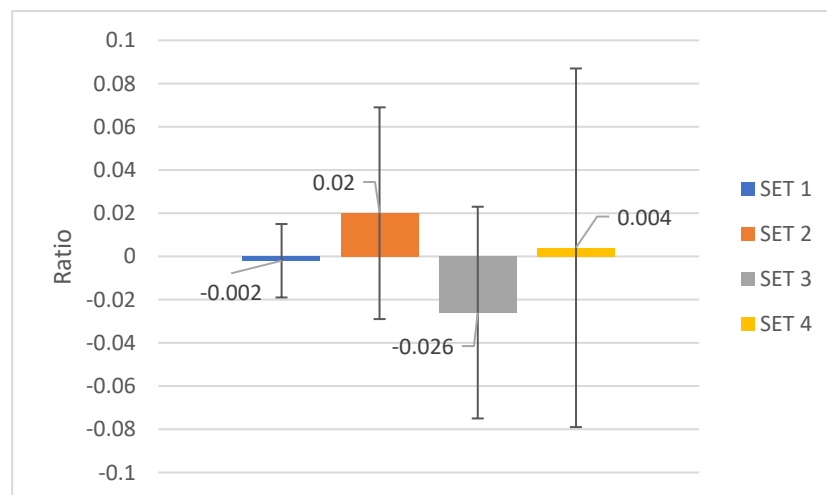
Gambar 4.6 Nilai rata-rata *yaw error* dari keempat *set* data



Gambar 4.7 Nilai rata-rata *yaw error rate* dari keempat *set* data



Gambar 4.8 Nilai rata-rata sudut *roll* dari keempat *set* data



Gambar 4.9 Nilai rata-rata rasio *joystick roll* dari keempat *set* data

Dari keempat gambar di atas, dapat dilihat untuk *set* data 1 hingga 3 memiliki kesamaan pola, yaitu ketika nilai rata-rata dari variabel sudut *roll*, *yaw error rate*, dan *joystick roll* bernilai negatif, nilai rata-rata dari variabel *yaw error* akan bernilai positif, begitupula sebaliknya. Namun pada *set* data ke-4, kesamaan pola ini tidak ditemukan, dikarenakan cakupan data yang terlalu luas akan menggeser nilai rata-rata pada *set* data ini.

Didapati juga bahwa pada variabel *yaw error rate*, nilai standar deviasi pada model *set* kedua hingga keempat memiliki nilai yang tinggi hingga memiliki cakupan persebaran data yang mirip. Berbeda halnya dengan ketiga variabel lainnya, yang mana cakupan persebaran data akan dominan pada tanda positif atau negatif dari nilai rata-ratanya dengan model *set* pertama memiliki cakupan data paling sempit, dan model *set* keempat memiliki cakupan data paling luas.

4.2.2 Normalisasi Data

Dalam memberikan masukan pada model *machine learning* apapun, data harus melewati proses normalisasi dengan alasan yaitu dengan melakukan normalisasi, setiap variabel akan mendapatkan pembobotan yang setara, sehingga tidak ada variabel yang akan mengarahkan performa model ke suatu arah yang berlainan akibat variabel tersebut memiliki nilai yang lebih besar dibandingkan variabel masukan lainnya.

Proses normalisasi yang digunakan pada tahapan ini adalah menggunakan teknik *Min-Max scaler*, yaitu sebuah teknik yang akan melakukan transformasi setiap *feature* agar memiliki rentang nilai tertentu. Rentang nilai yang dipilih adalah 0 hingga 1. Proses normalisasi ini dicapai dengan menerapkan persamaan (1), sedangkan proses denormalisasi dapat dilakukan dengan menerapkan persamaan (2). Proses normalisasi ini hanya diterapkan pada *features* saja, sedangkan *targets* akan dibiarkan pada nilai sesungguhnya.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.1)$$

4.2.3 Bayesian Optimization Menggunakan KerasTuner

Dalam menerapkan algoritme *Bayesian Optimization* untuk mencari *hyperparameter* terbaik, perlu dibuat terlebih dahulu *base model* atau model dasar yang nantinya melalui *hyperparameter space* yang didefinisikan. Algoritme BO akan mencoba menggunakan nilai dari rentang tersebut dan menggunakan informasi dari iterasi sebelumnya dalam menentukan *set hyperparameter* berikutnya. Model dasar tersebut diimplementasikan dengan menerapkan *pseudocode* yang tertampil pada Gambar 4.10.

FUNCTION 6: BASE MODEL	
<i>Input: hyperparameter space</i>	
1	DEFINE FUNCTION (<i>hyperparameter space</i>)
2	<i>model</i> ← Assign sequential model
3	<i>model.add</i> (<i>Dense</i> (<i>units</i> = <i>units_first</i> ,
4	<i>input_dim</i> =(<i>x_train.shape</i> [1]),
5	<i>kernel_initializer</i> = <i>initializer</i>))
6	FOR <i>i</i> in range (<i>Num_hidden</i>)
7	<i>model.add</i> (<i>Dense</i> (<i>units</i> = <i>units_hidden</i> ,
8	<i>activation</i> = <i>activation_hidden</i>)
9	END FOR
10	IF <i>drouput</i> == <i>TRUE</i>
11	<i>model.add</i> (<i>Dropout</i> (<i>rate</i> = <i>Dropout_rate</i>)
12	END IF
13	<i>model.add</i> (<i>Dense</i> (<i>units</i> = <i>y_train.shape</i> [1],
14	<i>activation</i> = <i>activation</i>))
15	END FUNCTION

Gambar 4.10 Pseudocode base model

Hyperparameter space yang dimaksud berisikan rentang dari setiap *hyperparameter* yang ingin dicoba. Tabel 4.1 di bawah ini menunjukkan *hyperparameter* apa saja yang akan diatur beserta ruang yang didefinisikan.

Tabel 4.2 *Hyperparameter space*

<i>Hyperparameter Name</i>	<i>Space</i>
<i>Units_first</i>	(min=8, max=32, step=4)
<i>Units_hidden</i>	(min=8, max=32, step=4)
<i>Initializer</i>	['zeros','glorot_normal','he_normal']
<i>Num_hidden</i>	[0,1,2]
<i>Activation</i>	['elu','LeakyReLU']
<i>Activation_hidden</i>	['elu','LeakyReLU']
<i>Optimizer</i>	['sgd', 'rmsprop', 'adam']
<i>Dropout</i>	True or False
<i>Dropout_rate</i>	(min=0.1, max=0.5, step=0.1)
<i>Learning_rate</i>	(min=1e-4, max=1e-1, sampling=log)

Setelah *based model* dan *hyperparameter space* telah didefinisikan, maka algoritme *Bayesian Optimization* yang disediakan oleh *library* KerasTuner dapat dijalankan dengan jumlah iterasi sebanyak 100 kali untuk setiap set data dengan objektif meminimalkan nilai *loss*. Dalam setiap iterasinya model akan dilatih sebanyak 5 *epochs*. Sebagai catatan, *base model* dan *hyperparameter space* tidak akan dirubah dalam proses *hyperparameter tuning* meskipun akan digunakan *dataset* yang berbeda-beda untuk setiap model. Tabel di bawah ini menunjukkan *set hyperparameter* terbaik untuk masing-masing set data pembelajaran.

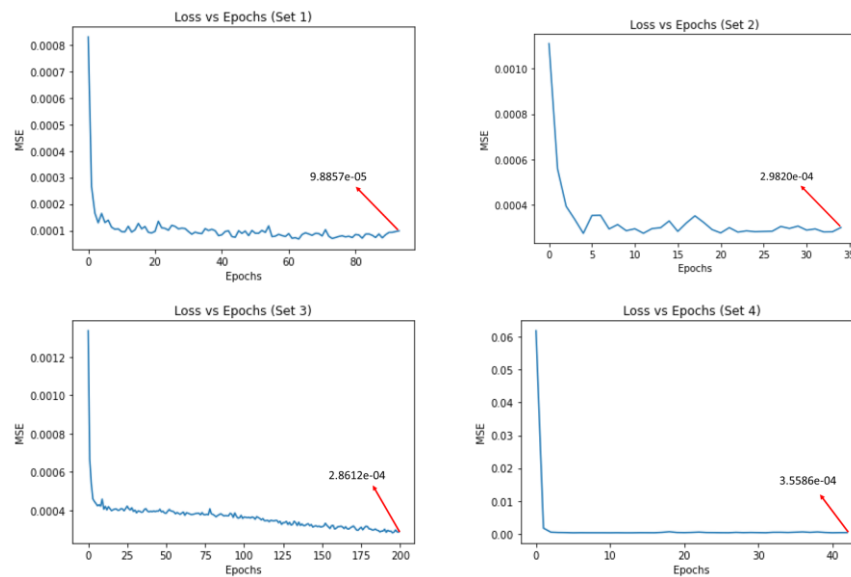
Tabel 4.3 *Set hyperparameter* terbaik

<i>Hyperparameter</i> <i>Name</i>	<i>Best</i> (Set 1)	<i>Best</i> (Set 2)	<i>Best</i> (Set 3)	<i>Best</i> (Set 4)
<i>Num_hidden</i>	1	0	2	0
<i>Units_first</i>	8	32	32	8
<i>Units_hidden</i>	32	-	32	8
<i>Initializer</i>	zeros	zeros	zeros	he_normal
<i>Activation</i>	elu	elu	LeakyReLU	elu
<i>Activation_hidden</i>	LeakyReLU	-	LeakyReLU	-
<i>Optimizer</i>	Adam	Adam	Adam	Adam
<i>Dropout</i>	True	True	False	False
<i>Dropout_rate</i>	0.5	0.5	-	-
<i>Learning_rate</i>	9.03e-3	2.56e-3	8.1e-4	0.1
<i>Loss for 5 epochs</i> (MSE)	1.01e-4	3.02e-4	2.4e-4	9.3e-4

4.2.4 Pembelajaran Model Menggunakan TensorFlow

Proses *training* dilakukan dengan menggunakan parameter-parameter yang terlampir pada Tabel 4.2 pada masing-masing set data pembelajaran. Setiap model akan dilatih maksimum sebanyak 200 *epochs*. Namun ditambahkan fungsi *callback* berupa *early stopping* yang akan menghentikan proses *training* bila dalam 30 *epochs* tidak terjadi penurunan *loss* yang signifikan. *Loss* berupa nilai *mean*

squared error pada setiap *epoch*-nya dari keempat buah model yang dilatih ditunjukkan melalui gambar berikut.



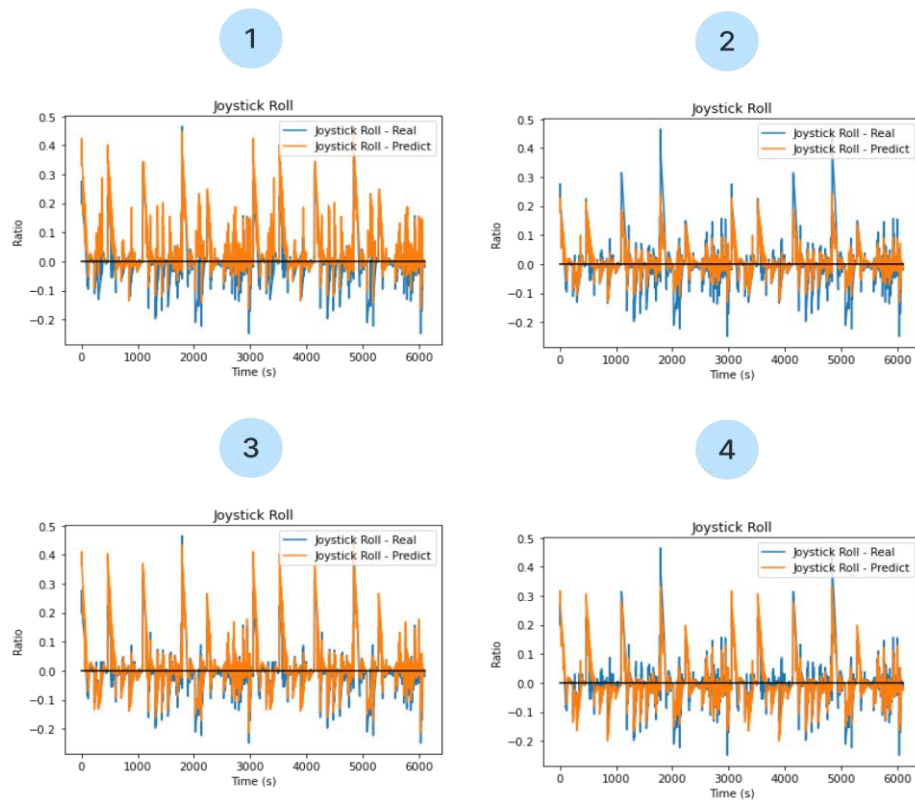
Gambar 4.11 Grafik *loss vs epochs*

Dari gambar di atas, dapat terlihat bahwa keempat model terus mengalami penurunan nilai MSE menuju nilai konvergen yang berbeda-beda. Namun, hanya model yang dilatih menggunakan set data ketiga yang diiterasi hingga 200 *epochs*. Sedangkan, ketiga model lainnya berhenti diiterasi di bawah 100 *epochs*.

Didapati juga bahwa model set kedua hingga keempat, nilai MSE pada akhir proses *training* berada pada orde 10^{-4} . Berbeda halnya dengan model set pertama, yang memiliki nilai MSE terkecil, yaitu pada orde 10^{-5} . Hal ini dikarenakan *set* data pertama memiliki jangkauan nilai *features* serta *targets* yang paling sempit dan seimbang antara nilai positif dan negatif, yang memungkinkan model dapat belajar lebih spesifik.

Selanjutnya, performa dari model akan diuji terlebih dahulu dengan membandingkan hasil prediksi *joystick roll* dengan data validasi. Hal yang dilihat pada tahapan ini adalah apakah prediksi dari model memiliki *trend* pergerakan *joystick* yang serupa dengan data validasi, khususnya apakah pada rentang nilai defleksi *joystick* seharusnya bernilai negatif, prediksi dari model juga memberikan nilai yang negatif dan sebaliknya.

Validasi menggunakan data pembelajaran ini dilakukan semata-mata hanya untuk memastikan model mampu memberikan prediksi yang baik dan siap diintegrasikan dengan model X-Plane. Keberhasilan dari kontroler berbasis DL ini sepenuhnya dinilai dari keberhasilan pesawat pada X-Plane dalam menyelesaikan misi yang didefinisikan. Data validasi yang digunakan adalah *dataset* yang berisi gabungan 10 data misi penerbangan atau sama dengan *set* data pembelajaran keempat. Gambar 4.47 menunjukkan perbandingan nilai defleksi *joystick* pada sumbu *roll* dari keempat model terhadap keseluruhan data pembelajaran.



Gambar 4.12 Plot prediksi *joystick roll* model DL vs data validasi

Gambar 4.12 menunjukkan bahwa keempat model DL mampu mengikuti *trend* pergerakan *joystick* pada sumbu *roll* dari data validasi dengan rentang cakupan nilai rasio yang berbeda-beda. Model pertama terlihat memberikan keluaran nilai rasio *joystick roll* yang sedikit melebihi nilai aktualnya atau *overshoot* dalam menjangkau nilai positif, namun cukup jauh dari nilai aktualnya atau *undershoot* dalam menjangkau nilai negatif.

Model kedua yang dilatih dengan data pembelajaran yang lebih banyak mencakup manuver berbelok dengan sudut *roll* positif memberikan hasil prediksi yang tidak *overshoot* dalam rentang nilai positif, tetapi *undershoot* pada rentang nilai negatif. Sedangkan, model ketiga yang dilatih dengan data pembelajaran yang lebih banyak mencakup manuver berbelok dengan sudut *roll* negatif memberikan hasil yang sedikit *undershoot* pada rentang nilai negatif, tetapi memberikan hasil yang *overshoot* dalam menjangkau nilai positif.

Hal ini sesuai dengan distribusi data yang ditampilkan pada Gambar 4.6 dan Gambar 4.7, bahwa data yang didominasi dengan data bernilai positif pada variabel sudut *roll* akan memberikan hasil rasio *joystick roll* yang didominasi data bernilai positif juga, begitupula sebaliknya. Sehingga, set ketiga ini seharusnya mampu memberikan hasil prediksi yang baik pada cakupan data yang bernilai negatif, sedangkan untuk cakupan data bernilai positif dapat dicapai lebih baik oleh *set* data kedua.

Namun, performa model secara keseluruhan dan kuantitatif dapat diamati melalui *metrics* berupa nilai *mean squared error* (MSE) dan *mean absolute error* (MAE) hasil prediksi model terhadap data validasi yang dihitung menggunakan persamaan 2.1 dan 2.2. Berikut ini ditampilkan nilai *metrics* tersebut dari setiap *set* model DL.

Tabel 4.4 Nilai metriks dari setiap model DL

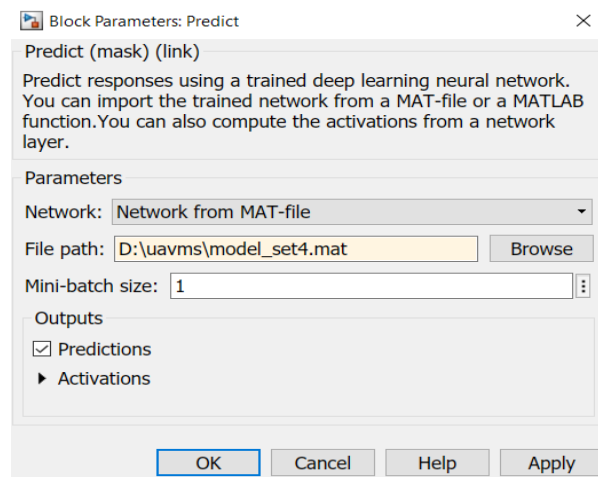
<i>Metrics</i>	<i>Model</i>			
	<i>Set 1</i>	<i>Set 2</i>	<i>Set 3</i>	<i>Set 4</i>
MSE	3.531e-03	1.391e-03	1.364e-03	8.748e-04
MAE	3.559e-02	2.043e-02	2.036e-02	1.993e-02

Secara kuantitatif, model *set* pertama memiliki nilai MSE dan MAE yang paling besar. Sedangkan, model *set* keempat memiliki nilai *metrics* yang terkecil, yang diakibatkan kesamaan data pembelajaran dan data validasi yang digunakan. Sehingga, wajar bila *set* keempat memiliki nilai *metrics* terendah. Dapat dilihat juga pada tabel di atas bahwa nilai kedua *metrics* pada model *set* kedua dan ketiga memiliki nilai yang mirip dan berada di antara nilai model *set* pertama dan keempat. Keempat model ini kemudian akan disimpan dalam format *.h5 yang mana berisikan struktur jaringan serta nilai dari *weight* dan *bias* dari setiap lapisan model.

4.2.5 Integrasi Model DL dengan X-Plane

Tahapan selanjutnya yaitu mengintegrasikan model berformat *.h5 pada Simulink agar sistem kendali *waypoint following* yang semula berbasis PID dapat digantikan menjadi berbasis DL. Untuk mengintegrasikan X-Plane dengan model yang telah dibuat, maka perlu dilakukan instalasi *Deep Learning Toolbox* pada MATLAB untuk memanfaatkan blok bernama “*predict*” pada Simulink. Pada blok ini, dibutuhkan model DL berformat *.mat, sehingga diperlukan konversi format data yang mulanya berformat *.h5 menjadi *.mat.

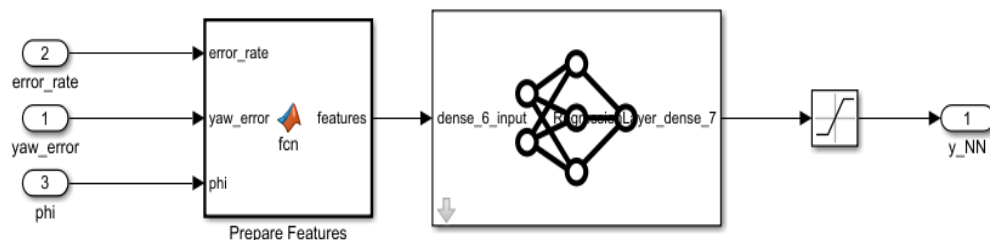
Proses konversi format data ini dapat dilakukan dengan mengunduh terlebih dahulu “*Deep Learning Toolbox Converter for TensorFlow Models*” yang menyediakan fungsi bernama “*importKerasNetwork*”. Dikarenakan model yang dikembangkan pada TensorFlow sudah didesain sedemikian rupa agar cocok dengan *layer* ataupun fungsi aktivasi yang disediakan oleh MATLAB, maka dengan menggunakan fungsi tersebut, model yang semula berformat *.h5 dapat dikonversikan menjadi format *.mat dan menghasilkan hasil prediksi yang serupa dengan model yang telah dikembangkan pada TensorFlow. Berikut ini adalah tampilan pengaturan dari blok “*predict*”.



Gambar 4.13 Pengaturan blok *predict*

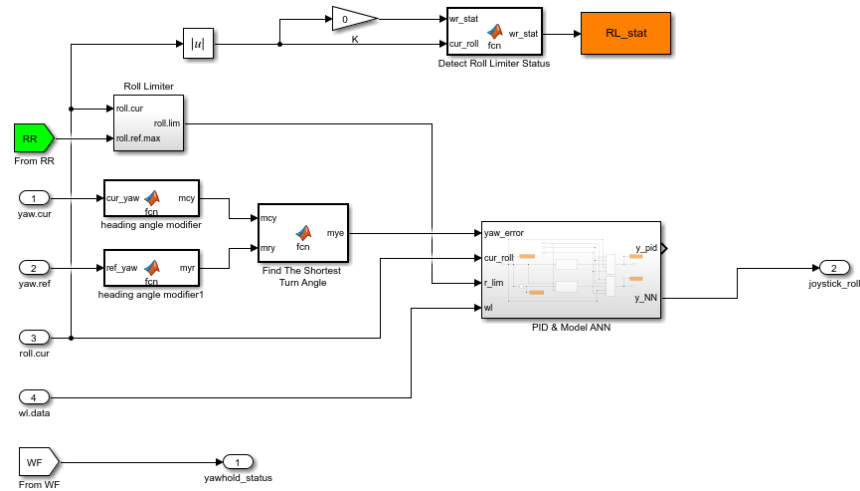
Blok ini menerima masukan berupa *file path* dari model yang ingin digunakan. Pada kasus ini akan dipilih “*Network from MAT-file*” dan letak penyimpanan dari model berformatkan *.mat tersebut dipilih. Nilai *mini-batch size* ditetapkan bernilai 1 dan *outputs* yang dipilih adalah “*predictions*”. Sehingga, setiap ada data masukan dari X-Plane, blok ini akan mengeluarkan hasil prediksi berdasarkan model yang dipilih.

Mengingat bahwa *features* dari setiap model yang telah dikembangkan telah dinormalisasi menggunakan metode *Min-Max Scaler*. Maka, akan dibuat sebuah *block* yang diberi nama “*Prepare Features*” dengan fungsi yaitu untuk melakukan pengskalaan nilai setiap *features* pada *set* data pembelajaran yang berkaitan dengan model yang dipilih menggunakan persamaan 4.1. Berikut ini ditampilkan pengimplementasian model DL pada Simulink.



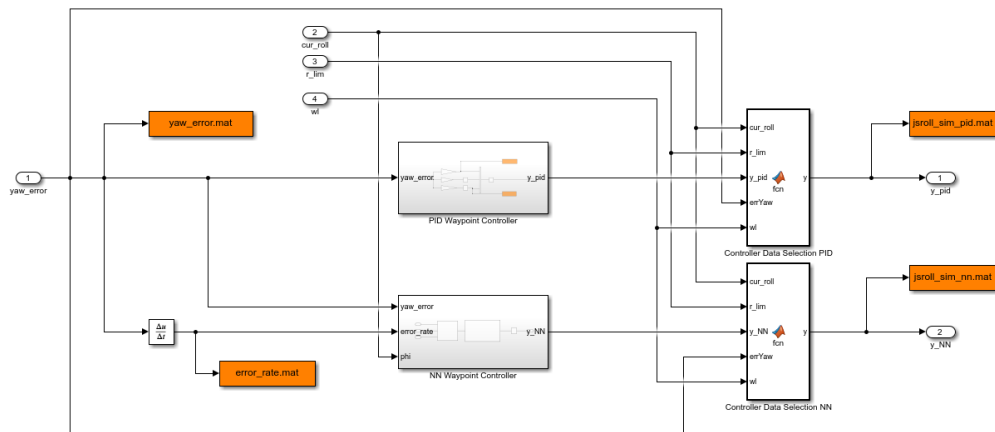
Gambar 4.14 Pengimplementasian model DL pada Simulink

Pengimplementasian diagram Simulink pada Gambar 4.9 dilakukan dengan melakukan sedikit modifikasi pada subsistem “waypoint controller”. Tampilan subsistem yang telah dimodifikasi ditampilkan sebagai berikut.



Gambar 4.15 Modifikasi subsistem *waypoint controller* pada Simulink

Pada diagram di atas, proses persiapan seluruh *features* dilakukan terlebih dahulu sebelum dikirimkan menuju *block* bernama “PID & Model DL” yang nantinya akan meneruskan salah satu perintah kendali menuju subsistem kendali. Diagram pada *block* ini dipaparkan sebagai berikut.



Gambar 4.16 Diagram pada *block* "PID & Model DL"

Pada subsistem yang tertampil pada Gambar 4.11, dapat dilihat bahwa subsistem ini terdiri atas dua buah subsistem yaitu subsistem yang bekerja berdasarkan *PID controller* dan subsistem yang memberikan perintah kendali berdasarkan model DL yang menerima 3 buah masukan yaitu sudut *yaw error*, *yaw error rate*, dan sudut *roll*.

Melalui bab ini, telah dibahas mengenai pra-pemrosesan data yang dilakukan pada hasil rekonstruksi data misi penerbangan serta pengkarakterisasian *set* data pembelajaran, sehingga didapati 4 buah model *deep learning* yang telah dilatih berdasarkan hasil dari iterasi *Bayesian Optimization*. Setiap model tersebut sudah diubah agar dapat dibaca oleh Simulink, sehingga proses simulasi dan evaluasi dari performa model *deep learning* untuk melaksanakan kendali lintas terbang dapat terlaksana dan dibahas pada bab selanjutnya.

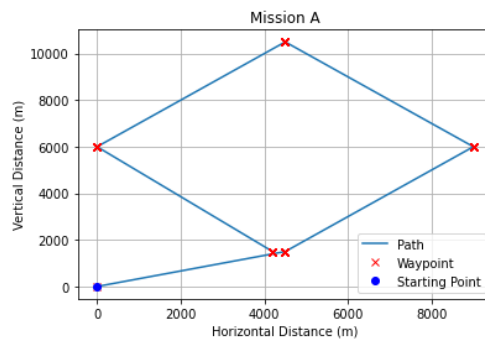
BAB V

HASIL SIMULASI DAN ANALISIS

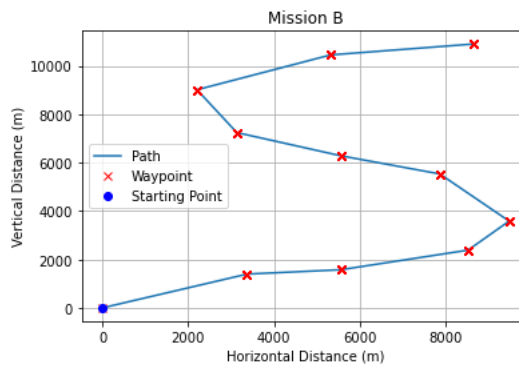
Pada bab ini, akan dibahas mengenai skenario simulasi misi lintas terbang yang akan dilaksanakan, beserta analisis dari hasil simulasi yang didapatkan pada setiap misi terbangnya.

5.1 Skenario Simulasi

Dalam rangka mengevaluasi dan membandingkan performa kontroler dari model DL dengan model PID. Akan dilakukan simulasi menggunakan 2 buah misi terbang, yaitu misi terbang A dan misi terbang B. Pada masing-masing misi, akan dilakukan simulasi sebanyak dua kali untuk melihat konsistensi dari kontroler. Proses simulasi dilakukan dengan memastikan kondisi awal dari setiap simulasi adalah serupa dengan menggunakan fitur “*Save Situation*” pada X-Plane dan digunakan konfigurasi persentase *throttle* yang serupa yaitu pada nilai 47%. Berikut ini ditunjukkan lintasan dari misi terbang A dan misi terbang B.



Gambar 5.1 Bentuk lintasan misi terbang A



Gambar 5.2 Bentuk lintasan misi terbang B

Misi terbang A terdiri dari 5 buah *waypoints* yang mengharuskan pesawat melakukan manuver berbelok ke arah kiri terus menerus. Sedangkan misi terbang B terdiri dari 10 buah *waypoints* dengan lintasan yang melibatkan lintasan garis lurus dan manuver berbelok ke arah kiri dan kanan dengan perubahan sudut yang ekstrem.

Pada proses simulasi menggunakan metode PID baik dalam menyelesaikan misi terbang A dan misi terbang B, digunakan nilai konstanta yang ditampilkan pada Tabel 5.1 di bawah ini. Keempat model DL akan menggunakan nilai konstanta tersebut terkecuali pada sistem kendali *yaw hold*, dikarenakan sistem kendali ini telah disubstitusi menggunakan model DL.

Tabel 5.1 Nilai konstanta setiap sistem kendali pada kedua misi uji

Konstanta	Sistem Kendali		
	<i>Yaw Hold</i>	<i>Roll Limiter</i>	<i>Wing Leveler</i>
<i>Proportional</i>	-0.0035	-0.09	-0.003
<i>Integral</i>	0	0	0
<i>Derivative</i>	-0.0308	-0.0078	-0.0007

Dari susunan *waypoints* yang terdiri dari koordinat X dan Y, dapat dicari sudut terkecil yang dibentuk dari dua buah *waypoint* terhadap sumbu horizontal menggunakan persamaan trigonometri berikut.

$$\theta = \tan^{-1} \left(\frac{y}{x} \right) \quad (5.1)$$

Dari nilai sudut θ , dapat dicari besarnya perubahan sudut *yaw* yang diperlukan pesawat dalam rangka mencapai garis lintasan selanjutnya. Kedua nilai tersebut baik pada misi terbang A maupun misi terbang B ditunjukkan melalui Tabel 5.2 dan Tabel 5.3.

Maksud perubahan sudut pada *waypoint* ke-n pada kedua tabel tersebut yaitu perubahan sudut yang terjadi untuk mencapai *waypoint*_(n) dari *waypoint*_(n-1). Perubahan sudut dengan nilai negatif bermakna pesawat perlu melakukan manuver berbelok dengan arah berlawanan arah jarum jam.

Dalam melakukan evaluasi performa model DL dalam melaksanakan misi lintas terbang, digunakan metrik berupa kedekatan pesawat dengan *waypoint* yang dituju yang merupakan metrik utama dalam misi lintas terbang. Ditambahkan juga metrik tambahan yaitu rata-rata jarak lintasan pesawat terhadap lintasan misi sepanjang misi berlangsung atau yang dapat disebut sebagai *cross track distance* (CTD) sebagai tolak ukur tambahan, meskipun nilai CTD yang semimum mungkin merupakan tujuan dari *fly-by waypoints*. Untuk menghitung kedekatan pesawat terhadap *waypoint*, diterapkan algoritme berikut ini.

FUNCTION 7: MINIMUM DISTANCE TO WAYPOINT

Input: df: pandas dataframe
Output: List consisting of minimum distance for each waypoint

```

1  DEFINE FUNCTION minimum_distance(df)
2      x_wp_List, y_wp_List  $\leftarrow$  List consisting of unique coordinates
        x and y
3      X_List, Y_List  $\leftarrow$  List containing aircraft x and y coordinates
4      Storage  $\leftarrow$  List consisting blank List with amount of number of
        waypoints
5      min_dist  $\leftarrow$  empty List to store minimum distance of all
        waypoints
6      Radius  $\leftarrow$  450
7      FOR index IN RANGE (number of waypoints)
8          FOR value IN RANGE (Length of df)
9              x_wp  $\leftarrow$  x_wp_List AT index
10             y_wp  $\leftarrow$  y_wp_List AT index
11             X  $\leftarrow$  X_List AT value
12             Y  $\leftarrow$  Y_List AT value
13             distance  $\leftarrow$  SQUARE ROOT  $((x\_wp - X)^2 + (y\_wp - Y)^2)$ 
14             IF distance  $\leq$  radius:
15                 APPEND distance to Storage AT index
16             END IF
17             FOR section IN RANGE (Length of storage)
18                 APPEND minimum value of storage AT section to
                    min_dist
19             END FOR
20         END FOR
21         RETURN min_dist
22 END FUNCTION

```

Gambar 5.3 Pseudocode minimum distance to waypoint

Sedangkan, untuk menghitung nilai CTD, pertama-tama akan dicari terlebih dahulu nilai gradien (m) dan *intercept* (c) dari setiap garis yang menyusun lintasan misi terbang. Kemudian nilai dari setiap koordinat x dan y dari garis yang terkait akan dikalkulasikan melalui persamaan berikut yang akan menjumlahkan keseluruhan nilai CTD dari misi yang dijalankan.

$$CTD = \sum_{i=1}^n \sum_{j=1}^n \frac{|-m_{(i)} * x_{(j)} + -c_{(i)} + y_{(j)}|}{\sqrt{(-m_{(i)})^2 + 1^2}} \quad (5.1)$$

Tabel 5.2 Detil koordinat *waypoint* dan perubahan sudut lintasan pada misi terbang A

<i>Waypoint</i>	0	1	2	3	4	5
X (m)	0	4,500	9,000	4,500	0	4,200
Y (m)	0	1,500	6,000	10,500	6,000	1,500
Sudut Terkecil (°)	0	18.43	45	-45	45	-46.97
Perubahan Sudut (°)	-	-18.43	-26.57	-90	-90	-88.3

Tabel 5.3 Detil koordinat *waypoint* dan perubahan sudut lintasan pada misi terbang B

<i>Waypoint</i>	0	1	2	3	4	5	6	7	8	9	10
X (m)	0	3,375	5,565	8,535	9,495	7,875	5,565	3,135	2,205	5,325	8,655
Y (m)	0	1,395	1,575	2,385	3,585	5,535	6,285	7,245	9,015	10,455	10,905
Sudut Terkecil (°)	0	22.46	4.7	15.26	51.34	-50.28	-17.99	-21.56	-62.28	24.78	7.7
Perubahan Sudut (°)	-	-22.46	17.76	-10.56	-36.08	-78.38	-32.29	3.57	40.72	92.94	17.08

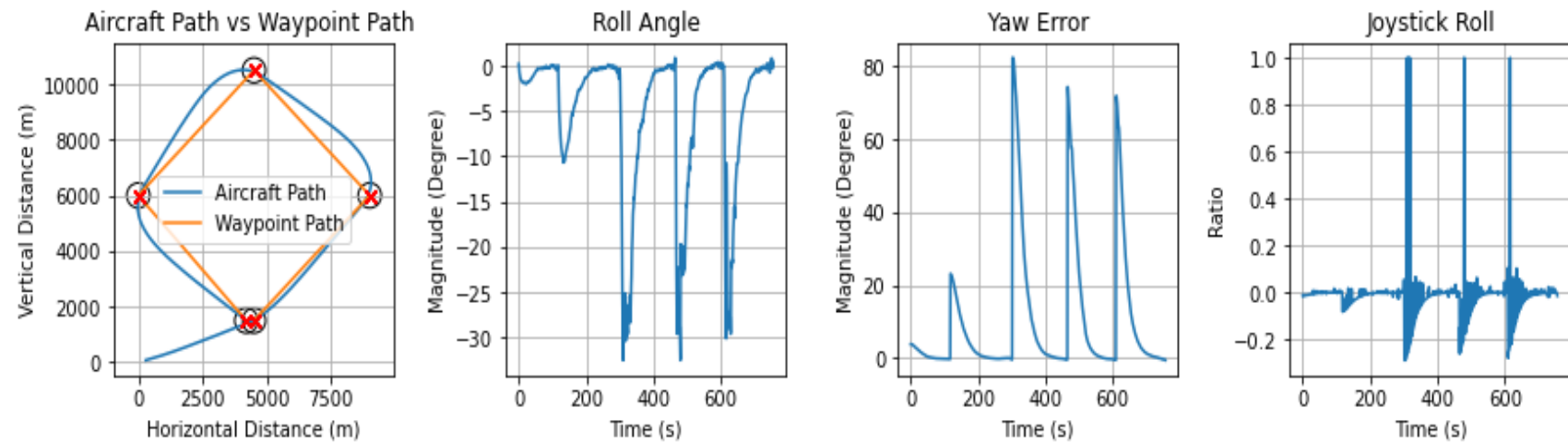
5.2 Hasil Simulasi Misi Terbang A

Misi terbang A disimulasikan sebanyak dua kali untuk masing-masing metode sistem kendali lintas terbang. Nilai dari metrik kedekatan dengan setiap *waypoint* pada misi terbang A untuk kelima buah metode yang digunakan ditampilkan melalui Tabel 5.4 hingga Tabel 5.8. Sedangkan, nilai metrik *cross track distance* ditampilkan melalui Tabel 5.9 hingga Tabel 5.13.

Berikut ini ditampilkan grafik lintasan pesawat, sudut *roll* dan *error* dari sudut *yaw* pesawat saat melakukan misi terbang A.

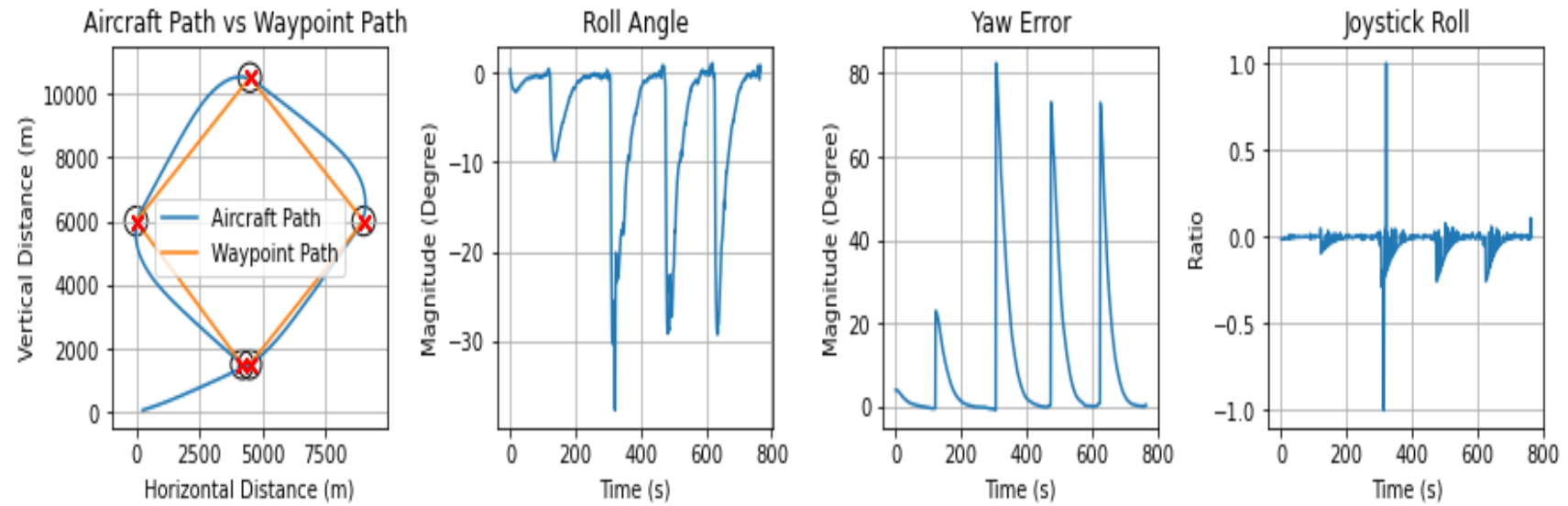
1. PID Controller

- Percobaan 1



Gambar 5.4 Grafik hasil simulasi metode PID percobaan 1 pada misi A

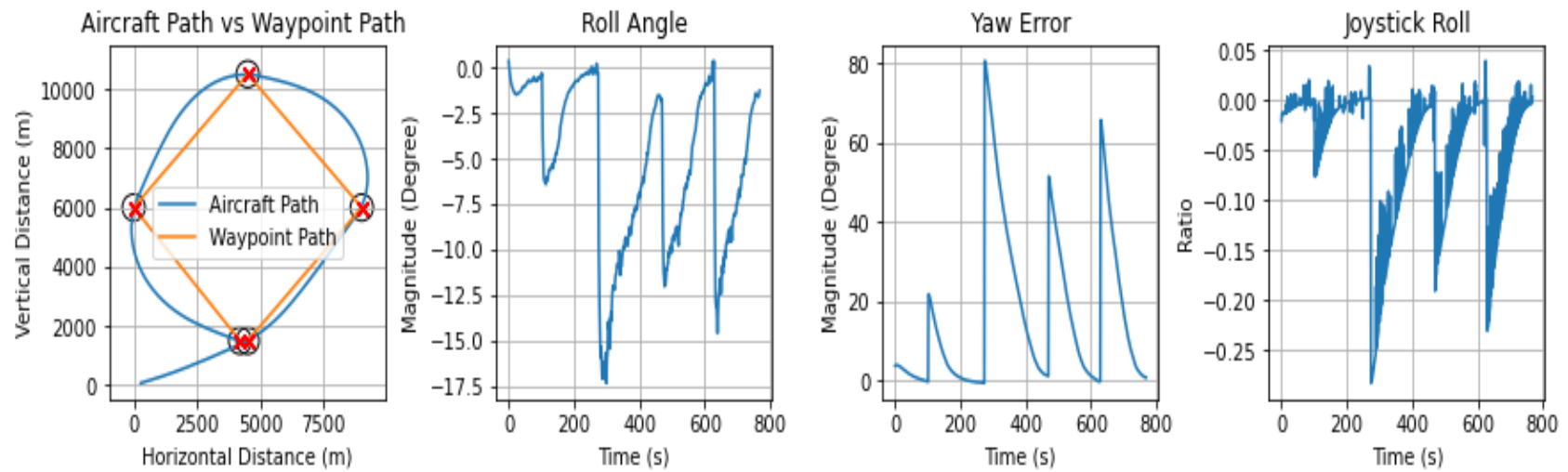
- Percobaan 2



Gambar 5.5 Grafik hasil simulasi metode PID percobaan 2 pada misi A

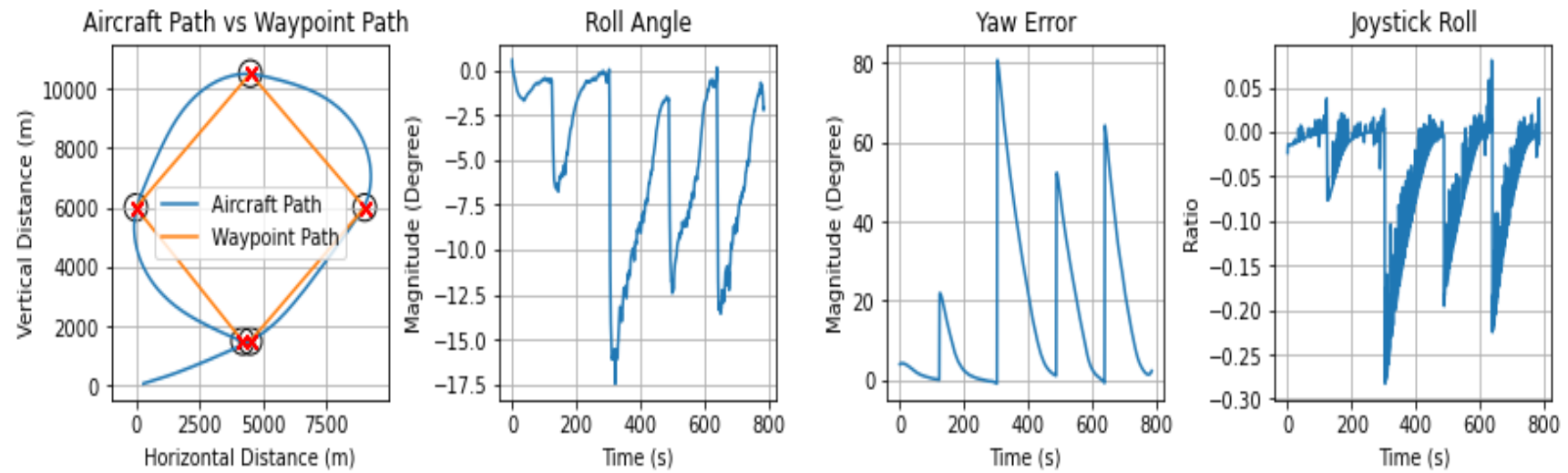
2. Model DL *Set 1*

- Percobaan 1



Gambar 5.6 Grafik hasil simulasi model DL *set 1* percobaan 1 pada misi A

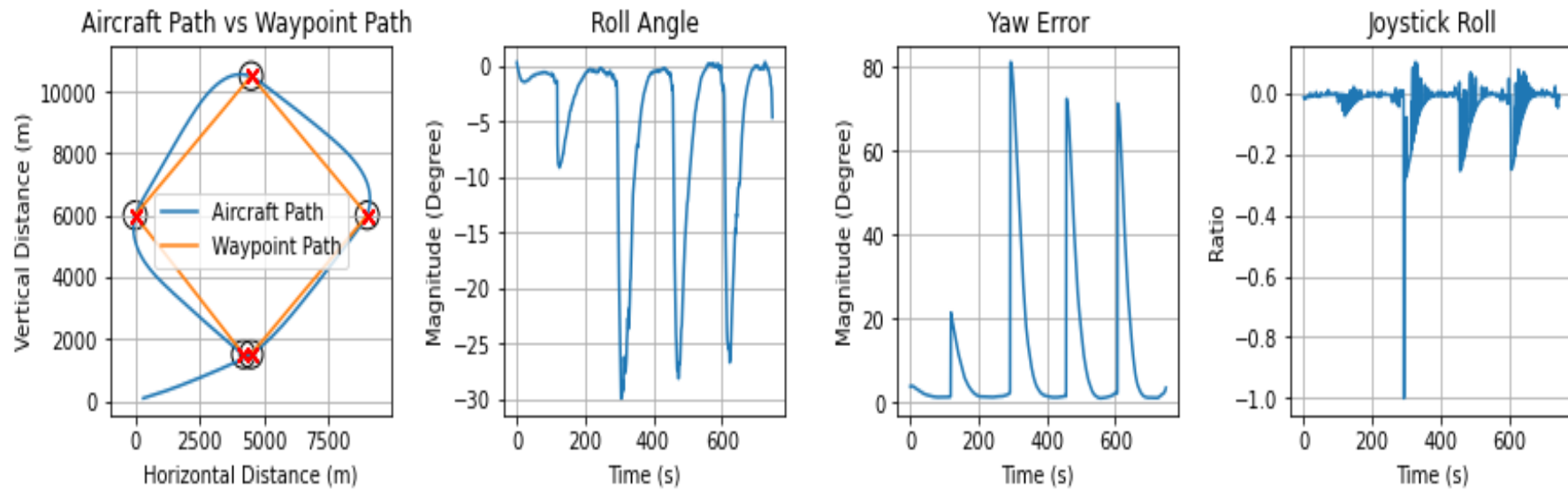
- Percobaan 2



Gambar 5.7 Grafik hasil simulasi model DL *set* 1 percobaan 2 pada misi A

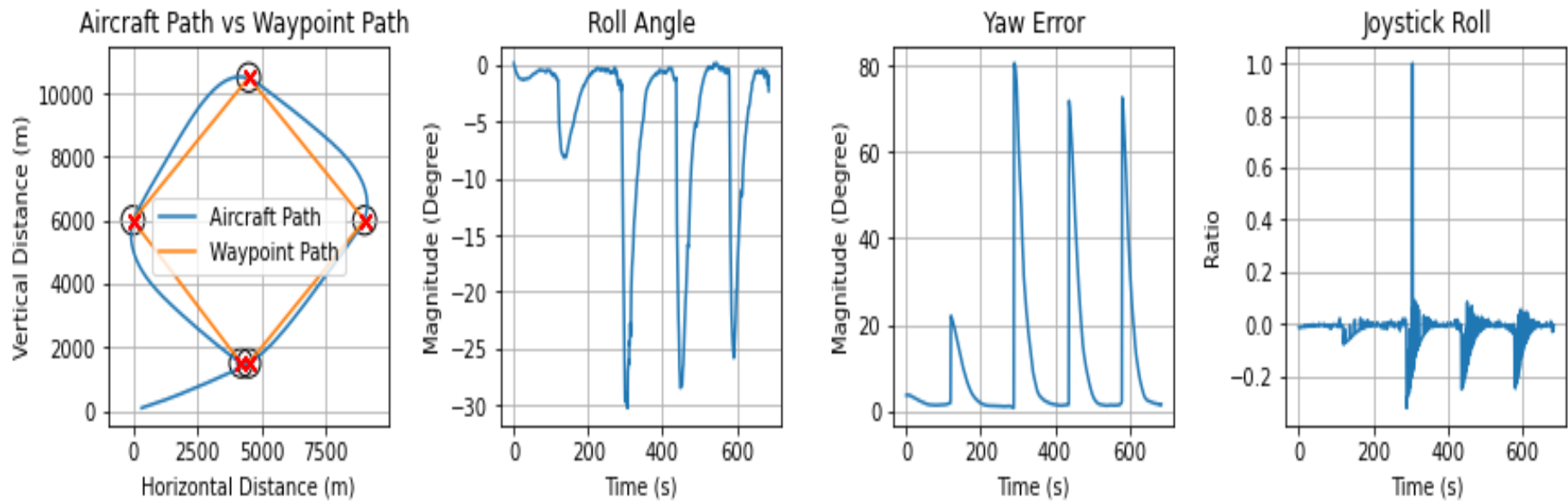
3. Model DL *Set 2*

- Percobaan 1



Gambar 5.8 Grafik hasil simulasi model DL *set 2* percobaan 1 pada misi A

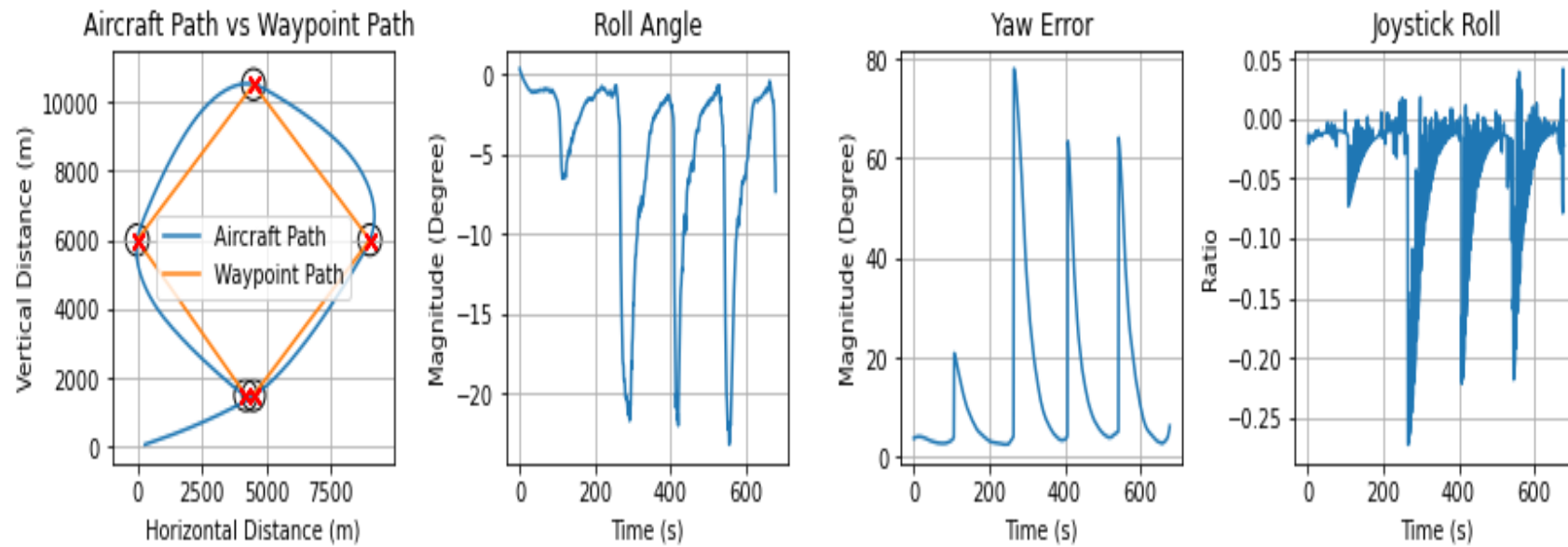
- Percobaan 2



Gambar 5.9 Grafik hasil simulasi model DL *set* 2 percobaan 2 pada misi A

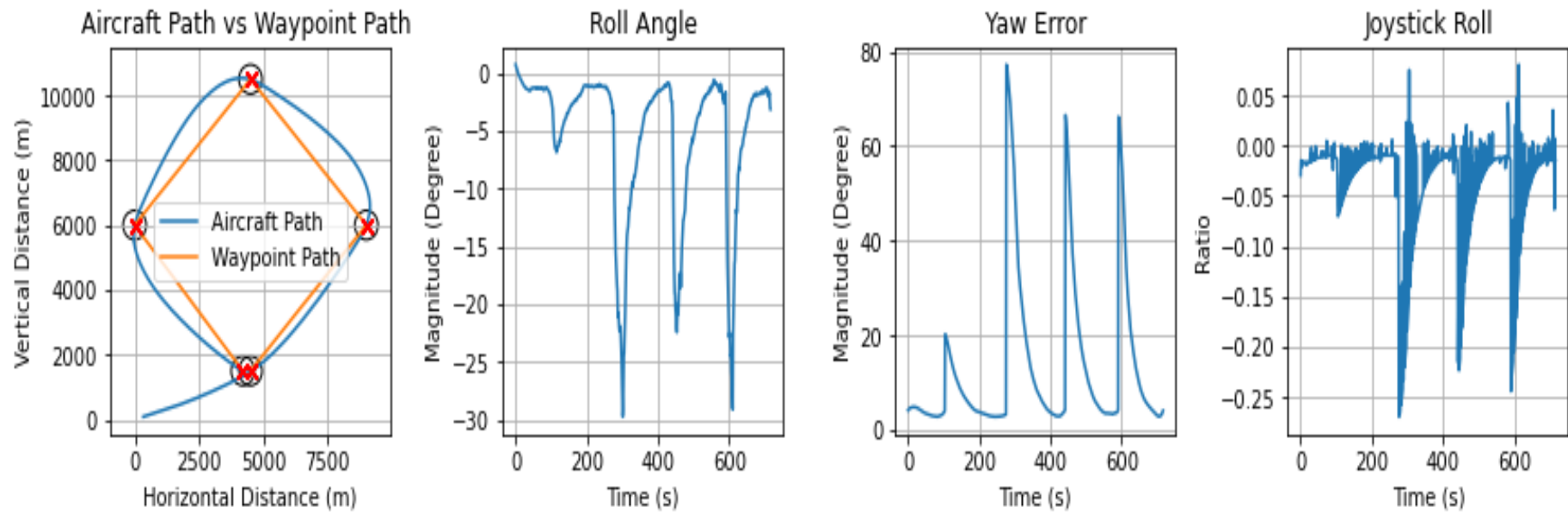
4. Model DL *Set 3*

- Percobaan 1



Gambar 5.10 Grafik hasil simulasi model DL *set 3* percobaan 1 pada misi A

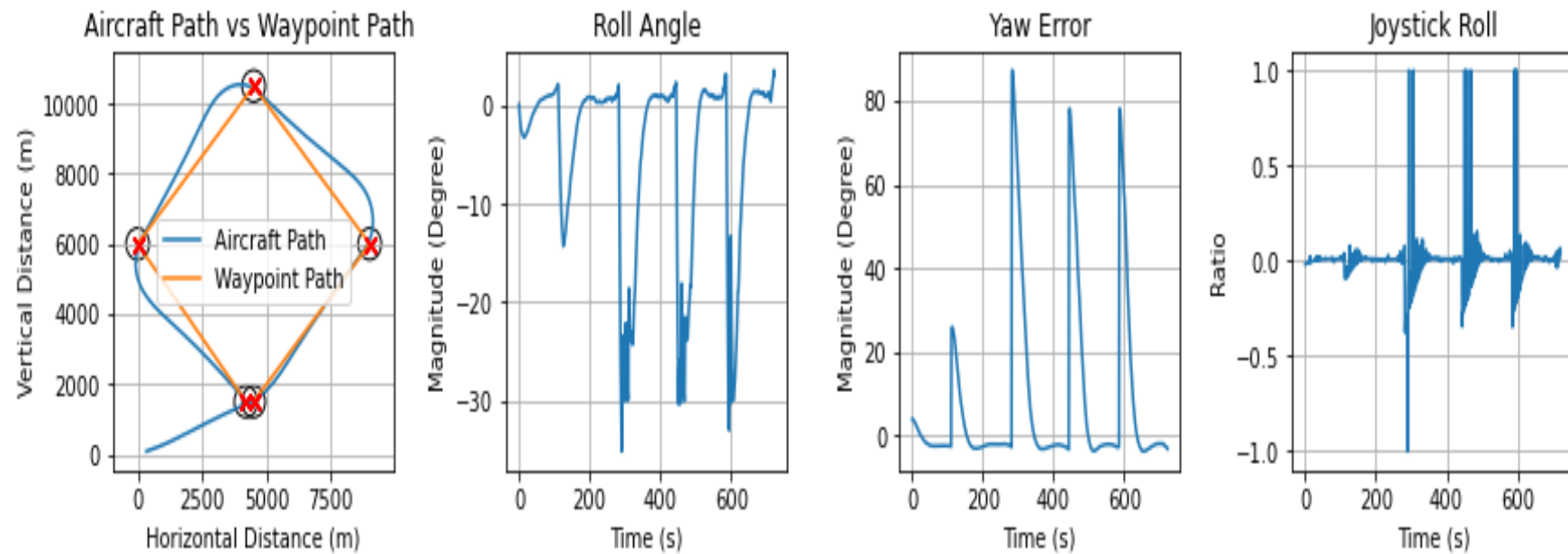
- Percobaan 2



Gambar 5.11 Grafik hasil simulasi model DL set 3 percobaan 2 pada misi A

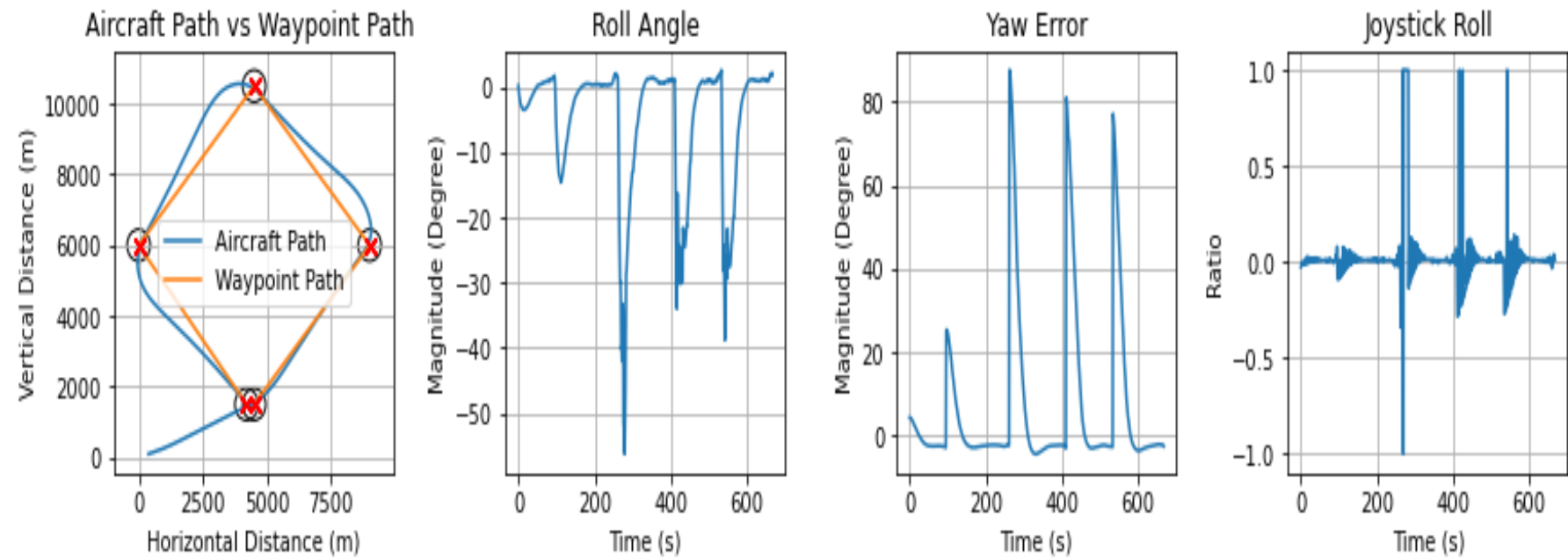
5. Model DL *Set 4*

- Percobaan 1



Gambar 5.12 Grafik hasil simulasi model DL *set 4* percobaan 1 pada misi A

- Percobaan 2



Gambar 5.13 Grafik hasil simulasi model DL set 4 percobaan 2 pada misi A

Tabel 5.4 Detil jarak minimum pesawat terhadap *waypoint* metode PID pada misi terbang A

Minimum Distance to <i>Waypoint</i> (m)						
PID	<i>Waypoint</i> Number					AVG ALL WP
	1	2	3	4	5	
TRIAL 1	20.71	57.7	48.88	49.36	10.89	37.51 ± 18.27
TRIAL 2	20.1	51.05	36.45	38.06	13.58	31.85 ± 13.42
AVG	20.41 ± 20.26	54.38 ± 52.72	42.67 ± 39.56	43.71 ± 40.89	12.24 ± 12.91	34.68 ± 15.85

Tabel 5.5 Detil jarak minimum pesawat terhadap *waypoint* model *set 1* pada misi terbang A

Minimum Distance to <i>Waypoint</i> (m)						
MODEL SET 1	<i>Waypoint</i> Number					AVG ALL WP
	1	2	3	4	5	
TRIAL 1	32.26	32.1	13.24	23.67	9.6	22.17 ± 9.39
TRIAL 2	8.28	35.08	12.63	27.82	6.88	18.14 ± 11.27
AVG	20.27 ± 14.28	33.59 ± 34.34	12.94 ± 12.79	25.75 ± 26.79	8.24 ± 7.56	20.16 ± 10.33

Tabel 5.6 Detil jarak minimum pesawat terhadap *waypoint* model *set 2* pada misi terbang A

Minimum Distance to <i>Waypoint</i> (m)						
MODEL SET 2	<i>Waypoint</i> Number					AVG ALL WP
	1	2	3	4	5	
TRIAL 1	12.57	17.5	25.58	13.95	23.56	22.17 ± 9.39
TRIAL 2	10.26	38.6	32.34	28.37	6.75	18.14 ± 11.27
AVG	20.27 ± 14.28	33.59 ± 34.34	12.94 ± 12.79	25.75 ± 26.79	8.24 ± 7.56	20.16 ± 10.33

Tabel 5.7 Detil jarak minimum pesawat terhadap *waypoint* model *set 3* pada misi terbang A

Minimum Distance to <i>Waypoint</i> (m)						
MODEL SET 3	<i>Waypoint</i> Number					AVG ALL WP
	1	2	3	4	5	
TRIAL 1	16.79	23.27	28.98	10.7	14.73	18.89 ± 6.48
TRIAL 2	19.83	5.73	12.74	11.49	32.74	16.51 ± 9.27
AVG	18.31 ± 19.07	14.5 ± 10.12	20.86 ± 16.8	11.1 ± 11.3	23.74 ± 28.24	17.7 ± 7.88

Tabel 5.8 Detil jarak minimum pesawat terhadap *waypoint* model *set 3* pada misi terbang A

Minimum Distance to <i>Waypoint</i> (m)						
MODEL SET 4	<i>Waypoint</i> Number					AVG ALL WP
	1	2	3	4	5	
TRIAL 1	34.24	55.28	62.83	53.57	11.25	43.43 ± 18.65
TRIAL 2	36.31	59.7	69.15	53.46	11.3	45.98 ± 20.38
AVG	35.28 ± 35.8	57.49 ± 58.6	65.99 ± 67.57	53.52 ± 53.49	11.28 ± 11.29	44.71 ± 19.52

Tabel 5.9 Detil nilai *cross track distance* setiap *section* metode PID pada misi terbang A

Cross Track Distance (m)						
PID	Section					AVG ALL SECTION
	1	2	3	4	5	
TRIAL 1	45.76	123.71	518.94	470.96	459.61	323.8 ± 197.75
TRIAL 2	50.42	140.36	589.59	484.83	481.02	349.24 ± 212.81
AVG	48.09 ± 2.33	132.04 ± 8.33	554.27 ± 35.33	477.9 ± 6.94	470.32 ± 10.71	336.52 ± 205.42

Tabel 5.10 Detil nilai *cross track distance* setiap *section* model *set 1* pada misi terbang A

Cross Track Distance (m)						
MODEL SET 1	Section					AVG ALL SECTION
	1	2	3	4	5	
TRIAL 1	65.48	202	1068.07	612.55	855.61	560.74 ± 659.79
TRIAL 2	67.72	193.68	1069.49	639.85	812.97	556.74 ± 654.55
AVG	66.6 ± 1.12	197.84 ± 4.16	1068.78 ± 0.71	626.2 ± 13.65	834.29 ± 21.32	558.74 ± 657.18

Tabel 5.11 Detil nilai *cross track distance* setiap *section* model *set 2* pada misi terbang A

Cross Track Distance (m)						
MODEL SET 2	Section					AVG ALL SECTION
	1	2	3	4	5	
TRIAL 1	71.04	143.66	545.16	507.66	482.59	350.02 ± 200.46
TRIAL 2	66.6	161.66	568.65	456.05	552.78	361.15 ± 207.53
AVG	68.82 ± 2.22	152.66 ± 9	556.91 ± 11.75	481.86 ± 25.81	517.69 ± 35.1	355.59 ± 204.03

Tabel 5.12 Detil nilai *cross track distance* setiap *section* model *set 3* pada misi terbang A

Cross Track Distance (m)						
MODEL SET 3	Section					AVG ALL SECTION
	1	2	3	4	5	
TRIAL 1	89.42	229.87	777.96	599.79	529.23	445.25 ± 250.85
TRIAL 2	101.34	224.25	697.23	597.69	524.62	429.03 ± 227.52
AVG	95.38 ± 5.96	227.06 ± 2.81	737.6 ± 40.37	598.74 ± 1.05	526.93 ± 2.31	437.14 ± 239.47

Tabel 5.13 Detil nilai *cross track distance* setiap *section* model *set 4* pada misi terbang A

Cross Track Distance (m)						
MODEL SET 4	Section					AVG ALL SECTION
	1	2	3	4	5	
TRIAL 1	30.36	67.96	513.33	409.41	413.85	286.98 ± 198.06
TRIAL 2	34.96	62.86	362.35	444.51	374.14	255.76 ± 171.44
AVG	32.66 ± 2.3	65.41 ± 2.55	437.84 ± 75.49	426.96 ± 17.55	394 ± 19.86	271.37 ± 185.23

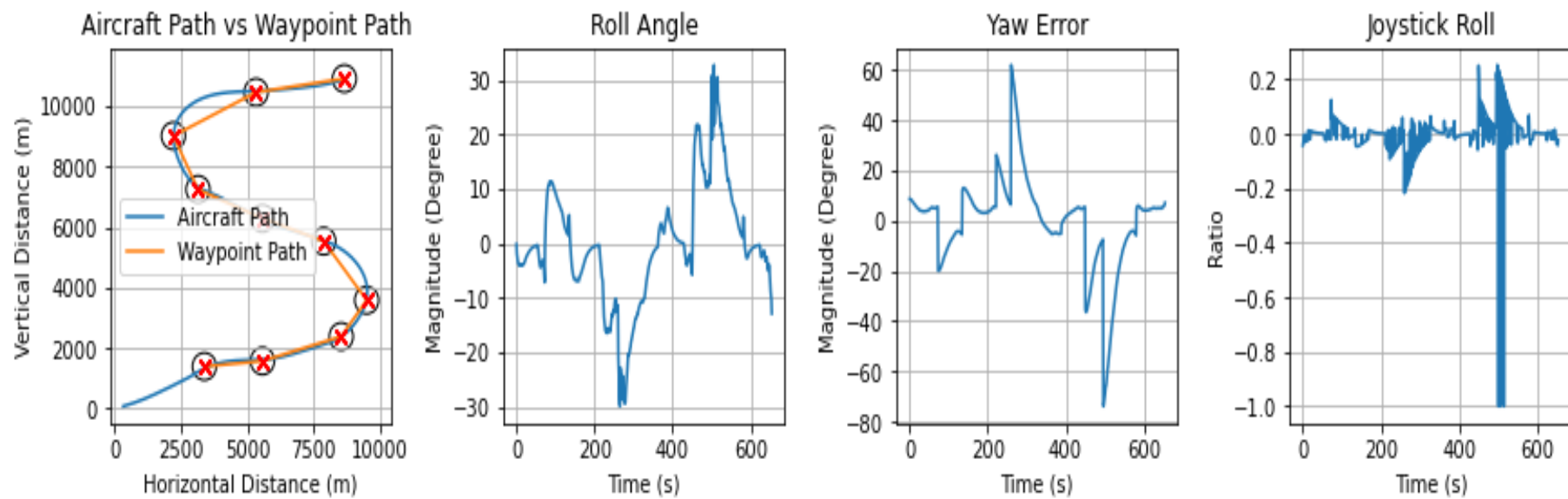
5.3 Hasil Simulasi Misi Terbang B

Nilai dari metrik kedekatan dengan setiap *waypoint* pada misi terbang B untuk metode PID dan model yang dilatih menggunakan *set* data pertama hingga keempat ditampilkan melalui Tabel 5.14 hingga Tabel 5.18. Sedangkan, nilai metrik *cross track distance* untuk setiap *waypoint* dari semua metode yang digunakan ditampilkan melalui Tabel 5.19 hingga Tabel 5.23.

Berikut ini ditampilkan grafik lintasan pesawat, sudut *roll* dan *error* dari sudut *yaw* pesawat saat melakukan misi terbang A.

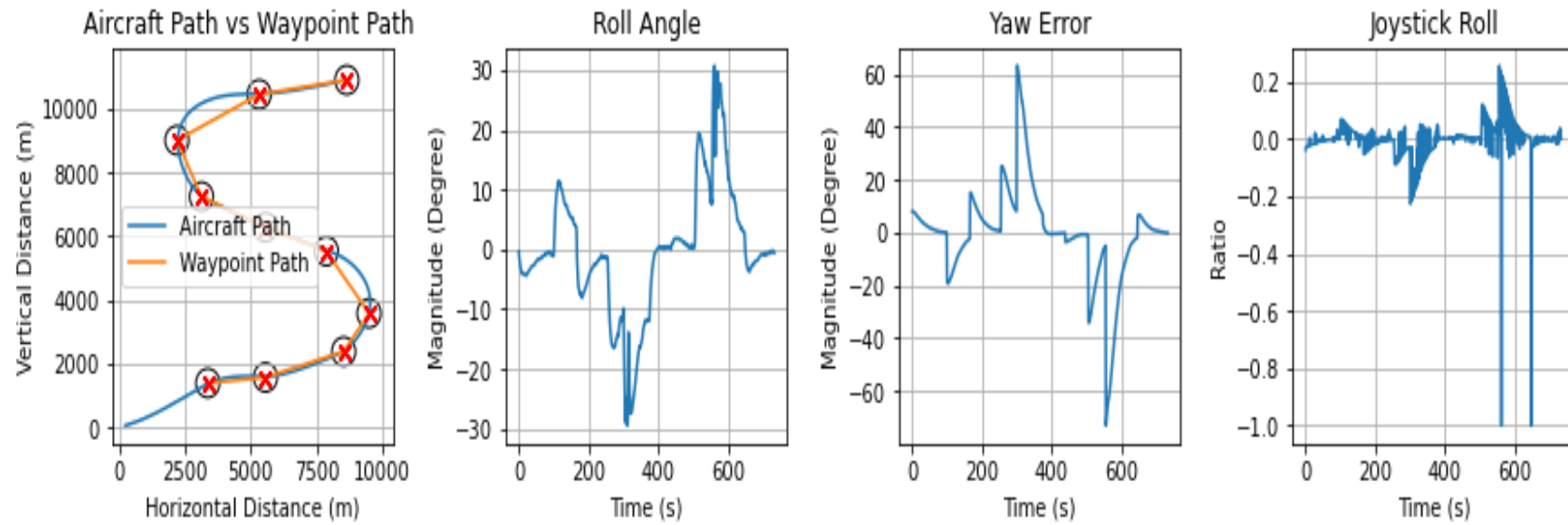
1. PID Controller

- Percobaan 1



Gambar 5.14 Grafik hasil simulasi metode PID percobaan 1 pada misi B

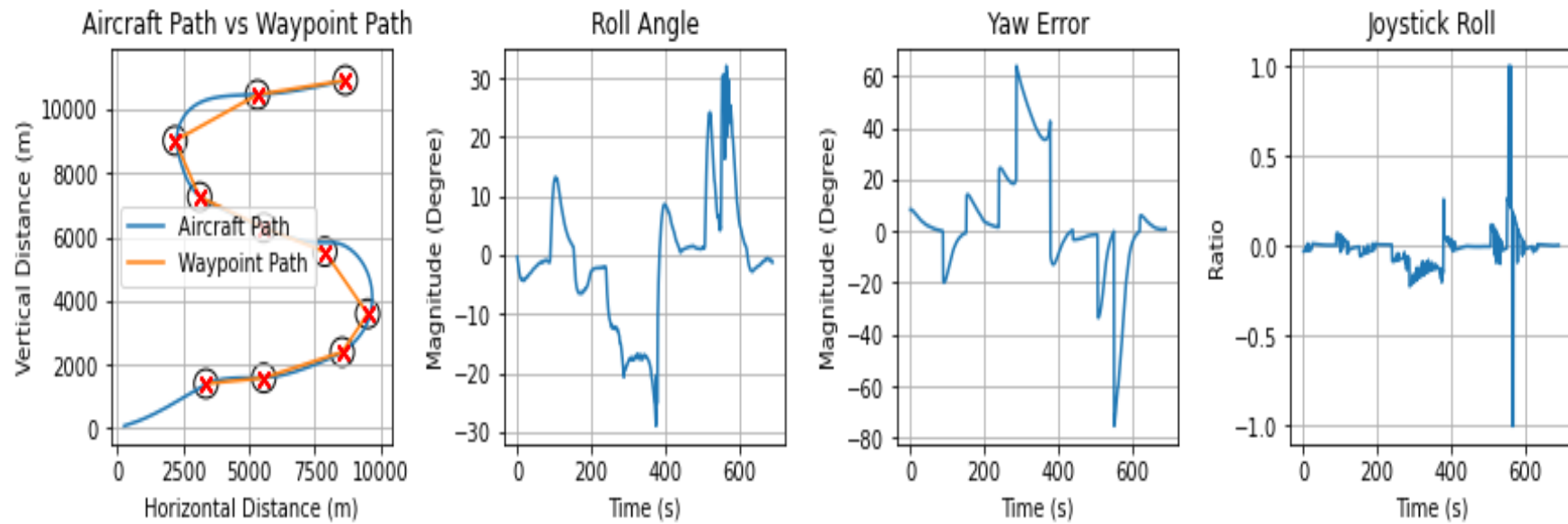
- Percobaan 2



Gambar 5.15 Grafik hasil simulasi metode PID percobaan 2 pada misi B

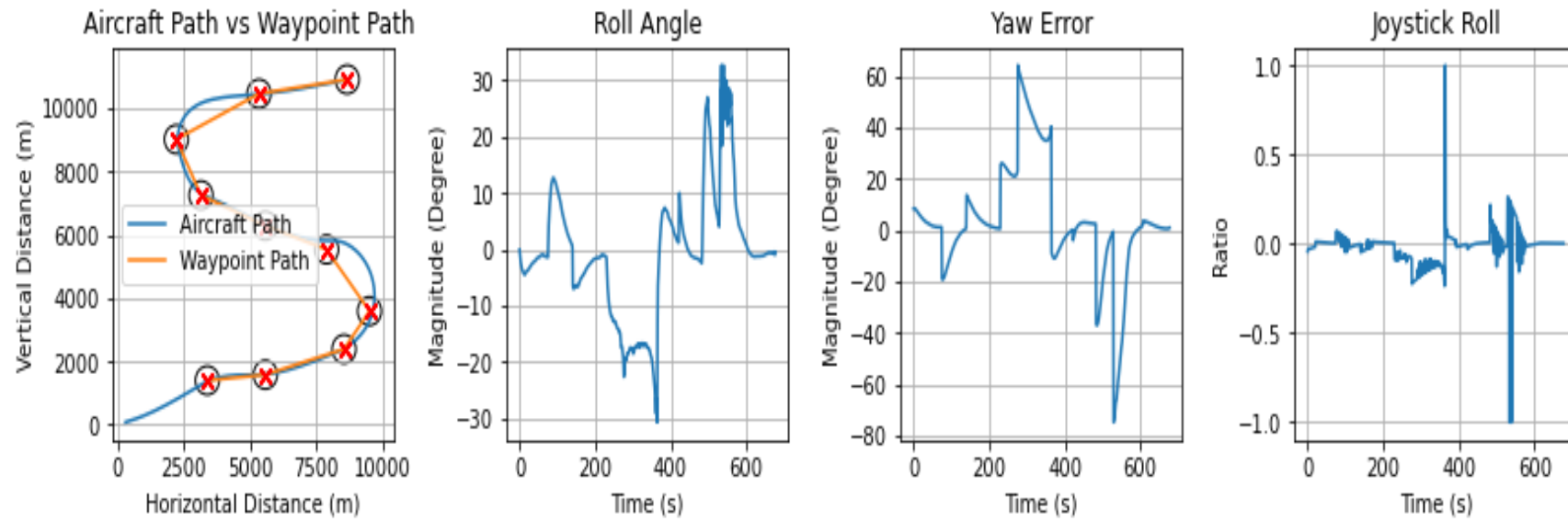
2. Model DL *Set 1*

- Percobaan 1



Gambar 5.16 Grafik hasil simulasi model DL *set 1* percobaan 1 pada misi B

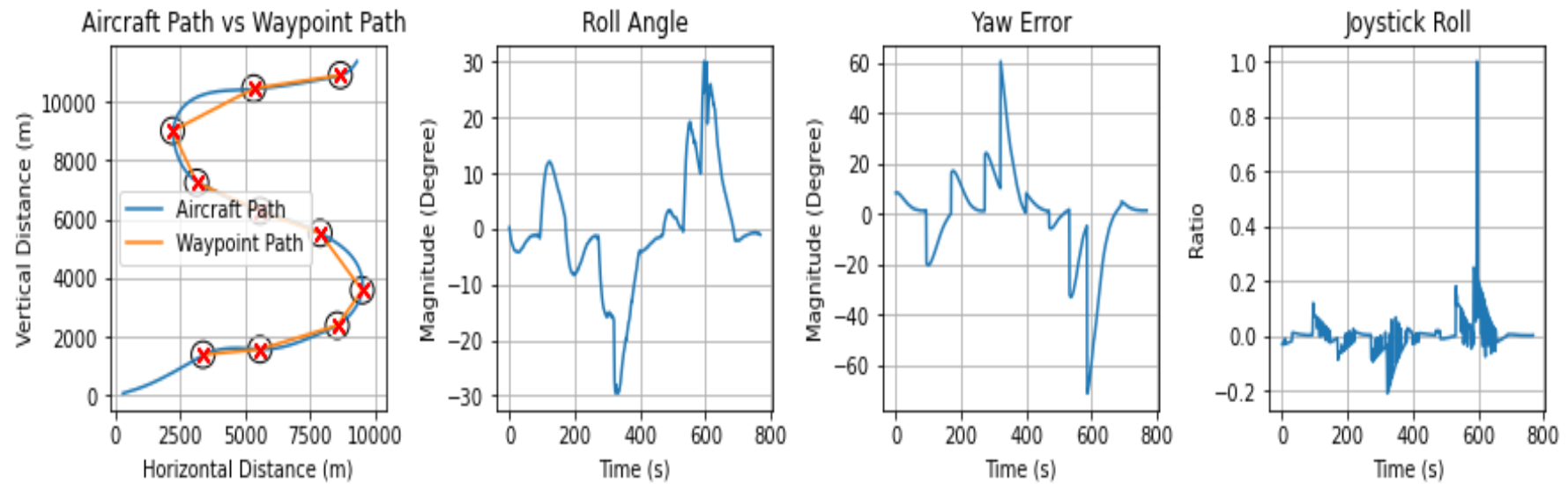
- Percobaan 2



Gambar 5.17 Grafik hasil simulasi model DL *set 1* percobaan 2 pada misi B

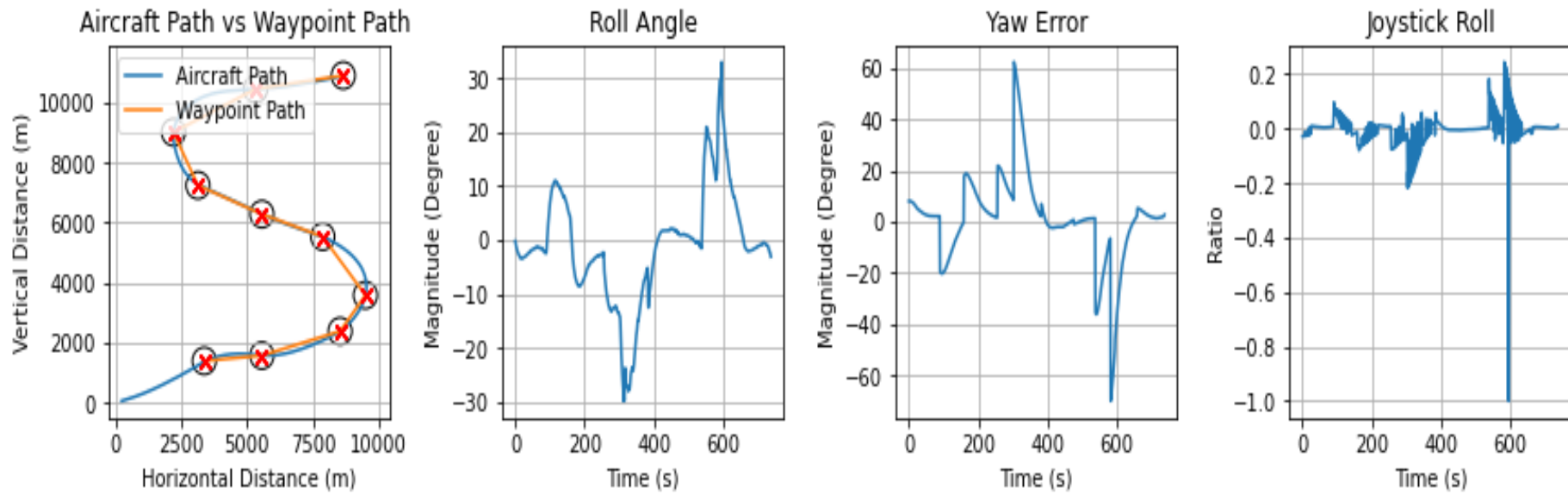
3. Model DL *Set 2*

- Percobaan 1



Gambar 5.18 Grafik hasil simulasi model DL *set 2* percobaan 1 pada misi B

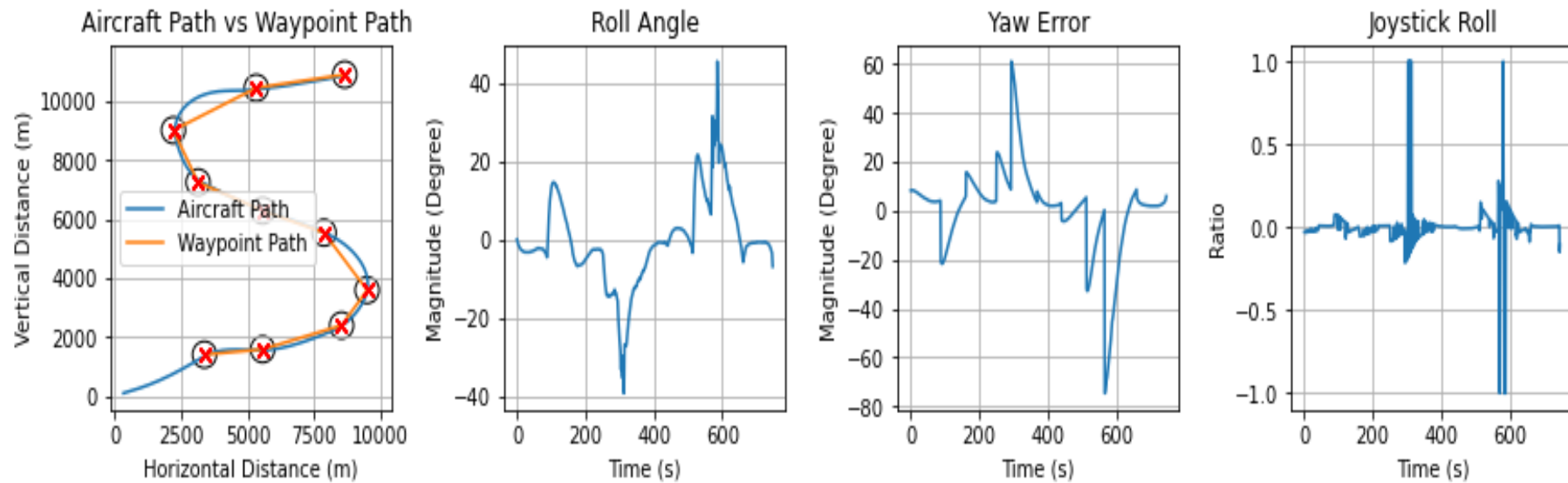
- Percobaan 2



Gambar 5.19 Grafik hasil simulasi model DL *set 2* percobaan 2 pada misi B

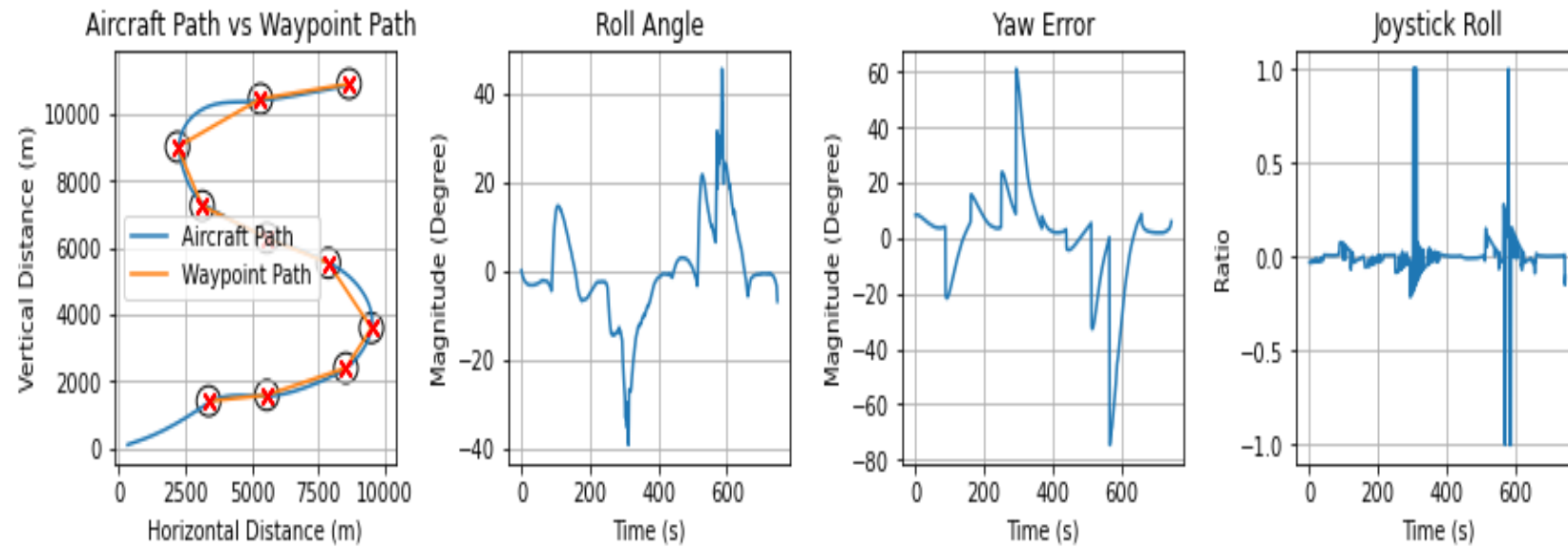
4. Model DL *Set 3*

- Percobaan 1



Gambar 5.20 Grafik hasil simulasi model DL *set 3* percobaan 1 pada misi B

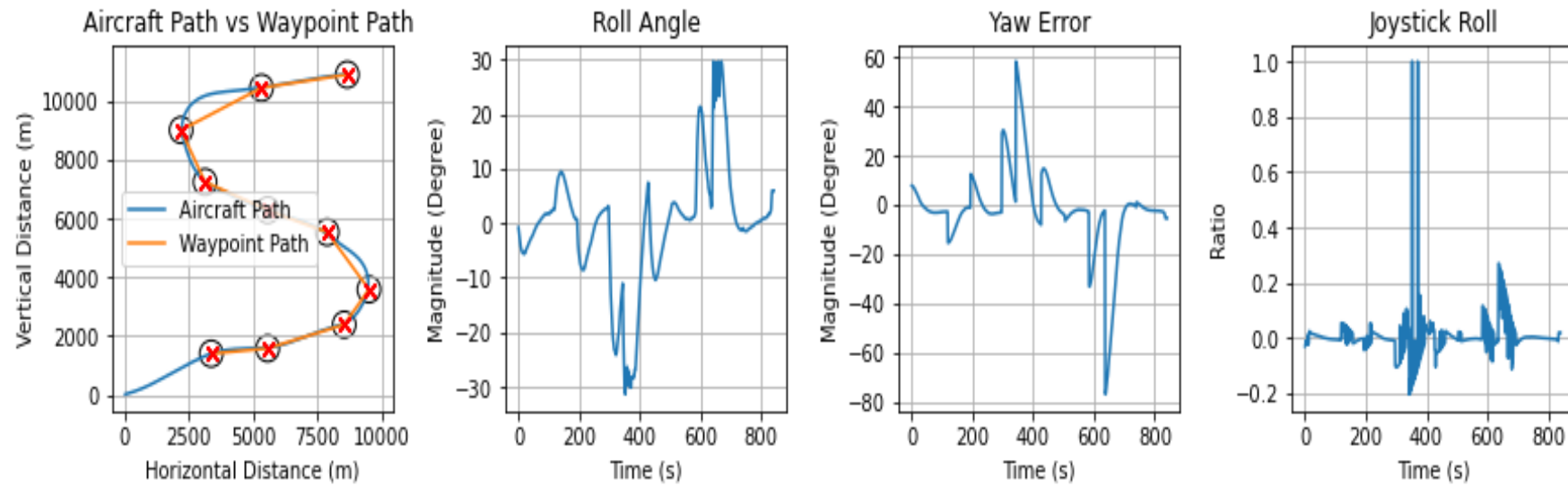
- Percobaan 2



Gambar 5.21 Grafik hasil simulasi model DL *set 3* percobaan 2 pada misi B

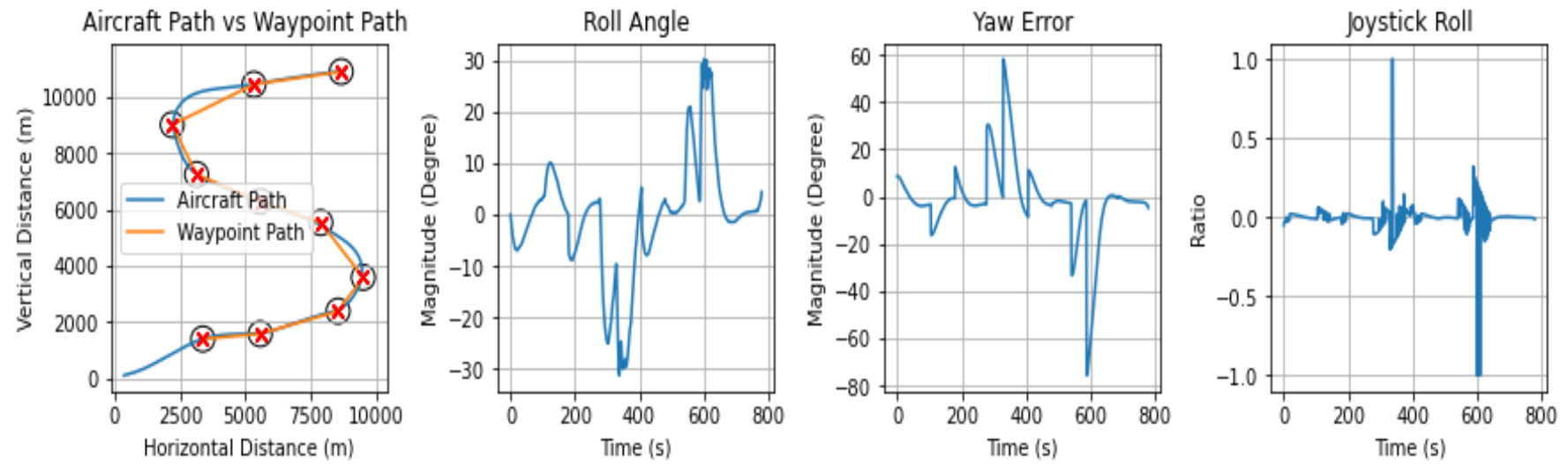
5. Model DL *Set* 4

- Percobaan 1



Gambar 5.22 Grafik hasil simulasi model DL *set* 4 percobaan 1 pada misi B

- Percobaan 2



Gambar 5.23 Grafik hasil simulasi model DL *set* 4 percobaan 2 pada misi

Tabel 5.14 Detil jarak minimum pesawat terhadap *waypoint* metode PID pada misi terbang B

Minimum Distance to <i>Waypoint</i> (m)											
PID	<i>Waypoint</i> Number										AVG ALL WP
	1	2	3	4	5	6	7	8	9	10	
TRIAL 1	61.38	41.98	15.78	11.5	21.41	24.55	67.51	20.11	37.34	24.44	32.6 ± 18.16
TRIAL 2	16.7	26.83	21.07	12.9	36.46	16.71	23.19	28.62	23.85	2.7	20.9 ± 8.84
AVG	39.04 ± 22.34	34.41 ± 7.57	18.43 ± 2.65	12.2 ± 0.7	28.94 ± 7.52	20.63 ± 3.92	45.35 ± 22.16	24.37 ± 4.25	30.6 ± 6.75	13.57 ± 10.87	26.75 ± 14.28

Tabel 5.15 Detil jarak minimum pesawat terhadap *waypoint* model *set 1* pada misi terbang B

Minimum Distance to <i>Waypoint</i> (m)											
MODEL SET 1	<i>Waypoint</i> Number										AVG ALL WP
	1	2	3	4	5	6	7	8	9	10	
TRIAL 1	19.89	9.62	7.91	97.9	281.07	17.41	18.63	58.98	8.26	5.83	52.55 ± 81.09
TRIAL 2	31.1	13.84	13.23	120.55	278.57	29.08	49.5	51.76	25.53	16.62	62.98 ± 77.92
AVG	25.5 ± 5.61	11.73 ± 2.11	10.57 ± 2.66	109.23 ± 11.33	279.82 ± 1.25	23.25 ± 5.84	34.07 ± 15.44	55.37 ± 3.61	16.9 ± 8.64	11.23 ± 5.4	57.77 ± 79.52

Tabel 5.16 Detil jarak minimum pesawat terhadap *waypoint* model *set 2* pada misi terbang B

Minimum Distance to <i>Waypoint</i> (m)											
MODEL SET 2	<i>Waypoint</i> Number										AVG ALL WP
	1	2	3	4	5	6	7	8	9	10	
TRIAL 1	23.02	2.79	9.63	9.48	6.98	15.09	32.03	22.75	27.35	20.08	16.92 ± 9.1
TRIAL 2	27.25	4.08	14.17	9.52	16.78	13.28	30.77	8.67	17.09	9.14	15.08 ± 7.96
AVG	25.14 ± 2.12	3.44 ± 0.65	11.9 ± 2.27	9.5 ± 0.02	11.88 ± 4.9	14.19 ± 0.91	31.4 ± 0.63	15.71 ± 7.04	22.22 ± 5.13	14.61 ± 5.47	16 ± 8.55

Tabel 5.17 Detil jarak minimum pesawat terhadap *waypoint* model *set 3* pada misi terbang B

Minimum Distance to <i>Waypoint</i> (m)											
MODEL SET 3	<i>Waypoint</i> Number										AVG ALL WP
	1	2	3	4	5	6	7	8	9	10	
TRIAL 1	44.47	38.35	17.62	20.57	35.9	28.61	62.31	70.39	43.65	15.18	37.71 ± 17.45
TRIAL 2	44.66	34.37	11.91	13.31	15.23	26.39	66.72	62.61	56.01	23.37	35.46 ± 19.76
AVG	44.57 ± 0.09	36.36 ± 1.99	14.77 ± 2.86	16.94 ± 3.63	25.57 ± 10.34	27.5 ± 1.11	64.52 ± 2.21	66.5 ± 3.89	49.83 ± 6.18	19.28 ± 4.1	36.59 ± 18.64

Tabel 5.18 Detil jarak minimum pesawat terhadap *waypoint* model *set 4* pada misi terbang B

Minimum Distance to <i>Waypoint</i> (m)											
MODEL SET 4	<i>Waypoint</i> Number										AVG ALL WP
	1	2	3	4	5	6	7	8	9	10	
TRIAL 1	4.3	25.04	40.07	42.4	55.15	24.18	14.17	31.41	13.93	13.93	26.46 ± 15
TRIAL 2	5.3	29.49	44.9	57.16	80.62	23.72	14.56	42.35	21.55	22.74	34.24 ± 21.23
AVG	4.8 ± 0.5	27.27 ± 2.23	42.49 ± 2.42	49.78 ± 7.38	67.89 ± 12.74	23.95 ± 0.23	14.37 ± 0.2	36.88 ± 5.47	17.74 ± 3.81	18.34 ± 4.4	30.35 ± 18.38

Tabel 5.19 Detil nilai *cross track distance* setiap *section* metode PID pada misi terbang B

Cross Track Distance (m)											
PID	Section										AVG ALL SECTION
	1	2	3	4	5	6	7	8	9	10	
TRIAL 1	116.26	79.32	69.12	97.43	339.1	27.82	33.48	155.83	445.29	59.45	142.31 ± 132.11
TRIAL 2	88.06	84.8	71.18	101.36	372.3	19.7	15.39	153.46	434.72	39.19	138.02 ± 138.96
AVG	102.16 ± 14.1	82.06 ± 2.74	70.15 ± 1.03	99.4 ± 1.97	355.7 ± 16.6	23.76 ± 4.06	24.44 ± 9.05	154.65 ± 1.19	440.01 ± 5.29	49.32 ± 10.13	140.17 ± 135.58

Tabel 5.20 Detil nilai *cross track distance* setiap *section* model *set 1* pada misi terbang B

Cross Track Distance (m)											
MODEL SET 1	Section										AVG ALL SECTION
	1	2	3	4	5	6	7	8	9	10	
TRIAL 1	96.87	71.22	85.16	130.98	582.52	104.95	23.73	121.28	423.09	47.13	168.69 ± 175.88
TRIAL 2	100.62	63.76	61.79	143.74	583.59	119.65	41.54	136.43	383.35	37.86	167.23 ± 173.89
AVG	98.75 ± 1.88	67.49 ± 3.73	73.48 ± 11.69	137.36 ± 6.38	583.06 ± 0.54	112.3 ± 7.35	32.64 ± 8.91	128.86 ± 7.58	403.22 ± 19.87	42.5 ± 4.63	167.96 ± 174.89

Tabel 5.21 Detil nilai *cross track distance* setiap *section* model *set 2* pada misi terbang B

Cross Track Distance (m)											
MODEL SET 2	Section										AVG ALL SECTION
	1	2	3	4	5	6	7	8	9	10	
TRIAL 1	112.11	94.04	109.77	116.75	292.42	37.82	24.22	164.26	404.68	45.71	140.18 ± 114.44
TRIAL 2	101.3	100	127.03	100.39	319.4	24.69	13.16	182.78	389.57	48.89	140.72 ± 117.82
AVG	106.71 ± 5.41	97.02 ± 2.98	118.4 ± 8.63	108.57 ± 8.18	305.91 ± 13.49	31.26 ± 6.56	18.69 ± 5.53	173.52 ± 9.26	397.13 ± 7.56	47.3 ± 1.59	140.45 ± 116.14

Tabel 5.22 Detil nilai *cross track distance* setiap *section* model *set 3* pada misi terbang B

Cross Track Distance (m)											
MODEL SET 3	Section										AVG ALL SECTION
	1	2	3	4	5	6	7	8	9	10	
TRIAL 1	131.85	58.71	109.96	108.29	373.89	22.25	17.33	115.96	414.89	64.58	141.77 ± 131.91
TRIAL 2	131.1	67.09	113.17	108.56	300.21	29.73	16.75	121.14	386.68	67.8	134.22 ± 112.34
AVG	131.48 ± 0.38	62.9 ± 4.19	111.57 ± 1.61	108.43 ± 0.13	337.05 ± 36.84	25.99 ± 3.74	17.04 ± 0.29	118.55 ± 2.59	400.79 ± 14.11	66.19 ± 1.61	138 ± 122.52

Tabel 5.23 Detil nilai *cross track distance* setiap *section* model *set 4* pada misi terbang B

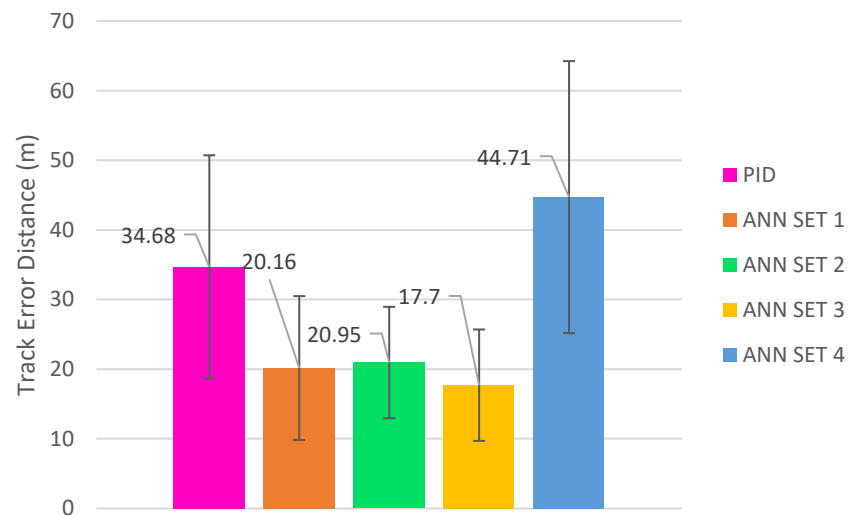
Cross Track Distance (m)											
MODEL SET 4	Section										AVG ALL SECTION
	1	2	3	4	5	6	7	8	9	10	
TRIAL 1	37.22	58.55	27.35	112.69	232.58	47.18	36.64	122.26	418.21	24.22	111.69 ± 119
TRIAL 2	69.3	60.98	26.77	110.14	232.87	48.46	33.54	130.39	395.18	26.45	113.41 ± 111.5
AVG	53.26 ± 16.04	59.77 ± 1.22	27.06 ± 0.29	111.42 ± 1.28	232.73 ± 0.14	47.82 ± 0.64	35.09 ± 1.55	126.33 ± 4.06	406.7 ± 11.52	25.34 ± 1.12	112.55 ± 115.31

5.4 Analisis

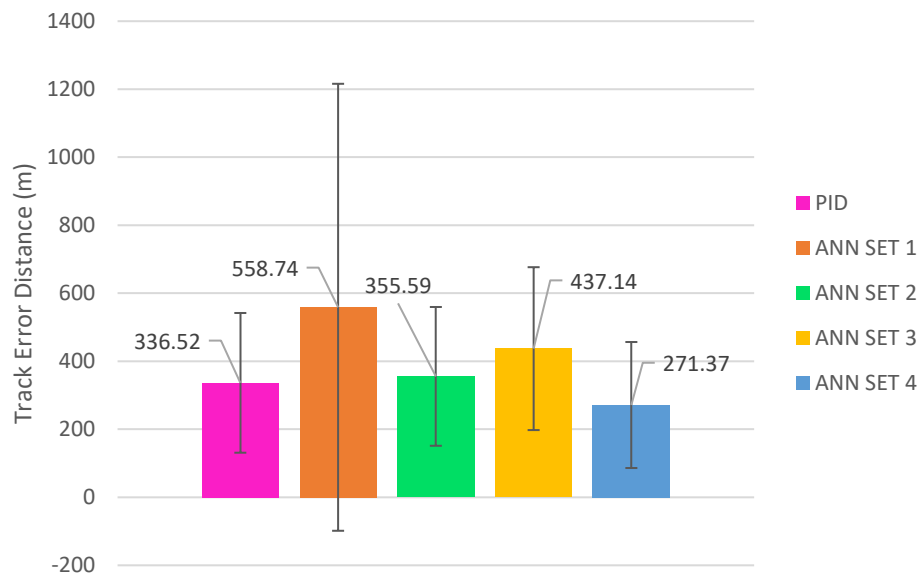
Proses analisis dilakukan dengan membandingkan performa keempat buah model DL terhadap kontroler lintas terbang yang menggunakan strategi *PID controller* pada dua buah misi terbang. Nilai dari dua percobaan pada masing-masing misi dan metode akan dirata-ratakan sebelum dilakukan perbandingan metrik berupa jarak terdekat terhadap setiap *waypoint* yang penulis kategorikan sebagai metrik A dan *cross track distance* sebagai metrik B. Selain itu, akan ditinjau juga karakter rentang sudut *roll* yang dilakukan pesawat pada setiap *section* lintasan untuk melihat pengaruh karakterisasi data pembelajaran terhadap karakter kontroler dalam memberikan kendali manuver *roll*.

5.4.1 Misi Terbang A

Dari nilai rata-rata jarak minimum pesawat dan nilai *cross track distance* dengan keseluruhan *waypoint* pada misi terbang A yang telah dipaparkan pada Tabel 5.4 hingga Tabel 5.13, dapat disusun dua grafik berikut untuk memudahkan membandingkan nilai dari 5 buah metode yang digunakan.

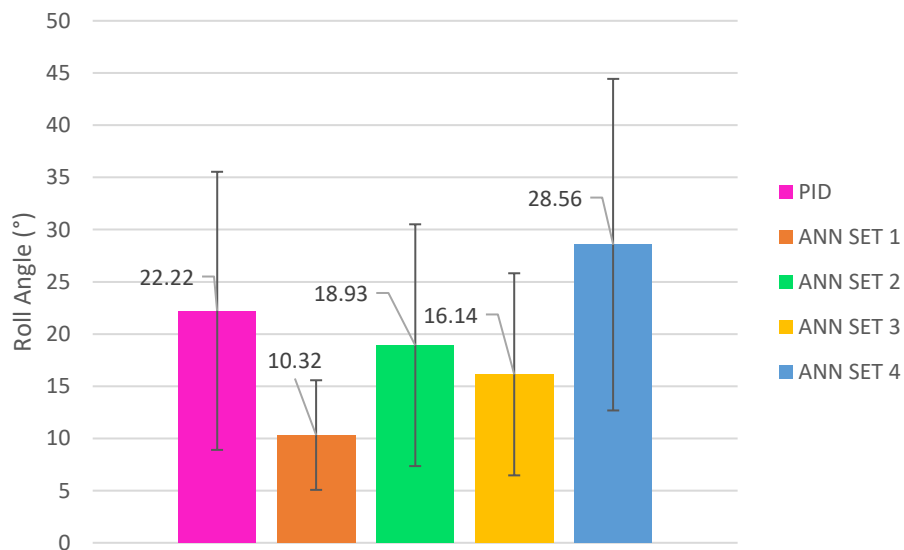


Gambar 5.24 Grafik nilai rata-rata jarak min. terhadap seluruh *waypoint* pada misi terbang A



Gambar 5.25 Grafik nilai rata-rata CTD terhadap seluruh *waypoint* pada misi terbang A

Selain dua grafik di atas, ditampilkan juga grafik yang berinformasikan nilai rata-rata rentang sudut *roll* dari setiap metode pada Gambar 5.26 guna melihat hubungan antara kedua metrik tersebut dengan manuver yang dilaksanakan.



Gambar 5.26 Grafik nilai rata-rata sudut *roll* terhadap seluruh *waypoint* pada misi terbang A

Dari Gambar 5.24 dan Gambar 5.25, ditemukan sebuah tren berkebalikan dimana ketika nilai rata-rata rentang sudut *roll* menempati posisi pertama sebagai nilai yang terbesar, maka didapat nilai rata-rata CTD pada peringkat pertama sebagai nilai yang terkecil, begitu seterusnya hingga posisi terakhir. Jika dihitung, nilai korelasi antar dua variabel ini bernilai -0.97 sehingga dapat dikatakan metrik B memiliki korelasi negatif yang hampir sempurna terhadap variabel rentang sudut *roll*. Didapat juga bahwa metrik A memiliki tren yang serupa dengan rentang sudut *roll* berupa korelasi positif yang bernilai 0.88 dan berkorelasi negatif dengan nilai CTD bernilai -0.76.

Hal yang menjelaskan nilai korelasi antar metrik ini yaitu dengan memiliki keagresifan kontrol manuver *roll* yang lebih tinggi, maka pesawat dapat mencapai posisi sejajar dengan *waypoint* yang dituju lebih cepat. Sehingga, pesawat akan cenderung lebih cepat mendekati garis lintasan yang terbentuk antara *waypoint* sebelumnya dan *waypoint* yang dituju, menghasilkan nilai CTD lebih rendah. Aksi kendali yang terlalu responsif ini memberikan *trade-off* berupa nilai kedekatan dengan *waypoint* yang meningkat.

Dari Gambar 5.24, dapat terlihat bahwa model DL *set* keempat memiliki nilai rata-rata metrik A serta nilai standar deviasi yang paling tinggi. Sedangkan, model *set* pertama hingga ketiga terlihat memiliki nilai rata-rata serta persebaran yang lebih rendah dibandingkan dengan metode PID, dengan model *set* ketiga yang memberikan nilai rata-rata terkecil perihal nilai jarak minimum terhadap keseluruhan *waypoint* pada misi terbang A.

Jika diamati pada Gambar 5.25, sesuai dengan kecenderungan yang sempat disinggung, didapat bahwa model *set* keempat memiliki nilai rata-rata metrik B yang paling rendah, ketika nilai metrik A bernilai paling tinggi. Jika dibandingkan bentuk lintasan yang dibentuk pesawat oleh model *set* keempat yang ditampilkan pada Gambar 5.12 dan Gambar 5.13 dengan bentuk lintasan pesawat yang dilakukan oleh model *set* pertama dengan metrik B tertinggi, dapat terlihat bahwa model *set* pertama memiliki bentuk lintasan yang lebih berbentuk oval, sedangkan pada *set* keempat lintasan yang dibentuk lebih berbentuk seperti bujur sangkar.

Untuk melihat peningkatan atau penurunan performa model DL dibandingkan dengan metode PID perhal jarak minimum terhadap setiap *waypoint* yang menyusun misi terbang A. Dilakukan pengurangan nilai rata-rata metrik ini pada setiap *set* model DL dengan nilai rata-rata percobaan PID, yang ditampilkan melalui tabel berikut.

Tabel 5.24 Perbedaan jarak min. dengan setiap *waypoint* dari metode PID pada misi terbang A

WP	1	2	3	4	5	AVG ALL WP
Turn Angle (°)	-18.43	-26.57	-90	-90	-88.3	-62.66
Minimum Distance to WP PID (m)						
AVG PID	20.41	54.38	42.67	43.71	12.24	34.68
Minimum Distance to WP Difference from PID (m)						
AVG SET1	-0.14	-20.79	-29.73	-17.96	-4.00	-14.52
AVG SET2	-8.99	-26.33	-13.71	-22.55	2.92	-13.73
AVG SET3	-2.10	-39.88	-21.81	-32.61	11.50	-16.98
AVG SET4	14.87	3.11	23.32	9.81	-0.96	10.03

Selisih nilai yang bertanda negatif ditandai dengan kolom berwarna hijau, yang mengindikasikan bahwa model pada *set* terkait berhasil memberikan peningkatan performa dibandingkan dengan metode PID. Dapat dilihat bahwa model *set* pertama hingga ketiga memberikan perbaikan nilai kedekatan dengan *waypoint* pertama hingga keempat. Sedangkan, model DL *set* keempat memberikan hasil yang lebih buruk pada keempat *waypoint* tersebut dan hanya mengalami perbaikan sebesar 0.96 meter atau 8% dari metode PID saat melintasi *waypoint* kelima.

Melalui Tabel 5.24, terlihat bahwa model *set* pertama merupakan model satu-satunya yang memberikan peningkatan nilai metrik A pada kelima *waypoint* penyusun misi terbang A. Namun, dalam hal rata-rata dari keseluruhan *waypoint*, model *set* pertama tetap kalah unggul sekitar 7% terhadap referensi metode PID dibandingkan model *set* ketiga.

Perlu diketahui bahwa peningkatan atau penurunan nilai dari setiap metrik pada suatu *waypoint* tertentu saling berhubungan dengan manuver yang dilakukan untuk mencapai *waypoint* sebelumnya. Sehingga, akan dilihat pola yang diberikan dari kelima metode ini dari setiap *waypoint* yang menyusun misi terbang A.

Pola dari kedekatan pesawat terhadap *waypoint* untuk metode PID yaitu akan meningkat dari *waypoint* pertama, lalu turun, naik dan turun kembali pada *waypoint* terakhir. Pola yang sama ditemukan pada model DL *set* pertama. Sedangkan, model DL *set* kedua dan keempat memiliki pola yang sama yaitu peningkatan pada dua *waypoint* setelah *waypoint* pertama, dan diikuti dengan penurunan hingga *waypoint* terakhir. Model *set* ketiga memiliki pola yang unik yaitu diawali dengan penurunan pada saat memasuki *waypoint* kedua lalu dilanjutkan dengan peningkatan, kemudian penurunan lagi pada *waypoint* keempat dan diakhiri dengan kenaikan nilai.

Jika nilai dari *waypoint* kedua dan keempat dari model DL *set* ketiga ini diamati, didapati bahwa pada *waypoint* inilah didapati peningkatan jarak minimum yang sangat tinggi yang berdampak meningkatnya nilai rata-rata pada keseluruhan *waypoint*. Didapati juga bahwa dalam pergantian *waypoint* menuju *waypoint* kedua, terjadi peningkatan perubahan sudut lintasan ke arah kiri yang semula bernilai 18.43° menjadi 26.57° dengan selisih sebesar 8.14° . Sedangkan, pada pergantian menuju *waypoint* keempat, tidak terjadi perubahan sudut lintasan yaitu tetap bernilai 90° ke arah kiri.

Berbeda halnya dengan metode PID dan model DL *set* pertama, yang mengalami peningkatan metrik A pada *waypoint* ketiga dan kelima. Dalam pergantian *waypoint* menuju *waypoint* ketiga, terjadi peningkatan perubahan sudut yang semula bernilai 26.57° ke arah kiri menjadi 90° ke arah kiri, dengan peningkatan sebesar 63.43° . Sedangkan, dalam pergantian *waypoint* menuju *waypoint* kelima, terjadi penurunan perubahan sudut yang semula bernilai 90° ke arah kiri menjadi -88.3° ke arah kiri, dengan penurunan sebesar 1.7° .

Melalui kedua pola perubahan sudut lintasan yang telah dipaparkan di atas, dapat ditarik sebuah kesimpulan naif yaitu model DL *set* ketiga dalam melaksanakan misi terbang A dapat meminimalkan jarak terhadap *waypoint* ketika terjadi perubahan sudut yang semakin negatif dengan nilai yang kecil dan konstan. Sedangkan, perilaku yang berkebalikan dihasilkan oleh kendali menggunakan metode PID dan model DL *set* pertama yaitu peningkatan nilai kedekatan terhadap *waypoint* terjadi ketika muncul perubahan sudut yang semakin negatif dengan nilai yang besar dan ketika perubahan sudut lintasan yang tidak semakin negatif.

Selain meninjau dari metrik kedekatan pesawat terhadap setiap *waypoint*, akan dilihat pengaruh terhadap metrik *cross distance* dari setiap *section* melalui tabel berikut, dimana $section_{(n)}$ adalah rentang ketika pesawat membentuk lintasan dari $waypoint_{(n-1)}$ menuju $waypoint_{(n)}$.

Tabel 5.25 Perbedaan *cross track distance* dari metode PID pada misi terbang A

Section	1	2	3	4	5	AVG ALL SECTION
Turn Angle (°)	-18.43	-26.57	-90	-90	-88.3	-62.66
Cross Track Distance PID (m)						
AVG PID	48.09	132.04	554.27	477.9	470.32	336.52
Cross Track Distance Difference from PID (m)						
AVG SET1	18.51	65.80	514.51	148.30	363.97	222.22
AVG SET2	20.73	20.62	2.64	3.96	47.37	19.07
AVG SET3	47.29	95.02	183.33	120.84	56.61	100.62
AVG SET4	-15.43	-66.63	-116.43	-50.94	-76.32	-65.15

Seperti yang telah dijelaskan sebelumnya bahwa nilai CTD memiliki korelasi negatif yang tinggi dengan rentang nilai sudut *roll*, berikut ini disajikan tabel yang memaparkan rentang sudut *roll* dari setiap *section* misi terbang A.

Tabel 5.26 Perbedaan rentang sudut *roll* dari metode PID pada misi terbang A

Section	1	2	3	4	5	AVG ALL WP
Turn Angle (°)	-18.43	-26.57	-90	-90	-88.3	-62.66
Roll Angle PID (°)						
AVG PID	2.42	10.48	36.04	31.62	30.54	22.22
Roll Angle Range Difference from PID (°)						
AVG SET1	-0.38	-3.77	-19.54	-19.15	-16.70	-11.90
AVG SET2	-0.70	-2.06	-6.22	-3.15	-4.32	-3.29
AVG SET3	0.95	-4.64	-11.40	-10.06	-5.27	-6.08
AVG SET4	2.88	5.97	11.54	3.44	7.88	6.34

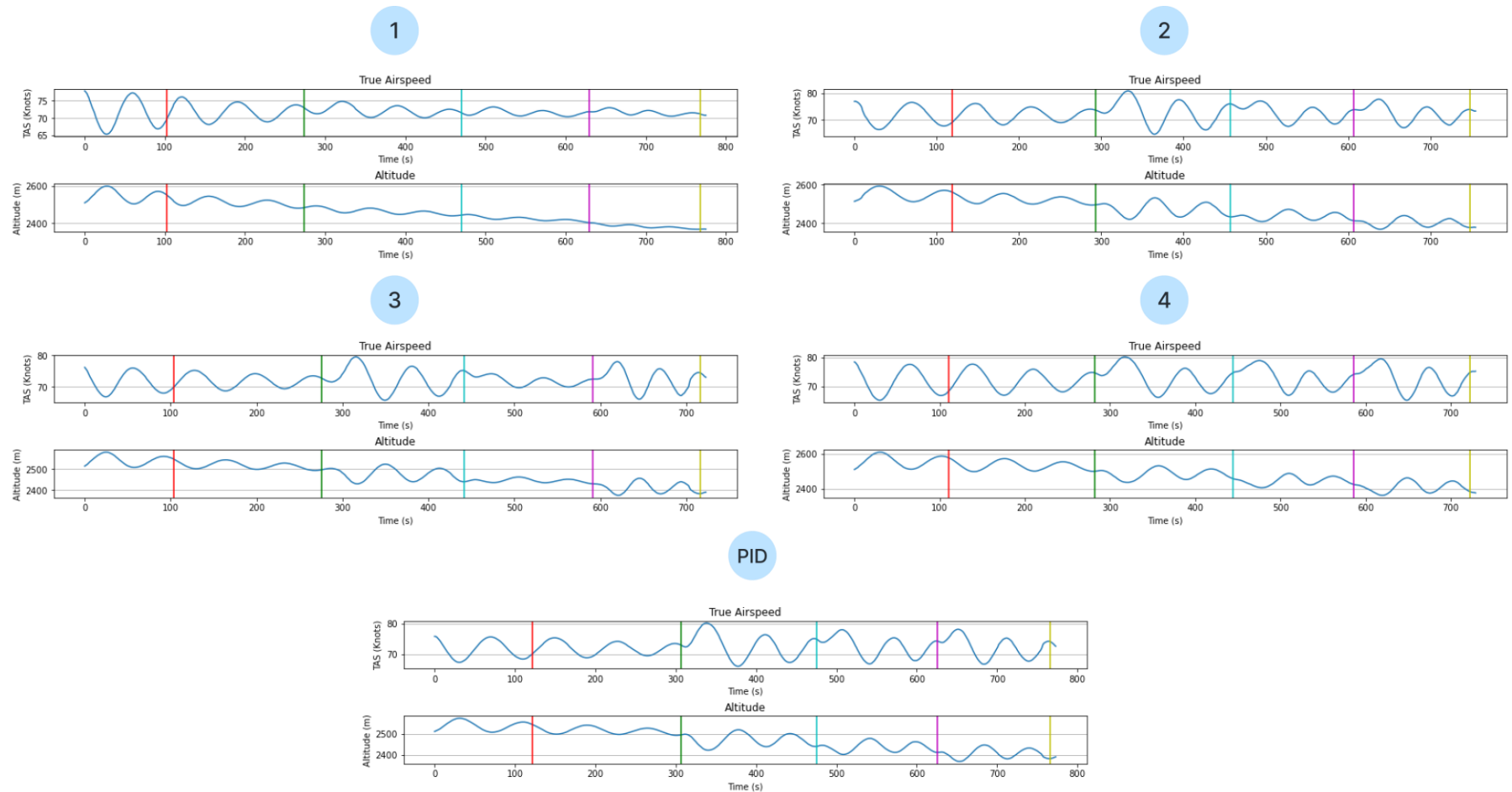
Melalui kedua tabel diatas, dapat terlihat bahwa hanya model *set* keempat saja yang mampu memberikan pengurangan nilai CTD sebesar 19% realtif terhadap hasil metode PID pada keseluruhan *waypoint*, yang mana hal ini terjadi dikarenakan rentang sudut *roll* yang dilakukan pada setiap *section* jauh lebih luas dibandingkan yang terjadi pada metode PID.

Berbeda halnya dengan model DL *set* pertama, kedua, dan ketiga yang mana pada setiap *section* memiliki rentang sudut *roll* yang lebih rendah dibandingkan metode PID, terkecuali pada model *set* ketiga *section* pertama, dengan nilai selisih bertandakan positif yang hanya bernilai kurang dari 1° saja. Terlihat dari Tabel 5.27, ketiga model ini memberikan selisih CTD yang bernilai positif pada seluruh *waypoint*, yang berartikan berperforma lebih buruk dibanding metode PID.

Secara keseluruhan *waypoint*, didapati bahwa model DL *set* ketiga memiliki nilai rentang sudut *roll* yang paling rendah dari semua metode. Hal ini sesuai dengan *plot* prediksi *joystick roll* model DL terhadap data validasi yang berasal dari metode PID untuk model *set* ketiga yang sebelumnya ditampilkan pada

Gambar 4.12. Dari grafik ini, terlihat bahwa dalam menjangkau nilai rasio *joystick roll* yang negatif, model ini memberikan hasil prediksi yang *undershoot* dibanding PID, dan *overshoot* pada rentang nilai positif. Sehingga, dalam melakukan misi terbang A yang seluruhnya melibatkan pergerakan *joystick* ke arah kiri atau negatif, akan didapatkan pengendalian yang tidak seagresif metode PID.

Dalam rangka meninjau lebih dalam apa yang terjadi selama proses pelaksanaan misi terbang, disajikan grafik dari *true airspeed* (TAS) dan *altitude* terhadap waktu pada Gambar 5.27, dengan garis vertikal menandakan waktu dimana pesawat melintasi *waypoint*.



Gambar 5.27 TAS dan ketinggian terhadap waktu pada misi terbang A

Melalui Gambar 5.27, dapat dilihat bahwa untuk kelima metode yang diuji, TAS berhubungan erat dengan *altitude*, dimana ketika ketinggian terbang semakin meningkat, maka TAS semakin menurun, begitupula sebaliknya. Hal ini disebabkan semakin tingginya posisi terbang pesawat, densitas udara semakin menurun. Densitas yang semakin menurun menyebabkan tekanan juga menurun. Dikarenakan TAS memanfaatkan sensor *pitot tube* yang merasakan tekanan yang menurun, maka nilai dari TAS akan menurun ketika ketinggian terbang meningkat dan sebaliknya.

Kelima metode menunjukkan terjadinya osilasi pada nilai TAS sepanjang melakukan misi lintas terbang. Namun, amplitudo osilasi yang terjadi pada model *set* data pembelajaran pertama terlihat paling minim dibandingkan metode lainnya. Sebelumnya, Tabel 5.24 menunjukkan bahwa model *set* pertama mampu memberikan peningkatan nilai metrik A pada seluruh *waypoint* penyusun misi terbang A. Sehingga, dapat dikatakan secara naif bahwa kestabilan secara longitudinal yang mempengaruhi ketinggian terbang pesawat turut berkontribusi pada metrik A dan B dalam melaksanakan misi lintas terbang A. Dengan catatan bahwa relasi ini belum terbukti kebenarannya dan perlu dilakukan penelitian lebih lanjut.

Dari semua yang telah dibahas di atas, dapat ditarik kesimpulan bahwa untuk kasus misi terbang A, untuk mencapai metrik jarak minimum terhadap *waypoint* terbaik, dapat digunakan model dengan *set* data pembelajaran yang memiliki rentang sudut *roll* dari -40° hingga 10° atau dominan terdiri dari karakter manuver *roll* ke arah kiri, dengan *trade-off* yaitu meningkatnya nilai dari CTD akibat karakter kontroler yang kurang agresif. Sedangkan, untuk mencapai nilai metrik *cross track distance* yang terbaik, dapat digunakan model *set* pembelajaran keempat yang memiliki karakter sudut *roll* dari -40° hingga 40° dengan *trade-off* berupa semakin jauh kedekatan pesawat dengan *waypoint* yang dituju.

Namun, jika ingin diamati dari kelima metode yang digunakan, manakah yang memberikan keseimbangan antara metrik jarak minimum terhadap *waypoint* dan juga CTD. Dilakukan standarisasi masing-masing metrik menggunakan teknik

Min-Max scaler menggunakan persamaan 4.1, lalu diberikan bobot sebesar 50% untuk setiap metriknya sehingga didapati informasi yang disajikan melalui tabel berikut.

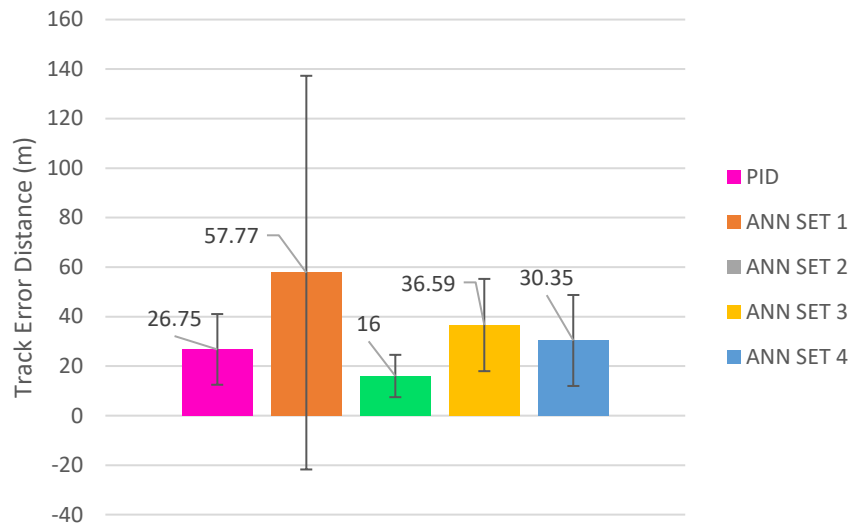
Tabel 5. 27 Nilai bobot total metrik pada misi terbang A

Metode	Scaled Min. Dist (A)	Scaled CTD (B)	0.5A + 0.5B
PID	0.63	0.23	0.43
DL set 1	0.09	1	0.55
DL set 2	0.12	0.29	0.21
DL set 3	0	0.58	0.29
DL set 4	1	0	0.5

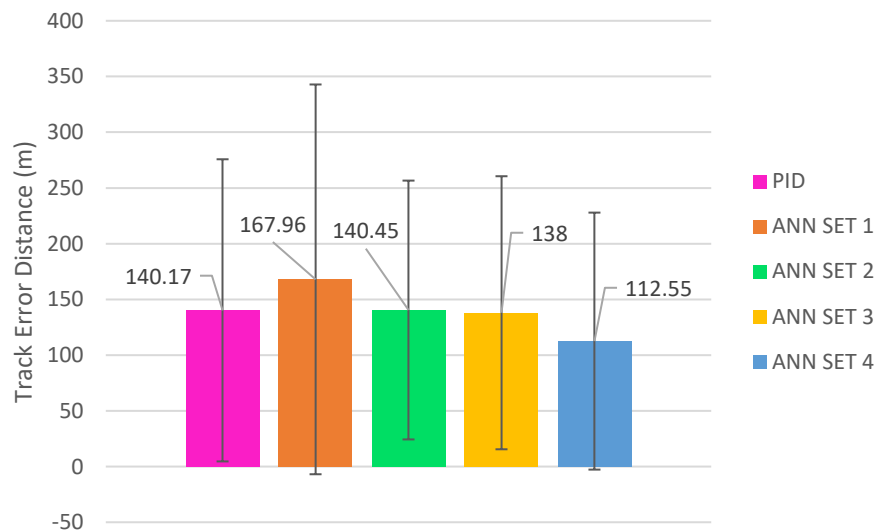
Nilai total dari pembobotan diwakili oleh kolom yang diberi nama “0.5A + 0.5B”. Nilai terkecil dari kolom ini menandakan performa yang paling seimbang dari kedua metrik A & B, yang mana dipegang oleh model DL *set* kedua dengan total nilai pembobotan bernilai 0.21. Perlu diingat bahwa hasil yang diberikan dari setiap model DL yang dilatih menggunakan *set* data pembelajarannya masing-masing, merupakan hasil dari *Bayesian Optimization* sebanyak 100 iterasi untuk setiap modelnya.

5.4.2 Misi Terbang B

Serupa dengan apa yang dilakukan pada analisis misi terbang A. Akan dibuat terlebih dahulu dua grafik berikut berdasarkan nilai rata-rata jarak minimum pesawat dan nilai *cross track distance* dengan keseluruhan *waypoint* pada misi terbang B yang telah dipaparkan pada Tabel 5.14 hingga Tabel 5.23.

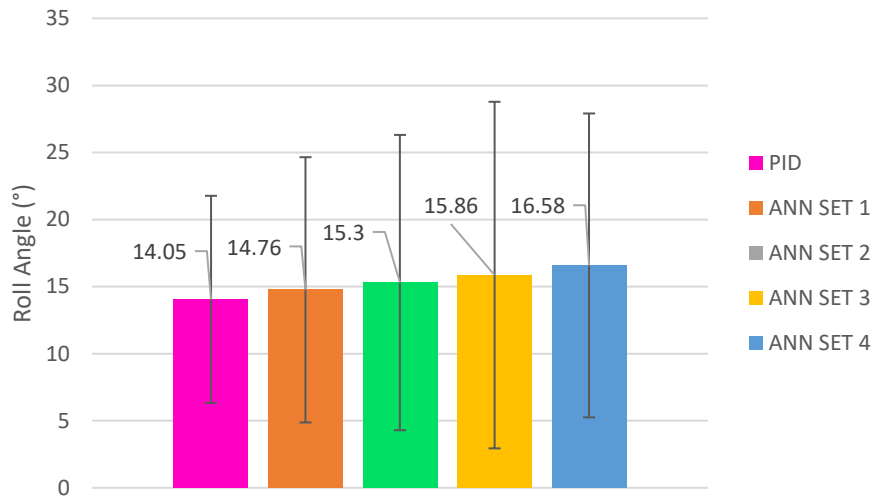


Gambar 5.28 Grafik nilai rata-rata jarak min. terhadap seluruh *waypoint* pada misi terbang B



Gambar 5.29 Grafik nilai rata-rata CTD terhadap seluruh *waypoint* pada misi terbang B

Selain dua grafik di atas, ditampilkan juga grafik yang berinformasikan nilai rata-rata rentang sudut *roll* dari setiap metode pada gambar di bawah ini guna melihat hubungan antara kedua metriks tersebut dengan manuver yang dilaksanakan.



Gambar 5.30 Grafik nilai rata-rata sudut *roll* terhadap seluruh *waypoint* pada misi terbang B

Dari Gambar 5.28 dan Gambar 5.29, ditemukan sebuah tren yang serupa pada misi terbang A perihal hubungan metriks B dengan rentang sudut *roll*, yang mana didapati nilai korelasinya sebesar -0.67, tidak senegatif yang didapati pada misi terbang A.

Ditemukan juga hal yang berkebalikan dari nilai korelasi metriks A terhadap metriks B dan rentang sudut *roll*. Korelasi antara metriks A terhadap metriks B yang bernilai -0.76 pada misi terbang A, pada misi terbang B bernilai positif 0.62. Begitu pula yang terjadi pada korelasi metriks A dengan rentang sudut terbang, yang pada misi terbang A bernilai positif tinggi yaitu pada angka 0.88, pada misi terbang B nilai ini menjadi negatif dengan nilai 0.12.

Hal yang menyebabkan perbedaan nilai korelasi antara misi terbang A dan misi terbang B yaitu dikarenakan jumlah *waypoint* penyusun misi B lebih banyak serta mengharuskan melakukan manuver yang eskترم ke kiri dan juga ke kanan, berbeda halnya dengan misi A yang memerlukan manuver ke kiri saja. Sehingga,

nilai rata-rata yang dihasilkan bisa sangat variatif yang menyebabkan perubahan tanda pada nilai korelasi.

Dari Gambar 5.28, dapat terlihat bahwa model DL *set* pertama memiliki nilai rata-rata kedekatan terhadap keseluruhan *waypoint* serta nilai standar deviasi yang paling tinggi naun juga memiliki nilai tertinggi pada metriks B. Hal ini sesuai dengan nilai korelasi antara kedua metriks ini yang bernilai 0.62.

Untuk melihat peningkatan atau penurunan perfroma model DL perihal jarak minimum terhadap setiap *waypoint* yang menyusun misi terbang B. Dilakukan pengurangan nilai rata-rata metriks ini pada setiap *set* model DL dengan nilai rata-rata percobaan PID, yang mana disajikan melalui Tabel 5.28.

Melalui Gambar 5.28, dapat dilihat bahwa hanya model *set* kedua yang memiliki nilai rata-rata serta persebaran yang lebih rendah pada metriks A dibandingkan dengan metode PID pada misi terbang B. Secara lebih detil pada setiap *waypoint*-nya yang ditunjukkan melalui Tabel 5.29 di bawah, dapat dilihat bahwa model DL *set* kedua mampu memberikan peningkatan nilai kedekatan terhadap *waypoint* pertama hingga kesembilan relatif terhadap metode PID. Sedangkan, pada *waypoint* kesepuluh, didapati selisih yang bernilai positif 1.04 meter, yang mana nilai sekecil ini tidak terlalu berarti.

Model dengan nilai metriks A terburuk pada misi terbang B dipegang oleh model *set* pertama, meskipun dari tabel di bawah dapat terlihat bahwa model *set* ini memberikan perbaikan nilai metriks A sebesar 40% relatif terhadap metode PID pada 6 dari 10 *waypoint* yang ada. Namun, pada *waypoint* keempat dan kelima, yang merupakan *section* dimana pesawat perlu melakukan manuver ke kiri dengan tajam, model ini kesulitan untuk mendekati *waypoint* ini sehingga memberikan nilai kedekatan yang bernilai ratusan meter.

Tabel 5.28 Perbedaan jarak min. dengan setiap *waypoint* dari metode PID pada misi terbang B

WP	1	2	3	4	5	6	7	8	9	10	AVG ALL WP
Turn Angle (°)	-22.46	17.76	-10.56	-36.08	-78.38	-32.29	3.57	40.72	92.94	17.08	-4.32
Minimum Distance to WP PID (m)											
AVG PID	39.04	34.41	18.43	12.2	28.94	20.63	45.35	24.37	30.6	13.57	26.75
Minimum Distance to WP Difference from PID (m)											
AVG SET1	-13.54	-22.68	-7.86	97.03	250.88	2.62	-11.28	31.00	-13.70	-2.34	31.02
AVG SET2	-13.90	-30.97	-6.53	-2.70	-17.06	-6.44	-13.95	-8.66	-8.38	1.04	-10.75
AVG SET3	5.53	1.95	-3.66	4.74	-3.37	6.87	19.17	42.13	19.23	5.71	9.84
AVG SET4	-34.24	-7.14	24.06	37.58	38.95	3.32	-30.98	12.51	-12.86	4.77	3.60

Seperti yang sebelumnya sudah disampaikan bahwa peningkatan atau penurunan nilai dari metrik A dan B pada suatu *waypoint* saling berhubungan dengan manuver yang dilakukan untuk mencapai *waypoint* sebelumnya. Didapati bahwa pada *waypoint* keempat dan kelima, terjadi perubahan sudut lintasan ke kiri dengan *step* yang tinggi, yaitu dari nilai negatif 10.56° menuju nilai -36.08° dan dilanjutkan dengan perubahan sudut lintasan sebesar -78.38° pada *waypoint* kelima.

Perubahan sudut ke arah kiri dengan *step* besar inilah yang membuat model *set* pertama yang belajar dari *set* data pembelajaran dengan karakter manuver *roll* paling sempit, kurang agresif dalam rangka mencapai kedua *waypoint* tersebut. Dapat dilihat juga melalui grafik pada Gambar 5.29, nilai rata-rata dari rentang sudut *roll* yang dapat mewakili keagresifan perintah kendali dari model, menunjukkan model DL *set* pertama memiliki nilai yang terkecil, dan terus naik hingga model *set* keempat yang dilatih menggunakan *set* data pembelajaran paling luas. Karakter pengendalian yang kurang agresif dari model *set* pertama ini juga sudah terlihat pada misi terbang A, yang mana menghasilkan lintasan yang berbentuk *oval* dengan nilai CTD yang paling tinggi.

Selain meninjau dari metrik kedekatan pesawat terhadap setiap *waypoint*, akan dilihat relasi dari metrik A terhadap metrik B yang dipengaruhi oleh keagresifan pesawat dalam melakukan manuver pada setiap *section*-nya melalui Tabel 5.30 dan Tabel 5.31.

Tabel 5.29 Perbedaan *cross track distance* dari metode PID pada misi terbang B

Section	1	2	3	4	5	6	7	8	9	10	AVG ALL SECTION
Turn Angle (°)	-22.46	17.76	-10.56	-36.08	-78.38	-32.29	3.57	40.72	92.94	17.08	-4.32
Cross Track Distance PID (m)											
AVG PID	102.16	82.06	70.15	99.4	355.7	23.76	24.44	154.65	440.01	49.32	140.17
Cross Track Distance Difference from PID (m)											
AVG SET1	-3.41	-14.57	3.33	37.96	227.36	88.54	8.20	-25.79	-36.79	-6.82	27.79
AVG SET2	4.55	14.96	48.25	9.17	-49.79	7.50	-5.75	18.87	-42.88	-2.02	0.28
AVG SET3	29.32	-19.16	41.42	9.03	-18.65	2.23	-7.40	-36.10	-39.22	16.87	-2.17
AVG SET4	-48.90	-22.29	-43.09	12.02	-122.97	24.06	10.65	-28.32	-33.31	-23.98	-27.62

Tabel 5.30 Perbedaan rentang sudut *roll* dari metode PID pada misi terbang B

Section	1	2	3	4	5	6	7	8	9	10	AVG ALL SECTION
Turn Angle (°)	-22.46	17.76	-10.56	-36.08	-78.38	-32.29	3.57	40.72	92.94	17.08	-4.32
Roll Angle Range PID (°)											
AVG PID	5.5	12.24	11.16	10.7	19.7	14.05	6.55	22.7	29.71	8.23	14.05
Roll Angle Range Difference from PID (°)											
AVG SET1	-1.36	1.69	-3.71	7.15	-6.59	10.95	0.19	2.48	2.25	-6.07	0.71
AVG SET2	-1.69	1.10	0.29	1.55	5.35	-5.16	-3.00	-2.12	2.80	13.35	1.25
AVG SET3	-1.34	6.41	-7.00	0.57	7.87	-1.71	-0.71	3.37	14.38	-3.78	1.81
AVG SET4	3.88	-2.80	0.66	16.47	17.85	0.34	-3.75	-4.24	1.69	-4.84	2.53

Dari segi nilai *cross track distance* atau metrik B, terlihat bahwa model DL *set* keempat mampu memberikan hasil yang lebih baik relatif terhadap metode PID sebanyak 20% yang berasal dari peningkatan nilai pada 7 dari 10 *waypoint* penyusun misi terbang B. Didapati juga bahwa model *set* ketiga juga mampu memberikan perbaikan pada metrik B meskipun hanya sebesar 2% relatif terhadap metode PID.

Dari Tabel 5.30, terlihat bahwa keagresifan perintah manuver *roll* dari model DL *set* keempat khususnya pada *waypoint* keempat dan kelima, berdampak pada peningkatan metrik B sebanyak 35% relatif terhadap metode PID pada *waypoint* kelima yang membutuhkan manuver berbelok kiri yang ekstrem dengan jarak menuju *waypoint* yang cukup singkat dibandingkan manuver berbelok ke arah kanan dari *waypoint* ke-8 dan ke-9.

Menariknya, model DL *set* kedua yang dilatih dominan terhadap manuver *roll* ke arah kanan, mampu memberikan nilai kedekatan terhadap *waypoint* yang mencakup manuver *roll* ke kiri dan ke kanan. Sedangkan model DL *set* ketiga yang seharusnya berperforma baik pada metrik A ketika manuver yang perlu dilakukan adalah *roll* ke arah kiri, malah memberikan hasil yang lebih buruk dibanding metode PID. Hal ini mungkin disebabkan perubahan sudut lintasan dari misi B yang dimulai dengan manuver ke arah kiri dilanjutkan ke arah kanan, kemudian dilanjutkan dengan manuver berbelok ke kiri lagi.

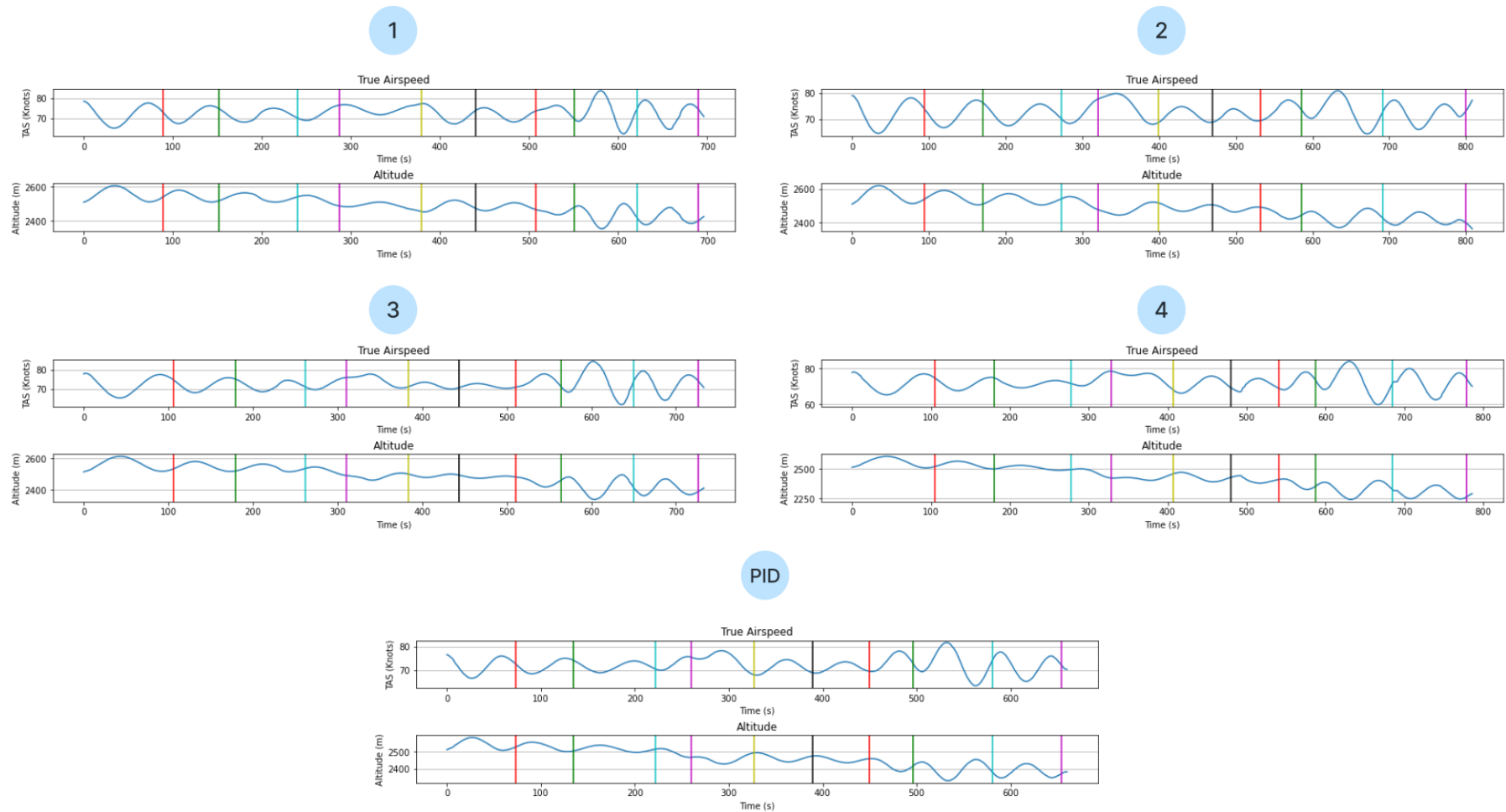
Seperti yang ditunjukkan pada Gambar 4.12 yaitu model DL *set* ketiga cenderung mengalami *overshoot* pada rentang pergerakan rasio *joystick roll* positif atau berbelok kanan. Hal ini terjadi juga ketika pesawat akan memasuki *waypoint* kedua, yang mana pada *waypoint* ini didapati nilai rentang sudut *roll* paling tinggi, begitu pula dengan nilai metrik A. Pergerakan berlebihan ini mengakibatkan model kesulitan untuk mencapai *waypoint* selanjutnya meskipun manuver yang dibutuhkan adalah ke arah kiri, dan *error* ini merambat terus pada *waypoint* seterusnya.

Overshoot yang sempat disinggung dalam melakukan manuver ke kanan pada model *set* ketiga juga terlihat pada *waypoint* ke-8 dan ke-9. Pada kedua

waypoint tersebut, rentang sudut *roll* yang dihasilkan terlihat paling tinggi, yang berdampak pada mengecilnya nilai CTD sebesar 23% dan 9% relatif terhadap metode PID.

Pola yang ditemukan pada misi terbang B memiliki perbedaan dengan yang didapati pada misi terbang A. Perbedaan itu adalah pada misi terbang A, ketika nilai metrik A bernilai minimum, didapati nilai metrik B yang maksimum. Namun, pada misi terbang B yang lebih kompleks, tidak terlihat pola yang demikian. Model DL *set* pertama yang dilatih menggunakan *set* data pembelajaran yang mencakup sudut *roll* yang sempit, memberikan perintah kendali yang paling tidak agresif sehingga didapati nilai CTD paling besar dan juga berperforma paling buruk pada metrik A.

Hal yang dilakukan pada analisis misi terbang A juga dilakukan pada misi terbang B, yaitu melakukan *plot* dari *true airspeed* (TAS) dan *altitude* terhadap waktu yang ditunjukkan pada Gambar 5.31, dengan garis vertikal menandakan waktu dimana pesawat melintasi *waypoint*.



Gambar 5.31 TAS dan ketinggian terhadap waktu pada misi terbang B

Dengan meninjau Gambar 5.31, didapati bahwa model *set* kedua yang memiliki performa berkaitan dengan metrik A terbaik memiliki grafik *true airspeed* terhadap waktu yang secara visual terlihat memiliki gerakan sinusoidal yang lebih mulus dibandingkan keempat metode lainnya. Dikarenakan sistem kendali lintas terbang tidak didesain untuk mencakup matra longitudinal, maka baik pada misi terbang A dan misi terbang B dari setiap metode akan memiliki tren pergerakan TAS yang berosilasi akibat osilasi pada ketinggian. Osilasi yang tidak menentu inilah yang membuat kecepatan terbang pesawat tidak stabil saat melakukan koreksi terhadap *waypoint* yang dituju, sehingga tidak dapat ditarik keterkaitan yang pasti antara TAS dengan metrik A dan B.

Dari semua yang telah dibahas di atas, dapat ditarik kesimpulan bahwa untuk kasus misi terbang B, dapat digunakan model DL yang dilatih menggunakan *set* data pembelajaran kedua untuk mencapai kedekatan terhadap setiap *waypoint*. Sedangkan, untuk mencapai nilai metrik *cross track distance* yang terbaik, dapat digunakan model *set* pembelajaran keempat yang memiliki karakter sudut *roll* dari -40° hingga 40° .

Perihal metode manakah yang memberikan keseimbangan antara metrik jarak minimum terhadap *waypoint* dan juga CTD. Dilakukan tahapan yang sama saat mengevaluasi metrik pada misi terbang A, sehingga didapati informasi yang disajikan melalui tabel berikut.

Tabel 5.31 Nilai bobot total metrik pada misi terbang B

Metode	Scaled Min. Dist (A)	Scaled CTD (B)	$0.5A + 0.5B$
PID	0.26	0.5	0.38
DL set 1	1	1	1
DL set 2	0	0.5	0.25
DL set 3	0.49	0.46	0.48
DL set 4	0.34	0	0.17

Didapati dari Tabel 5.32 di atas bahwa model DL yang dilatih menggunakan *set* data pembelajaran yang paling luas atau *set* data keempat memberikan performa pengendalian kendali lintas terbang yang paling seimbang antara metriks A & B dengan total nilai pembobotan sebesar 0.17 dengan kontribusi dari metriks B yang paling baik serta metriks A yang berada pada posisi ketiga terbaik dibandingkan metode lainnya pada misi terbang B yang lebih kompleks dibandingkan misi terbang A.

Melalui Tabel 5.27 dan Tabel 5.32, didapati fakta bahwa model DL *set* pertama berperforma paling buruk dalam memberikan keseimbangan antara metriks A dan B pada kedua misi terbang yang diujikan, dimana nilai CTD pada model ini selalu memberikan nilai yang terbesar.

Dari seluruh proses analisis yang dilakukan pada kedua misi terbang yang digunakan, didapati bahwa semakin luasnya informasi yang dimiliki dari data pembelajaran tidak selalu dapat memberikan hasil yang terbaik. Setiap model DL memiliki keunggulan yang berbeda bergantung misi terbang yang dilakukan. Hal ini terlihat pada misi terbang A, dimana *set* data pembelajaran yang didominasi oleh penerbangan dengan sudut *roll* bernilai negatif mampu memberikan keseimbangan metriks A dan B yang terbaik dari setiap metode karena misi terbang A seluruhnya terdiri dari manuver berbelok ke arah kiri. Begitupun pada misi terbang B, dimana bentuk lintasan yang kompleks yang melibatkan manuver berbelok ke arah kiri dan kanan yang ekstrem memerlukan *set* data pembelajaran yang mencakup pergerakan sudut *roll* yang luas untuk mencapai keseimbangan metriks A dan B yang terbaik.

Hal menarik yang ditemukan pada penelitian ini yaitu meskipun data pembelajaran yang digunakan berbasis metode PID, beberapa model DL ditemukan mampu memberikan performa yang lebih baik dibandingkan metode PID. Hal ini mungkin diakibatkan oleh penghalusan data yang dilakukan untuk mendapatkan kualitas data yang lebih baik terutama ketika sistem kendali *roll limiter* dalam keadaan aktif, digunakannya sudut *roll* pesawat sebagai *features*, dan sifat dari model DL yang adalah non linier.

Pada bab lima, telah dibahas mengenai hasil simulasi yang dilakukan pada keempat model *deep learning* yang dilatih menggunakan *set* data pembelajaran yang berbeda-beda dan telah dianalisis karakter pengendalian serta pengaruh dari karakter data pembelajaran terhadap performa dari model yang dibangun. Kesimpulan serta saran yang didapat melalui penelitian ini akan dibahas pada bab selanjutnya.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang diperoleh dari tugas akhir ini adalah:

1. Sistem kendali lintas terbang pesawat jet bisnis berbasis *deep learning* telah dibentuk dengan memanfaatkan data simulasi berupa potongan misi yang dikarakterisasi berdasarkan sudut *roll* pesawat, menghasilkan empat buah model DL yang digunakan sebagai pemberi sinyal rasio defleksi *joystick* pada sumbu *roll* untuk mengubah sikap pesawat.
2. Beberapa sistem kendali lintas terbang berbasis DL mampu memberikan keseimbangan nilai metrik kedekatan dengan *waypoint* dan *cross track distance* yang lebih baik dibandingkan metode *PID controller* meskipun data pembelajaran yang digunakan berbasis metode PID. Hal ini mungkin diakibatkan oleh penghalusan data yang dilakukan untuk mendapatkan kualitas data yang lebih baik terutama ketika sistem kendali *roll limiter* dalam keadaan aktif, digunakannya sudut *roll* pesawat sebagai *features*, dan sifat dari model DL yang adalah non linier.
3. Model DL *set* kedua dan ketiga mampu memberikan peningkatan sebesar 51% dan 33% secara berurut pada misi terbang A. Sedangkan, pada misi terbang B, model DL *set* keempat dan kedua mampu memberikan peningkatan sebesar 55% dan 34% terhadap metode PID.
4. Karakter pengendalian sistem kendali lintas terbang berbasis DL berkaitan erat dengan karakter data pembelajaran yang digunakan untuk melatih model. Pada misi terbang A, dikarenakan lintasan misi sepenuhnya disusun atas manuver berbelok kiri, *set* data pembelajaran yang memberikan performa terbaik dan seimbang antara kedua metrik adalah data yang didominasi oleh sudut *roll* negatif. Sedangkan, pada misi terbang B yang melibatkan manuver ke kiri dan ke kanan secara ekstrem, model yang dilatih menggunakan *set* data pembelajaran dengan cakupan sudut *roll* yang paling luas pada kasus ini dari -40° hingga 40° memberikan performa terbaik.

6.2 Saran

Beberapa saran yang diajukan untuk mengembangkan penelitian ini lebih lanjut adalah sebagai berikut.

1. Penggunaan data dari pesawat aktual ketika melakukan lintas terbang dengan harapan model DL dapat mengimitasi perintah kendali yang pilot lakukan dalam melakukan manuver berbelok mengikuti *waypoint*.
2. Penambahan jumlah *set* data pembelajaran dengan karakterisasi data yang lebih sempit dengan harapan didapatkan relasi yang lebih mendalam mengenai pengaruh karakter data pembelajaran terhadap performa dari model DL.
3. Pengujian pada misi terbang yang lebih bervariasi untuk melihat seberapa *robust* sistem pengendalian lintas terbang berbasis DL.
4. Penambahan sistem kendali pengatur *throttle* untuk mengubah kecepatan pesawat bergantung pada seberapa tajam manuver yang akan dilakukan.
5. Penggunaan variabel akselerasi (G) dalam karakterisasi data pembelajaran agar efek kecepatan terbang dan sudut *roll* pesawat dijadikan konsiderasi pembelajaran model.
6. Pengembangan sistem kendali lintas terbang yang melibatkan matra *longitudinal* sehingga kemampuan pengendalian pesawat dapat disempurnakan.

DAFTAR PUSTAKA

- [1] Donald McLean, "Aircraft Flight Control", in *Automatic Flight Control Systems*. Hertfordshire: Prentice Hall International, 1990, pp 1-15.
- [2] Federal Aviation Administration, "Area Navigation (RNAV) and Required Navigation Performance (RNP)" in *Aeronautical Information Manual*. USA: FAA, 2010, ch. 1, section 2. [Online]. Available: <https://tfmlearning.faa.gov/publications/atpubs/AIM/Chap1/aim0102.htm>
- [3] J. Sembiring, "Rancang-Bangun UAV Simulator". LAPAN, Jakarta, Indonesia, Tech. Rep., Jun. 2015.
- [4] H. S. Nugraha and R. A. Sasongko, "The Development of a Computer Simulation Platform for Evaluating Mission Performance of a Low-cost UAV," in *Regional Conference on Mechanical and Aerospace Technology*, Feb. 2010, pp.1-10.
- [5] A. Kaviyarasu, A. Saravanakumar and M. L. Venkatesh, "Hardware in Loop Simulation of a Way Point Navigation Using Matlab/Simulink and X-Plane Simulator," 2019 International Conference on Intelligent Sustainable Systems (ICISS), 2019, pp. 332-335, doi: 10.1109/ISS1.2019.8908103.
- [6] E. Capello, G. Guglieri, F. Quagliotti, "A Waypoint-Based Guidance Algorithm for mini UAVs" in *IFAC Proceedings Volumes*, vol. 46, Nov. 2013, pp 120-125. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667015402812>
- [7] K. Ogata, "PID Controllers and Modified PID Controllers" in *Modern Control Engineering*, 5th ed. Upper Saddle River, New Jersey, United States: Prentice Hall, 2010, pp. 567-648.
- [8] M. Aamir, "On replacing PID with ANN Controller for DC Motor Position Control," arXiv:1312.0148 [cs.SY], Nov 2013.
- [9] Parkale, Yuvraj. "Comparison of ANN Controller and PID Controller for Industrial Water Bath Temperature Control System using MATLAB Environment", vol. 53, no. 2, pp. 1-6, Sep. 2012, doi: 10.5120/8390-1967.

- [10] J. Ariss and S. Rabat, "A Comparison Between a Traditional PID Controller and An Artificial Neural Network Controller in Manipulating a Robotic Arm", Bachelor thesis, Dept. Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, 2019. [Online]. Available: <http://kth.diva-portal.org/smash/get/diva2:1351191/FULLTEXT01.pdf>
- [11] L. Fausett, "Introduction" in Fundamentals of Neural Networks: Architectures, Algorithms and Applications. New Jersey: Prentice Hall International, 1994, pp 1-37.
- [12] F. Chollet, "Fundamentals of Deep Learning" in *Deep Learning with Python*. Shelter Island: Manning Publications, 2018, pp 1-4.
- [13] A. Zayegh and N. Al Bassam, "Neural Network Principles and Applications" in *Digital Systems*. London: Pearson, 2018, pp 115-132.
- [14] S. Kim et al. "Deep Learning for Bayesian Optimization of Scientific Problems with High-Dimensional Structure," arXiv:2104.11667 [cs.LG], Apr 2021.
- [15] M. Abadi et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. [Online]. Available: <https://www.tensorflow.org/>
- [16] O'Malley et al., KerasTuner. <https://github.com/keras-team/keras-tuner>, 2019. [Online]. Available: <https://github.com/keras-team/keras-tuner> -> 16
- [17] Muhammad, H. and Y.I. Jenie. (2010). Diktat Kuliah AE3231 Dinamika Terbang, Fakultas Teknik Mesin dan Dirgantara, Institut Teknologi Bandung, Bandung.
- [18] L'avionnaire. "Flight Controls." lavionnaire.fr. <https://www.lavionnaire.fr/AngFlightControl.php> (accessed Jul. 1, 2022).
- [19] Institut Teknologi Bandung. (2022). Introduction to Flight Measurement Sensors.
- [20] R. A. Sasongko (2021). Handsout Kuliah AE3200 Teknik Kendali Chapter 6. Fakultas Teknik Mesin dan Dirgantara, Institut Teknologi Bandung, Bandung.
- [21] E.H. J. Pallet and S. Coyle, "Principles of Flight" in *Automatic Flight Control*. Oxford: Blackwell Publishing, 1993, pp 1-55.

- [22] European Airspace Concept Handbook for PBN Implementation, 3rd ed., Eurocontrol, 2013. [Online]. Available: [https://www.icao.int/EURNAT/Other%20Meetings%20Seminars%20and%20Workshops/PBN%20TF/PBNTF%20-%20EUROCONTROL%20RAISG1/PBNTF_ECTL_RAISG1_IP07_European-ac-handbook-pbn-implement-2013-web%20\(ID%2040765\).pdf](https://www.icao.int/EURNAT/Other%20Meetings%20Seminars%20and%20Workshops/PBN%20TF/PBNTF%20-%20EUROCONTROL%20RAISG1/PBNTF_ECTL_RAISG1_IP07_European-ac-handbook-pbn-implement-2013-web%20(ID%2040765).pdf)
- [23] M. Tooley and D. Watt. “Inertial Navigation Systems” in *Aircraft Communications and Navigation Systems*, 2nd ed. Oxon: Routledge, 2018, pp 223-238.
- [24] Laminar Research Press Releases. “Laminar Research Announce Release Version of X-Plane 11.” x-plane.com. <https://www.x-plane.com/x-world/laminar-research-press-releases/> (accessed Jul. 4, 2022).
- [25] M. Finlay. “Small But Mighty: The Story of The Cirrus Vision SF50.” simpleflying.com. <https://simpleflying.com/cirrus-vision-sf50-story/> (accessed Jul. 4, 2022).
- [26] GlobalAir.com. “Cirrus Aircraft Vision SF50.” globalair.com. <https://www.globalair.com/aircraft-for-sale/specifications?specid=1254> (accessed Jul. 4, 2022).